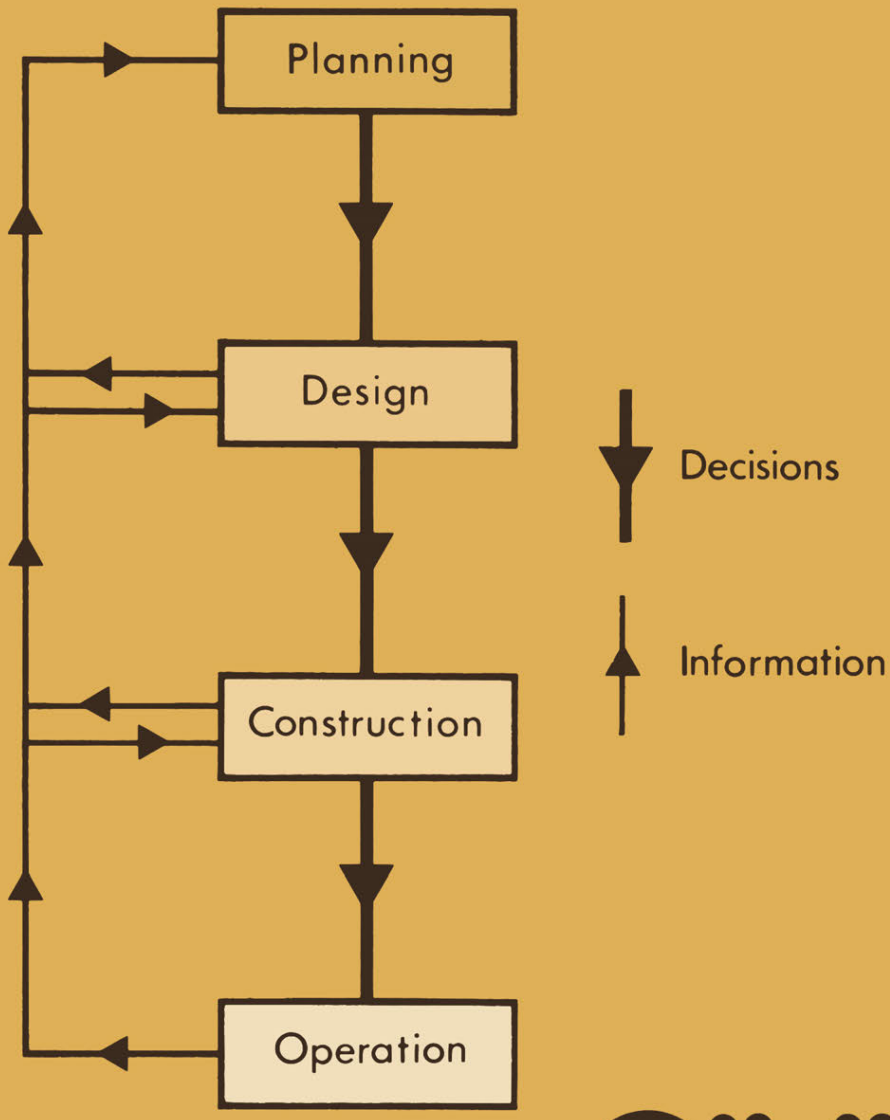


CIVIL ENGINEERING SYSTEMS

ANDREW B. TEMPLEMAN

www.EngineeringBooksPDF.com

CIVIL ENGINEERING SYSTEMS

www.EngineeringBooksPDF.com

Other engineering titles from Macmillan Education

Malcolm Bolton, **A Guide to Soil Mechanics**

J. G. Croll and A. C. Walker, **Elements of Structural Stability**

J. A. Fox, **An Introduction to Engineering Fluid Mechanics, Second Edition**

N. Jackson (ed.), **Civil Engineering Materials, Second Edition**

W. H. Mosley and J. H. Bungey, **Reinforced Concrete Design**

Stuart S. J. Moy, **Plastic Methods for Steel and Concrete Structures**

Ivor H. Seeley, **Civil Engineering Quantities, Third Edition**

Ivor H. Seeley, **Civil Engineering Specification, Second Edition**

J. D. Todd, **Structural Theory and Analysis**

E. M. Wilson, **Engineering Hydrology, Second Edition**

www.EngineeringBooksPDF.com

Civil Engineering Systems

Andrew B. Templeman

Department of Civil Engineering
University of Liverpool

M

www.EngineeringBooksPDF.com

© Andrew B. Templeman 1982

All rights reserved. No part of this publication may be reproduced or transmitted, in any form or by any means, without permission.

First published 1982 by
THE MACMILLAN PRESS LTD
London and Basingstoke
Companies and representatives
throughout the world

Typeset in 10/12 pt Press Roman by
MULTIPLEX techniques ltd, Orpington, Kent

ISBN 978-0-333-28510-7 ISBN 978-1-349-86099-9 (eBook)
DOI 10.1007/978-1-349-86099-9

The paperback edition of the book is sold subject to the condition that it shall not, by way of trade or otherwise, be lent, resold, hired out, or otherwise circulated without the publisher's prior consent in any form of binding or cover other than that in which it is published and without a similar condition including this condition being imposed on the subsequent purchaser.

www.EngineeringBooksPDF.com

CONTENTS

Preface *viii*

1 Systematic Decision-making in Civil Engineering **1**

1.1 What is Civil Engineering Systems? 1
1.2 The Civil Engineering Project 4
1.3 Systematic Decision-making 11
1.4 Mathematical Decision-making Models 14
Summary 16

2 Systematic Mathematical Modelling – Linear Problems **17**

Introduction 17
Example 2.1 – Earthmoving Operations 17
Example 2.2 – Precasting Plant 22
Example 2.3 – Rigid-Plastic Design of Frameworks 25
2.4 The General Linear Programming Problem 30
Summary 37

3 Solution Techniques for Linear Problems **38**

Introduction 38
3.1 The Simplex Method for Linear Programming Problems 38
3.2 Sensitivity Analysis and LPs 58
3.3 Duality in Linear Programming 63
3.4 Other Methods for Solving LP Problems 66
3.5 Negative Variables 67
3.6 Problems with Integer or Discrete-valued Variables 70
3.7 Civil Engineering Uses for Linear Programming 73
Example 3.8 – Water Resource Management 75
Summary 78
Bibliography 78
Exercises 78

www.EngineeringBooksPDF.com

4 Project Planning Methods, Networks and Graphs **82**

Introduction 83
4.1 Construction Planning Networks 83
4.2 Linear Programming and Construction Planning Networks 97
4.3 Resource Allocation and Project Control 102
4.4 Generalised Network Problems 105
4.5 Directed Networks 108
4.6 Undirected Networks and Graphs 119
Summary 125
Bibliography 125
Exercises 126

5 Serial Systems and Dynamic Programming **129**

Introduction 129
Example 5.1 – A Critical Path Problem 129
5.2 Generalisation of the Network Approach 135
Example 5.3 – Allocating a Tower Crane 144
Example 5.4 – A Purification Process 148
Example 5.5 – Drainage Design 152
5.6 Further Aspects of Dynamic Programming 157
Summary 163
Bibliography 163
Exercises 164

6 Systematic Design and Non-linear Problems **167**

Introduction 167
6.1 Systematic Design 167
6.2 Simple Design Examples 171
6.3 Features of Non-linear Programming Problems 176
6.4 Engineering and Mathematical Viewpoints on Non-linear Optimisation 184
Summary 185

7 Non-linear Unconstrained Optimisation Methods **186**

Introduction 186
7.1 The Classical Differential Method 186
7.2 Zeroth-order Methods 190
7.3 First-order Methods 213
7.4 Second-order Methods 224
7.5 Appropriate Methods for Engineering Problems 232
Summary 234
Bibliography 234
Exercises 235

www.EngineeringBooksPDF.com

8 Non-linear Constrained Optimisation Methods **237**

Introduction 237
8.1 Simple Solution Devices 238
8.2 Lagrange Multiplier Methods 241
8.3 Penalty Function Methods 249
8.4 Linearisation Methods 257
8.5 Direct Numerical Search Methods 263
8.6 Geometric Programming 265
Summary 286
Bibliography 287
Exercises 287

9 Non-linear Optimisation in Civil Engineering **291**

Introduction 291
9.1 Example – A Pumped Pipeline 292
9.2 Micro-design of Engineering Elements 297
9.3 Design of Multi-element Structural Systems 301
9.4 Other Non-linear Problems 307
Summary 309
Bibliography 309

10 Probabilistic Decision-making **310**

Introduction 310
10.1 Deterministic and Probabilistic Quantities 311
10.2 Probabilistic Decision-making Problems 313
10.3 Random Variables and their Properties 315
10.4 The Use of Expected Values for Decision-making 325
10.5 Maintenance and Replacement Problems 340
10.6 Reliability 343
Summary 355
Bibliography 355
Exercises 356

Solutions to Exercises 360

Index 366

www.EngineeringBooksPDF.com

PREFACE

Operations research, management science, mathematical optimisation and statistical decision-making are specialised disciplines which have blossomed since the Second World War. They are all concerned with quantitative methods for the solution of decision-making, planning and control problems in industrial and commercial enterprises. Many of the methods are applicable to a wide range of civil engineering problems and the profession is gradually accepting some of them and benefiting from their use. This book introduces some of the methods and concepts of these specialised disciplines which are particularly useful and applicable to practical civil engineering problems.

Civil engineering systems is, however, far more than a convenient holdall for diverse specialist mathematical methods. Civil engineering systems is concerned with decision-making processes within the civil engineering profession. It provides a logical, comprehensive framework for the study of civil engineering decision-making, and consequently many techniques from other disciplines which are concerned with decision-making naturally find a place in civil engineering systems.

The book is based on lecture courses given over a number of years to civil engineering students at the University of Liverpool. These courses present the practice of civil engineering as a creative, decision-making process for which a systematic approach and a knowledge of some efficient decision-making methods are invaluable. The material in this book is aimed at final-year undergraduate and master's degree levels although some of the topics could easily and appropriately be taught earlier. The book assumes a knowledge of simple differential calculus, vectors and matrices but all the mathematical methods described are developed simply and are self-contained. Only an elementary knowledge of technological theory and analysis, for example, structural mechanics and hydromechanics, is assumed. An important feature of the book is that civil engineering considerations are always uppermost. All mathematical methods are developed in a rigorous mathematical fashion but are only developed when a number of practical civil engineering problems have clearly

www.EngineeringBooksPDF.com

demonstrated the need for a mathematical solution method. The theoretical aspects are illustrated as much as possible with detailed examples drawn from civil engineering. The arrangement of the book is as follows.

Chapter 1 is of an introductory nature, characterising civil engineering as a decision-oriented profession and examining the nature of the decisions that have to be made during the planning, design, construction and operation phases of a civil engineering project. The underlying aim of making the best possible decisions is presented as a process of optimisation. A four-stage systematic approach to decision-making, used frequently throughout the book, is introduced.

The next three chapters deal with linear decision-making models and methods. Chapter 2 uses civil engineering examples to illustrate the systematic approach and derives linear programming problems for each example. The nature of LP problems is examined. This leads naturally into chapter 3 where the simplex method for solving LP problems is presented. Several aspects of linear programming and its uses in civil engineering are examined. Chapter 4 deals with networks. It describes the critical path method of construction planning in its usual form, and then shows the basic linearity of the method by relating it to linear programming. The linearity is then used to examine other network problems and some simple graph problems are explained.

Chapter 5 covers dynamic programming in a non-classical fashion. A construction planning example is solved by constructing a network of possible solution policies. Methods from chapter 4 are then used to find an optimal path through the network and the algorithm is then generalised to become the DP method. Several further civil engineering problems are described and solved to illustrate many aspects of dynamic programming.

Chapters 6 to 9 are concerned with non-linear decision-making models and methods. Chapter 6 shows by simple examples that almost all civil engineering design problems are non-linear. Some general characteristics of non-linear optimisation problems are examined. Chapter 7 deals with solution methods for unconstrained optimisation problems and chapter 8 with methods for constrained optimisation. These chapters are the most mathematical in the book with very little civil engineering content. Chapter 9 balances the two previous chapters by concentrating on the civil engineering applications of non-linear optimisation. Several examples are studied in detail.

Chapter 10 deals with uncertainty in the decision-making process. The nature of the solutions to be expected when statistical information is introduced into a problem is examined and several statistical decision-making methods are presented using civil engineering examples. The concepts of reliability-based decision-making are examined.

Many of the chapters have a bibliography which suggests specialised texts for further reading. Also many chapters have a final section of problems for the reader to solve. For each problem the briefest of numerical solutions is provided at the back of the book. My experience is that students tend not to attempt to

www.EngineeringBooksPDF.com

solve problems unless they have some way of telling whether their solutions are right or wrong.

The most difficult aspect of writing this book has been the conscious omission of useful and interesting topics. Those included are probably the major ones of interest to civil engineers, but who could argue that, for example, queuing theory or inventory theory are not also of use in civil engineering? They are omitted with reluctance along with many equally relevant and useful topics. The book is an introductory one to a very wide and diverse discipline. I hope that it will encourage others to explore this field for themselves and to be rewarded by the pleasure and stimulus which I have found there.

ANDREW TEMPLEMAN

www.EngineeringBooksPDF.com

1 SYSTEMATIC DECISION-MAKING IN CIVIL ENGINEERING

1.1 WHAT IS CIVIL ENGINEERING SYSTEMS?

Civil engineering is a creative profession. The role of the civil engineer is essentially one of *synthesis*, planning and designing, moulding and shaping the domestic and industrial environment. In order to create and synthesise, civil engineers must be fully aware of how the materials they use and the artefacts they build will behave under working conditions. The education of a civil engineer is consequently much dominated by learning how things behave and how that behaviour may be determined by *analysis*. Knowledge of disciplines such as structural mechanics, hydromechanics, soil mechanics and their associated analysis techniques is an essential prerequisite for a civil engineer. Essential though a knowledge of analysis is, however, it is a mistake to think that civil engineering is an analytically dominated profession. Quite the opposite is true: analysis is important only as an adjunct of the process of synthesis.

Faced with a completely designed civil engineering project most civil engineering graduates should be able to analyse how it will behave under working conditions. Usually this is done by establishing a *mathematical model* of the project which embodies the known mechanical laws (equilibrium, compatibility, material properties, conservation of energy). This mathematical model is then manipulated and solved to yield values for the required behaviour parameters (stresses, displacements, flows). Usually the analysis yields a unique set of results which can be checked against ranges of acceptable values from codes of practice and other sources.

Faced with the problem of designing the same project instead of analysing it, things are very different and much more difficult. It would be very convenient if the process of analysis were completely reversible, but of course it is not. It is not possible to start with a set of acceptability criteria such as are specified by codes of practice, and to work backwards to a unique structure or project which satisfies those criteria, without at many stages making *design decisions* about the shape and dimensions of the structure. In design, a set of acceptability criteria

www.EngineeringBooksPDF.com

does not define a unique solution except in the most trivial of examples. Generally there are very many widely different designs that will satisfy the acceptability criteria and a single design will only be obtained by making decisions which eventually eliminate all alternatives but one. Design is, therefore, a *decision-making process*, unlike analysis which allows no scope for choice or decision-making.

It is this decision-making aspect of design which makes it so daunting to civil engineering students. How can beginners in the profession make the right decisions? Clearly an injudicious decision at any stage might ultimately lead to a design that is unnecessarily difficult or expensive to build or even to one that fails to meet the acceptability criteria. How can an ability to make good decisions be learned?

Until recently many university courses in civil engineering have treated this question with extreme caution and have concentrated on analysis instead of getting to grips with decision-making and design. In support of this policy it can be argued that good decision-making and good design should be learned through experience, and that the best experience is gained in practical design offices not in universities. There is certainly some truth in this but this should not imply that practical experience is the only route to good decision-making. There are many quantitative techniques available that permit good decisions to be made according to logical, planned principles. It is these techniques with which civil engineering systems is concerned.

More precisely civil engineering systems is concerned with quantitative decision-making techniques of use in the planning, design, construction and operation of civil engineering projects. It is essentially concerned with the synthesis rather than with the analysis aspects of civil engineering. It relies on the same mathematical modelling approach as analysis. Instead of constructing a mathematical model for analysis and manipulating it to yield behaviour parameters, a different mathematical model is constructed for synthesis purposes so that the manipulation and solution yields design decisions (numbers, sizes, configurations). Civil engineering systems is concerned with mathematical models of this synthesis type.

The noun 'system' has several distinct meanings and it is useful to define it here in the sense in which it is used in the phrase 'civil engineering systems'. The relevant definition of 'system' in *The Concise Oxford Dictionary* is: 'Method, organization, considered principles of procedure, (principle of) classification'. Closely related to this is the adjective 'systematic' which is defined as: 'Methodical, according to a plan, not casual or sporadic or unintentional, classificatory'. In the sense of these definitions civil engineering systems is the study of systematic methods and procedures used in civil engineering with particular emphasis on decision-making. Perhaps a better phrase which focuses the subject more clearly is *systematic decision-making in civil engineering*.

Important features of the dictionary definitions are the words 'classification' and 'classificatory'. Civil engineering systems is concerned with examining the mathematical structures of decision-making methods for all kinds of civil engin-

www.EngineeringBooksPDF.com

eering problems, identifying similar mathematical forms and classifying decision-making methods according to these forms. An analogy can be drawn with structural mechanics. There is an infinite variety of structural forms – bridges, beams, columns, plates, shells, etc. – yet the analysis of all these different structures under applied load may be performed using only a few techniques. Examination of the behaviour of these structures has shown that although all the structures are different they behave in similar ways. Thus a knowledge of linear elasticity, elasto-plastic theory, rigid-plastic theory and buckling is sufficient to enable an analysis to be performed on many structures. Analysis is, therefore, classificatory. Civil engineering systems attempts to do the same sort of classification with decision-making problems. It attempts to establish a basic framework encompassing all decision-making methods so that any decision-making problem can be examined, classified and, with the aid of a few appropriate methods, solved.

Earlier it was suggested that there is no substitute for experience in making good decisions. This is true but may be qualified. To derive maximum benefit from experience it must be possible to relate that experience to some fundamental, logical principles. There must be some context or framework into which the experience may be fitted. The slavish following of precedents leads only to stagnation. Civil engineering systems attempts to create a basic framework for decision-making, not as a substitute for experience but to enhance experience by providing a context for it.

As a separate discipline, civil engineering systems is of very recent origin. Indeed, the systematic study of any form of decision-making was virtually non-existent until the Second World War. At that time large armies, navies and air forces were deployed on a worldwide scale. The problems of supplying them, controlling and co-ordinating their tactical and strategic operations into an over-all plan were immense. It was soon evident that *ad hoc* planning on the basis of past experience was inadequate to control everything. New mathematically based planning and forecasting methods were required to ensure that production of war materials of all forms met the anticipated demands and was supplied to the right places at the right times. When the War finished it was realised that these same methods and new ones could be used in peace-time to plan industrial regeneration. Around 1950, so great was the research activity in planning methods that it became a separate discipline known as *operations research*, or *OR*. OR, as its name implies, is concerned with the planning, assessment and control of operating systems such as industrial production or commerce. Interest in methods for the design of these systems rather than in their operation led to *systems engineering*, a product of the 1960s. These two disciplines of OR and systems engineering supply both the philosophical *raison d'être* and the methodology of civil engineering systems. To be precise, decision-making in design is covered by systems engineering, and decision-making in planning, construction, operation and management is covered by OR although in reality the two overlap and merge into an over-all systematic approach.

www.EngineeringBooksPDF.com

This book is intended as an introduction to the use of systems and OR concepts for civil engineers. The scope of these disciplines is much wider than that of this book. Here only those methods which have relevance to civil engineering are introduced; less relevant topics in systems and OR are omitted. Also, necessarily, some useful topics such as inventory theory, queuing theory and others are not included because of space limitations. This book is, therefore, a beginner's guide to systems and OR in civil engineering. The interested reader who wishes to know more than can be covered here must turn to the books and articles listed in the bibliographies to each chapter. These texts are not specifically intended for civil engineering readers but are interdisciplinary in nature. Some are mathematical, others are of a general engineering nature, others have a commercial flavour. In the author's opinion this is a distinct advantage. One of the attractive features of the methods and concepts of civil engineering systems is that they are obviously of as much relevance outside civil engineering as inside; they provide an extra insight into how the rest of the world works. At this point, however, it is necessary to return to civil engineering and examine in more detail the many decisions which have to be made to translate an idea into a working civil engineering artefact.

1.2 THE CIVIL ENGINEERING PROJECT

All civil engineering projects have four distinct phases

(1) planning
(2) design
(3) construction
(4) operation.

The relative importance of these phases varies depending on the nature and size of a particular project but all are present to some extent, even in the simplest of projects. A major project such as a new airport or sewage-treatment facility involves many decisions in all phases. It is instructive to examine each of the four phases in turn and to determine the sorts of decisions which have to be made. It is helpful in doing this if in some major project such as the airport or treatment works is borne in mind.

1.2.1 The Planning Phase

The planning phase of a project precedes all the other three phases but can overlap the design phase. The planning phase starts with the idea for the project and examines that idea from many angles. Perhaps the most important decisions which must be made are those concerned with whether or not to pursue the idea for the project any further. In order to make this decision many questions must be answered. Among these are the following

Is the project needed?
What will the project cost?

www.EngineeringBooksPDF.com

What will the benefits of the project be?
Where will it be located?
How big will it be?
What impact will it have on the environment?
Who will pay for the project?
How will the work be financed?
What alternatives are available?
What are their quantitative advantages and disadvantages?

The list is by no means exhaustive. The important thing about these questions is that they are very general yet they each require very specific answers.

In the case of major projects the over-all planning decisions may not be made by civil engineers. For example, the final decision whether or not to proceed with a new airport project might be made at national government level. Other decisions in project planning are often made by local committees. The major decisions are, therefore, sometimes made on grounds other than those of civil engineering viability. Nevertheless, although civil engineers may not make the actual planning decisions, they still carry a major responsibility for providing technical information that will enable others to make the decisions in an informed way. Many different disciplines and professions may be involved in providing specialist advice about aspects of the project. Questions about the size and cost of the project and its alternatives are the concern of civil engineers. Questions about the likely benefits of the project require information of a socio-economic nature. Environmental considerations also require specialist assessments. Planning is essentially a group activity. Within the group each member must be able to supply advice on his own speciality and must also be aware of the activities of other specialists.

Ideally, the objective in planning is to provide the decision-makers with alternatives. Several possible schemes of different sizes and using different methods and concepts should be presented. Each scheme should be technically viable and should have broad total cost estimates associated with it. For each scheme a balance sheet of costs, benefits, advantages and disadvantages should be prepared by the planning group as a whole. Given this information the decision-making body may then make its selection.

For the civil engineering members of a planning group these alternative schemes take the form of *feasibility studies*. A civil engineering feasibility study is itself an exercise in decision-making. For a project such as the new airport, several different schemes must be evaluated in terms of civil engineering viability and cost. For each scheme many decisions are needed. How many runways should there be and what length and what orientations should they have? What terminal facilities, maintenance facilities and car parks should there be and how big should they all be and how should they be laid out? Some of these decisions encroach on the design phase but at the planning level detailed design is not usually required. Decisions only need to be made such that over-all cost esti-

www.EngineeringBooksPDF.com

mates can be drawn up for each scheme. Thus feasibility studies are as much concerned with cost as with technical feasibility. Indeed, almost all schemes can be made technically feasible but sometimes this can only be done at great cost. Feasibility studies are therefore very much exercises in *cost modelling*. All other things being equal, the least cost scheme usually attracts especial attention.

If several schemes are presented at the planning stage it is important to maintain comparability among them. If one scheme has been worked out in great detail and special efforts have been made to improve and refine it, whereas another scheme has only been very roughly evaluated, the job of choosing one or the other is made more difficult. By how much would the second scheme improve as a result of more care and attention? Ideally, therefore, there should be some common basis for comparison among the alternatives. As this book will demonstrate, such a basis does exist if a systematic approach to decision-making is used. All participants in the planning process should be motivated by the same desire: to make the best possible decisions. By whatever means this is achieved, formally or informally, mathematically or through experience, this desire is an *optimum-seeking* motivation. Civil engineering systems is much concerned with processes of *optimisation*.

1.2.2 The Design Phase

The design phase may be considered to start when the major planning phase decisions have been made. Clearly a decision to proceed with the project must have been made. The other major decisions of the planning phase should have selected a particular scheme from the alternatives and this scheme becomes a framework within which the design phase is conducted. Taking the airport example, the planning decisions should have determined an over-all scheme. A site should be known. Numbers, lengths and orientations of runways should be known. Capacities of terminal facilities for passengers and freight, car-parking capacities, aircraft handling and maintenance capacities should also be known. In other words a design brief or specification should be available. The design phase consists of providing a complete design which fulfils that specification. Clearly the design phase is a major civil engineering responsibility.

It is helpful to consider the design phase as having two parts – *macro-design* and *micro-design*. The design brief is usually sufficiently general to allow considerable scope for creative civil engineering design. For example, in the airport scheme the brief may include the provision of underground car-parking for five thousand cars. The macro-design stage must determine how this requirement is to be satisfied. This requires many decisions to be made. For instance

How many car parks should there be?
How big should each one be?
Where should they be located?
How many underground storeys should there be in each car park?

www.EngineeringBooksPDF.com

What should be the layout of access roads?
What should be their proximity to terminal facilities?
What materials should be used?

The decisions that must be made in macro-design are very similar to those for feasibility studies in the planning phase. They narrow down a general requirement to a much more detailed design and produce a very specific brief for the micro-design stage. Micro-design is then concerned with the detailed design of the elements of the project: member sizes, arrangements, joints, etc.

In the macro-design stage major considerations in decision-making are those of cost and technical suitability. Cost models are important for deciding between alternatives but are not the only means available. There are still inputs to the decision-making process from outside the design office. Design decisions cannot be made independently of the means of construction and this implies an overlap with the construction phase. Also there must be consideration given to the users of the designed project. Does it relate well or badly to the over-all project plan as defined by the planning phase? The objectives of macro-design are to make the best possible decisions in the light of all influencing factors – again a process of optimisation.

In the micro-design stage the influence of the planning phase is much reduced but the detailed design of all the project elements is influenced by the needs of the construction phase, for example, whether the design will be easy and cheap to build, and by the parallel design stages of non-civil elements. Typically, civil engineering designers must compromise with others over electrical, mechanical, heating and ventilation design requirements. The governing motivation is always to make the best possible decisions. The end product of the design phase should be complete plans and drawings for the entire project. This end product forms the starting brief for the construction phase.

1.2.3 The Construction Phase

The construction phase turns the project design into reality and the civil engineering contractor is responsible for the construction work. His over-all objective is to complete the construction work within an estimated time, according to the design and generally as efficiently as possible so as to maximise his profits on the contract. In order to do this the contractor must plan his operations very carefully and has many decisions to make. Typically, these include

What is the best order for the various construction activities?
How long will each activity take?
What plant is needed for each activity?
How many men are needed for each activity?
How are the men and machinery allocated among activities?
How much will each activity cost?
Will the materials be available when required?
Is there sufficient money available to pay for everything?

www.EngineeringBooksPDF.com

Very few of the contractor's decisions are technological ones. Frequently contractors negotiate minor design changes in order to make the construction easier. Also there are usually some unforeseen difficulties, perhaps associated with groundwater or unexpected subsoil properties, etc., which require technological expertise. There is, therefore, some overlap with the design phase. Most of the contractor's decisions, however, concern logistics rather than technology. Furthermore the contractor has to make each decision very many times over as the construction work alters. Yesterday's allocations of men to tasks must be reviewed and modified to take account of the changed position of the construction work today. The contractor, therefore, is faced with regular intensive decision-making. Most of these decisions are concerned with ensuring that men, machinery, materials and money are available as required and with allocating these resources among the many construction activities which require them.

The construction phase perhaps benefits more than any other from the use of systematic planning methods. The interrelationships among men, machinery, materials, time and money are so complicated and are so much affected by supply and demand for these resources that the control of construction work on an *ad hoc* basis for a large project is almost impossible. It is, therefore, natural that many of the methods described in this book are of most use in the construction phase. It is also no coincidence that the construction phase remains a most fertile area for new methods of decision-making. It is obvious that the over-all objective in the construction phase is optimisation in the widest sense of doing the best possible with available resources.

1.2.4 The Operation Phase

The last of the four phases of a civil engineering project, the operation phase, is sometimes overlooked yet it is often the most important of the four. All projects have an operation phase; in some it is obvious and important, in others less so. For a new airport or sewage-treatment works the operation phase is the culmination of the project. The first three phases can only be judged to have been successful if the completed project can be operated efficiently. Although the civil engineers concerned with design and construction are not usually actively involved in the operation phase, other civil engineers may be.

Design and construction is usually associated with the private sector of the civil engineering industry. Civil engineering, however, also has a large public sector. Very many civil engineers work for local authorities, water authorities and nationalised industries. The operation phase is of particular interest to this group of civil engineers who are responsible for the running of essential services such as water supply, sewage and solid waste disposal or for the planning and operation of public transport, maintenance of road and rail systems, etc. It is perhaps artificial to class operation as the fourth phase of a project since it usually involves different groups of people but this classification has the merit of providing a reminder that, when all design and construction work is finished, there is still much work to be done by civil engineers.

www.EngineeringBooksPDF.com

The determination of efficient operating policies dominates the operation phase. These require decision-making on a regular, planned basis if the policies are to remain efficient as operational circumstances alter. Plant, vehicles and the service infrastructure must be maintained, repaired and replaced systematically. Long-term changes in demand for services such as water or transport must be forecast and planned for. Improvements in performance of new equipment must be monitored and operational activities changed to reflect these improvements. Many of the methods described in this book are applicable to decision-making in an operational context and are, therefore, particularly relevant to the work of civil engineers in the public sector.

1.2.5 Comments

Several general comments may be made about the four phases of a project. From the initial idea through to the operation of the completed project countless numbers of decisions, large and small, must be made. The success of the entire project depends on the right decisions being made at all stages.

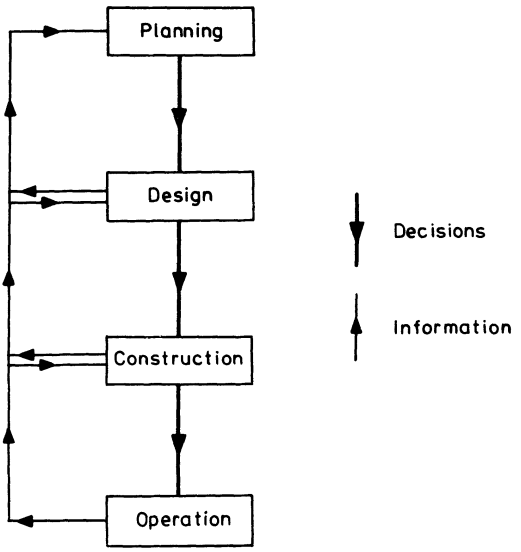


Figure 1.1 The sequential nature of a project

The four phases of a project are essentially sequential as far as decision-making is concerned. Figure 1.1 shows this. The planning phase isolates a specific plan from a general idea. The design phase operates within the limits of this plan and isolates a complete detailed design. The construction phase is carried out within the limits of the detailed design and converts it into a completed project. The

www.EngineeringBooksPDF.com

operation phase is limited by what the three preceding phases have provided. Thus each phase establishes firm criteria for the subsequent phase. If the entire project is to be carried out efficiently it is important that these interface criteria between phases are right. There must be a reverse flow of information between and among the phases to ensure this.

The sequential nature of the decisions associated with figure 1.1 places extra weight on those decisions made early in the sequence. An injudiciously chosen over-all plan affects all decisions in the design, construction and operation phases and may irrevocably impair the efficiency and value of the whole project. A poorly organised construction phase may cost time and money but is not usually as damaging to the project as a whole. Planning decisions are, therefore, the most important of all but because of the many and varied inputs to the planning process they are also the most difficult decisions to make. Subsequent phases become increasingly constrained and decision-making can consequently be more precise.

All the project phases are oriented towards the single goal of producing a completed project that functions efficiently. Each phase taken separately also has efficiency as an underlying theme. This general desire always to do the best with available resources, to produce the best possible plan or design and to make the best possible decisions can be classified as a process of optimisation. Optimisation is often thought of in a numerical or mathematical sense. In fact it is far wider than this. Optimisation means the selection of the best from a number of alternatives. Sometimes logical mathematical methods can be used to make this selection and such methods are classed as *formal* optimisation. At other times engineers may select the best from alternatives using past experience. This is just as much an optimisation process although it is an *informal* one; the necessary retrieval, evaluation and comparison of alternatives is done by the engineer's brain instead of by a computer. Optimisation is, therefore, fundamental to all decision-making processes and is treated in detail in this book.

Analysis, in the sense of the calculation of the response of an artefact to external loads, plays an essential though subservient role in the phases of a civil engineering project. Without analysis many projects could not be undertaken and no project could be efficiently completed. It is the means by which feasible alternatives are separated from infeasible ones; a filter on the decision-making process. Occasionally analysis and technological aspects dominate decision-making; examples of this are projects such as the Sydney Opera House and the Concorde aircraft. More usually, however, it is cost in its many forms that dominates decision-making. Even the two high-technology projects mentioned above were not executed regardless of cost. It is too restricting to view civil engineering simply in terms of applied science and technology. Decision-making in civil engineering demands technological skill but is equally dependent on an appreciation of costs and economic factors. In this book it is assumed that the reader has a basic knowledge of the analysis of civil engineering works. Analysis is presented here as one element of a tool-kit for decision-making in which other elements have equal prominence.

www.EngineeringBooksPDF.com

Section 1.2 has demonstrated that civil engineering is a decision-oriented profession. So many decisions have to be made in the phases of any project that it is entirely logical to ask whether any systematic approach to decision-making exists. The next section proposes a general framework within which decision-making can be logically done.

1.3 SYSTEMATIC DECISION-MAKING

Pure scientists have a framework known as the *scientific method* which provides a means of organising and expediting their work. The scientific method has four elements

(1) observation
(2) hypothesis
(3) testing
(4) concluding.

Much has been written about the scientific method. Although the four elements are very simple to understand and present, when considered together they form a very powerful means of describing and organising all aspects of pure science. The purpose of this section is to propose a general framework for decision-making that has the same power and unity for decision-makers as the scientific method has for scientists.

A systematic decision-making approach may be stated in the form of four questions

(1) What decisions must be made?
(2) How are the decisions related and what external factors limit them?
(3) What criteria determine whether the decisions made are good or bad?
(4) How can the best decisions be made?

This book is devoted to showing, by examples, that these four questions are fundamental to all decision-making and provide a logical framework for making decisions. They constitute a *systematic decision-making method*. Answers must be supplied to these questions. It is contended in this book that, if the questions can be supplied with answers, sufficient information will be amassed to permit good decisions to be made for any problem.

The first question asks what decisions are needed. This can sometimes be difficult to answer, particularly in the case of a complex, interacting problem. The question immediately demands careful thought about the particular problem. What answers is this systematic process expected to give? In the case of structural design the decisions needed (or the answers we would like from this process) are things like numbers of members, orientations, dimensions, etc. – everything necessary to produce a dimensioned plan. Some of these quantities, however, may already be known. It is necessary to find out exactly what is known and to remove it from the list of things that must be determined. In the case of planning a large concrete-pour on site, for example, this separation of the

www.EngineeringBooksPDF.com

known from the unknown is important. Is the planning problem concerned with finding out what plant is needed or is it concerned with how to operate existing plant effectively? Answering question 1 requires careful thought about and 'analysis' of the problem and the determination of a list of essential decisions that would comprise a solution to it. In civil engineering these decisions usually concern quantifiable things such as men, reinforcing areas, beam dimensions, capacities, etc. In view of the fact that many decision-making problems can be solved numerically or mathematically it is useful to allocate an algebraic variable name to each quantity or decision for which a value is sought. These variables can then be used in a numerical decision-making model of the problem.

Having determined what decisions must be made, the second question asks how all these decisions or unknown quantities are related and what external factors limit them. In structural design, for example, beam depth may be required not to exceed some specified value; this is a direct limit upon the value of a decision. Codes of practice may limit axial stresses to some prescribed value. This provides a relationship between breadth and depth of a member. Quantities may be related together in many ways such as by moment of inertia relationships, bending deflections, etc. All possible factors which may affect the design should be listed and examined carefully to extract relationships between and among the design quantities that are to be found. In the concrete-pour example the different types of plant to be used will have different limiting capacities, cycle times, operating requirements, all of which provide limits for the unknown quantities that are to be found. All factors that might affect the pour operation should be listed, examined and quantified. If algebraic variables have been associated with each decision in question 1, it is clear that question 2 requires that algebraic functional relationships among the variables are written down. These relationships are most often inequalities, for example, some decision quantity or combination of quantities must be less than (or greater than) a known limit value. In mathematical terms the answer to question 2 is a complete mathematical model that describes all relationships between variables which might conceivably affect their values.

When question 2 has been answered, the decision-making problem has been thoroughly analysed in great detail and it should now be possible to select values for the unknown quantities (decisions) that satisfy all the relationships and limits. Occasionally it will be discovered that values to satisfy all the restrictions cannot be found. For example, in the concrete-pour, the concrete-batching plant may not be able to produce enough concrete to complete the pour in the time available or there may not be enough vehicles to carry the concrete between batching plant and site. In this case no solution to the problem as posed exists. The engineer must return to the start of the process and re-examine the problem in detail, namely 'What extra decisions are needed to ensure a feasible operation?', 'What extra plant or vehicles are needed?', 'How does this change the mathematical model?'.

More usually the difficulty in selecting values for the decisions after question

www.EngineeringBooksPDF.com

2 is not that no satisfactory values exist, but rather that too many possible values exist. There are too many solutions to the problem. In this case some extra criterion is needed to determine a unique solution to the problem from all the available alternatives. Question 3 asks what this criterion is which distinguishes good decisions from bad ones and good design from poor design. The answer to this question requires more thought about the objectives of the decision-making problem. Cost is very often adopted as a distinguishing criterion. If it is possible to do something satisfactorily in a number of ways then the way that does it most cheaply should be chosen. The use of cost as a decision-making criterion has several difficulties associated with it. It is generally used in the form of minimum total cost or maximum profit but, whichever is chosen, it can often be difficult to define precisely what is meant by 'cost'. As an example, consider the cost of buying a private motor car. The purchase price of a number of alternatives may easily be ascertained but this is not necessarily representative of the true cost of each car. A better picture of total cost would be given by adding to the present purchase price several extra cost items representing estimates of running costs, insurance, maintenance and repair costs for the anticipated lifetime of each car. These extra costs are often hard to estimate because they will be incurred in the future. They will, therefore, be affected by that unknown factor, inflation. The lifetimes of different cars are also different. Is it worth paying twice the purchase price for a car that will last much longer and require less maintenance than another? All these factors are present in the total cost of civil engineering plant, vehicles and artefacts. Also, most large projects are financed by loans which must be repaid with interest over long periods. Loan interest should, therefore, also be included as a total cost item. The use of minimum cost as a decision-making criterion is, therefore, not easy but nevertheless it is the most often used criterion, usually in a crude and imprecise form. The estimation of true cost which takes into account the changing value of money with time is not considered in this book in any detail but clearly it is necessary for some decision-making problems. Ideally the cost criterion should always be sufficiently representative of true costs to allow decisions to be made with accuracy. Thus in deciding upon a replacement policy for mechanical excavators, for instance, factors such as interest rates and inflation are important and should be included in the cost model. In choosing a depth for a concrete beam these factors are not important and should not be included. Cost modelling is a very large topic which can only be mentioned without further study in this book.

Cost is by no means the only criterion which may be used. The concrete pour provides an example of an alternative criterion. In large concrete-pours time can be more important than cost. It is often essential to complete the pour as quickly as possible to ensure uniform hardening, thus minimum time may be chosen as a criterion. Ideally, of course, it would be nice to be able to choose a criterion such as minimum cost in minimum time but this is impossible. Minimum time implies high cost and vice versa so these two criteria conflict. One way of circumventing this conflict is to select one criterion, say minimum time, as being the dominant

www.EngineeringBooksPDF.com

one and to assign a limiting value to the cost. The limiting cost then becomes another relationship among variables within the mathematical model. The decision problem then becomes one of finding values for the decision variables which satisfy all the relationships among variables including the cost limit and which, at the same time, minimise the time criterion.

Whatever criterion is chosen for the decision problem it should be expressed in the form of a function of the decision variables. Question 4 then asks how the best decisions may be made. It is sometimes found that by this stage the decision-making problem has been analysed in such depth that it is easy to choose values for the decisions that are optimal or almost optimal without recourse to any formal methods. The systematic decision-making approach has guided the thought processes to isolate the relevant elements and discard the irrelevancies to such an extent that good decisions are now obvious. Often, however, this is not the case and some formal mathematical solution to the decision model is needed. This book contains many solution methods for mathematical decision models.

The systematic decision-making method was stated in the form of four questions, but in examining the questions further the answers to them were expressed in mathematical terms. The answers contribute all the necessary elements of a mathematical decision model; decision variables, sets of relationships among the variables, an efficiency criterion expressed as a function of the variables and a solution method. The systematic method may, therefore, also be stated in a mathematical modelling form in which the steps correspond exactly to the four questions. The mathematical modelling form is

(1) assign variables
(2) construct relationships among the variables
(3) select a criterion and express it as a function of the variables
(4) solve the problem.

It is useful now to examine briefly the forms of some of these mathematical models.

1.4 MATHEMATICAL DECISION-MAKING MODELS

The three components of a decision-making model are the variables, the efficiency criterion and the relationships among the variables. There are several ways of incorporating these components into a mathematical statement. One form is

$$\left.\begin{array}{ll} \text{Find values for the variables } x_i, & \\ \text{such that relationships of the form} & i = 1, \ldots, N \\ g_j(x_1, x_2, \ldots, x_N) \left(\begin{matrix}\leqslant \\ = \\ \geqslant\end{matrix}\right) C_j & j = 1, \ldots, M \\ \text{are satisfied and the function} & \\ f(x_1, x_2, \ldots, x_N) \rightarrow \text{minimum (or maximum)} & i = 1, \ldots, N \end{array}\right\} \quad (1.1)$$

www.EngineeringBooksPDF.com

In problem 1.1 there are N variables $x_i, i = 1, \ldots, N$, which are the decision variables for which values must be found (step 1 or question 1). Sometimes these variables are expressed as the N components of a vector $\boldsymbol{x}$, thus $\boldsymbol{x} \equiv x_i, i = 1, \ldots, N$, There is a total of M separate relationships among the variables and limits on combinations of variables (step 2 or question 2). Each relationship or combination is expressed as a function $g_j, j = 1, \ldots, M$ of the variables $\boldsymbol{x}$ and each is less than or equal to, equal to, or greater than or equal to some known constant $C_j, j = 1, \ldots, M$. For each relationship j the appropriate inequality sense or equality is known, as is the value of C_j. The function $f(x_1, x_2, \ldots, x_N)$ is the criterion which determines the goodness or merit of the variable values (step 3 or question 3). Another way of expressing this problem is

$$\left.\begin{array}{ll} \text{Minimise (or maximise) } f(\boldsymbol{x}) & \\ \text{subject to} & g_j(\boldsymbol{x}) \left(\begin{matrix}\leqslant \\ = \\ \geqslant\end{matrix}\right) C_j \end{array}\right\} \qquad (1.2)$$

The form 1.2 is that used most often in this book.

Problem 1.2 is a mathematical problem of *optimisation* since its purpose is to find the optimum value of some efficiency criterion. The function $f(x_i)$ which is to be minimised or maximised is called the *objective function* or *merit function* since it quantifies the merit of a set of decisions. The M relationships which must also be satisfied by the variable values are referred to as *constraints*. The general model forms 1.1 and 1.2 are fundamental to most civil engineering decision-making problems. It is usually possible to cast a problem into these forms. They are not the only useful problem formats; there are some problems that do not easily fit these forms and others that may more easily be solved by ignoring these forms. Nevertheless the systematic approach to decision-making described earlier and explained in this book has these problem forms at its core. All decision-making problems should be approached with the goal of trying to establish mathematical models which have these forms.

The functions $f(\boldsymbol{x})$ and $g_j(\boldsymbol{x})$ may have very many forms. The classificatory aspects of civil engineering systems centre around the forms of these functions. If all the functions f and g_j are linear functions of the decision variables the problem is classed as one of *linear optimisation* or *linear programming* as it is usually called. The words optimisation and programming are synonymous in this context. A *program* is a solution method or solution plan by means of which a problem is solved (step 4 or question 4). If any of the functions f or g_j are non-linear in the variables, the problem is one of *non-linear optimisation* or *non-linear programming*. Taken together, linear and non-linear functions might appear to constitute an all-embracing classification but there are other ways of classifying problems. Problems may be classified as *deterministic* or *non-deterministic*. A deterministic problem is one in which the variables and functions represent quantities that can be expressed or determined uniquely. In a non-deterministic or *probabilistic* problem they represent quantities that can only be expressed or

www.EngineeringBooksPDF.com

determined as statistical distributions of values. Most of this book is concerned with deterministic problems but chapter 10 explores the non-deterministic area in which the problem forms 1.1 and 1.2 are less relevant.

Within the broad classes mentioned above of linear, non-linear, deterministic and non-deterministic problems there are many sub-classes depending on the nature of the variables and functions. Usually, the variables x_i, $i = 1, \ldots N$, are *continuous-valued*, that is, they may have any real value. Sometimes variables may be required to be *integer-valued* (for example, if x is the number of men needed to perform a task, x must obviously have only integer values), or *discrete-valued* (if x is the depth of a rolled-steel beam it can only have certain discrete values which correspond to the depths of available rolled beams). Each sub-classification requires its own solution techniques and this book examines many such methods for solving problems 1.1 and 1.2. The most important aspect of civil engineering systems is that, although there is an infinite number of different practical decision-making problems, they may nearly all be classified by the systematic approach into a relatively small number of mathematical problem types. This book demonstrates how this classification process works for a wide range of practical civil engineering problems and shows how mathematical solution methods can be devised for several major problem types.

SUMMARY

Civil engineering systems is concerned with logical, numerate, systematic decision-making methods for use in all aspects of civil engineering. It is a natural outcome of the application of well-established methods of operations research, management science and mathematical optimisation to civil engineering problems. The nature of civil engineering decision-making was examined in relation to a typical large civil engineering project. A project may be divided into four over-all phases; planning, design, construction and operation. In each phase many decisions must be made to establish interface criteria or limits within which the next phase is carried out. The types of decisions to be made in each phase are very different yet the entire project is characterised by the need to make the best possible decisions at all times. Decision-making is, therefore, a process of optimisation in its widest sense.

A systematic way of approaching all decision-making problems was proposed in the form of four questions. Provision of answers requires careful thought and analysis and is often sufficient to enable good decisions to be made without the need for formal methods. Often, however, formal decision-making methods are required. The answers to the four questions can be expressed in mathematical terms and a mathematical decision model constructed. Different forms of these models were proposed, each of which represents an optimisation problem. A brief examination of different classes of optimisation problems was made and will be elaborated on in the rest of this book.

www.EngineeringBooksPDF.com

2 SYSTEMATIC MATHEMATICAL MODELLING — LINEAR PROBLEMS

In chapter 1 the idea of a systematic approach to problem solving and decision making was introduced. In this chapter it is applied to three very different practical problems which are typical of those encountered during the life of a civil engineering project. The systematic approach enables a mathematical model to be constructed for each problem so that much of the superficial complexity disappears. Once the mathematical structure of each problem has been deduced, an important feature becomes evident; the mathematical structure of all three problems is the same, consisting only of linear functions of the variables.

This commonality of mathematical structure among widely different practical problems is important because it implies that a single solution technique designed to handle linear functions can be used to solve all three problems. In fact the solution technique has far wider applications than merely the three chosen examples. Linear programming, as the method is called, is applicable to very many practical problems in all four phases of a civil engineering project and in many aspects of everyday life. This chapter examines the nature of linear programming problems and provides the necessary groundwork for understanding the simplex algorithm for solving linear programming problems which is described in chapter 3.

Firstly, however, some practical problems which give rise to linear mathematical models are examined. Example 2.1 is an earthworks problem typical of those often occurring in the construction phase of a large civil engineering project. Example 2.2 is drawn from the operation phase and considers how to plan the operation of a precasting plant. Example 2.3 is a design problem which arises in the economic design of rigid-plastic frameworks.

EXAMPLE 2.1 – EARTHMOVING OPERATIONS

It is necessary to use a fleet of large earthmoving vehicles to level a large and uneven site prior to the start of construction operations. The objective is to

www.EngineeringBooksPDF.com

move the earth between cut and fill locations in such quantities that the site is levelled as cheaply as possible.

There are three areas on the site, A, B and C, at which cut material is produced in the following quantities: location A produces 5000 m^3, B produces 7000 m^3 and C produces 9000 m^3. There are four locations, D, E, F and G, at which fill material is required in the following quantities: location D requires 2000 m^3, E requires 6000 m^3, F requires 8000 m^3 and G requires 4000 m^3. There is also a convenient dump, H, to which excess cut material may be taken. Distances in km units between each of the cut sources A, B, C and each of the fill destinations D, E, F, G and the dump H have been measured from the site plan and are tabulated in table 2.1.

Table 2.1 Distances (km) between cut and fill locations

		Fill locations				
		D	E	F	G	H
Cut locations	A	0.7	0.3	0.2	0.4	0.9
	B	0.6	0.5	0.8	0.3	1.1
	C	0.3	0.2	0.5	0.7	0.8

Applying the systematic approach outlined in chapter 1 the first step is to determine what decisions have to be made and then to assign a mathematical variable to each decision. Material from a cut source may be taken in any quantity to any of the four fill locations or to the dump H. In order to start levelling operations it will be necessary to determine exactly how much material should be taken from each cut source to each possible fill destination. Each of these quantities represents a decision which has to be made. The complete list of all the quantities to be carried between sources and destinations is a *transportation schedule.* Variables can now be assigned to each of the decisions, that is, to each item of the transportation schedule. Let x_{ij} be the quantity of material in m^3 carried between source i, $i \equiv$ A, B, C and destination j, $j \equiv$ D, E, F, G, H. There are therefore a total of 15 variables for this problem: $x_{AD}, x_{AE}, x_{AF}, x_{AG}, x_{AH}, x_{BD}, x_{BE}, x_{BF}, x_{BG}, x_{BH}, x_{CD}, x_{CE}, x_{CF}, x_{CG}, x_{CH}$.

Having assigned these variables, the second step of the systematic approach asks what restrictions are imposed on the variables by the practical problem and it requires a mathematical model to be constructed which completely specifies all such relationships. The main practical requirement of the problem is that the site should be levelled. This implies that all the material produced at a cut location must be carried away and also that the requisite amounts of fill material at each location must be provided. Unless these conditions are met the

www.EngineeringBooksPDF.com

site will not be levelled and the corresponding transportation schedule would be unacceptable. The mathematical statements of these requirements may be obtained as follows.

Consider first the requirement that all cut material produced at the cut sites should be carried away. Consider cut source A. This produces 5000 m^3 of material. x_{AD} m^3 of this may be carried away from A to fill location D, x_{AE} from A to E, x_{AF} from A to F, etc. The total volume carried away to all possible destinations from A must equal 5000 m^3, thus the following may be written

$$x_{AD} + x_{AE} + x_{AF} + x_{AG} + x_{AH} = 5000 \tag{2.1}$$

Similarly at cut source B the total volume carried away from B, represented by the sum of all variables having B as the first subscript, must equal the cut produced of 7000 m^3. Thus

$$x_{BD} + x_{BE} + x_{BF} + x_{BG} + x_{BH} = 7000 \tag{2.2}$$

A similar expression can be written for cut source C as

$$x_{CD} + x_{CE} + x_{CF} + x_{CG} + x_{CH} = 9000 \tag{2.3}$$

Equations 2.1 to 2.3 ensure that all cut material is carried away from the cut sources. It is now necessary to ensure that all fill sections receive the required amounts of fill and this is done in a fashion similar to that above.

Consider fill location D which requires 2000 m^3 of material. Material can arrive at D from each of the three sources, A, B and C and the total amount arriving at D will be $(x_{AD} + x_{BD} + x_{CD})$. Thus to satisfy the fill requirement at D the following must hold

$$x_{AD} + x_{BD} + x_{CD} = 2000 \tag{2.4}$$

At fill location E, 6000 m^3 is required. The total amount of material arriving at E is represented by the sum of all variables having E as second subscript. For these quantities to balance it is necessary that

$$x_{AE} + x_{BE} + x_{CE} = 6000 \tag{2.5}$$

Similar equations represent the balance at F and G between the incoming material and the fill requirement

$$x_{AF} + x_{BF} + x_{CF} = 8000 \tag{2.6}$$

$$x_{AG} + x_{BG} + x_{CG} = 4000 \tag{2.7}$$

Equations 2.4 to 2.7 ensure that all material requirements at fill locations are met. Consequently any set of values of the 15 variables which satisfies all the equations 2.1 to 2.7 will represent a possible transportation schedule which will

www.EngineeringBooksPDF.com

level the site. These seven equations are, therefore, a complete mathematical specification of the restrictions on an acceptable transportation schedule.

A quick summation of the totals of cut material produced and fill material required shows that there is a surplus of $1000\,m^3$ cut material. This surplus must be conveyed to the dump H and equations 2.1 to 2.3 include variables x_{AH}, x_{BH} and x_{CH} to take care of this. It is not necessary to include an extra equation requiring dump H to receive a total of $1000\,m^3$ surplus material; equations 2.1 to 2.7 will implicitly satisfy this unwritten equation. Had the problem indicated a deficit of total cut material rather than a surplus then the equations would have been somewhat different. Variables x_{AH}, x_{BH} and x_{CH} would not have been needed since all available cut material would be required at the fill locations, so these variables would have been absent from equations 2.1 to 2.3. The need for extra fill must be met by conveying material from the dump H to the fill locations D, E, F and G. Thus variables x_{HD}, x_{HE}, x_{HF} and x_{HG} could have to be added to the left-hand sides of equations 2.4 to 2.7 respectively.

Since fifteen variables may satisfy the seven equations 2.1 to 2.7 in an infinite number of ways, there is an infinite number of possible transportation schedules for this problem. Just one of them, derived in a casual fashion is shown in Table 2.2.

Table 2.2 Quantities carried (m^3) between cut and fill locations

		Fill locations				
		D	E	F	G	H
Cut locations	A	1000	1000	1000	1000	1000
	B	1000	2000	3000	1000	0
	C	0	3000	4000	2000	0

Step three of the systematic approach requires the definition of a criterion for measuring the efficiency of a set of decisions. In this example the objective of the contractor is to move the earth around as cheaply as possible. Cost is, therefore, a suitable criterion for distinguishing the merit of a transportation schedule. An appropriate measure of cost here is the total operating costs of the vehicles. Since the costs of both the fuel used and a driver's time are dependent on the distance travelled, it can be assumed that the total cost of transporting a unit volume of material is proportional to the distance it is carried. Table 2.1 gives distances between all cut and fill locations and by using it a mathematical function representing a measure of the total cost of any transportation schedule can be derived. Since x_{ij} is the amount of material carried between source i and destination j, and if d_{ij} is the distance between i and j from table 2.1, the 'cost' of moving x_{ij} over d_{ij} will be $d_{ij}\,x_{ij}$. The total cost of an entire transportation

www.EngineeringBooksPDF.com

schedule will be found by summing $d_{ij}\,x_{ij}$ over all possible routes ij. For example, the total cost, C is

$$\begin{aligned} C = {} & 0.7x_{AD} + 0.3x_{AE} + 0.2x_{AF} + 0.4x_{AG} + 0.9x_{AH} + 0.6x_{BD} \\ & +0.5x_{BE} + 0.8x_{BF} + 0.3x_{BG} + 1.1x_{BH} + 0.3x_{CD} + 0.2x_{CE} \\ & +0.5x_{CF} + 0.7x_{CG} + 0.8x_{CH} \end{aligned} \tag{2.8}$$

Using the values for quantities x_{ij} given in the casually chosen transportation schedule of table 2.2, this schedule may be costed using equation 2.8 and it turns out to have a 'cost' of 10 800 km m^3 units. In order to get a real monetary cost it is necessary to specify a cost coefficient which multiplies equation 2.8 and represents the monetary cost of transporting 1 m^3 of material over 1 km. In the present example, however, this coefficient is not vital; the use of km m^3 units as a measure of cost is perfectly valid for comparing different schedules. A better schedule will always have a lower value of C from equation 2.8 regardless of the units used.

Having devised a schedule that will level the site (table 2.2) and having costed it, it is tempting at this stage either simply to go ahead and implement it, or better, to devise one or two further possible schedules, cost them and select the best for immediate implementation. This is all too frequently done but the full benefits of the mathematical modelling exercise are not felt until the systematic approach is pursued to its logical conclusion. Since it is easy to cost out different possible schedules it is logical to try to find the very best possible schedule, the one with an absolute lowest cost. This, the fourth step of the systematic approach, requires that the best possible decisions are made. In this problem the best decisions would be that set of values of the 15 variables which satisfy all the equations 2.1 to 2.7 and at the same time give a minimum value of the total cost function 2.8. Expressed more formally, if $\boldsymbol{x}$ is the vector of variables whose fifteen elements are the x_{ij}s, the problem is to

$$\begin{aligned} &\text{Minimise equation 2.8 over variables } \boldsymbol{x} \\ &\text{while satisfying equations } \begin{matrix} 2.1 \\ \vdots \\ 2.7 \end{matrix} \\ &\text{with all variables } x_{ij} \geqslant 0 \end{aligned} \tag{2.9}$$

Methods for solving problem 2.9 are considered in detail in chapter 3 but some features of the problem are worth noting here. Firstly, all the functions in equation 2.8 and in the left-hand sides of equations 2.1 to 2.7 are linear functions of the variables. For this reason problem 2.9 is classified as a linear programming (LP) problem. Secondly, if problem 2.9 is solved using the methods described in chapter 3 the best transportation schedule turns out to level the site at a cost of only 7200 km m^3 units, that is, the best schedule saves 33% per cent of the cost of the schedule shown in table 2.2. Thus it is economically very worthwhile pursuing the systematic approach to its logical conclusion and any method that

www.EngineeringBooksPDF.com

solves problem 2.9 clearly has value. Before examining solution methods, however, consider another type of practical problem.

EXAMPLE 2.2 – PRECASTING PLANT

A precasting plant produces one element in three different strength grades depending on the proportions of cement and aggregate used. The plant has three production lines and it is necessary to plan the future production of elements so that the company satisfies the expected demand for elements from its existing stores of raw materials and also maximises its profits on the operation.

All the elements produced weigh 500 kg but the amounts of cement, large aggregate and fine aggregate in each type vary according to the three strength specifications. Table 2.3 gives the constituents of each of the three element types.

Table 2.3 Weights (kg) of constituents of each element type

	Type of element		
	1	2	3
Cement	50	75	50
Large aggregate	300	250	275
Fine aggregate	150	175	175

The plant has stocks of raw materials as follows: 10 000 kg of cement costing 2.5 units of cost per kg, 40 000 kg of large aggregate costing 1.0 units/kg and 30 000 kg of fine aggregate costing 1.5 units/kg.

The three production lines are all different and can produce elements at different added costs. Table 2.4 shows which types of element may be made on each production line and the additional manufacturing cost per element.

Table 2.4 Production line details

	Element type produced	Extra cost units/element
Line 1	1	500
	2	500
Line 2	1	1000
	2	750
	3	500
Line 3	3	1000

www.EngineeringBooksPDF.com

The commercial demand is for at least 20 elements of each type. Each production line can produce a maximum of 80 elements before it must be shut down for maintenance purposes. The selling prices of the elements are as follows: 2000 units for type 1 elements, 1600 units for type 2 elements and 1750 units for type 3 elements.

At first sight this problem appears to be a bewildering mass of information with little order to it and it seems to bear no similarity whatsoever to the earthworks example. In order to establish some order to this problem the systematic approach may be used. The first stage is to determine what decisions have to be made and to assign variables to them. In order to start production of elements it is necessary to know how many elements of each type are to be produced. Furthermore it is necessary to know on which production line a particular element type is to be manufactured. The production plan, therefore, consists of a list of the numbers of each element type to be produced on each production line. If x_{ij} is the number of elements of type i produced on line j, the list of variables is then $x_{11}, x_{12}, x_{21}, x_{22}, x_{32}$ and x_{33}, that is, a total of six variables.

Stage two of the systematic approach requires that all the relationships among the variables imposed by the problem are examined. In this example they arise from several distinct considerations. Firstly it is specified that the commercial demand for at least 20 elements of each type must be met. Expressed mathematically this requires that

$$\text{Demand for type 1}, x_{11} + x_{12} \geqslant 20 \tag{2.10}$$

$$\text{Demand for type 2}, x_{21} + x_{22} \geqslant 20 \tag{2.11}$$

$$\text{Demand for type 3}, x_{32} + x_{33} \geqslant 20 \tag{2.12}$$

Secondly, it is necessary that the elements are produced from existing stocks of raw materials. Thus the total available quantities of cement, large and fine aggregates must not be exceeded. Each type of element uses a different quantity of cement shown in table 2.3 and this leads to a mathematical requirement that

$$50x_{11} + 50x_{12} + 75x_{21} + 75x_{22} + 50x_{32} + 50x_{33} \leqslant 10\,000 \tag{2.13}$$

A similar expression is obtained for large aggregate

$$300x_{11} + 300x_{12} + 250x_{21} + 250x_{22} + 275x_{32} + 275x_{33} \leqslant 40\,000 \tag{2.14}$$

and for fine aggregate

$$150x_{11} + 150x_{12} + 175x_{21} + 175x_{22} + 175x_{32} + 175x_{33} \leqslant 30\,000 \tag{2.15}$$

www.EngineeringBooksPDF.com

Finally, the mathematical model must include the fact that each production line can only produce 80 elements before it is shut down for maintenance. This leads to a restriction for each production line as follows

$$\text{For line } 1, x_{11} + x_{21} \leqslant 80 \tag{2.16}$$

$$\text{For line } 2, x_{12} + x_{22} + x_{32} \leqslant 80 \tag{2.17}$$

$$\text{For line } 3, x_{33} \leqslant 80 \tag{2.18}$$

The nine relationships 2.10 to 2.18 represent all the stated restrictions upon acceptable values of the six variables. In order to find a feasible production plan it is necessary only to find values for all the six variables which satisfy the relationships 2.10 to 2.18. As in the earthworks example 2.1, there are very many possible sets of values representing feasible production plans. The problem requires that the company profit be maximised so it is sensible to construct a profit function in order that the profit on any production plan may be easily assessed. To evaluate profit it is first necessary to calculate element costs which are of two types; materials costs and manufacturing costs. Consider first the materials cost per element of each type. For type 1 the constituents are given in table 2.3 which when combined with the unit costs of the materials gives a materials cost per element of type 1 as (50 x 2.5) + (300 x 1.0) + (150 x 1.5) = 650 units. For type 2 the materials cost per element is (75 x 2.5) + (250 x 1.0) + (175 x 1.5) = 700 units. Similarly for type 3 elements the cost of materials per element works out to be 662.5 units. To these material costs must be added the manufacturing costs given in table 2.4. A type 1 element may be produced either on line 1 at an extra cost of 500 units which, when added to the material cost of 650 units, gives a total cost of 1150 units or on line 2 at an extra cost of 1000 units giving a total cost of 1650 units. For type 2 elements the total costs are 1200 units per element if produced on line 1 and 1450 units per element if produced on line 2. For type 3 elements the total costs are 1162.5 units per element if produced on line 2 and 1662.5 units if produced on line 3. The selling prices of all units are given in the problem and this now enables a profit to be calculated for each element produced on each line as follows. If p_{ij} is the profit per element of type i produced on line j, then p_{11} is equal to the selling price of a type 1 element of 2000 units less the total cost price of a type 1 element produced on line 1 of 1150 units, that is, p_{11} = 850 units. Similarly p_{12} = 2000 - 1650 = 350 units, p_{21} = 1600 - 1200 = 400 units, p_{22} = 150 units, p_{32} = 587.5 units, p_{33} = 87.5 units. The total profit function, P, is now constructed by summing $p_{ij} \cdot x_{ij}$ over all i, j combinations, thus

$$P = 850.0x_{11} + 350.0x_{12} + 400.0x_{21} + 150.0x_{22} + 587.5x_{32} + 87.5x_{33} \tag{2.19}$$

Given any feasible production plan, that is, values of the variables satisfying inequalities 2.10 to 2.18, the profit to be expected from that plan may be

www.EngineeringBooksPDF.com

assessed from equation 2.19. As in the earthworks example a feasible production plan may be selected quite easily. For example

$$x_{11} = x_{12} = x_{21} = x_{22} = x_{32} = x_{33} = 20$$

satisfies all the inequalities 2.10 to 2.18. Substitution into equation 2.19 gives a profit of 48 500 units. Other possible plans may be costed similarly but a systematic approach should now entail finding the best possible plan which maximises the profit. In mathematical terms if $\boldsymbol{x}$ is the vector of variables whose six elements are the x_{ij}s the problem becomes

$$\begin{aligned} &\text{Maximise equation 2.19 over variables } \boldsymbol{x} \\ &\text{subject to non-violation of inequalities 2.10 to 2.18} \\ &\text{with all variables } x_{ij} \geqslant 0 \end{aligned} \tag{2.20}$$

It is seen in problem 2.20 that, as in the earthworks problem, all the functions in the left-hand sides of equations 2.1 to 2.18 and in the cost function 2.19 are again linear functions of $\boldsymbol{x}$. Thus problem 2.20 is also classified as a linear programming problem and any solution method for LP problems should be able to solve both examples so far studied. If problem 2.20 is solved it is found that the best production policy leads to an expected profit of 95 318 units, that is, an increase of around 96 per cent on that from the randomly selected policy. Once again the pursuit of the systematic approach to its logical conclusion shows great benefits and points to a need to study how to solve linear programming problems. A final example drawn from the design phase of a project demonstrates a further area of application of linear programming.

EXAMPLE 2.3 – RIGID-PLASTIC DESIGN OF FRAMEWORKS†

The portal frame shown in figure 2.1 must be designed in steel on a rigid-plastic basis to have a factor of safety of 2.0 against total collapse under the loading shown. The two columns are to be of identical section while the beam may have a different section.

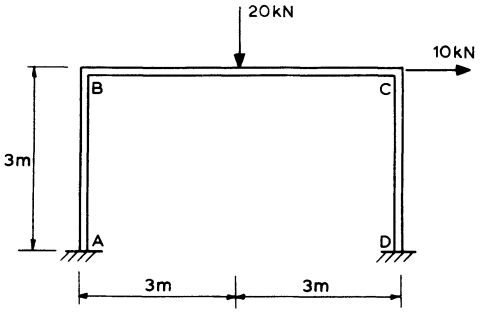


Figure 2.1

† An excellent treatment of minimum weight plastic design is to be found in B. G. Neal, *The Plastic Methods of Structural Analysis* (Science Paperbacks, 1965)

www.EngineeringBooksPDF.com

Since the member lengths are known, the design process consists of selecting appropriate member cross-sectional sizes. As a rigid-plastic design is required, an appropriate choice for a measure of the size of a member is its fully plastic moment. The designer has therefore to make decisions upon the fully plastic moment of the beam member, to which variable M_1 may be assigned, and of the column members, to which variable M_2 is allocated. Restrictions on the values that the two variables may take come from the requirements that the structure must have a factor of safety of 2.0 against collapse under the given loading. It is necessary to ensure that in any possible collapse mode of the frame the work done on the frame by the factored applied loads does not exceed the energy capacity of the plastic deformations (rotations at plastic hinges) of the frame. Figure 2.2 shows the six possible collapse mechanisms of the frame and the energy-balance requirement associated with each kinematic mechanism. There are three general failure mechanisms possible: a beam mechanism (a and b in figure 2.2), a sway mechanism (c and d) and a combined mechanism (e and f) in which both the beam and sway failure occur simultaneously. The reason for there being two mechanisms for each general type is that it is not yet known whether the beam is weaker than the column, in which case $M_1 < M_2$, or whether the column is weaker than the beam, in which case $M_1 > M_2$. The hinges at joints B and C of the frame will always occur in the weaker member at the joint since less energy will be needed to produce a plastic hinge in that member. Associated with each of the six possible collapse mechanisms in figure 2.2 is a relationship between the work done by the factored applied loads on the mechanism and the energy absorbed by the deformations. For example, consider the mechanism shown in figure 2.2b. The work done by the loads on that mechanism is all done by the 40 kN load (20 kN actual load x 2.0 safety factor). The horizontal load does no work because the frame does not deform horizontally in this mechanism. The work done by the 40 kN load is

$$\begin{aligned}\text{work done} &= 40\ \text{kN} \times 3\ \text{m} \times \theta \\ &= 120\theta\ \text{kN m}\end{aligned}$$

This assumes that deformations are small so that $\sin\theta \approx \theta$. The energy absorbed by the two hinges at the tops of the columns is equal to their fully plastic moments multiplied by the angular rotations at the hinges. Thus

$$\begin{aligned}\text{energy absorbed (column hinges)} &= 2 \times M_2 \times \theta \\ &= 2M_2\theta\ \text{kN m}\end{aligned}$$

Similarly for the hinge in the middle of the beam

$$\begin{aligned}\text{energy absorbed (beam hinge)} &= M_1 \times 2\theta \\ &= 2M_1\theta\ \text{kN m}\end{aligned}$$

In order that the frame shall either be safe or just collapse in this mechanism it is

www.EngineeringBooksPDF.com

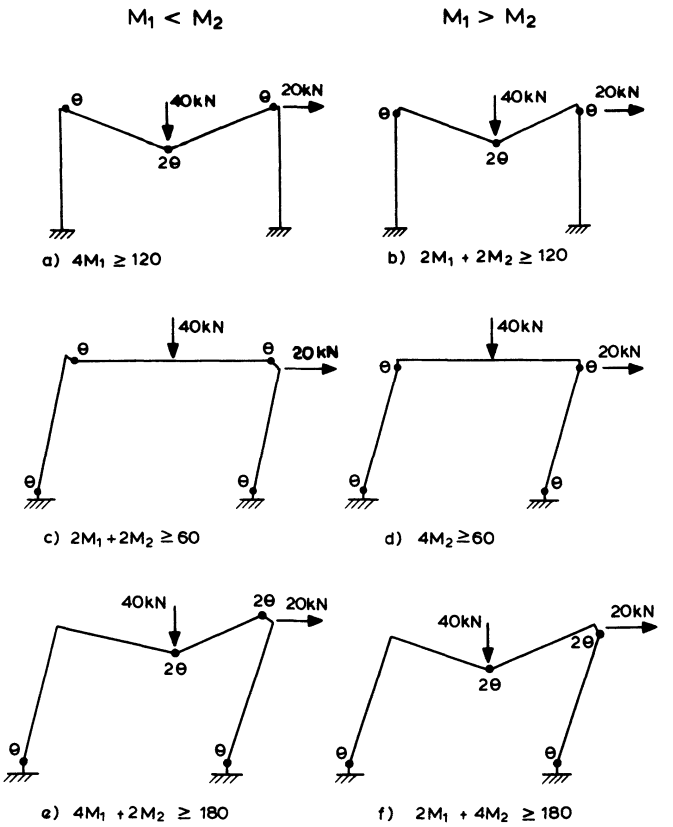


Figure 2.2 Possible collapse mechanism for the portal frame of figure 2.1

required that the total energy absorbed in all the hinges should be at least equal to the work done by the factored applied loads. Therefore

$$2M_1\theta + 2M_2\theta \geqslant 120\theta$$

and, cancelling the θs, this gives the requirement that

$$2M_1 + 2M_2 \geqslant 120$$

Similar relationships can be obtained from all the other possible collapse mechanisms and these are given under each mechanism in figure 2.2. These six relationships form a set of restrictions on the values that the designer can choose for M_1 and M_2. It is noted that they all have linear functions of the variables as their left-hand sides.

Since all the energy-balance mechanisms lead to inequalities of the form that the left-hand side $\geqslant$ a constant, a designer can always select a safe design by choosing high values for M_1 and M_2, that is, selecting large size members for the

www.EngineeringBooksPDF.com

frame. If this is done all the inequalities will be satisfied in a strictly > sense and the resulting design will have a factor of safety in excess of the required value of 2.0. Such a design, however, composed of unnecessarily large members, would be needlessly expensive. A major design criterion for frames such as this is cost and a logical design goal would be to find values for M_1 and M_2 that do not violate any of the six mechanism conditions and at the same time minimise a measure of the cost of the frame.

Since the cost of steel is largely proportional to the weight of steel and weight is proportional to volume, volume of steel used provides a useful measure of the cost of a frame. Lengths of the columns and beam are known so, if A_1, A_2 are the cross-sectional areas of the beam and columns respectively, the volume of the frame shown in figure 2.1 is

$$V = 6A_1 + 2 \times 3A_2 \text{ m}^3 \tag{2.21}$$

The design variables that have been selected, however, are M_1, M_2, the fully plastic moments and not A_1 and A_2. In order to remain consistent A_1 and A_2 must now be expressed as functions of M_1 and M_2. For British rolled-steel sections the relationship between the cross-sectional area of a beam or column member and its fully plastic moment is shown in figure 2.3 to be of the approximate form

$$A = CM^{0.6} \tag{2.22}$$

in which C is a constant. Furthermore, although equation 2.22 represents a smooth curve it only has very small curvature and figure 2.3 shows that the form

$$A = a + bM \tag{2.23}$$

in which a and b are suitably chosen constants, approximates equations 2.22

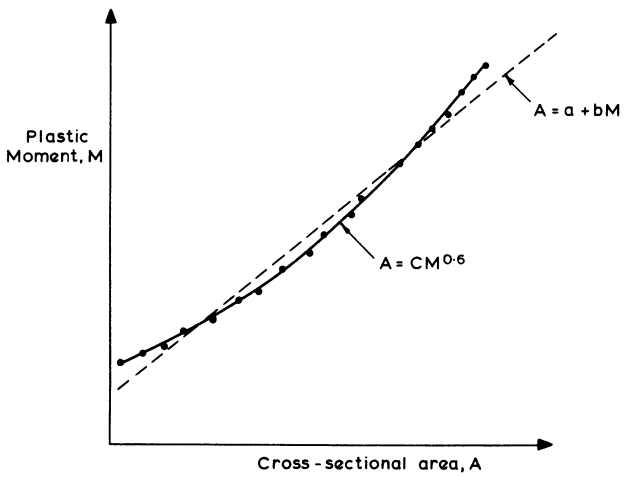


Figure 2.3 Relationships between M and A for rolled steel sections

www.EngineeringBooksPDF.com

very closely throughout the range of available sections. Thus if equation 2.23 is substituted into the volume function (2.21) the following is obtained

$$V = 6\,(a + bM_1) + 2 \times 3\,(a + bM_2)$$
$$V = 12a + b\,(6M_1 + 6M_2) \tag{2.24}$$

The element $12a$ in equation 2.24 is a constant as is the factor b. The only variable measure in equation 2.24 is the factor $(6M_1 + 6M_2)$. This therefore provides a measure of the cost of the frame which can be minimised. The entire optimum design process can then be expressed formally as

$$\left.\begin{array}{l}\text{Minimise } V' = 6M_1 + 6M_2 \text{ over variables } M_1, M_2 \\ \text{subject to non-violation of the inequalities} \\ \qquad \begin{array}{rl} 4M_1 & \geqslant 120 \\ 2M_1 + 2M_2 & \geqslant 120 \\ 2M_1 + 2M_2 & \geqslant 60 \\ 4M_2 & \geqslant 60 \\ 4M_1 + 2M_2 & \geqslant 180 \\ 2M_1 + 4M_2 & \geqslant 180 \\ M_1, M_2 & \geqslant 0 \end{array}\end{array}\right\} \tag{2.25}$$

Problem 2.25 is composed solely of linear functions of the variables M_1 and M_2 and so, like problems 2.9 and 2.20, can be classified as a linear programming problem. The differences among the three problems are quite small; one involves maximisation of a function, the other two are posed as minimisations. One problem involves equalities in the relationships among variables, the other two involve inequalities. These differences turn out to be very minor ones which are completely overwhelmed by the strong similarities among all three problems.

The three physical problems of earthmoving, planning the production of a precasting plant and the plastic design of a portal frame structure at first sight appear to be totally different and to have nothing in common. One might have been excused for believing that they were all 'one-off' problems, each requiring its own specific solution technique. The most interesting and important fact to have emerged from the application of a systematic analysis approach to all three problems is that they all have the same mathematical skeleton and are in fact closely similar. As will be seen shortly they may all be solved using only one solution method. Very many other practical problems could have been drawn from all aspects of civil engineering and presented here as further examples of how systematic analysis may be applied to yield linear programming problems similar in form to the three above. Further consideration of practical problems, however, must wait until the end of chapter 3. It is now necessary to turn attention to the mathematical aspects of linear programming. It is valuable to examine the detailed nature of linear programming problems in order to try to understand them and to develop logical solution methods which are of general applicability.

www.EngineeringBooksPDF.com

2.4 THE GENERAL LINEAR PROGRAMMING PROBLEM

The general mathematical statement of a linear programming problem is as follows

$$\left.\begin{aligned} &\text{Minimise } f = \sum_{i=1}^{N} a_i x_i \text{ over variables } x_i,\ i = 1, \ldots, N \\ &\text{subject to} \\ &g_j \equiv \sum_{i=1}^{N} b_{ji} x_i \geqslant c_j \qquad j = 1, \ldots, M_1 \\ &h_j \equiv \sum_{i=1}^{N} b_{ji} x_i \leqslant c_j \qquad j = M_1 + 1, \ldots, M \\ &\quad x_i \geqslant 0 \qquad\qquad i = 1, \ldots, N \end{aligned}\right\} \tag{2.26}$$

In problem 2.26 f is the linear function to be minimised over variables x_i, $i = 1, \ldots, N$ and the function f is often referred to as the *objective* function or *merit* function. Coefficients a_j, $i = 1, \ldots, N$ are known constants. Problem 2.26 is expressed solely as a minimisation problem. If a specific problem requires the maximisation of function f then it can immediately be transformed to one of the minimisation of a function f_1 which is equal to (-f). Figure 2.4 shows that the maximisation of a linear function $f = kx$ limited by the restrictions $x_L \leqslant x \leqslant x_u$ occurs at exactly the same point, $x = x_u$, as is obtained by minimising $f_1 = -kx$ with the same restrictions $x_L \leqslant x \leqslant x_u$.

There is thus no need to consider maximisation problems separately; the minimisation format of problem 2.26 also implies maximisation. The conditions

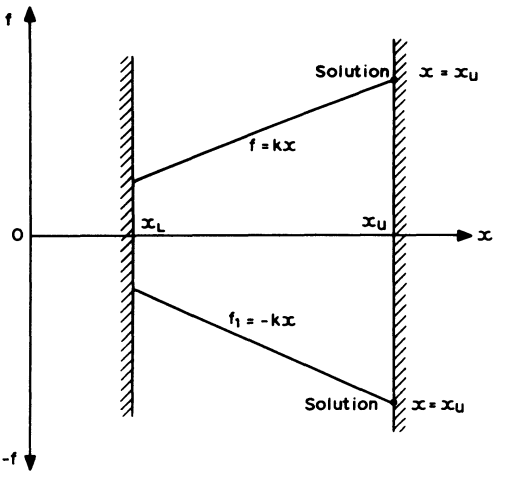


Figure 2.4 Maximisation of f is equivalent to minimisation of $f_1 = -f$

www.EngineeringBooksPDF.com

or restrictions which values of the variables must not violate are usually called *constraints* and in problem 2.26 there are M such constraints. Each constraint has a linear function of the variables at its left-hand side and the coefficients b_{ji} are again assumed to be known real constants, as are the right-hand sides of all constraints, $c_j, j = 1, \ldots, M$. It is conventional to arrange the constraints so that none of the c_js is negative. To this end the M constraints have been partitioned into two sets; the first set, g_j, contains M_1 inequalities all of which are of the $\geqslant$ type, the second set is labelled h_j and contains $(M - M_1)$ constraints all of which are $\leqslant$ inequalities. The final statement of problem 2.26 requires that values of the variables must always be non-negative, that is, positive or zero.

If the three example problems are compared with the general form of problem 2.26 it is seen that they all fit the general form very well. The only possible area for doubt lies with the earthworks problem constraints which are all equalities. The general form of problem 2.26 does not appear to cater for equality constraints. In fact, equality constraints form a special case of the inequalities in problem 2.26 and it will be shown later that they simplify the solution process. Consequently, problem 2.9 does fit the general form of problem 2.26.

Problem 2.26 is a general statement of linear programming in N dimensions; there are N variables x_i for which optimal values are sought. In order to shed light upon the nature of linear programming problems it is useful to examine a simple two-variable problem since this permits a graphical representation to be used. Problem 2.27 is a two-dimensional example which fits the general form of problem 2.26.

$$
\begin{aligned}
&\text{Minimise } f = 6x_1 + 4x_2 \\
&\text{subject to the constraints} \\
&\qquad g_1 \equiv 2x_1 + 2x_2 \geqslant 60 \\
&\qquad g_2 \equiv 2x_1 + 4x_2 \geqslant 80 \\
&\qquad g_3 \equiv 4x_1 \qquad\; \geqslant 60 \\
&\qquad g_4 \equiv \qquad 4x_2 \geqslant 20 \\
&\qquad h_5 \equiv 3x_1 + 2x_2 \leqslant 120 \\
&\qquad\qquad x_1, x_2 \quad \geqslant 0
\end{aligned}
\tag{2.27}
$$

Consider the constraints of problem 2.27. The non-negativity requirements, $x_1, x_2 \geqslant 0$ mean that any graphical representation of this problem must be confined to the first quadrant only. Consider constraint g_1. If this were written as an equality, $2x_1 + 2x_2 = 60$, this represents a straight line. Figure 2.5 shows the first quadrant with this equation represented as a solid straight line. Values of x_1 and x_2 that satisfy the equality $2x_1 + 2x_2 = 60$ lie on the straight line and are termed *feasible for* g_1 since they do not violate constraint g_1. Values of x_1 and x_2 that give to the function $2x_1 + 2x_2$ a value greater than 60 still satisfy the inequality constraint g_1. Such points lie on the side of the straight line opposite to the origin and are also termed feasible for g_1. Conversely, values of x_1 and x_2

www.EngineeringBooksPDF.com

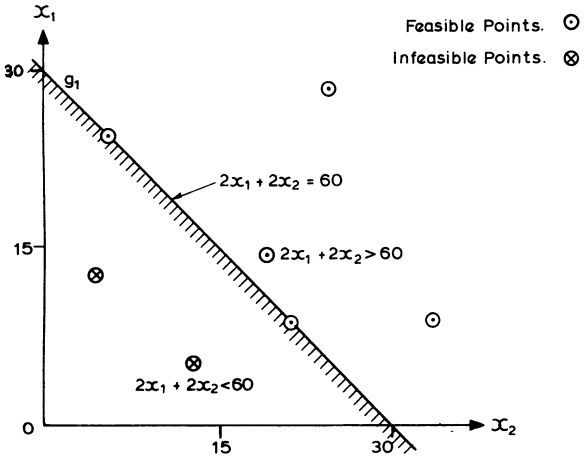


Figure 2.5

which give the function $2x_1 + 2x_2$ a value of less than 60 will lie on the origin side of the straight line. Since they violate the inequality such points are termed *infeasible for* g_1 and, to show that such points are not acceptable as solution points, the solid line has been shaded on the infeasible side in figure 2.5. If constraint g_3 is now added as shown in figure 2.6 it too marks out a feasible and an infeasible zone. Some points which were feasible for g_1 are now infeasible for g_3, thus the new constraint has cut down the size of the region which is feasible to both constraints. This process of adding constraints is extended to all constraints in problem 2.27 and figure 2.7 shows the resulting graph.

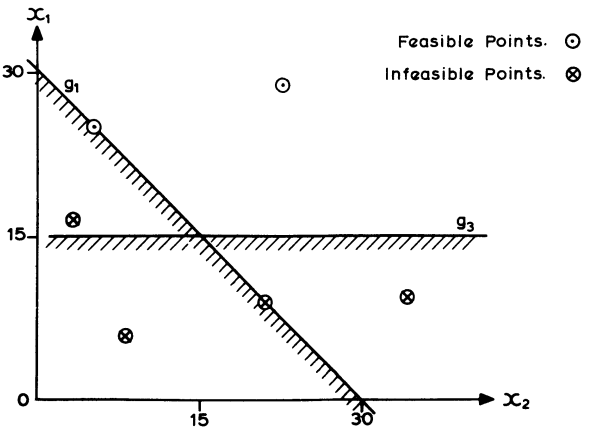


Figure 2.6

www.EngineeringBooksPDF.com

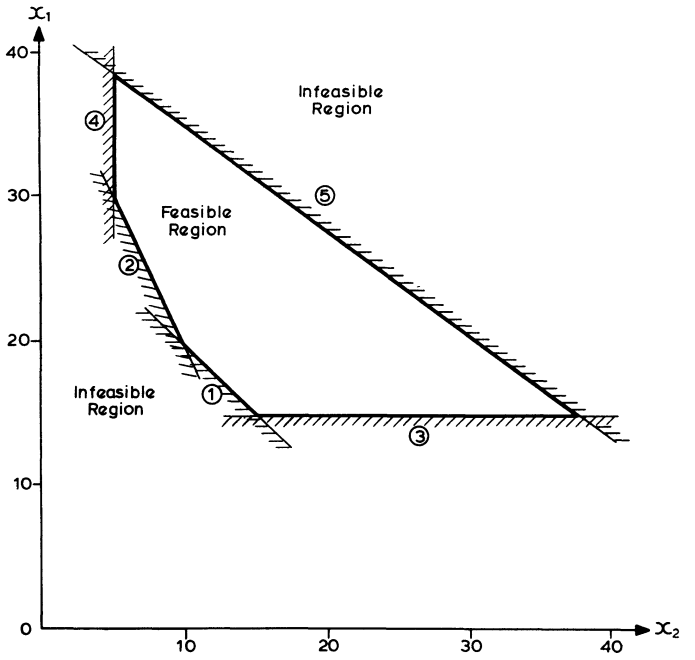


Figure 2.7 The constraint set of problem 2.27

The complete set of constraints marks out a *feasible region* and an *infeasible region.* A point within the feasible region or on its boundaries satisfies all the constraints of problem 2.27, is termed a *feasible point* and is acceptable as a possible solution point. A point that violates any one or more constraints, that is, is infeasible for at least one constraint, is termed an *infeasible point* and is not acceptable as a solution point since it lies somewhere within the shaded infeasible region. In the example shown in figure 2.7 the feasible region is *closed* since it is entirely closed in by boundaries. It is possible to have *open* feasible regions; if, for example, h_5 were omitted the feasible region would be open.

The solution point for the problem must lie within the feasible region or on its boundaries. The objective function, $f = 6x_1 + 4x_2$, is now used to locate the precise position of the solution point. The function $f = 6x_1 + 4x_2$ represents a plane and figure 2.8 shows the feasible region with straight line contours of the plane $f = 6x_1 + 4x_2$ superimposed upon it. This plane slopes downwards towards the origin and the arrow shows the general direction of reducing values of f. The problem demands the location of that point in the feasible region which gives f its lowest value. Clearly this must be the point X on figure 2.8. One mental visualisation of figure 2.8 consists of imagining the plane f to be a gentle downward slope and the boundaries of the feasible region are represented as fences

www.EngineeringBooksPDF.com

or walls on that slope. A ball released at the top of the slope, that is, somewhere along the wall h_5 will roll downhill until it reaches another wall or fence. It will then roll around the walls always in a downwards sense until it comes to rest at the lowest point, in the pocket formed by the intersection of walls g_1 and g_3, that is, at X. Point X is the solution of problem 2.27 and from figure 2.8 this point is $x_1^* = x_2^* = 15$. The value of f at this point is $f^* = 150$, where * denotes optimality.

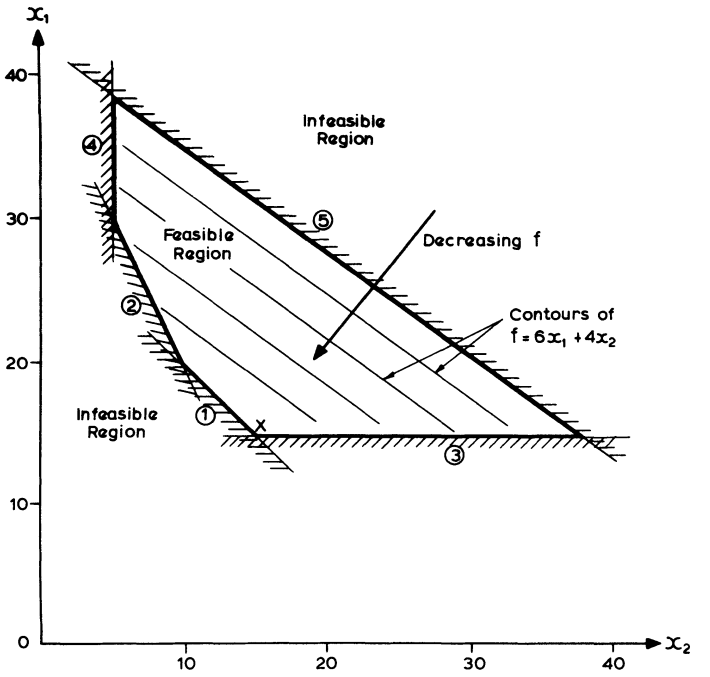


Figure 2.8 Graphical representation of problem 2.27.

The graphical solution method is, of course, only suitable for problems in two variables. For problems in more than two variables, numerical or algebraic solution methods are needed. The graphical method just examined, however, does illustrate some important features of linear programming problems and points the way towards quick numerical solution methods. The visualisation of a ball rolling down a sloping plane leads to the logical conclusion that unless the ball is brought to rest against a boundary it will roll on, theoretically, forever. Thus if a real, finite solution is to be obtained for an LP problem that solution point will lie on the boundary of the feasible region. Furthermore, in the example the ball came to rest at a junction of two boundaries, that is, at a *vertex* of the constraints. The only alternative possiblity is shown in figure 2.9 depicting a two-

www.EngineeringBooksPDF.com

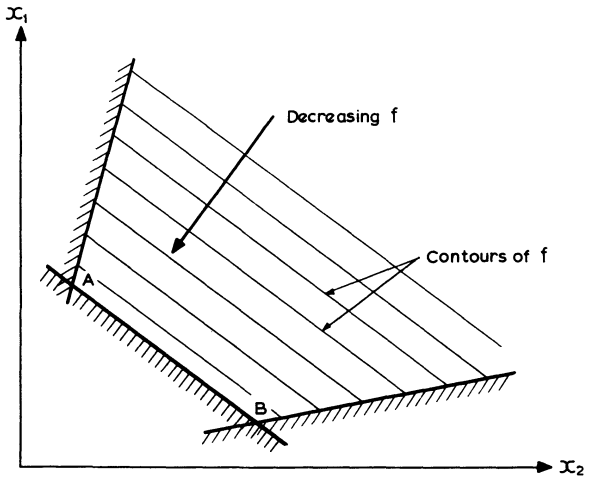


Figure 2.9 All points on constraint boundary AB are optimal

variable problem in which the lowest constraint boundary runs parallel to the objective function contours. Here any point between A and B is a solution point since it is feasible and gives the lowest value to f. The two ends of the constraint, however, vertices A and B, are just as good as any other point between them so it is still valid to say that a solution will be found at a vertex of the constraints. This principle that has been deduced from a two-variable example is applicable to problems in N variables also and is so important for general solution methods as to warrant repetition as a general principle.

A solution for a linear programming problem will always be found at a constraint vertex.

In order to devise a numerical solution procedure for general LP problems in N variables the above principle may be used. Since constraint vertices are all candidate solution points a possible solution method would be to locate all possible constraint vertices for a problem and to select that one which is feasible and gives the lowest value of the objective function f.

In the two-variable example, figure 2.8, a constraint vertex is formed by the intersection of two constraint boundaries. It has already been shown that constraint boundaries represent constraints satisfied as equalities. Thus in problem 2.27 the vertex X is numerically found by solving the two constraints g_1 and g_3 written as equalities, that is, point X has co-ordinates (x_1, x_2) given by the solution of

$$\left.\begin{aligned} 2x_1 + 2x_2 &= 60 \\ 4x_1 \quad\quad &= 60 \end{aligned}\right\}$$

www.EngineeringBooksPDF.com

For a *two*-variable problem a constraint vertex is found by solving any *two* constraints as simultaneous equations. For a *three*-variable LP problem it requires three co-ordinates to specify a point so a constraint vertex requires *three* constraints to be solved as simultaneous equations. Similarly for an N-variable LP problem a constraint vertex is found by solving N simultaneous constraint equations in N unknown variables.

In the two-variable example, problem 2.27, there were seven constraints in total (g_1 to g_4, h_5 and the two non-negativity requirements $x_1 \geqslant 0$ and $x_2 \geqslant 0$). In order to locate all possible vertices of this constraint set, it is necessary to examine all possible ways in which two constraints may be selected from the total of seven and solved as simultaneous equations. The number of ways in which two may be selected from seven is

$$ {}_7C_2 = \frac{7!}{2!\,(7-2)!} = \frac{7!}{2!\,5!} = 21 $$

So the example problem has 21 possible vertices, each of which is evaluated by solving two simultaneous equations. This presents quite a formidable task if all possible vertices are to be calculated. Furthermore, figure 2.8 shows that only five of these twenty-one vertices are feasible points, the remaining sixteen corresponding to infeasible points such as $x_1 = 15, x_2 = 12.5$, the vertex formed by constraints g_2 and g_3. If all twenty-one possible vertices are evaluated the next step must be to check whether each vertex is feasible or infeasible. This requires that each vertex is substituted into the constraints of problem 2.27 and if the vertex point violates any of the seven constraints it must be discarded as infeasible. This process of checking each of the twenty-one vertices against the set of seven constraints and retaining only the feasible vertices is also quite a tedious task. Since only five out of twenty-one vertices are feasible this means that more than 75 per cent of the calculations made so far in evaluating and checking vertices are wasted. The final task is a fairly simple one of evaluating the objective function at each of the feasible vertices and selecting as a solution point the vertex with the lowest value of f.

The numerical procedure described above is quite practicable but very laborious. A very high proportion of the calculations made are unproductive and are eventually discarded. Even for a two-variable problem the process is a lengthy one. For problems in more than two variables the process is even longer. The precasting plant, example 2.2, had six variables and a total of fifteen constraints including non-negativity requirements. The total number of constraint vertices is then

$$ {}_{15}C_6 = \frac{15!}{6!\,9!} = 5005 $$

To evaluate each of these 5005 vertices requires the solution of six simultaneous linear equations in six unknowns. This is clearly a major task as is the equally lengthy process of checking each vertex against the full constraint set to deter-

www.EngineeringBooksPDF.com

mine its feasibility. Once again we could expect to have to discard something over 75 per cent of all the vertices before finally finding the solution of the problem.

The larger the problem is, the more impossible it becomes to use a solution procedure based on locating all constraint vertices. A better method is required and fortunately one exists – the *simplex method* – which has found universal favour. The simplex method is also based on the use of constraint vertices but instead of locating and testing all the vertices the method first of all concentrates on finding any *feasible* vertex. Once a feasible vertex has been found the simplex method moves to another feasible vertex in a direction in which the objective function is decreased. This process continues until no new feasible vertex can be found with a lower value of f. The simplex method is very rapid because it calculates very little information which is later discarded. The method is discussed in detail in chapter 3.

SUMMARY

This chapter has shown that a systematic approach greatly simplifies the difficulties of constructing precise mathematical models of seemingly amorphous practical civil engineering problems. Three widely different problems have shown a common mathematical skeleton composed only of linear functions. Logical pursuance of the systematic approach to its conclusion showed considerable economic benefits in each example and pointed out the need to be able to solve linear programming problems. The nature of LP problems was examined by means of a simple graphical example and a method based on locating constraint vertices was used to demonstrate the advantages of the simplex method to be described in the next chapter.

www.EngineeringBooksPDF.com

3 SOLUTION TECHNIQUES FOR LINEAR PROBLEMS

Chapter 2 has established the general form of linear programming problems and has given some insight into the philosophy of their solution. This chapter deals in detail with the most widely used solution method for linear programming problems, the simplex method. Having presented the simplex method, consideration is given to some of the interesting features of linear programming such as duality, sensitivity analysis, problems with integer or discrete variables, etc.

Finally, towards the end of the chapter attention is redirected towards practical engineering problems and an attempt is made to classify those areas of civil engineering decision-making that naturally give rise to linear programming problems. Some guidelines are given on those areas of civil engineering practice in which linear programming is of most use as a decision-making aid.

3.1 THE SIMPLEX METHOD FOR LINEAR PROGRAMMING PROBLEMS

The simplex method solves LP problems that can be expressed in the general form of problem 2.26 established in chapter 2. For convenience this general form is restated below.

$$\left.\begin{aligned} &\text{Minimise} \quad f = \sum_{i=1}^{N} a_i x_i \qquad \text{over variables } x_i,\ i = 1, \ldots, N \\ &\text{subject to } g_j \equiv \sum_{i=1}^{N} b_{ji} x_i \geqslant c_j \qquad j = 1, \ldots, M_1 \\ &\qquad h_j \equiv \sum_{i=1}^{N} b_{ji} x_i \leqslant c_j \qquad j = M_1 + 1, \ldots, M \\ &\qquad x_i \geqslant 0 \qquad\qquad\qquad\qquad i = 1, \ldots, N \end{aligned}\right\} \tag{3.1}$$

www.EngineeringBooksPDF.com

In order to demonstrate how the simplex method works in solving problem 3.1 it is again convenient to use the same two-variable example, problem 2.27 of chapter 2, that is

$$\left.\begin{array}{lll} \text{Minimise} & f = 6x_1 + 4x_2 & \\ \text{subject to} & g_1 \equiv 2x_1 + 2x_2 \geqslant 60 & \\ & g_2 \equiv 2x_1 + 4x_2 \geqslant 80 & \\ & g_3 \equiv 4x_1 \qquad \geqslant 60 & \\ & g_4 \equiv \qquad 4x_2 \geqslant 20 & \\ & h_5 \equiv 3x_1 + 2x_2 \leqslant 120 & \\ & x_1, x_2 \geqslant 0 & \end{array}\right\} \tag{3.2}$$

The graphical solution of this problem is shown in figure 2.8.

3.1.1 Preparatory Work – The Simplex Table

The simplex method starts by introducing extra variables which seemingly increase the dimensionality of the problem and make it more complicated. Each constraint that is not an equality is converted into an equality constraint by adding to or subtracting from the left-hand side of the constraint a new non-negative variable, different for each constraint. In problem 3.1 the M_1 constraints g_j each require the left-hand side to be greater than or equal to the right-hand side. To make the left-hand side equal to the right-hand side the left-hand side must be reduced, so a non-negative variable is *subtracted from* the left-hand side. The $(M - M_1)$ constraints h_j all have their left-hand sides less than or equal to their right-hand sides so to convert these into equalities a new non-negative variable must be *added to* each left-hand side. The general form of the modified constraint set is then

$$\left.\begin{array}{ll} g_j \equiv \sum_{i=1}^{N} b_{ji}x_i - x_{N+j} = c_j & j = 1, \ldots, M_1 \\ h_j \equiv \sum_{i=1}^{N} b_{ji}x_i + x_{N+j} = c_j & j = M_1 + 1, \ldots, M \\ x_i \geqslant 0 & i = 1, \ldots, N + M \end{array}\right\} \tag{3.3}$$

The constraint set for the numerical example corresponding to set 3.3 is

$$\left.\begin{array}{lll} g_1 \equiv 2x_1 + 2x_2 - x_3 & = 60 \\ g_2 \equiv 2x_1 + 4x_2 - x_4 & = 80 \\ g_3 \equiv 4x_1 - x_5 & = 60 \\ g_4 \equiv 4x_2 - x_6 & = 20 \\ h_5 \equiv 3x_1 + 2x_2 + x_7 & = 120 \\ x_i \geqslant 0 \quad i = 1, \ldots, 7 & \end{array}\right\} \tag{3.4}$$

www.EngineeringBooksPDF.com

The variables that have been added or subtracted in constraint sets 3.3 and 3.4 are usually called *slack variables* because they measure the amount of *slack* in a constraint. If an inequality constraint is actually satisfied as an equality then the constraint is said to be *active* or *tight*. The slack variables measure how much needs to be added to or subtracted from a constraint in order to make it tight. In graphical terms a slack variable gives a very rough measure of how far a point is away from a constraint boundary. Figure 3.1 shows just two constraints g_1 and g_3 from the numerical example. Slack variable x_3 is associated with constraint g_1 and x_5 with constraint g_3. Figure 3.1 shows how values of these slack variables change depending upon the position of a point. It should be noted that an infeasible point automatically leads to a negative value for at least one slack variable. Consequently, by ensuring that all variables (including slack variables) take only non-negative values, the feasibility of all points is guaranteed.

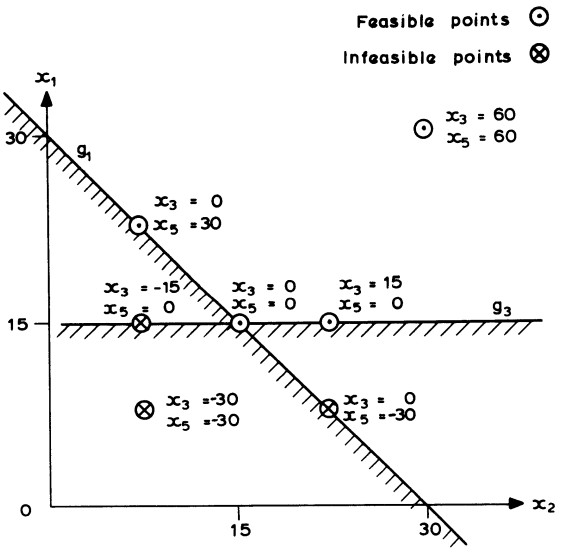


Figure 3.1 A graphical visualisation of slack variables

Although the inclusion of slack variables has increased the number of variables in the problem and hence has increased its algebraic complexity, it simplifies the solution by ensuring that only feasible points are considered.

It may be noted here that the constraints g_j, h_j have now been converted to equalities in set 3.3. In the earthworks example of chapter 2 all the constraints were naturally equalities and a measure of doubt arose as to whether they fitted the general LP form of problem 3.1. It is now seen that any inequalities in a problem are first of all converted to equalities in the simplex method so the fears

www.EngineeringBooksPDF.com

were groundless. In the earthworks problem no slack variables need to be added to or subtracted from the constraints which are already equalities.

The next stage of the simplex method is also a preparatory one. From each equality constraint in set 3.3 a variable is selected (a different one for each constraint) and is expressed as a linear function of the other variables. Usually the slack variables are selected as this simplifies the process. Constraint set 3.3 then becomes

$$\left.\begin{aligned} g_j \equiv x_{N+j} &= -c_j + \sum_{i=1}^{N} b_{ji}x_i \quad j = 1, \ldots, M_1 \\ h_j \equiv x_{N+j} &= c_j - \sum_{i=1}^{N} b_{ji}x_i \quad j = M_1 + 1, \ldots, M \\ x_i &\geqslant 0 \quad i = 1, \ldots, N+M \end{aligned}\right\} \tag{3.5}$$

For the numerical example this is a very simple transposition and constraint set 3.4 becomes

$$\left.\begin{aligned} g_1 \equiv x_3 &= -\ 60 + 2x_1 + 2x_2 \\ g_2 \equiv x_4 &= -\ 80 + 2x_1 + 4x_2 \\ g_3 \equiv x_5 &= -\ 60 + 4x_1 \\ g_4 \equiv x_6 &= -\ 20 \qquad + 4x_2 \\ h_5 \equiv x_7 &= \ 120 - 3x_1 - 2x_2 \\ x_i &\geqslant 0 \quad i = 1, \ldots, 7 \end{aligned}\right\} \tag{3.6}$$

This operation has effectively partitioned the variables into two types: those that appear only in the right-hand sides of the constraints and those that appear only in the left-hand side. The left-hand side variables are dependent for their values on the values that are chosen for the right-hand side variables. The names *decision variables* or *independent variables* are often given to the right-hand side variables, (x_1 and x_2 in constraint set 3.6), because their values may be freely chosen. The left-hand side variables are called *state variables* or *dependent variables* because their values or states are dependent on the values given to the right-hand side variables. At this stage the objective function f is introduced. Formally, f must be expressed as a minimisation problem, thus for maximisation problems all the coefficients in f must be multiplied by -1. Function f must be expressed as a function of the independent variables only (right-hand side variables).

With f added to the constraint set the problem can be written in matrix form as below, corresponding to sets 3.5 and 3.6

$$\left.\begin{aligned} [x_s] &= [c] + [b]\ [x_D] \\ \text{Min } f &= \qquad [a]\ [x_D] \end{aligned}\right\} \tag{3.7}$$

where $[x_s]$, $[x_D]$ are column vectors containing the state and decision variables respectively and $[a]$, $[b]$, $[c]$ contain the known coefficients of the problem.

www.EngineeringBooksPDF.com

This matrix form 3.7 is very convenient for use in computer programs for LP problems but for hand calculation it is often written out in tabular form. For the general algebraic problem the tabular form is as shown in table 3.1 with table 3.2 showing the equivalent table for the numerical example. Across the top of the table are the names of the decision variables. Down the left-hand side of the table are the names of the state variables plus the objective function name f. The coefficient matrices linking the two sets of variables go into the body of the table.

Table 3.1 Tabular form for the simplex method

		x_1	x_2		x_N
x_{N+1}	$-c_1$	b_{11}	b_{12}	...	b_{1N}
	⋮	⋮	⋮		⋮
x_{N+M_1}	$-c_{M_1}$	$b_{M_1 1}$	$b_{M_1 2}$	...	$b_{M_1 N}$
x_{N+M_1+1}	c_{M_1+1}	$-b_{M_1+1,1}$	$-b_{M_1+1,2}$		$-b_{M_1+1,N}$
	⋮	⋮	⋮		⋮
x_{N+M}	c_M	$-b_{M1}$	$-b_{M2}$		$-b_{MN}$
f		a_1	a_2	...	a_N

Table 3.2 First simplex table for numerical problem 3.2

		x_1	x_2	
x_3	- 60	2	2	25
x_4	- 80	2	4	30
x_5	- 60	4		60
x_6	- 20		4	30
x_7	120	-3	-2	5
f		6	4	230
		30	12.5	

Having set up a simplex table such as table 3.1 or table 3.2 the preparatory work is almost complete. In order to start the method any feasible point for the original problem must be known. In the case of the numerical example a feasible point can be selected quite simply as any point, say (30, 12.5), within the feasible region shown on figure 2.8. The values of these decision variables, $x_1 = 30$,

www.EngineeringBooksPDF.com

$x_2 = 12.5$, are placed on the simplex table 3.2 at the foot of the table under the corresponding variable name. Using these values of the decision variables the corresponding values of all the state variables, $x_3, \ldots, x_7$, and f may be calculated by a matrix multiplication operation. Thus the value of x_3 is

$$x_3 = -60 + 2 \times 30 + 2 \times 12.5 = 25$$

and the value of f at (30, 12.5) is

$$f = 6 \times 30 + 4 \times 12.5 = 230$$

These calculated values of $x_3, \ldots, x_7$ and f are placed on the right-hand side of the table on the row of the corresponding name. A check on the feasibility of the starting point (30, 12.5), is afforded by recalling that a feasible point is one that is itself non-negative and gives non-negative values to all dependent state variables. Thus the values of *all variables* around the edges of the table should be non-negative. In this numerical example a feasible point was easy to specify. Sometimes, particularly in much larger problems, it can be hard to find a feasible point at which to start the simplex method. A means of finding feasible points is described in section 3.1.6 of this chapter. For the present, however, assume that one is known.

3.1.2 Using the Simplex Table

The value of the objective function f at the selected feasible point is known, ($f = 230$ in the numerical example). The simplex table is used to reduce this value of f whilst still maintaining non-negative values for all decision and state variables. The value of f can only be altered by changing the value of one or more of the variables in the decision set, $x_1, \ldots, x_N$ in table 3.1, (x_1, x_2 in table 3.2). The coefficients $a_1, \ldots, a_N$ determine by how much f will change as $x_1, \ldots, x_N$ are changed. If the value of variable x_i is increased by δ then f will alter by $(a_i \times \delta)$. In the simplex method it is conventional and much simpler if only one decision variable is altered in value in each table. Which one should be chosen? As f is to be minimised it is logical to select a variable to be altered from among those that correspond to a negative coefficient, a. Then if that variable is increased by δ, f will change by $(a \times \delta)$, that is f will decrease since a is negative. If several decision variables have negative coefficients, a, in f, then it is sensible to select as the variable to change, the one that has the largest negative coefficient, a, since f will then decrease by the largest amount for a unit increase in the variable.

Turning to the numerical example in table 3.2 it is evident that neither of the objective function coefficients $a_1 = 6, a_2 = 4$, is negative. Thus neither x_1 nor x_2 may be increased from their present values of 30 and 12.5 respectively since such an increase would only increase the value of f above 230. If all the coefficients $a_1, \ldots, a_N$ are positive then f may be decreased only by decreasing the value of a decision variable. If all the decision variables $x_1, \ldots, x_N$ had the value

www.EngineeringBooksPDF.com

zero then clearly none of them could be decreased as this would violate the non-negativity requirement. In this circumstance the solution process would be terminated as the final solution of the LP problem would have been reached. If, however, with all coefficients $a_1, \ldots, a_N$ positive there are decision variables with positive values, then if any of these variables is decreased in value, f will also decrease. Again it is sensible to choose as the variable with a positive value to be decreased, the one corresponding to the largest positive coefficient, a, since this will cause the largest unit decrease of f.

In table 3.2 both the coefficients a_1 and a_2 are positive and furthermore both corresponding decision variables have positive values. Thus either x_1 or x_2 may be decreased in value and the logical choice is x_1 since unit decrease in x_1 causes f to reduce by 6 whereas a similar unit decrease in x_2 reduces f by only 4. Thus x_1 is chosen as the variable to be reduced in value.

By how much should x_1 be reduced? The lower limit on the value of x_1 imposed by non-negativity is $x_1 = 0$ and this gives a maximum potential reduction of x_1 of 30. Reducing x_1, however, not only causes f to reduce but also causes changes in the state variables $x_3, \ldots, x_7$. Those state variables with a positive coefficient, b, of x_1, that is, x_3, x_4, x_5, will reduce in value as x_1 is reduced and further restrictions on the decrease of x_1 will be imposed by the need to keep all state variables non-negative. State variable x_3 will be driven down to a value of zero by a reduction in the value of x_1 of 12.5 since

$$12.5 = \frac{\text{present value of } x_3, (25)}{\text{coefficient of } x_1 \text{ in } x_3, (2)}$$

Thus x_1 may not be reduced by more than 12.5. State variable x_4 is driven to zero by a reduction of

$$\frac{\text{present value of } x_4}{\text{coefficient of } x_1 \text{ in } x_4} = \frac{30}{2} = 15$$

in x_1. This reduction, however, could not occur since the reduction in x_1 has already been limited to 12.5 from consideration of state variable x_3. State variable x_5 reaches zero when x_1 is reduced by

$$\frac{\text{present value of } x_5}{\text{coefficient of } x_1 \text{ in } x_5} = \frac{60}{4} = 15$$

Thus x_5 imposes no new restriction on the amount by which x_1 may be decreased. Variable x_1 is absent from the equation defining state variable x_6 so whatever is done to x_1 will not affect x_6. Variable x_7 has a negative coefficient which indicates that the value of x_7 will increase as x_1 is reduced and so x_7 cannot possibly become negative. It is therefore concluded that the maximum amount by which x_1 may be reduced is 12.5. This will give a new value to x_1 of (30 - 12.5) = 17.5. Table 3.3 shows table 3.2 with the new point (17.5, 12.5) added below the first point (30, 12.5).

www.EngineeringBooksPDF.com

Table 3.3 Table 3.2 after the first simplex operation

		x_1	x_2		
x_3	- 60	(2)	2	25	0
x_4	- 80	2	4	30	5
x_5	- 60	4		60	10
x_6	- 20		4	30	30
x_7	120	-3	-2	5	42.5
f		6	4	230	155
		30	12.5		
		17.5	12.5		

Values of all of the state variables $x_3, \ldots, x_7$ and f are calculated at the new point and are shown as the new right-hand column. It is immediately seen that f has considerably reduced to a value of 155 and that all variables remain non-negative. The value of x_3 is zero. x_3 is the slack variable associated with constraint g_1 and its zero value indicates that g_1 is satisfied as an equality by the new point (17.5, 12.5). Figure 3.2 shows graphically what has just been done numeri-

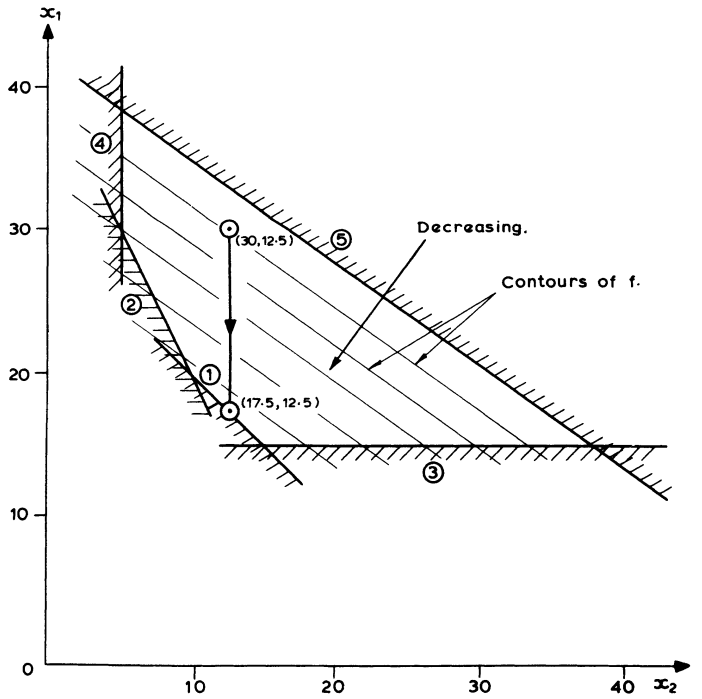


Figure 3.2 Graphical representation of the first simplex operation

www.EngineeringBooksPDF.com

cally. Starting from the point (30, 12.5) within the feasible region the value of x_1 has been altered causing movement along a line parallel to the x_1-axis in such a direction as to decrease f until the boundary formed by constraint g_1 is encountered at the point (17.5, 12.5).

Examining table 3.3 again, having changed the value of decision variable x_1, it is tempting now to try to alter decision variable x_2 and perhaps to reduce f further. In order to reduce f by changing x_2, the value of x_2 must be reduced. If x_2 is decreased in value, however, state variable x_3 will also be decreased below its present value of zero. As this is not permitted, x_2 cannot be altered. The presence of x_3 with a value of zero among the state variables clearly presents a barrier to further progress. The simplex method now requires that variable x_3 is removed from the state set of variables where it is presenting such a nuisance and that it is replaced by another variable from the decision set that has a strictly positive value. This operation is known as *pivoting*.

3.1.3 Pivoting

When any state variable reaches the value zero it should be removed from the state set and exchanged with a variable from the decision set that has a strictly positive value. Usually the decision variable chosen for this exchange is that variable which, in the previous simplex operation, was altered so as to drive the state variable to zero. In table 3.3 the coefficient 2 is encircled since this was the key coefficient by means of which decision variable x_1 drove state variable x_3 to zero. Now x_3 must leave the state set and become a decision variable while x_1 does the opposite.

The actual pivoting operation could be demonstrated on the general algebraic problem, table 3.1, but the algebra is not attractive. A better, more understandable demonstration of pivoting is afforded by the numerical example. Referring to table 3.3, we need to swap the positions in the table of variables x_1 and x_3. The equation linking them is given by the first row of the table, that is

$$x_3 = -60 + 2x_1 + 2x_2$$

A simple algebraic operation switches x_1 to the left-hand side where it becomes a dependent variable and x_3 to the right-hand side where it becomes a decision variable

$$x_1 = 30 + \tfrac{1}{2}x_3 - x_2 \tag{3.8}$$

The relationship 3.8 can be entered as the x_1 row of a new simplex table, table 3.4, in which the positions of variable names x_1 and x_3 are switched.

State variables $x_4, \ldots, x_7$ must be altered to reflect the fact that x_1 is no longer a decision variable. From table 3.3

$$x_4 = -80 + 2x_1 + 4x_2$$

www.EngineeringBooksPDF.com

Table 3.4 Second simplex table – table 3.3 after pivoting

		x_3	x_2	
x_1	30	$\frac{1}{2}$	-1	17.5
x_4	- 20	1	2	5
x_5	60	2	-4	10
x_6	- 20		4	30
x_7	30	$-\frac{3}{2}$	1	42.5
f	180	3	-2	155
		0	12.5	

x_1 from equation 3.8 must now be substituted into this equation in order to express x_4 as a function of decision variables x_3 and x_2, that is

$$x_4 = -80 + 2\,(30 + \tfrac{1}{2}x_3 - x_2) + 4x_2$$

$$x_4 = -20 + x_3 + 2x_2$$

This is inserted as the second row of table 3.4. A similar operation gives the remaining state variables x_5, x_6, x_7 in terms of the new decision variables x_3 and x_2. Thus the body of table 3.4 is filled in. A new form for the objective function, f, is constructed in similar fashion; thus from table 3.3

$$f = 6x_1 + 4x_2$$

substitute equation 3.8

$$f = 6\,(30 + \tfrac{1}{2}x_3 - x_2) + 4x_2$$

therefore

$$f = 180 + 3x_3 - 2x_2$$

The present values of the decision variables, $x_3 = 0, x_2 = 12.5$ are now entered at the foot of the table and, using these values, values of all the state variables, $x_1, x_4, \ldots, x_7$ and f are calculated and entered as the right-hand column. The values of *all* variables and f should, of course, be the same as in table 3.3. This affords a partial check on the algebraic accuracy of the pivoting operation. Table 3.4 is now complete and can be used for a second simplex operation in an exactly similar manner to that adopted for table 3.2. The pivoting operation on a simplex table is somewhat laborious in hand calculation and presents scope for algebraic errors. Although it has been presented here as a numerical operation there is a general algebraic equivalent and the whole pivoting operation can be performed quickly and accurately with the aid of a short computer subroutine.

www.EngineeringBooksPDF.com

3.1.4 Numerical Example – Solution

It is now possible to complete the solution of the numerical example in problem 3.2. Simplex and pivoting operations are repeated exactly as has already been outlined and a brief description of the solution follows.

In table 3.4 an inspection is made for any negative coefficients, a, in the objective function f, since increasing the corresponding decision variable will cause f to decrease. One such negative coefficient is present: that of -2 for decision variable x_2. Thus x_2 is selected as the decision variable to be increased. Decision variable x_3 remains zero. A limit on the increase in value of x_2 is provided by non-negativity requirements for the state variables. Only state variables x_1 and x_5 decrease in value as x_1 increases, by virtue of having negative coefficients (-1 and -4). The limit upon the increase of x_1 is provided by variable x_5 whose present value of 10 is reduced to zero as x_1 increases by 2.5. Thus x_2 is increased by 2.5 to the value of 15. Table 3.5 shows the second simplex table at the new point $x_3 = 0, x_2 = 15$, with corresponding values of the state variables and f.

Table 3.5 Second simplex table after alteration of value of x_2

		x_3	x_2		
x_1	30	$\frac{1}{2}$	-1	17.5	15
x_4	- 20	1	2	5	10
x_5	60	2	(-4)	10	0
x_6	- 20		4	30	40
x_7	30	$-\frac{3}{2}$	1	42.5	45
f	180	3	-2	155	150
		0	12.5		
		0	15		

This table now represents the solution point. The values of $x_1 = 15, x_2 = 15$ and $f = 150$ is the solution previously obtained graphically (figure 2.8). Two slack variables, x_3 and x_5 both have the value zero. x_3 and x_5 are each associated with constraints g_1 and g_3 respectively, thus these two constraints must be simultaneously satisfied as equalities. This represents the vertex X on figure 2.8.

Although in this example it is easy to recognise a known solution point, there is nothing obvious in table 3.5 which allows the termination of the simplex operations. The simplex method itself has not yet terminated and the calculations must be continued until no further calculation is possible. State variable x_5 has been driven to zero by decision variable x_2 so the simplex method requires that x_5 be removed from the state set and replaced by x_2. Another pivoting operation

www.EngineeringBooksPDF.com

must be carried out to swap the positions of x_5 and x_2. From table 3.5 it can be seen that

$$x_5 = 60 + 2x_3 - 4x_2$$

Hence

$$x_2 = 15 + \tfrac{1}{2}x_3 - \tfrac{1}{4}x_5 \qquad (3.9)$$

This becomes the new third row of a new simplex table, table 3.6.

Table 3.6 Third simplex table – solution terminates

		x_3	x_5	
x_1	15		$\frac{1}{4}$	15
x_4	10	2	$-\frac{1}{2}$	10
x_2	15	$\frac{1}{2}$	$-\frac{1}{4}$	15
x_6	40	2	-1	40
x_7	45	-1	$-\frac{1}{4}$	45
f	150	2	$\frac{1}{2}$	150
		0	0	

This relationship for x_2 must be substituted into all existing equations for state variables x_1, x_4, x_6, x_7 and into the objective function f, that is, for x_1

$$x_1 = 30 + \tfrac{1}{2}x_3 - x_2$$

substitute equation 3.9

$$x_1 = 30 + \tfrac{1}{2}x_3 - (15 + \tfrac{1}{2}x_3 - \tfrac{1}{4}x_5)$$

therefore

$$x_1 = 15 + \tfrac{1}{4}x_5$$

which becomes the first row of table 3.6. Similarly for f

$$f = 180 + 3x_3 - 2x_2$$

Substituting equation 3.9

$$f = 180 + 3x_3 - 2(15 + \tfrac{1}{2}x_3 - \tfrac{1}{4}x_5)$$

www.EngineeringBooksPDF.com

therefore

$$f = 150 + 2x_3 + \tfrac{1}{2}x_5$$

which is the new objective function for table 3.6. The table may be completed in this fashion. Values of the decision variables $x_3 = 0, x_5 = 0$ are inserted and dependent variable values are evaluated along with the value of f from them. A check against table 3.5 shows the same values for all variables.

This third simplex table is now ready for the start of a further cycle. It becomes very quickly evident, however, that no further operations are possible. Examining the coefficients, a, of the objective function, f, no negative coefficients are found. Thus f cannot be decreased by an *increase* in either x_3 or x_5. Also, since x_3 and x_5 both have zero value, neither of them may be *decreased*. Since no increase or decrease is possible in any decision variable the solution of the problem must have been reached and the simplex method terminates here. The solution of the problem 3.2 can be read from the table 3.6 as $x_1^* = 15, x_2^* = 15; f^*$ (min) = 150. Figure 3.3 shows graphically the path traced out from the initial feasible point (30, 12.5) to the solution point (15, 15).

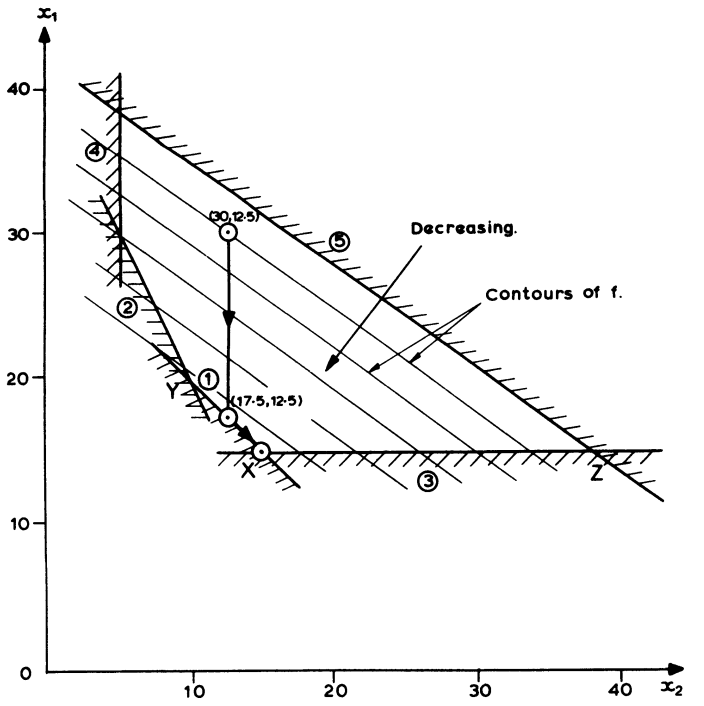


Figure 3.3 Path traced by the simplex method solution

www.EngineeringBooksPDF.com

3.1.5 A Flowchart for the Simplexing Operation

The numerical example just solved was small but nevertheless its hand solution is not particularly rapid. As the number of variables and constraints increases hand calculation becomes increasingly tedious. It has already been mentioned that the pivoting operation is amenable to computer programming. This applies to the whole of the simplex method. Indeed very many good computer programs are available for the solution of LP problems expressed in the general form of problem 3.1. To demonstrate how the calculations on a simplex table such as table 3.1 or 3.2 might be computerised, figure 3.4 shows a flowchart of the logical operations involved in selecting a decision variable to alter in value, calculating

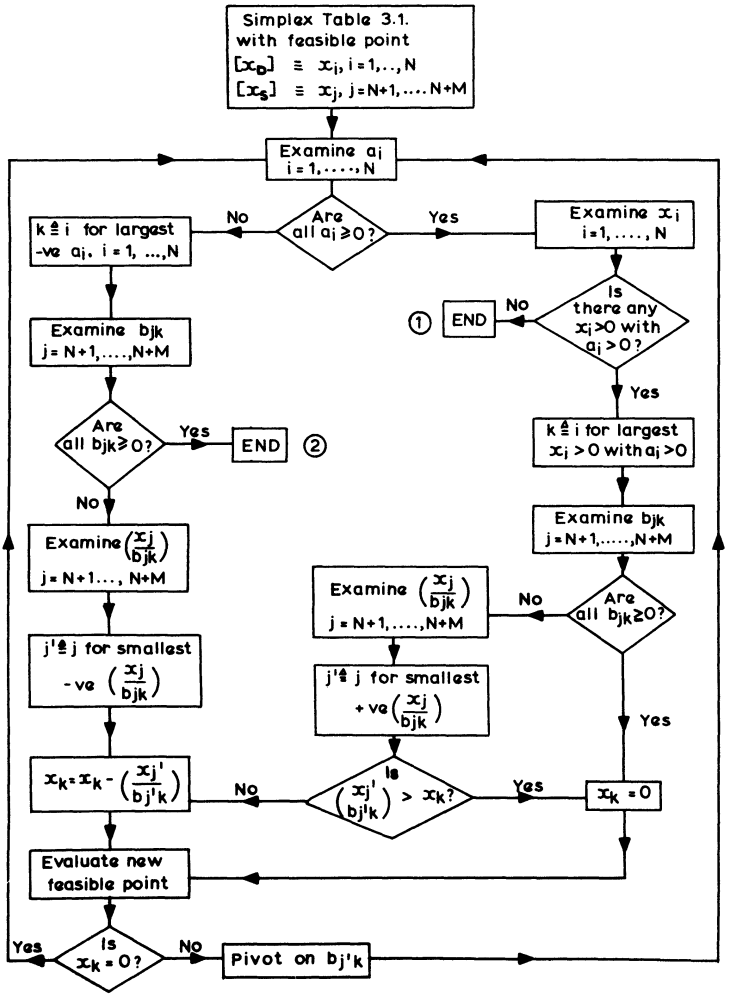


Figure 3.4 Flowchart of the simplexing operation

www.EngineeringBooksPDF.com

how much it may alter, and moving to a new point. The flowchart commences with a simplex table in the form of table 3.1 and assumes that a feasible point is known, that is, values of the decision variables $x_i, i = 1, \ldots, N$ and values of the state variables $x_j, j = N + 1, \ldots, N + M$ are known and are all non-negative. The flowchart traces exactly the manipulations and calculations described in section 3.1.2. Each simplex operation ends, if necessary, with a pivoting operation and then returns to examine the next new simplex table. The pivoting operation has not been included in the flowchart because in algebraic form it is not very concise.

The flowchart terminates in one of two ways. The END statement labelled 1 is the conventional solution point at which no objective function coefficient is negative and all decision variables have the value zero. This corresponds to a correct solution to the problem. The END statement labelled 2 corresponds to a failure to solve the problem. This statement would be reached in the circumstances shown in figure 3.5. Here the objective function slopes away from the constraint vertices into an open region and movement in the direction of the arrow is then unlimited. In the event that this statement is reached, a thorough check of the formulation of the problem is necessary because the most likely cause of this failure is an error in the way the problem is written.

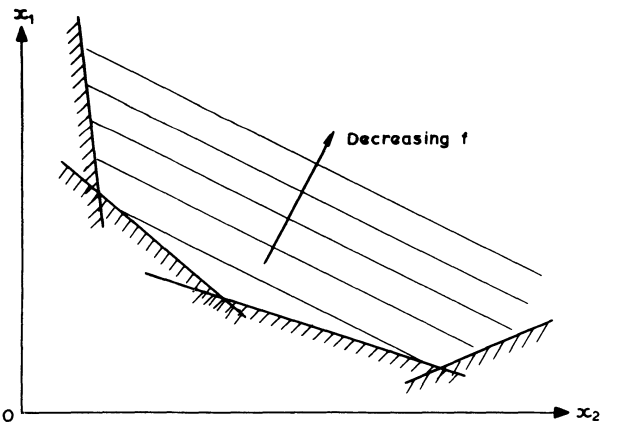


Figure 3.5 An unbounded solution to an LP problem

3.1.6 Finding an Initial Feasible Point – Simplex Phase I

In order to solve LP problems by the simplex method it is necessary to know an initial feasible point from which to start the simplexing operations. In the treatment so far it was assumed that a feasible point was known – sometimes this is the case and a feasible point for a numerical problem can often be spotted by a close examination of the constraints. In general, however, this approach is not good enough and it is necessary to have some reliable method for finding a feas-

www.EngineeringBooksPDF.com

ible point, if one exists, for any LP problem no matter how large or complicated the problem is. One ingenious method of finding a feasible point is to use the simplex method itself. This is often called the *simplex phase I* process to distinguish it from the use of the simplex method to actually solve an LP problem, which becomes *simplex phase II*. Naturally this use of the simplex method both to find a feasible point for a problem and then to solve the problem is computationally compact and efficient and it has consequently become very popular.

Consider the general LP problem 3.1 and the numerical example problem 3.2. The first preparatory stage for a simplex solution is to convert all inequality constraints to equalities using slack variables. Constraint set 3.3 for the general case and set 3.4 for the numerical example result from this. The next stage is normally to partition the variables into decision and state variables giving constraint sets 3.5 and 3.6. If an initial feasible point must be found, the partitioning is not done at this stage. Instead, constraint sets 3.3 and 3.4 are first modified by the introduction of *artificial variables*. To each constraint in sets 3.3 and 3.4 a separate non-negative artificial variable is *added* to the left-hand side: $x'_{N+M+j}, j = 1, \ldots, M$ in set 3.3; $x'_8, \ldots, x'_{12}$ in set 3.4. Thus set 3.3 becomes

$$\left.\begin{aligned} g_j &\equiv \sum_{i=1}^{N} b_{ji}x_i - x_{N+j} + x'_{N+M+j} = c_j \quad j = 1, \ldots, M_1 \\ h_j &\equiv \sum_{i=1}^{N} b_{ji}x_i + x_{N+j} + x'_{N+M+j} = c_j \quad j = M_1 + 1, \ldots, M \\ & \qquad\qquad x_i \geqslant 0 \quad i = 1, \ldots, N + 2M \end{aligned}\right\} \tag{3.10}$$

and set 3.4 becomes

$$\left.\begin{aligned} g_1 &\equiv 2x_1 + 2x_2 - x_3 + x'_8 = 60 \\ g_2 &\equiv 2x_1 + 4x_2 - x_4 + x'_9 = 80 \\ g_3 &\equiv 4x_1 - x_5 + x'_{10} = 60 \\ g_4 &\equiv 4x_2 - x_6 + x'_{11} = 20 \\ h_5 &\equiv 3x_1 + 2x_2 + x_7 + x'_{12} = 120 \\ & \qquad x_i \geqslant 0 \quad i = 1, \ldots, 12 \end{aligned}\right\} \tag{3.11}$$

The artificial variables in sets 3.10 and 3.11 are distinguished by the superscript $'$ for reasons that will become apparent later. They remain, however, normal non-negative variables.

If the partitioning is done now, taking the artificial variables as state (dependent) variables and the problem variables and slacks as decision (independent) variables, set 3.10 becomes

$$\left.\begin{aligned} g_j &\equiv x'_{N+M+j} = c_j - \sum_{i=1}^{N} b_{ji}x_i + x_{N+j} \quad j = 1, \ldots, M_1 \\ h_j &\equiv x'_{N+M+j} = c_j - \sum_{i=1}^{N} b_{ji}x_i - x_{N+j} \quad j = M_1 + 1, \ldots, M \\ & \qquad\qquad x_i \geqslant 0 \quad i = 1, \ldots, N + 2M \end{aligned}\right\} \tag{3.12}$$

www.EngineeringBooksPDF.com

and, for the numerical example, set 3.11 becomes

$$\left.\begin{aligned} g_1 \equiv x'_8 &= 60 - 2x_1 - 2x_2 + x_3 \\ g_2 \equiv x'_9 &= 80 - 2x_1 - 4x_2 + x_4 \\ g_3 \equiv x'_{10} &= 60 - 4x_1 + x_5 \\ g_4 \equiv x'_{11} &= 20 - 4x_2 + x_6 \\ h_5 \equiv x'_{12} &= 120 - 3x_1 - 2x_2 - x_7 \\ & x_i \geqslant 0 \quad i = 1, \ldots, 12 \end{aligned}\right\} \tag{3.13}$$

If values of zero are now assigned to all decision variables, the state variables (the artificials) will all be positive, that is, $x'_{N+M+j} = c_j, j = 1, \ldots, M$ in the general case, and $x'_8 = 60, x'_9 = 80, x'_{10} = 60, x'_{11} = 20, x'_{12} = 120$ in the numerical example.

The artificial variables in constraint sets 3.10 and 3.11 should not really be there. These constraint sets will only correspond to those of the original problems 3.3 and 3.4 if each of the artificial variables has the value zero and is effectively absent. Thus it is necessary somehow to reduce each artificial variable to zero whilst still satisfying the constraint equations 3.10 and 3.11. Obviously this can only be done by transferring value from the artificial variables to the problem and slack variables. If this can be done in such a way that the problem and slack variables take non-negative values and all the artificial variables become zero this will then correspond to a feasible point for the original problem. The simplex operation is a convenient way of altering the values of variables and sets 3.12 and 3.13 are already partitioned to allow a simplex table to be drawn up. Only one thing is missing – an objective function to be minimised. Since the object of simplexing is to reduce the value of each artificial variable from its present positive value to zero, a suitable objective would be to minimise the sum of the artificial variables. The substitute objective function for the simplex phase I method is therefore

$$f' = \sum_{j=1}^{M} x'_{N+M+j} \tag{3.14}$$

Table 3.7 shows the first simplex table corresponding to the constraint set 3.13 for the numerical example. The row containing the substitute objective function f', as in equation 3.14, is evaluated by summing the coefficients of all artificial variables in the rows above. Values of zero are assigned to all decision variables and non-negative values are calculated using them for all the state (artificial) variables. A simplex operation as described in section 3.1.2 can now be done on this table with the aim of reducing f' from its present value of 340. Variable x_2 is selected as the decision variable to be altered. It is increased in value to 5 at which point state variable x'_{11} reaches its base value of zero. A pivot on the encircled coefficient of -4 swaps the positions of x_2 and x'_{11} and sets up the next simplex table 3.8. Since variable x'_{11}, which has reached the value zero, is an artificial variable, we obviously do not want that variable to change in value again.

www.EngineeringBooksPDF.com

Table 3.7 First simplex table for phase I

		x_1	x_2	x_3	x_4	x_5	x_6	x_7		
x'_8	60	-2	-2	1					60	50
x'_9	80	-2	-4		1				80	60
x'_{10}	60	-4				1			60	60
x'_{11}	20		(-4)				1		20	0
x'_{12}	120	-3	-2					-1	120	110
f'	340	-11	-12	1	1	1	1	-1	340	280
		0	0	0	0	0	0	0		
		0	5	0	0	0	0	0		

Once this variable has entered the decision set after pivoting it is thus erased from the problem, that is, the column corresponding to x'_{11} is removed from the table. Table 3.8 demonstrates this. The coefficients of f' are once again given by summing the coefficients of the artificial variables. Note that the coefficients of x_2 are not included in this summation since, although x_2 is a state variable, it is not an artificial variable.

Table 3.8 Second simplex table for phase I

		x_1	x_3	x_4	x_5	x_6	x_7		
x'_8	50	-2	1			$-\frac{1}{2}$		50	20
x'_9	60	-2		1		-1		60	30
x'_{10}	60	(-4)			1			60	0
x_2	5					$\frac{1}{4}$		5	5
x'_{12}	110	-3				$-\frac{1}{2}$	-1	110	65
f'	280	-11	1	1	1	-2	-1	280	115
		0	0	0	0	0	0		
		15	0	0	0	0	0		

The second simplex operation (increasing x_1 to 15) is shown in table 3.8. A pivot brings x_1 into the state set and x'_{10} into the decision set where, since x'_{10} is an artificial variable with a value of zero, all reference to it is deleted. Thus table 3.9 is constructed. Similar simplex operations and pivots set up tables 3.10 and 3.11. In table 3.11 the simplex operation increases decision variable x_4 until

www.EngineeringBooksPDF.com

Table 3.9 Third simplex table for phase I

		x_3	x_4	x_5	x_6	x_7		
x'_8	20	1		$-\frac{1}{2}$	$-\frac{1}{2}$		20	5
x'_9	30		1	$-\frac{1}{2}$	(-1)		30	0
x_1	15			$\frac{1}{4}$			15	15
x_2	5				$\frac{1}{4}$		5	12.5
x'_{12}	65			$-\frac{3}{4}$	$-\frac{1}{2}$	-1	65	50
f'	115	1	1	$-\frac{7}{4}$	-2	-1	115	55
		0	0	0	0	0		
		0	0	0	30	0		

Table 3.10 Fourth simplex table for phase I

		x_3	x_4	x_5	x_7		
x'_8	5	1	$-\frac{1}{2}$	$-\frac{1}{4}$		5	5
x_6	30		1	$-\frac{1}{2}$		30	30
x_1	15			$\frac{1}{4}$		15	15
x_2	12.5		$\frac{1}{4}$	$-\frac{1}{8}$		12.5	12.5
x'_{12}	50		$-\frac{1}{2}$	$-\frac{1}{2}$	(-1)	50	0
f'	55	1	-1	$-\frac{3}{4}$	-1	55	5
		0	0	0	0		
		0	0	0	50		

the last remaining artificial variable x'_8 is driven to zero. The value of the substitute objective function f' becomes zero as expected. A pivot sets up table 3.12 and the deletion of x'_8 completes the removal of all the artificial variables. Values of all the remaining problem and slack variables are non-negative and therefore represent a feasible point for the original problem. At this stage the real objective function to be minimised, $f = 6x_1 + 4x_2$, is inserted at the foot of the table written in terms of the decision variables x_3 and x_5, that is

$$f = 6x_1 + 4x_2$$

$$= 6(15 + \tfrac{1}{4}x_5) + 4(15 + \tfrac{1}{2}x_3 - \tfrac{1}{4}x_5)$$

$$= 150 + 2x_3 + \tfrac{1}{2}x_5$$

www.EngineeringBooksPDF.com

Table 3.11 Fifth simplex table for phase I

		x_3	x_4	x_5		
x_8'	5	1	$\textcircled{-\frac{1}{2}}$	$-\frac{1}{4}$	5	0
x_6	30		1	$-\frac{1}{2}$	30	40
x_1	15			$\frac{1}{4}$	15	15
x_2	12.5		$\frac{1}{4}$	$-\frac{1}{8}$	12.5	15
x_7	50		$-\frac{1}{2}$	$-\frac{1}{2}$	50	45
f'	5	1	$-\frac{1}{2}$	$-\frac{1}{4}$	5	0
		0	0	0		
		0	10	0		

Table 3.12 First simplex table for phase II

		x_3	x_5	
x_4	10	2	$-\frac{1}{2}$	10
x_6	40	2	-1	40
x_1	15		$\frac{1}{4}$	15
x_2	15	$\frac{1}{2}$	$-\frac{1}{4}$	15
x_7	45	-1	$-\frac{1}{4}$	45
f	150	2	$\frac{1}{2}$	150
		0	0	

Phase II now begins with table 3.12 and its feasible starting point. Normally more simplex and pivoting operations would be necessary to reduce f even further but in this case table 3.12 represents the solution point since neither x_3 nor x_5 may be altered without increasing f. Table 3.12 is identical with table 3.6 described earlier.

The artificial variable method provides a reliable means of finding an initial feasible point for an LP problem. It is obviously amenable to computerisation and does not require much extra programming beyond that already needed for the simplex phase II method. It is worth noting here that LP problems that have no feasible region can be formulated. The artificial variable method would detect such a problem by terminating the phase I solution process with a positive value of the substitute objective function, f'. Also the length of the phase I solution

www.EngineeringBooksPDF.com

process can be reduced by means of a simple short cut. This consists of not adding artificial variables to all constraints but only adding them to constraints which were originally of the $\geqslant$ or = type. Thus the constraint set 3.11 for the numerical example need not have included artificial variable x'_{12} in constraint h_5. For this constraint x_7 could be taken as a state variable since, when all the other problem variables and slacks are allocated the value zero, x_7 will have the non-negative value 120. By not adding an artificial variable to $\leqslant$ constraints (the hs) the total number of artificial variables is reduced and hence the number of simplex operations and pivots necessary to eliminate them will also be reduced.

3.2 SENSITIVITY ANALYSIS AND LPs

In order to solve a linear programming problem posed in the standard form of problem 3.1, it is necessary to know precise numerical values for all the coefficients, a_i, b_{ji}, c_j. The solution is determined by those values. In practice those coefficients usually represent physical resources of some kind and it frequently occurs that the resources change and so the problem changes also. How can we find out whether the solution will change and if so by how much? One way of doing this would be to resolve the problem using the changed values of the constants. This would certainly answer the question but if the original problem were a large one with many variables and constraints it may be expensive to do this. Also if only one or two of the constants are changed this might seem to be a rather wasteful process. Is there any way of calculating the effect on the solution of changes in the coefficients without resolving the entire problem? *Sensitivity analysis* provides methods of doing this.

Several methods have been proposed which allow the effects of changes in the objective function coefficients, constraint coefficients and constraint bounds to be calculated without completely resolving the problem. A comprehensive study of these is beyond the scope of this book but may be found in more specialist linear programming textbooks such as those by Dantzig, Gass, Hadley, Garvin, etc. For many civil engineering applications the coefficients that tend to be the most susceptible to change are perhaps the objective function coefficients, a_i, $i = 1, \ldots, N$ which often represent unit costs, and the constraint bounds, c_j, $j = 1, \ldots, M$, which frequently represent resource limits. The effects of changes in these are now examined.

3.2.1 Changes in the Objective Function Coefficients, a_i, $i = 1, \ldots, N$

Suppose that the numerical example 3.2 has been solved and the solution point $x_1^* = x_2^* = 15$ is known and represents some policy. Suppose that the objective function coefficients of 6 for x_1 and 4 for x_2 represent unit costs that are susceptible to change. By how much may the coefficients of 6 and 4 change without affecting the optimal policy $x_1^* = x_2^* = 15$? Referring to figure 3.3 which shows

www.EngineeringBooksPDF.com

the numerical example in graphical form, the values of the objective function coefficients determine the orientation of the contours of f. Changing the values of the coefficients will cause the orientation of the contours to alter. The coefficient values of 6 and 4 locate point X as the optimum. Any ball rolling down the plane of the objective function will be trapped at X. If the orientation of the plane f is altered the ball will remain at X until the contours change sufficiently to allow the ball to roll downwards from X to a different vertex, say Y or Z. Movement of the optimum from its present position to a new vertex would be represented algebraically by another simplex operation and pivot. Thus limits upon the objective function coefficients that are necessary for X to remain optimal can be found by examining the final table of the simplex method.

Table 3.6 is the terminating table of the simplex method for this problem and the objective function f used there corresponds to

$$f = 6x_1 + 4x_2$$

Suppose that a different objective function had been used, say

$$f = c_1x_1 + c_2x_2$$

The coefficients of f in table 3.6 would be, for the new objective function

$$f = c_1\ (15 + \tfrac{1}{4}x_5) + c_2\ (15 + \tfrac{1}{2}x_3 - \tfrac{1}{4}x_5)$$

that is

$$f = (15c_1 + 15c_2) + \tfrac{1}{2}c_2x_3 + \left(\frac{c_1}{4} - \frac{c_2}{4}\right)x_5$$

If these are substituted into the f-row in table 3.6 the foot of the table becomes as shown in table 3.13.

Table 3.13 Changes in objective function coefficients

		x_3	x_5	
f	$15(c_1 + c_2)$	$\frac{1}{2}c_2$	$\frac{1}{4}(c_1 - c_2)$	$15(c_1 + c_2)$
		0	0	

This table must remain the final table of the simplex method. Values of the decision variables x_3 and x_5 are both zero and will increase, leading to another simplex operation/pivot, unless both corresponding coefficients of f are positive. Thus for variable x_3 it is essential that

$\frac{1}{2}c_2 > 0$, that is, c_2 must be +ve

www.EngineeringBooksPDF.com

and for x_5 it is required that

$$\tfrac{1}{4}(c_1 - c_2) > 0, \text{ that is, } c_1 \text{ must be} > c_2$$

Thus it can be stated that for this numerical example the solution point will remain at $x_1^* = x_2^* = 15$ providing $c_1 > c_2 > 0$. It can easily be deduced on figure 3.3 that if c_2 becomes negative the solution point will move from X to Z, or that if $c_1 < c_2 > 0$ the solution point will move from X to Y or perhaps beyond Y. Although the solution point remains at X provided that $c_1 > c_2 > 0$ the objective function value will alter and will be given by $15\ (c_1 + c_2)$. If it is known that the changes in unit costs, c_1 and c_2, are such that the requirement $c_1 > c_2 > 0$ cannot be met then the optimum policy must change. The new optimum policy can then be found by inserting new values for c_1 and c_2 into f at the foot of table 3.13 and continuing the simplex operations until a new solution is found.

3.2.2 Changes in the Constraint Limits, $c_j, j = 1, \ldots, M$

Suppose that having solved the numerical example 3.2 it is desired to examine what happens to the solution as one of the constraint limits is altered, corresponding perhaps to a change in some resource. Graphically, referring to figure 3.3, alteration of the right-hand side of a constraint causes the constraint boundary to be displaced parallel to itself. This displacement may be either into the feasible region or into the infeasible region, depending on the sign of the change in the right-hand side. On figure 3.3 the solution lies at point X, the intersection of constraints g_1 and g_3. Clearly constraints g_2, g_4 and h_5 can tolerate some parallel displacement of their boundaries without affecting the solution in any way, but constraints g_1 and g_3 cannot be moved without altering the solution. Thus the treatment of slack constraints at the optimum must be different from that of tight constraints.

Firstly consider constraints that are slack at the solution. These constraints, g_2, g_4 and h_5 in the numerical example, have associated with them slack variables x_4, x_6 and x_7. In the final simplex table for the numerical example, table 3.6, each of the slack variables appears in the set of state variables and, because the constraints in which they appear are slack, each of these variables has a non-zero value. If the right-hand side of a constraint in set g_j is changed by an amount Δ this is equivalent to changing the value of the slack variable in that constraint by $-\Delta$. If the constraint whose limit is changed is in the h_j set (constraints of the $\leqslant$ form) then a change of Δ in the right-hand side is equivalent to changing the slack variable by $+\Delta$. In the final simplex table 3.6 slack variable x_4 has the value of 10. If the right-hand side of constraint g_2 were changed from 80 to $(80 + \Delta)$ then the value of x_4 in table 3.6 will become $(10 - \Delta)$. The optimality of the table would be in no way affected by this change. The only feature that need be of concern is the feasibility of all variables. Clearly if Δ is negative, x_4 will always be positive and feasible but if Δ has a value greater than 10 then x_4 will

www.EngineeringBooksPDF.com

become negative and infeasible. Thus the solution of the problem remains valid provided that the constraint right-hand side in g_2 does not exceed 90. Similar consideration of constraints g_4 and h_5 shows the solution remaining unchanged provided that the right-hand side of g_4 does not exceed 60 and the right-hand side of h_5 exceeds 75. Thus examination of the values of slack variables in the state set of the final simplex table enables limits to be calculated for the permissible alterations to the right-hand sides of all slack constraints at the optimum.

What would happen if a desired change to one of the right-hand sides violated feasibility? As an example, suppose it is planned to change the right-hand side of g_2 from 80 to 100. This would change the value of x_4 from 10 to -10 in the final table. Variable x_4 is now infeasible as is shown in table 3.14. A simplex operation must be carried out to restore feasibility of all variables so decision variable x_3 is increased to the value of 5 which restores x_4 to zero. Since state variable x_4 has the value zero, a pivot is carried out exchanging x_3 and x_4. The new table turns out to be optimal (and, of course, feasible) and the new solution to the problem is $x_1^* = 15, x_2^* = 17.5; f^* = 160$. Figure 3.6 shows graphically the changed problem with a right-hand side of 100 in g_2 and the corresponding new solution.

Table 3.14 Restoration of feasibility after changing a constraint limit

		x_3	x_5		
x_1	15		$\frac{1}{4}$	15	15
x_4	-10	2	$-\frac{1}{2}$	-10	0
x_2	15	$\frac{1}{2}$	$-\frac{1}{4}$	15	17.5
x_6	40	2	-1	40	50
x_7	45	-1	$-\frac{1}{4}$	45	40
f	150	2	$\frac{1}{2}$	150	160
		0	0		
		5	0		

Now examine the effects of changing the right-hand side of a tight constraint at the solution point. Figure 3.3 shows that if either of the boundaries g_1 or g_3 is moved parallel to themselves the solution point X must become either infeasible or non-optimal. Suppose it is desired to change the right-hand side of constraint g_1 from 60 to $(60 + \Delta)$. Since this is a constraint in the g_j set (with $\geqslant$ sign) this is equivalent to altering the value of slack variable x_3 by $-\Delta$. Table 3.6 remains exactly in its present form except for the value of decision variable x_3 which has become $-\Delta$ instead of zero. The point represented by this new table

www.EngineeringBooksPDF.com

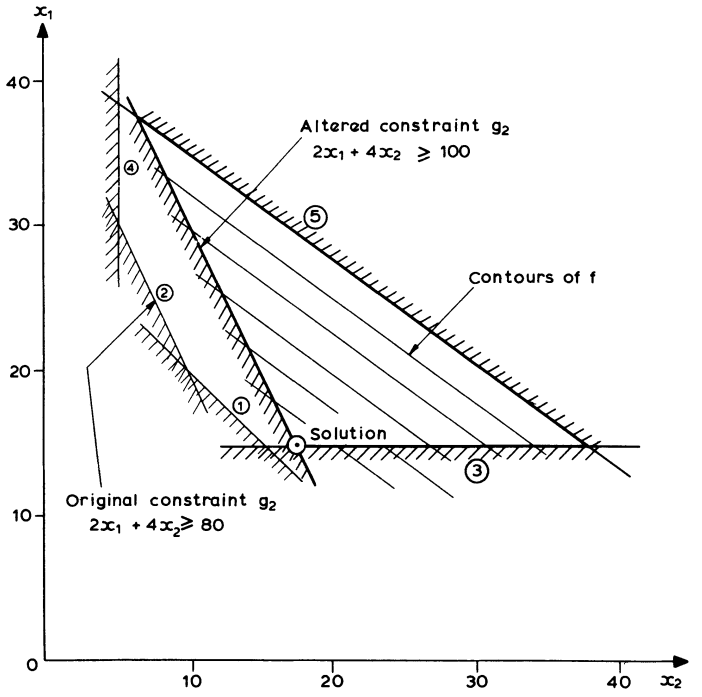


Figure 3.6 Effect of changing the right-hand side of a constraint

is obviously infeasible and to restore feasibility variable x_3 must be increased by Δ to a value of zero. This changes the values of all state variables and f as shown in table 3.15. Table 3.15 is thus restored to optimality and feasibility provided that the state variable values of $10 + 2\Delta$, $15 + \frac{1}{2}\Delta$, $40 + 2\Delta$ and $45 - \Delta$ remain non-negative. This is ensured by values of Δ within the range $-5 \leqslant \Delta \leqslant 45$ which correspond to values for the right-hand side of constraint g_1 between 55 and 105. For a particular value of Δ within the range given above the solution of the problem will be $x_1^* = 15, x_2^* = 15 + \frac{1}{2}\Delta; f^* = 150 + 2\Delta$. If Δ is required to have a value outside the range $-5 \leqslant \Delta \leqslant 45$ some state variable must become infeasible and further simplex operations and pivots will be needed to restore feasibility and ensure optimality.

In examining the effects of changes in the right-hand sides of constraints it has been shown that changes made to some constraints (the slack ones) have no effect on the optimal policy or the value of the objective function but changes made in tight constraints have very direct influence on the values in the solution. In a situation where the problem corresponds to a particular civil engineering operation this knowledge can be important. The constraint limits reflect the levels of resources, supply and demand and it is very useful to know which limits

www.EngineeringBooksPDF.com

Table 3.15 Alteration of a tight constraint limit

		x_3	x_5		
x_1	15		$\frac{1}{4}$	15	15
x_4	10	2	$-\frac{1}{2}$	10	$10+2\Delta$
x_2	15	$\frac{1}{2}$	$-\frac{1}{4}$	15	$15+\frac{1}{2}\Delta$
x_6	40	2	-1	40	$40+2\Delta$
x_7	45	-1	$-\frac{1}{4}$	45	$45-\Delta$
f	150	2	$\frac{1}{2}$	150	$150+2\Delta$
		$-\Delta$	0		
		0	0		

are the critical, sensitive ones and which have little effect on the optimal decisions. Very often exact values for constraint limits are not known. Approximate values may perhaps be estimated quite quickly but often much time and effort must be spent in order to obtain accurate values. This sort of sensitivity analysis determines which particular limits are the critical ones upon which it may be worthwhile spending time and effort to estimate accurate values. For the non-critical limits often the quick approximate values will suffice.

3.3 DUALITY IN LINEAR PROGRAMMING

An important feature of linear programming is that every linear programming problem has associated with it another linear programming problem called its *dual*. Consider the problem

$$\left.\begin{aligned} &\text{Minimise } f=\sum_{i=1}^{N} a_i x_i \text{ over variables } x_i,\ i=1,\ldots,N \\ &\text{subject to } \sum_{i=1}^{N} b_{ji} x_i \geqslant c_j \qquad j=1,\ldots,M \\ &\qquad\qquad x_i \geqslant 0 \qquad\qquad i=1,\ldots,N \end{aligned}\right\} \tag{3.15}$$

Problem 3.15 is a slightly modified form of the standard linear programming problem 3.1 in which all the $\leqslant$ constraints in set h_j have been converted to $\geqslant$ constraints in set g_j by multiplying throughout by -1. Consequently, in problem

www.EngineeringBooksPDF.com

3.15 some of the c_js may be negative. Associated with problem 3.15 is its *symmetric dual* problem

$$\left.\begin{aligned} &\text{Maximise } F = \sum_{j=1}^{M} c_j y_j \text{ over variables } y_j, j = 1, \ldots, M \\ &\text{subject to } \sum_{j=1}^{M} b_{ji} y_j \leqslant a_i \qquad i = 1, \ldots, N \\ &\qquad\qquad y_j \geqslant 0 \qquad\qquad j = 1, \ldots, M \end{aligned}\right\} \tag{3.16}$$

Problems 3.15 and 3.16 have many similarities. The objective function coefficients of one problem are the constraint bounds of the other and vice versa. One problem has N variables and M constraints, the other has M variables and N constraints. It is simple to demonstrate that the dual of the dual problem 3.16 is the primal problem 3.15.

The most important feature of the symmetric primal-dual pair of problems 3.15 and 3.16 is that the solutions of both are also closely related. In fact, if problem 3.15 is solved and f^* is the numerical value of f at its minimum constrained solution point, then

$$f^* = F^* \tag{3.17}$$

where F^* is the numerical value of F at the solution point of problem 3.16. Also, at the solution points of both problems, if the jth dual variable, y_j, is positive, the jth primal problem constraint will be satisfied as an equality; if y_j is zero then constraint j in the primal problem will be satisfied as an inequality.

The mathematical proofs of the above statements are too technical to introduce at this point. The equivalence of problems 3.15 and 3.16 may be demonstrated by examining the Lagrangian functions of both problems and by exploiting the nature of Lagrangian functions using the methods described later in this book. For the present, it is necessary to accept on trust that adequate mathematical proofs of the equivalence of problems 3.15 and 3.16 exist. Specialist textbooks such as that by Hadley may be referred to for detailed proofs.

From the point of view of the need to solve LP problems, duality is very important. Any LP problem that has the form of problem 3.15 may be either solved directly or its dual problem may be solved and the primal solution evaluated once the dual solution is known. The primal problem 3.15 is set in N-dimensional space with M constraints. The dual problem 3.16 is set in M-dimensional space with N constraints. If M and N differ greatly then it can sometimes be much quicker to solve the dual problem than to solve the primal. As an example of this consider an LP problem of the form of problem 3.15 with two variables x_1 and x_2 but twenty constraints, $j = 1, \ldots, 20$. When slack variables are added to all constraints to make them equalities there will be a total of 22 variables. Since all constraints are of the $\geqslant$ type the addition of artificial variables

www.EngineeringBooksPDF.com

for the simplex phase I routine requires a further 20 variables. Thus the simplex table for the first iteration of the phase I routine will have 22 decision variables and 20 state variables. Furthermore the body of the table will comprise 20 x 22 = 440 coefficients b_{ji}. Its shape will be as shown in table 3.16.

Table 3.16 Simplex table for primal problem (phase I)

	← 22 decision variables →
↑ 20 state variables ↓	440 elements
f'	

Examination of the dual problem for a primal with two variables and twenty constraints shows that the dual problem 3.16 has twenty variables $y_j, j = 1, \ldots,$ 20, but only two constraints, $i = 1, 2$. When slack variables are introduced to make the constraints into equalities the number of variables rises to 22. With $\leqslant$ constraints, however, no simplex phase I routine is necessary as was noted at the end of section 3.1.6. An initial feasible point is readily available in the form of $y_j = 0, j = 1, \ldots, 20, y_{j+i} = a_i, i = 1, 2; j = 20$. Thus the first simplex table for the dual problem (phase II) will have 20 decision variables, 2 state variables and only 40 elements in the body of the table. Its shape will be as shown in table 3.17.

Table 3.17 Simplex table for dual problem (phase II)

	← 20 decision variables →
y_{21} y_{22}	40 elements
F	

Comparison of these primal and dual problems demonstrates the great computational advantages to be gained by solving the dual rather than the primal. The dual requires no phase I routine, it is easier to formulate and requires much less computer storage than the primal. In the solution process the simplexing and pivoting operations will be much quicker for the smaller dual problem. If the dual problem is solved in preference to the primal, a final necessary stage is that

www.EngineeringBooksPDF.com

of isolating the corresponding primal problem solution. The primal objective function value at the solution point, f^* is directly given by equation 3.17. Optimal values of the primal variables x_i^*, $i = 1, \ldots, N$ are found by the solution of the simultaneous equations formed by the primal constraints in problem 3.15 whose identifiers j correspond to dual variables y_j with positive values. In the above example table 3.17 shows that two state variables, y, may be expected to be positive. Thus two of the twenty primal constraints corresponding to the two positive dual variables must be solved as equalities to yield optimal values of the two primal variables, x_1^* and x_2^*.

Used in the manner outlined above, duality can be a very useful tool in the solution of LP problems. The choice of whether to solve an LP problem directly or indirectly via its dual problem depends on the relative numbers of variables and constraints. When these numbers are about the same, direct solution is recommended. When the number of primal constraints is much greater than the number of primal variables, the dual problem can be far more economical to solve than the primal.

The usefulness of the dual is not restricted merely to computational efficiency; it can be of value in sensitivity analysis of the solution of an LP problem. Section 3.2.1 showed how it is possible to study the effects of changes in the objective function coefficients of an LP problem. Section 3.2.2 dealt with changes in the constraint limits. In this section 3.3 it has been shown that primal objective function coefficients become dual constraint limits and vice versa. Thus the dual problem can be used to extract sensitivity information about the primal problem. For example, if a primal problem has been solved indirectly via its dual problem it can be much easier to examine changes in the primal constraint limits by studying changes in the dual objective function coefficients instead.

3.4 OTHER METHODS FOR SOLVING LP PROBLEMS

Section 3.1 has described the simplex method for solving linear programming problems. It is probably the most widely understood method and is frequently used in practice. Most large computing facilities will have standard library programs for the solution of LP problems and the standard simplex method will usually be among these. It is possible, however, to improve upon the standard simplex method, and several other methods are often used to solve LP problems because they make more economical use of computer facilities. For example, in the simplex method a complete new table is constructed at each pivoting operation but very few of the elements of the new table are actually used in the next simplexing operation. The computation of large numbers of elements that are not used is inefficient. Consequently the *revised simplex* method was developed which uses a restructured solution method based on the standard simplex method in which very few redundant calculations are made. Another simplex-based method allows more than one decision variable to be altered in each simplexing operation. The *dual simplex* method and the *primal-dual* method use the duality

www.EngineeringBooksPDF.com

relationships described in section 3.3 to produce rapid, computationally efficient LP computer methods. Thus a computer library may offer a choice of algorithms for solving linear programming problems but the potential user need not be daunted by the choice. All the methods will be based upon the simplex technique described in this chapter and each one will be especially efficient in solving particular types of LP problems. Close examination of a particular LP problem and considerations of whether or not sensitivity analyses will be needed should allow an appropriate library algorithm to be chosen. If the choice is not clear then the standard simplex method should always provide a solution to an LP problem although it may not be especially efficient.

Section 3.6 deals with special classes of LP problems in which values of variables are required to be either integer values or values taken from a discrete set. If a rigorous computer solution of these problems is required, special algorithms must be used since the simplex method will not operate under these restrictions. Another instance in which a purpose-made algorithm is strongly preferred to the simplex method is for so-called *transportation problems* in which all constraints are naturally equalities. The earthworks example in chapter 2 is an example of this class of problem. Special algorithms exploiting the fact that all constraints are equalities are very much more efficient than the standard simplex method which will, nevertheless, produce a solution.

3.5 NEGATIVE VARIABLES

The linear programming problems that have been examined so far have all involved variables that must have non-negative values, that is, zero or positive values. Usually, in practical problems this is a quite natural requirement because the variables often represent real resources of men, money, time, etc., for which a negative value has no meaning. The simplex method as described in section 3.1 is based on the assumption that all variables must be non-negative. Occasionally, however, problems arise in which the variables are defined in such a way that negative values would have meaning and would be acceptable in a solution. For example, a variable might be defined as the amount of money needed over and above some basic investment in a particular facility or process which is to be modified. Here a negative value for that variable would indicate that the basic investment could be reduced. This might be a perfectly acceptable policy but it poses the question of how the simplex method should be modified to allow a variable to have a negative value.

There are many ways of handling negative variables in LP problems and no concrete rules are proposed here. Rather a set of alternative methods is discussed. Suppose that an LP problem has been formulated in the form of problem 3.1 except that one of the variables, which may be called x_N, is not restricted in sign. All other variables x_i, $i = 1, \ldots, N-1$ must be non-negative, but x_N must be allowed to be negative, zero or positive as the solution demands. If the problem is solved by the simplex method exactly as described in section 3.1 the solution

www.EngineeringBooksPDF.com

method will automatically impose a non-negativity requirement upon all variables including x_N and it will not be possible to find a solution, if one exists, with x_N negative. Figure 3.7 demonstrates this on a simple two-variable problem in which x_2 is to be permitted to have a negative value if it is necessary. The problem constraints and objective function contours clearly show the problem solution at point X at which x_2 has a negative value. If the simplex method is used to solve the problem, a new constraint $x_2 \geqslant 0$ will be imposed by the solution process leading to the location of an incorrect solution to the original problem at point Y where $x_2 = 0$.

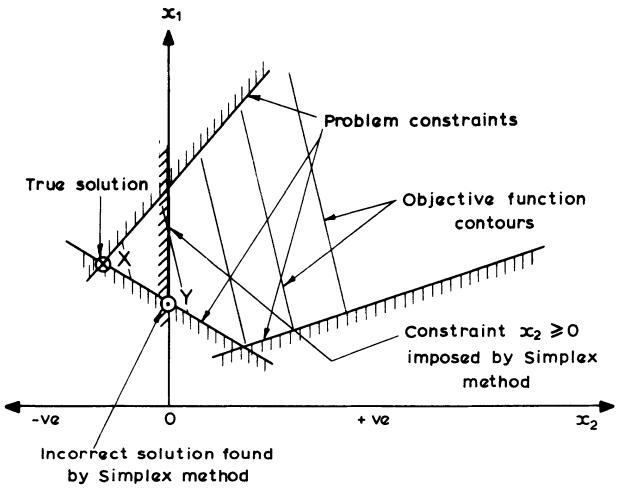


Figure 3.7 The simplex method finds a false solution for a negative variable

One way of overcoming this is to modify the simplex method and carefully to monitor variable x_N as its value is altered through successive simplex tables. Whenever variable x_N is adjusted in value either as a decision or as a state variable it should not automatically be given a base value of zero but should be allowed to take negative values if the solution process requires this. This approach would solve the problem correctly and is quite useful in small hand-solved problems. It does, however, require careful monitoring of the variables particularly if not just one but several variables are permitted to have negative values. In automatic computation, however, it is a cumbersome process requiring additions to the basic computer program to carry out the monitoring of variables. This increases program size and run time and is generally inefficient.

Another approach is to re-examine the formulation of the problem and to redefine variable x_N in such a way that it does not have negative values. Sometimes it is convenient to define a new variable $\tilde{x}_N$ to replace x_N so that

$$\tilde{x}_N = x_N + C$$

where C is an arbitrary constant with a positive value chosen large enough so that

www.EngineeringBooksPDF.com

whatever negative value x_N might have $\tilde{x}_N$ will still be positive. Wherever x_N appears in problem 3.1 it is replaced by $\tilde{x}_N - C$ and the new LP problem resulting from this is then solved with all variables $x_i, i = 1, \ldots, N-1$ and $\tilde{x}_N$ non-negative. Figure 3.8 shows how this works for the two-variable problem shown in figure 3.7. Introducing $\tilde{x}_2 = x_2 + C$ has the effect of moving the problem (or the axes) so that the problem now lies entirely within the positive quadrant. The simplex method would now locate correctly the solution at point X. The only difficulty in making the substitution is that of choosing a high enough value for C to ensure that the problem really is located in the positive quadrant. Frequently a reconsideration of the physical problem represented by the LP problem will suggest new formulations that do not require any variable to be allowed to have negative values.

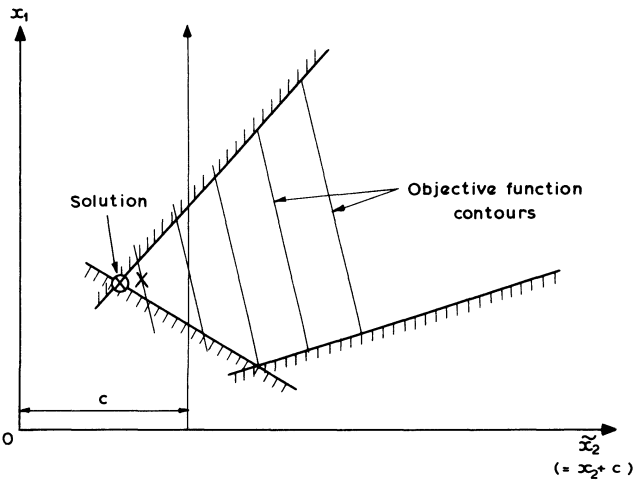


Figure 3.8 Changing variables in figure 3.7 locates the correct solution

A final and attractive possibility is that variable x_N might be redefined and replaced by the difference of two new variables $x_{N'}$ and $x_{N''}$, both of which are required to be non-negative, that is

$$x_N = x_{N'} - x_{N''}$$

This substitution for x_N is made wherever x_N appears in the LP problem which is then solved by the straightforward simplex method optimising over variables x_i, $i = 1, \ldots, N-1, x_{N'}$ and $x_{N''}$ all of which must be non-negative. A final solution in which $x_{N'} > x_{N''}$ implies a positive x_N; if $x_{N''} > x_{N'}$ with both non-negative the solution implies a negative x_N. The disadvantage of this substitution is that it increases the number of variables in the problem and therefore can increase the solution time required. Advantages are that the simplex solution method is not altered in any way and that the substitution does not need any arbitrary constants to be estimated.

www.EngineeringBooksPDF.com

3.6 PROBLEMS WITH INTEGER OR DISCRETE-VALUED VARIABLES

It is sometimes necessary in the final solution of an LP problem that some or all of the variables should have integer values. For instance solutions requiring that 6.8 men should be assigned to a task or that 4.3 precast concrete panels should be produced are clearly not acceptable as final policy. For these quantities only integer values are required. Similarly there is sometimes a requirement that a certain variable or variables in the solution of an LP problem should take on values chosen from a discrete set of available values. An example of this is afforded by example 2.3 of chapter 2 which is concerned with the design of steel frameworks. Mild steel rolled sections such as might be used to build the frame of figure 2.1 are commonly available only in a range of discrete sizes. Each available section has a unique fully plastic moment value and in order to ensure that an available section is chosen in the solution it is necessary to require that values of M_1 and M_2 should be selected from the set of discrete values corresponding to the available sections.

If a linear programming problem is to be solved over integer values for all variables it is classed as an *integer LP* problem. If only some of the variables take integer values while others are continuous the problem is classed as a *mixed integer LP* problem. Similar definitions apply to discrete-valued variables. A further type of linear programming problem is that known as a *zero-one* or a *binary* LP problem in which the variables may take only values of zero or one. Problems involving integer or discrete variables are sometimes reformulated in binary form in order to make use of the specialised solution methods available for these problems.

The rigorous solution of LP problems involving integer or discrete variables is difficult. Simple algorithmic concepts such as the simplex method cannot be used and the development of alternative solution methods has been the subject of much research. The interested reader should consult books such as that of Garfinkel and Nemhauser for further details of this highly specialised field. Despite the research which has been done it must be reported that the rigorous solution methods for LP problems with non-continuous variables are generally expensive in terms of computer resources and are applicable only to relatively small problems. Although problems of a civil engineering origin have been solved rigorously over non-continuous variables using specialised techniques this is by no means common practice. As a consequence of their complexity and of the fact that they are rarely used in civil engineering practice these solution methods will not be considered further.

It should be stressed that the methods mentioned above are concerned with the *rigorous* solution of non-continuous LP problems, that is, methods that may be mathematically proven to yield only the correct integer or discrete-valued solution to a problem. When rigorous solution methods are not available, or when their use is prohibitively expensive, engineers often have to make compromises. Indeed much of the art of engineering lies in knowing when, where and how to make the right compromise between rigour and practicality. In the case of non-

www.EngineeringBooksPDF.com

continuous LP problems, since rigorous solutions are difficult to obtain, engineers often compromise on rigour. Perhaps the most widely used concept in these non-rigorous methods is that of replacing a non-continuous function by a continuous one. The continuous LP problem is then solved and the solution point obtained is 'rounded' to the nearest integer or discrete solution point.

3.6.1 Rounded Solutions of LP Problems

The rounding approach to the solution of non-continuous linear programming problems is a very popular and simple means of obtaining a solution to otherwise difficult problems. Example 2.3 of chapter 2 is a good example of how rounding-off can be done. Variables M_1 and M_2 should, to be precise, take only discrete values from a specified set of available fully plastic moments throughout the solution process. Instead of doing this the LP problem is first of all solved in the normal way using the simplex method, allowing M_1 and M_2 to take any values they wish as if they were continuous variables. This yields a continuous solution $M_1{}^*, M_2{}^*$. In general $M_1{}^*$ and $M_2{}^*$ will not correspond to discrete available values but will fall between them. These continuous values must now be rounded to the nearest feasible discrete values with the most optimal alternative being chosen if several are available. Typically this rounding process consists of locating the discrete values which are just above and just below each of the continuous values. This gives four possibilities in this example, $M_1{}^{*+}, M_1{}^{*-}, M_2{}^{*+}$ and $M_2{}^{*-}$ where the extra superscript + or - indicates discrete values above or below the continuous value. Combining these values gives four candidate discrete solution points, $(M_1{}^{*+}, M_2{}^{*+})$, $(M_1{}^{*+}, M_2{}^{*-})$, $(M_1{}^{*-}, M_2{}^{*+})$ and $(M_1{}^{*-}, M_2{}^{*-})$. Each of these four candidate points must be substituted into all the problem constraints and discarded if it violates any constraint. If more than one of the four candidate solutions is feasible then the value of the objective function should be calculated for each feasible candidate and the most optimal chosen as the discrete solution. If none of the candidate points satisfies all the constraints then the search must be widened by selecting new discrete candidates, such as values $M_1{}^{*++}$ which represents the second discrete value above the continuous solution $M_1{}^*$. In this way a feasible discrete-valued solution point that is close to the continuous optimum is eventually obtained.

It might be thought that this rounding process which selects the discrete feasible point closest to a continuous optimum will always lead to a discrete optimum. Figure 3.9 shows that this is not so and, therefore, that the process is not rigorous. Figure 3.9 represents graphically the problem

$$\left.\begin{aligned} &\text{Maximise } f = 4x_1 + x_2 \\ &\text{subject to } 5x_1 + x_2 \leqslant 26.5 \\ &\qquad\quad x_1 + 5x_2 \leqslant 26.5 \\ &\qquad\quad x_1, x_2 \text{ are +ve integers} \end{aligned}\right\} \tag{3.18}$$

Strictly speaking, the objective function f only has a value at each of the discrete points shown in figure 3.9. If the integer requirement for x_1 and x_2 is relaxed,

www.EngineeringBooksPDF.com

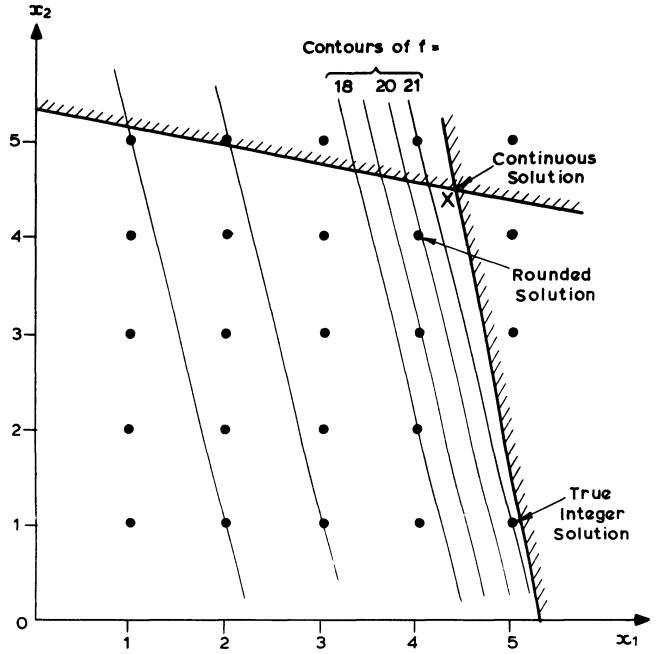


Figure 3.9 A rounded continuous solution is not always correct

however, and f is treated as a continuous function then the solution of problem 3.18 is found at the constraint vertex X where $x_1^* = x_2^* = 4.417$ and f has the value 22.083. Figure 3.9 shows that the nearest feasible discrete point to the continuous optimum is $x_1^{*-} = x_2^{*-} = 4$ at which f has the value 20. Close examination of figure 3.9 shows that this nearest feasible discrete point is not in fact the discrete optimum point of problem 3.18. The optimum solution is $x_1 = 5$, $x_2 = 1$ with $f = 21$. Thus in this example the true integer solution lies some considerable distance away from the nearest feasible point to the continuous optimum.

Problem 3.18 was somewhat artificially contrived in order to demonstrate the lack of rigour in the rounding process. In the great majority of practical problems in which rounding is used, however, behaviour such as that exhibited in figure 3.9 will not be found and the nearest feasible discrete point will probably also be the discrete optimum solution but no guarantee of this can be given. It should also be noted that even in the rather special problem 3.18 the objective function value of the nearest feasible point to the continuous optimum is only of the order of 5 per cent different from the value of f at the true integer optimum. If this problem were of an engineering origin a 5 per cent difference such as this might be quite acceptable and may be considerably less than possible inaccuracies in estimating values for the constants used in the LP problem.

www.EngineeringBooksPDF.com

The rounding process generally works well to provide rapid solutions to engineering problems involving discrete or integer variables. In the case of the steelwork design example of chapter 2 it is reasonable to represent M_1 and M_2 as continuous functions, rounding-off the solution, because the range of available rolled sections is such that they provide many discrete fully plastic moment values distributed evenly throughout the range. Whenever problems arise in which discrete-valued variables occur and they can be represented accurately by a continuous variable, rounding can be used with confidence. Conversely if only a few discrete values are available, perhaps with large and uneven gaps between them, then it is less easy to be confident that rounding will provide a truly optimum discrete solution.

3.7 CIVIL ENGINEERING USES FOR LINEAR PROGRAMMING

Applications of linear programming to practical civil engineering problems are so numerous and wide-ranging that only the more important uses can be mentioned here. A mere catalogue of applications would not be particularly valuable. Of far greater importance is that the reader should be aware of broad areas, types and classes of problems in which linear programming has been effectively used. The existence of certain well-defined characteristics in a problem should become recognisable and should intuitively suggest linear programming as a possible solution. Previous experience is invaluable in this area of spotting that a specific problem might be solved as a linear programming problem. No book can provide that experience but the following guidelines may be useful.

Returning to the three examples of chapter 2, the first of them was concerned with earthworks operations and clearing a site. Although this is an essential activity on all sites, the key to the use of linear programming lies in the existence of multiple sources of cut material and multiple destinations requiring fill. Linear programming can be used to find the best policy for the earthworks operations from the many possibilities available. The larger the site and the larger the earthworks operations, the more likely it will be that LP can be of use. Thus projects such as road construction, large industrial or port construction and earth dams are all likely areas for a detailed planning of the earthworks using LP to be of value.

It was mentioned in section 3.4 that the earthworks example with its equality constraints was an example of a much wider class of problems known as transportation problems for which special purpose LP algorithms are available. The earthworks example was concerned with moving earth between multiple sources and multiple destinations. Of course the commodity being moved need not necessarily be earth. The transportation problem covers all types of commodities and the linear programming formulation is the same for all of them. Three elements are essential in any transportation problem; multiple sources, multiple destinations and a linear cost relationship. Given these three elements a linear programming problem will enable an optimal transportation schedule to be

www.EngineeringBooksPDF.com

found. The economic life of a nation depends on transportation of commodities; coal between mines and power generation facilities, metal ores between mines and smelting plant, aggregates between quarries and sites, steel between steelworks and factories, manufactured goods between factories and warehouses and on to retail outlets. These are all transportation processes and as such are likely at some stage to be amenable to planning by some linear programming method. The range of applications of some form of LP is thus far wider than just civil engineering; it touches many aspects of everyday life. The important thing for civil engineers is to be able to recognise a transportation process and to think immediately in terms of a possible linear mathematical model for decision-making.

The second example in chapter 2 was concerned with the scheduling of the production of a precasting plant. This is clearly not a transportation process yet it allowed a linear mathematical model to be constructed and solved by LP. Further examination of the problem reveals that it has three essential elements; constraints representing demand, constraints representing the supply of resources (available materials and production capacity), and a linear cost or profit function. The close parallel with the three essential elements of a transportation LP problem is not coincidental. The precasting plant problem, while in no sense identical, is closely similar to transportation problems. In this problem resources had to be allocated among the multiple possible activities so that supply and demand restrictions were satisfied at least cost or maximum profit. The problem is typical of a very wide class of problems arising in any production process. There are many problems of this nature within civil engineering and an even greater number in manufacturing industry. The name 'resource allocation problems' is sometimes given to problems of this type and the resources to be allocated, so as to satisfy supply and demand, may be resources of money, labour, machinery or a combination of these.

Likely sources of resource allocation problems in the civil engineering sphere are in any repetitious fabrication processes such as are found in precast concrete production or steelwork prefabrication. Also, during construction site operations there are usually many different activities going on simultaneously each making demands for limited resources. Resource allocation problems arise here with the possibilities of using available resources of men, money and machinery to speed up some activities, slow down others and so complete the total project as economically as possible. In the planning phase, linear programming resource allocation can be used to plan efficient layouts. For example, in the planning of hospitals, the different types of activities (resources) such as operating theatres, wards, reception areas, casualty areas, outpatients, etc., may be allocated to different available areas on a plan so as to minimise communication distances within the hospital. In new town planning, resources such as industry, housing, schools, shops, etc., may similarly be allocated to different areas so that total travelling costs are minimised. Linear programming is involved in all these problems and the scope of possible applications of resource allocation within the civil engineering sphere is very large.

www.EngineeringBooksPDF.com

The third example of chapter 2 was that of steelwork design. In fact this example perhaps gives a false impression of the usefulness of linear programming for technological design. Most engineering design does not lead to linear mathematical models. Usually the models are highly non-linear. A more detailed study of the non-linear problems arising in detailed engineering design appears later in this book. There are some small exceptions to this general rule and the largest exception is probably that of the plastic design of structures. In plastic design the governing kinematic equations happen to be linear ones in the design variables and consequently there are several very useful applications for linear programming in plastic structural design. Even in the design example given, although the constraints were naturally linear in the fully plastic moment variables, the objective function is really a non-linear one which had to be approximated and forced into linear form for the purposes of the example. As a rule of thumb it is usual to look for non-linear problems in technological design and for linear problems in all the other aspects of planning, construction, operation and management.

To conclude this section on civil engineering uses of linear programming a final example, drawn from the area of water resource management, is described.

EXAMPLE 3.8 – WATER RESOURCE MANAGEMENT

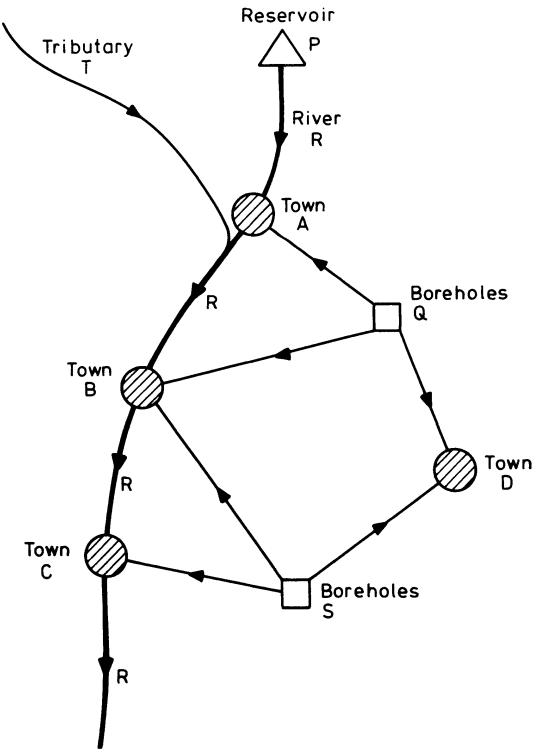


Figure 3.10 Water resources management example

www.EngineeringBooksPDF.com

Figure 3.10 shows the plan of a water authority's supply facilities and the towns served by them. Three towns A, B and C lie on the banks of a river, R, and they can abstract water from the river for treatment and use. The river is dammed upstream of the towns and the river flow is partially controlled by limiting the amount of water released from the reservoir, P, behind the dam. Just below town A a tributary, T, joins the river and contributes an uncontrolled flow to it. Towns A and B can also be served with water from boreholes at Q which tap an underground aquifer. Another set of boreholes at S can supply towns B and C. A smaller town, D, not on the river, is supplied solely by the boreholes at Q and/or S.

Towns A, B, C and D have a daily water demand of D_A, D_B, D_C and D_D, respectively. The boreholes at Q and S can supply a maximum of S_Q and S_S, respectively. The reservoir can release a maximum of S_P into the river and the tributary contributes S_T to the river flow. Riparian rights require that the river always has a flow of at least S_R in it.

The water authority's problem is how to operate or manage its facilities to satisfy all requirements as cheaply as possible. Formulate this as a linear programming problem.

This problem is clearly akin to a transportation problem. Water is the transported commodity between sources with limited supplies and destinations with demands. The river, in this problem, is both a source for towns A, B and C, and a destination for the reservoir P and tributary T. Therefore, let x_{ij} be the water quantity supplied to destination i = A, B, C, D, R by source j = Q, S, P, T, R. The variables in the LP problem will then be $x_{AR}, x_{BR}, x_{CR}, x_{AQ}, x_{BQ}, x_{BS}, x_{CS}, x_{DQ}, x_{DS}, x_{RP}$. Note that x_{RP}, the amount of water released to the river from the reservoir, is a variable because it can be controlled but there is no variable x_{RT} representing the tributary flow. The tributary flow cannot be controlled and will have the known value S_T.

The constraints arise in several ways. Firstly, the demand constraints for towns A, B, C and D can be expressed as

$$\text{for A}, x_{AR} + x_{AQ} = D_A \tag{3.19}$$

$$\text{for B}, x_{BR} + x_{BQ} + x_{BS} = D_B \tag{3.20}$$

$$\text{for C}, x_{CR} + x_{CS} = D_C \tag{3.21}$$

$$\text{for D}, x_{DQ} + x_{DS} = D_D \tag{3.22}$$

Riparian rights also produce demand constraints since the river flow in reaches PA, AB, BC and below C must all be at least equal to S_R. Thus

$$\text{for PA}, x_{RP} \geqslant S_R \tag{3.23}$$

$$\text{for AB}, x_{RP} - x_{AR} \geqslant S_R - S_T \tag{3.24}$$

$$\text{for BC}, x_{RP} - x_{AR} - x_{BR} \geqslant S_R - S_T \tag{3.25}$$

$$\text{below C}, x_{RP} - x_{AR} - x_{BR} - x_{CR} \geqslant S_R - S_T \tag{3.26}$$

www.EngineeringBooksPDF.com

Finally, there are supply constraints representing the supply capacities of the various sources. These are

$$\text{for P}, x_{RP} \leqslant S_P \tag{3.27}$$
$$\text{for Q}, x_{AQ} + x_{BQ} + x_{DQ} \leqslant S_Q \tag{3.28}$$
$$\text{for S}, x_{BS} + x_{CS} + x_{DS} \leqslant S_S \tag{3.29}$$

There are no direct supply limits from the river. Riparian rights 3.23 to 3.26 provide indirect limits on x_{AR}, x_{BR} and x_{CR}.

These linear relationships 3.19 to 3.29 represent all the problem constraints. A suitable objective function is that of cost. Water extracted from a river must be pumped and treated, from a borehole there will also be pumping, treating and transportation costs. Releasing water from a reservoir may have pumping costs associated with it and there may also be penalty costs depending upon how much or how little water is in the reservoir. Thus each of the variables x_{ij} will have a unit cost associated with it and generally all the unit costs will be different. For this example the cost function would be

$$\begin{aligned} C = {} & c_{AR}x_{AR} + c_{BR}x_{BR} + c_{CR}x_{CR} + c_{AQ}x_{AQ} + c_{BQ}x_{BQ} \\ & + c_{BS}x_{BS} + c_{CS}x_{CS} + c_{DQ}x_{DQ} + c_{DS}x_{DS} + c_{RP}x_{RP} \end{aligned} \tag{3.30}$$

This cost function should be minimised over non-negative variables $\boldsymbol{x}$ (obviously negative quantities of water are not permitted) subject to non-violation of constraints 3.19 to 3.29. This is an LP problem.

This example is typical of the problems which arise in the operation of water resource systems and it demonstrates the power of linear programming to represent and solve operational policy problems of a truly practical nature. This problem is slightly simplified but can be extended to cover other features of a typical system. For example this model assumes no losses of water from the river or infiltration and the reservoir is only crudely represented without any tributaries or catchment area. Nevertheless all these features could be incorporated in an LP model if desired. To use the model it is necessary to estimate all the demand and resource limits (the Ds and Ss) as well as the unit costs. Many of these will change in value over a period of time and regular updating of the operational policy which results from solving this problem is needed to ensure that the policy remains efficient and optimal. Sensitivity analysis can be useful in determining the stability of the policy. Thus water resource management depends heavily on systematic decision making and linear mathematical models.

The methods studied in this chapter have all been concerned with solving linear programming problems. It should not be inferred from this that all linear mathematical models must be LP problems. Linear algebra is a very large topic of which linear programming is only a part. This book, however, is concerned with decision-making methods and, of all the decision-making methods for linear problems, LP is by far the most important. In the next chapter network analysis methods are examined for the control and planning of construction projects. It is shown there that these network representations also yield linear mathematical

www.EngineeringBooksPDF.com

models. Many of these can be solved by linear programming, thus chapter 4 also contains many more practical examples of the use of linear programming in civil engineering.

SUMMARY

This chapter has described the simplex method for the solution of linear programming problems and has shown that the whole solution process is amenable to automatic computation. The sensitivity of optimum solutions to changes in the objective function coefficients and constraint bounds has been examined and it has been shown that the concept of duality is of importance both in sensitivity analysis and in efficient problem solving. Extensions of the simplex method and other methods have been described for handling negative variables, integer and discrete-valued variables and the usefulness of the concept of rounding the solutions of engineering problems has been discussed.

A brief survey of possible applications of linear programming to practical planning and decision-making problems has shown that linear programming models are ubiquitous in the planning, construction, operation and management phases of civil engineering projects. Linear programming is consequently a very important numerical technique with wide practical applications. This chapter has been restricted to linear programming problems and their solution yet the whole field of linear mathematical models is much broader and linear programming forms only a part of it. Chapter 4 examines further linear models.

BIBLIOGRAPHY

Dantzig, G. B., *Linear Programming and Extensions* (Princeton University Press, 1963)

Garfinkel, R. S., and Nemhauser, G. L., *Integer Programming* (Wiley, New York, 1972)

Garvin, W., *Introduction to Linear Programming* (McGraw-Hill, New York, 1960)

Gass, S. L., *Linear Programming Methods and Applications* (McGraw-Hill, New York, 1964)

Hadley, G. H., *Linear Programming* (Addison-Wesley, Reading, Mass., 1962)

EXERCISES

3.1 Solve the following problem by the simplex method using $x_1 = x_2 = 0$ as an initial feasible point.

$$\begin{aligned} &\text{Minimise } f = -5x_1 + 2x_2 \\ &\text{subject to } 2x_1 + x_2 \leqslant 9 \\ &\qquad\qquad x_1 - 2x_2 \leqslant 2 \\ &\qquad\qquad -3x_1 + 2x_2 \leqslant 4 \\ &\qquad\qquad x_1, x_2 \geqslant 0 \end{aligned}$$

www.EngineeringBooksPDF.com

3.2 Solve the following problem by the simplex method using $x_1 = x_2 = 12$ as an initial feasible point.

$$\begin{aligned}\text{Minimise } & f = x_1 + 3x_2 \\ \text{subject to } & x_1 + 4x_2 \geqslant 48 \\ & 5x_1 + x_2 \geqslant 50 \\ & 2x_1 - x_2 \leqslant 19 \\ & x_1, x_2 \geqslant 0\end{aligned}$$

3.3 Solve the following problem by the simplex phase I and phase II procedures.

$$\begin{aligned}\text{Minimise } & f = 5x_1 + x_2 \\ \text{subject to } & 2x_1 + x_2 \geqslant 2 \\ & -x_1 + x_2 \leqslant 1 \\ & x_1 + 2x_2 \geqslant 2 \\ & x_1, x_2 \geqslant 0\end{aligned}$$

3.4 Use the simplex method to solve the problem

$$\begin{aligned}\text{Maximise } & f = x_1 + x_2 - 2x_3 \\ \text{subject to } & 3x_2 - 2x_3 = 6 \\ & 1 \leqslant x_1 \leqslant 4 \\ & -1 \leqslant x_3 \leqslant 2 \\ & x_2 \geqslant 0\end{aligned}$$

3.5 Use the simplex method to solve the problem

$$\begin{aligned}\text{Minimise } & f = x_1 + 2x_2 - x_3 + x_4 \\ \text{subject to } & x_1 - x_2 + x_3 \leqslant 4 \\ & x_1 + x_2 + 2x_4 \leqslant 6 \\ & -x_2 + 2x_3 - x_4 \geqslant -2 \\ & x_1 - 2x_2 + x_3 \geqslant -2 \\ & x_1, x_4 \geqslant 0 \\ & x_2, x_3 \text{ unrestricted in sign}\end{aligned}$$

3.6 Solve the problem

$$\begin{aligned}\text{Maximise } & f = 2x_1 + x_2 + x_3 \\ \text{subject to } & x_1 + x_2 + x_3 \leqslant 10 \\ & x_1 + 5x_2 + x_3 \geqslant 20 \\ & x_1, x_2, x_3 \geqslant 0\end{aligned}$$

If the objective function is changed to $f = cx_1 + x_2 + x_3$ determine the range of values of c for which the previously derived solution remains optimal and express the maximum value of the objective function as a function of c.

www.EngineeringBooksPDF.com

3.7 A linear programming problem has the following constraints

$$\begin{aligned} 2x_1 + 2x_2 &\geqslant 60 \\ 2x_1 + 4x_2 &\geqslant 80 \\ 4x_1 &\geqslant 60 \\ 4x_2 &\geqslant 20 \\ 3x_1 + 3x_2 &\leqslant 120 \\ x_1, x_2 &\geqslant 0 \end{aligned}$$

The point $x_1^* = x_2^* = 15$ solves the LP problem giving the objective function a minimum value of 150. If the coefficient of x_1 in the objective function is doubled the point $x_1^* = x_2^* = 15$ still remains optimal although with a different value of the objective function. What is the objective function?

3.8 A precasting plant produces three types of wall panels. The first type requires 1 h of labour and 3 h of machinery time and sells at £13 per unit. The second type requires 2 h each of labour and machinery time and also contains 2 m^3 of an insulation material. This type sells at £22 per unit. One unit of the third type requires 2.5 h of machinery time, 1.5 h of labour, contains 4 m^3 of insulation material and sells for £30.

Labour rates are £2 per hour, machinery time costs £3 per hour and the insulation material costs £4 per m^3. In a particular work period the precasting plant has available 240 h of machinery time, 160 h of labour and 150 m^3 of insulation material. Demand is for at least ten of each type of panel.

How many of each type of panel should be produced during the work period so as to maximise profits assuming that all will be sold?

Labour unions are negotiating a wage deal which will effectively increase the labour rate from £2 per hour to £2.75 per hour. The company has to pay fixed overhead charges of £150 out of total profits during the work period. Can the company afford to agree to the new wage deal? If the new rates are implemented will the production schedule be altered and what will be the company's net profit or loss?

3.9 A ready-mixed concrete company has an order to supply 1000 m^3 of concrete for a large pour. The grading of the mix requires 30 per cent by volume large aggregate, 35 per cent medium aggregate and 20 per cent fines. The balance is cement and water. The concrete mix sells at £30 per m^3 and the company's objective is to fulfil the order at maximum profit. The company runs two mixing plants; A could supply up to 600 m^3 for the pour and B could supply up to 750 m^3.

Figure 3.11 shows the units A and B as square nodes on a network. Each unit buys its aggregate from three possible sources, C, D, and E, shown as circular nodes on the network. The links represent possible transportation routes between nodes and the number associated with each link is the cost in £ sterling of carrying 1 m^3 of any size of aggregate between the end nodes of the link.

www.EngineeringBooksPDF.com

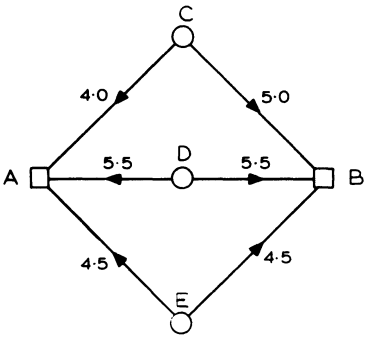


Figure 3.11

The sources C, D and E can produce aggregates in any quantities up to the limits shown in the table and at production costs as shown in £/m^3. These production costs are additional to any transportation costs.

Source	Large aggregate		Medium aggregate		Fine aggregate	
	m^3	£/m^3	m^3	£/m^3	m^3	£/m^3
C	200	5.0	150	4.5	–	–
D	125	5.3	150	6.0	100	6.5
E	–	–	150	8.0	150	7.0

Cement, water and mixing costs are the same at both batching units and total £15/m^3.

Formulate a linear programming problem which would enable the company to determine how to fulfil the order at maximum profit.

www.EngineeringBooksPDF.com

4 PROJECT PLANNING METHODS, NETWORKS AND GRAPHS

The initial stages of any civil engineering project – feasibility studies, macro-design and detailed design – define a goal. The construction phase translates that definition into reality and very careful planning is necessary to ensure that the goal is achieved efficiently and economically. Consequently the construction phase has its own planning methods which aim to specify precisely how the project will be carried out, determine what resources of materials, plant, labour and money will be required, schedule resources to be available at the right times and generally monitor the progress of the entire undertaking. One of the most powerful and widely used construction planning methods is that of network analysis. This chapter describes how networks are prepared for construction projects, how they are used to allocate resources and maintain control of the construction work.

A detailed examination of the full panoply of construction planning methods is, however, not the prime purpose of this chapter. Many books have already been written on construction planning and management and many of the techniques used will be familiar to students and graduates of civil engineering degree courses. This chapter views construction planning in a systems context and identifies the concept of a network as being fundamental to construction planning and as having considerable potential for use as a decision-making tool outside construction planning. Consequently only the network aspects of construction planning are considered here in detail. This chapter shows that construction planning networks are basically linear and that many network analysis problems can be solved by linear programming. This provides links with the earlier chapters. Other civil engineering problems that can be represented as networks are then examined and the linear nature of these problems is used to derive quick algorithms for solving longest and shortest path problems and network flow problems. Consideration is then given to some simple graph problems and solution methods exploiting linearity are suggested. The main thrust of this chapter is therefore towards demonstrating that the network representation and its associated mathematics are very powerful decision-making aids. There are many important

www.EngineeringBooksPDF.com

applications of network theory in civil engineering planning and management; construction planning networks are merely examples of a much wider class of problems.

4.1 CONSTRUCTION PLANNING NETWORKS

Since the Second World War the major development in construction planning techniques has been network analysis. There are today few civil engineers who are unfamiliar with CPM (critical path method) and some of its derivatives. Contrary to the beliefs of many civil engineers, however, CPM and network analysis methods do not have their origins in civil engineering. They are essentially imported techniques which have very firm roots in the mathematical theories of graphs and networks. The use of network analysis methods for construction planning can be set in a much broader perspective if these origins are examined and understood. Of course there are many other methods in addition to network analysis which are used as decision-making aids in construction planning. Bar charts or Gantt charts, for example, are much used to considerable effect. Indeed their use in conjunction with network methods increases the usefulness and value of both chart and network methods. Unfortunately space does not permit any examination of chart methods or other construction planning aids. Specialist construction planning texts, of which there are many, should be consulted for a comprehensive view of the topic. The purpose of this chapter is to offer a slightly different perspective on network analysis methods and to give a mathematical background to their use in construction planning. This section describes how construction planning networks are drawn, analysed and used.

4.1.1 Drawing a Network Diagram

During the feasibility studies and design stages of any civil engineering project considerable thought will have been given to possible construction methods which might be used. Indeed the final designs for a project are often dependent not just on how the project will look or behave when it is finished but on how it will be constructed. For example, in the detailed design of a structure the critical loadings which govern the design are frequently those which occur during construction when the structure is only partially completed and has not yet developed the resistive strength of its finished form. Therefore, when the time comes to plan the construction work in detail a general strategy for the construction should already be known.

A planning network is simply a means of representing this construction strategy in diagrammatic form. The drawing of a network diagram is an essential preliminary to allocating resources of time, money, labour, etc., to the various elements of the strategy. This section examines first of all how network diagrams are drawn and then how they are used to allocate resources. In practice, strategy

www.EngineeringBooksPDF.com

and resource allocation are not quite as independent as this. For example, completely different construction strategies, and hence different network diagrams, would be required for a project depending upon whether or not a major resource such as a tower crane will be available. In drawing a network diagram for a project, therefore, some general idea of the likely available resources is necessary, although the precise allocation of these resources is left until later in the planning process.

Network analysis requires that the over-all construction strategy should be analysed into a set of individual *activities*. An activity is an operation that requires resources and/or time for its performance. Thus 'fix column reinforcement' or 'pour concrete' or 'await concrete hardening' are construction activities. All activities involved in a construction project must be known and should be listed, preferably in an approximate order in which they must be carried out. Having listed all activities for a project their interrelationships must be studied. Each activity is considered in turn and the answers to three basic questions are sought

(1) Which other activities must be completed before this activity can start?
(2) Which other activities may proceed concurrently with this one?
(3) Which other activities cannot start until this activity is completed?

The answers to these questions require careful thought about precisely how the construction work will be done. One of the big advantages of network planning is that it imposes this need for thought and leaves little margin for uncertainty about the construction methods to be used.

Once the list of activities and their interrelationships are known, a network diagram can be drawn. In a network diagram each activity is represented by a separate arrow with the direction of the arrow representing the direction of time-flow through the activity. The tail of an arrow represents the commencement of an activity and the head represents its finish. At each end of an activity arrow there must be an *event*. An event is simply a point in time after the completion of all preceding activities and before the commencement of all succeeding activities. Events at each end of an activity arrow are represented by circles in this book. The head of an arrow must have a *head event* and the tail a *tail event*. Each activity is drawn on the network diagram in such a way that all interrelationships among activites are correctly represented. In drawing network diagrams the lengths and orientations of activity arrows are not of any significance. The shape and scale of the network is not fixed. For clarity and ease of interpretation, however, certain conventions are usually adopted. These are that, wherever possible, the over-all flow of time through a network should be from left to right and activity arrows should be composed of straight lines not curved ones. The vital element of network diagrams is that the logic of the interrelationships among activities must be correctly reproduced.

Some examples will clarify the points made above. Consider the laying of a long length of new sewerage beneath an existing road. This could be re-

www.EngineeringBooksPDF.com

presented by a single activity arrow as in figure 4.1 with the activity labelled as 'sewerage'. If the sewerage were a very small part of a much larger project then this representation might have some value as part of a much larger network diagram. Construction of the network diagram, however, is only a preliminary to its analysis and use for resource allocation and control. An activity such as figure 4.1 is not really of much value in assessing what resources need to be allocated to it or how long it will take to finish. For the network to be of value an operation such as this requires more detailed study and a more precise network representation. The necessary operations in laying a length of new sewerage are usually as follows

(1) excavating a trench for the pipe run
(2) laying and bedding individual pipe lengths
(3) sealing the joints between individual pipe lengths
(4) backfilling the trench
(5) reinstating the road surface.

These five activities are certainly more detailed than the single activity of figure 4.1. At first glance they appear to be sequential and this suggests a network diagram such as figure 4.2.

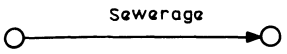


Figure 4.1 Representation of a single activity

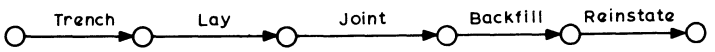


Figure 4.2 Simplistic representation of sewerage operations

Figure 4.2 would be much more useful than figure 4.1 at the analysis and resource allocation stage but it is still too simple because it does not represent the way in which long sewerage runs are laid in normal practice. Figure 4.2 implies by its logical arrangement that all excavation work must be completed before any pipe laying begins. Each activity must be completed before the succeeding activity can begin. Thus figure 4.2 represents a very time-consuming and inefficient way of laying sewerage. Still more thought is needed about the project before a truly representative and useful network diagram can be drawn.

Figure 4.3 represents a much better network diagram that corresponds more closely to construction practice. The long pipe run has been divided into smaller lengths of perhaps 75 to 100 m each. Four such lengths have been chosen for the purposes of the example and are denoted by subscripts 1 to 4. Starting at one end of the over-all length, the first length of trench is excavated (T_1). A soon as this is finished, pipe laying begins in that trench (L_1) while

www.EngineeringBooksPDF.com

excavation of the second length of trench continues (T_2). When the first length of pipe is laid, joint sealing begins on that length (J_1). As soon as the second length of trench is excavated, pipe laying begins in that trench (L_2) and excavation of the third length of trench commences simultaneously. Thus at any time there are several activities proceeding together and figure 4.3 depicts them all precisely. Figure 4.3, therefore, represents a useful and accurate statement in graphical form of the strategy used in laying the new sewerage.

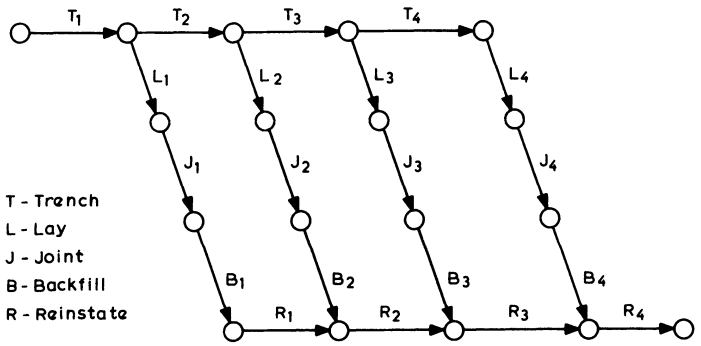


Figure 4.3 A more useful network for sewerage construction

Another example stresses the need to maintain the logic of the interrelationships among activities in the network diagram. The following activities for a project have the interrelationships shown.

Activities	Interrelationships
A	Independent
B	Independent
C	Depends on completion of A
D	Depends on completion of A
E	Depends on completion of A
F	Depends on completion of A and B
G	Depends on completion of C and D
H	Depends on completion of D
J	Depends on completion of E and F
K	Depends on completion of E and F
L	Depends on completion of G, H and J
M	Depends on completion of K and L

These activities lead to the network diagram shown in figure 4.4. In order to preserve the logic of the interrelationships in figure 4.4, two *dummy activities* have been introduced, shown as broken activity arrows. One of these links the

www.EngineeringBooksPDF.com

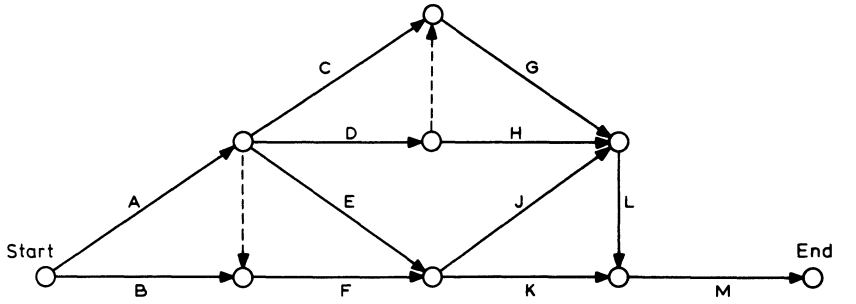
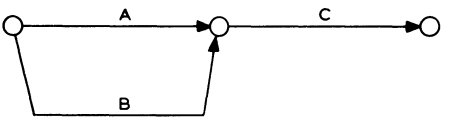


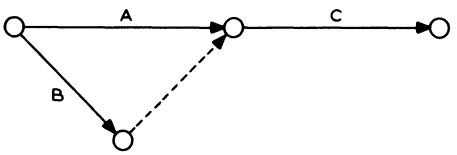
Figure 4.4 Network diagram with dummy activities

end events of activities A and B and is needed to ensure that activity F does not commence until both A and B are completed as required by the interrelationships. The other dummy activity is needed to show that activity G depends upon the completion of both C and D. A dummy activity is solely a logic-preserving artifice which consumes no resources or time. In figure 4.4 the event preceding activities A and B is termed a *start event* because it has no preceding activities. The event at the end of activity M is an *end event* because it has no succeeding activities. A planning network may have only one start event and one end event.

In drawing network diagrams for planning purposes it is usual to draw only one activity arrow between a pair of events. This ensures that activities are all unique with no possibility of ambiguity if an activity is specified by its start and end events only. If more than one activity is required between two events then dummy activities and extra events should be used. Figure 4.5 shows correct and incorrect representations of this.



a) Incorrect. Only one activity permitted between two events.



b) Correct. Dummy activity and extra event introduced.

Figure 4.5

www.EngineeringBooksPDF.com

Since planning network diagrams are essentially representations of logic, they should not contain any closed loops such as is shown in figure 4.6. A loop represents a set of illogical relationships; activities within loops could never be started. Illogical activity interrelationships, if they are present, usually become apparent at the network analysis stage and they should be removed. Essentially, illogicalities should not occur at all. If they do arise, more detailed thought about the construction strategy is necessary because either the strategy is an impossible one or, more likely, it has been incorrectly drawn on the network diagram.

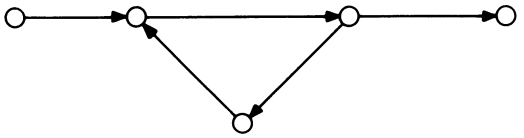


Figure 4.6 Closed loops are illogical and must not occur in a construction planning network

4.1.2 Analysing a Network Diagram

Having drawn a network diagram for a construction project the next step is to use it to study the time requirements of its constituent activities. Since only one activity exists between any pair of events, an activity could be specified by its start and end events if those are numbered. Firstly, therefore, all the events on the network must be numbered. In theory any random numbering scheme for the events could be used. Since time flow through an activity is in the direction from the start event to the end event it is usual to number the events in such a way that for all activities the preceding events have a lower event number than the succeeding events. As will be shown, many network problems of scheduling, resource allocation, etc., are amenable to automatic solution using a digital computer and it is sensible, therefore, to number the events in a way that is efficient from the point of view of computer data manipulation. The following very simple algorithm will number all the events in such a way that the start and end events of a construction network will have the lowest and highest event numbers, respectively, and all activities will have a succeeding event number higher than that of any preceding events.

4.1.2.1 Event-numbering Algorithm

Step 1: Give the start event of the network event number 1
Step 2: Give the next number in an ascending series to any unnumbered event whose predecessor events are all already numbered. Repeat Step 2 until all events have been numbered.

Having numbered all the events in a network the activities may now be referred to by the event numbers at their ends.

www.EngineeringBooksPDF.com

4.1.2.2 Allocation of Activity Durations

The next stage in network analysis is a most crucial one and it requires that a time duration should be specified for all activities within the network. Up to this point the network drawing process has been entirely logical. The sequence of events and activities has been determined by the nature of the specific construction project and there has been very little scope for choice. Now the nature of the network analysis process changes and for the first time an element of choice enters the process. In order to specify, as required, a time duration for every activity some detailed knowledge of available resources of men, money and plant is needed. Furthermore, these resources are closely related; the duration of any activity may usually be shortened by using more men and extra plant, which implies extra cost, or increased by withdrawing resources. Thus it can be concluded that activity durations are closely linked to activity costs and resource availability. These relationships will be examined later in this chapter but in order to make progress now it will be assumed that it is possible to estimate for each activity an 'average, likely duration'.

As soon as activity durations have been estimated for all activities in the network it is possible to calculate times at which the network events can occur. For each event two event times are important – the earliest and the latest event times. The *earliest event time* is the earliest possible time after the start of the project at which that event could occur while satisfying the logic of the network.

4.1.2.3 Earliest Event Time Algorithm

Define E_i to be the earliest event time for event i and t_{ij} to be the duration of the activity occurring between events i and j with $i < j$. Then, for a network with n events

Step 1: $E_1 = 0$

Step 2: For $j = 2, 3, \ldots, n$

$$E_j = \text{maximum } \{E_i + t_{ij}\}$$

with the maximisation extending over all activities which terminate at event j.

The earliest event time algorithm ends by assigning an earliest event time to the end event, E_n. This is clearly the earliest completion time for the project. It may be possible, however, for some events to occur at a time later than their earliest event times and still to achieve a project completion time of E_n. The *latest event time* is the latest possible time after the start of a project at which the event could logically occur and a project completion time of E_n be achieved.

4.1.2.4 Latest Event Time Algorithm

Define L_i to be the latest event time for event i in a network with n events.

www.EngineeringBooksPDF.com

Activity durations are as defined in the earliest event time algorithm. Then

Step 1: $L_n = E_n$

Step 2: For $i = n - 1, n - 2, \ldots, 2, 1$

L_i = minimum $\{L_j - t_{ij}\}$

with the minimisation extending over all activities which start at event i.

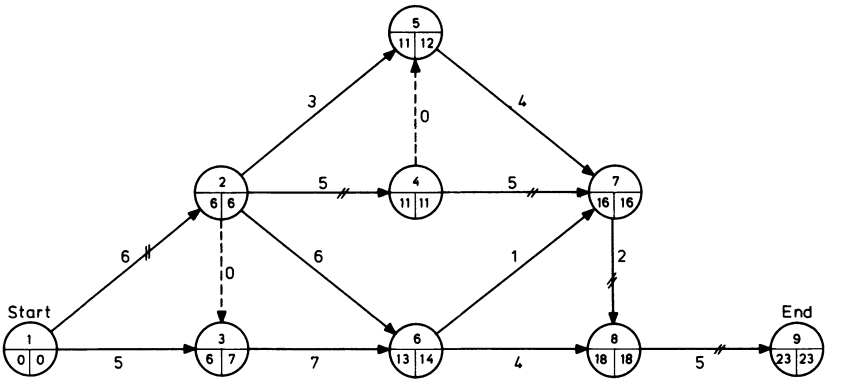


Figure 4.7 Event numbers and times for the network of figure 4.4

The operation of these algorithms may be studied by means of an example. Figure 4.7 shows the network of figure 4.4 with activity durations added alongside each activity. The event-numbering algorithm is used first to number all the events so that for each activity the head event number is larger than the tail event number. The start event is first allocated the number 1. In this book event numbers are placed in the upper half of the event circles; elsewhere other conventions are adopted. Having numbered event 1, the events at the heads of the two activity arrows leading out of event 1 are considered. Event 2 is numbered followed by event 3. Note that these two event numbers cannot be reversed. The event numbered 3 on figure 4.7 has two activity arrows leading into it and the tail events of *both* those arrows must be numbered before the head event can be numbered. Events 4 and 5 may not be reversed for the same reason. The event-numbering algorithm is straightforward in operation and will isolate illogicalities in the network such as closed loops, figure 4.6, which cannot be numbered logically. In general there is not a unique set of event numbers for a network. In figure 4.7, for instance, the events numbered 4, 5 and 6 could have been logically numbered 5, 6 and 4, respectively, and other entirely logical numbering patterns can be found for this example.

Once the events are numbered, event times are allocated to them starting with the earliest event times. In this book earliest event times are placed in the south-

www.EngineeringBooksPDF.com

west quadrant of the event circles. The earliest event time algorithm first allocates a time of zero to the earliest time of event 1. It then considers event 2. Only one activity arrow leads into event 2 so the earliest time for event 2 is given by the earliest time of the event at the tail of that activity, $E_1 = 0$, plus the activity duration, $t_{12} = 6$, that is, $E_2 = 6$. Moving to event 3, two activity arrows lead into this event so possible earliest event times for event 3 must be calculated for each incoming path and the larger value selected. Thus

$$E_3 = \text{maximum} \begin{Bmatrix} E_1 + t_{13} \\ E_2 + t_{23} \end{Bmatrix}$$

$$= \text{maximum} \begin{Bmatrix} 0 + 5 \\ 6 + 0 \end{Bmatrix}$$

$$E_3 = 6$$

Thus the earliest event time algorithm embodies the principle that an event cannot take place until *all* preceding activities have been completed. Event 7 cannot occur until all the three activities leading into it are completed. To find the earliest time for even 7 we must, therefore, examine three possibilities

$$E_7 = \text{maximum} \begin{Bmatrix} E_5 + t_{57} \\ E_4 + t_{47} \\ E_6 + t_{67} \end{Bmatrix}$$

$$= \text{maximum} \begin{Bmatrix} 11 + 4 \\ 11 + 5 \\ 13 + 1 \end{Bmatrix}$$

$$E_7 = 16$$

In this way earliest event times can be calculated and entered on the network diagram. The end event, event 9, is found to have an earliest time of 23 units. This represents the absolute minimum time necessary to complete all the activities on the network and hence complete the project. At this point it should be noted that the value of 23 for the earliest time of the end event is equal to the sum of the durations of activities 1-2, 2-4, 4-7, 7-8 and 8-9. These activities form a continuous path through the network from the start event to the end event. This path is the longest duration path; all other paths are composed of activities whose durations sum to less than 23 units.

The minimum duration for the project, 23 time units, was calculated assuming that all activities start at their earliest possible times. It is not always convenient or economical in a construction project to start all activities at their earliest possible times. Consequently it is very useful to know by how much the starts of activities may be delayed without affecting the total duration of the project. The latest event time algorithm is used to give this information. The algorithm starts by assuming that the project duration of 23 time units must be achieved.

www.EngineeringBooksPDF.com

It sets the latest time of the end event, event 9, to 23 units, that is, $L_9 = 23$. It then moves backwards to event 8. Only one activity arrow leads out of event 8 so the latest time for event 8 is given by the latest time of the event at the head of that activity arrow, $L_9 = 23$, minus the duration of that activity, $t_{89} = 5$, that is, $L_8 = 18$. Latest event times are placed in the south-east quadrant of the event circles. Moving backwards to event 7, again only one activity arrow leaves event 7. Therefore

$$\begin{aligned} L_7 &= L_8 - t_{78} \\ &= 18 - 2 \\ L_7 &= 16 \end{aligned}$$

Moving backwards again to event 6, two arrows lead out of this event so two possible latest event times must be calculated corresponding to each path out of event 6 and the smaller value selected. Thus

$$\begin{aligned} L_6 &= \text{minimum} \begin{Bmatrix} L_7 - t_{67} \\ L_8 - t_{68} \end{Bmatrix} \\ &= \text{minimum} \begin{Bmatrix} 16 - 1 \\ 18 - 4 \end{Bmatrix} \\ L_6 &= 14 \end{aligned}$$

The principle embodied in the latest event time algorithm is that each event should be timed as late as possible and still allow *all* subsequent activities to be completed within a specified project duration.

Examination of the events on figure 4.7 shows that some of them have identical earliest and latest times, for example, events 1, 2, 4, 7, 8 and 9. Clearly, if the earliest and latest times for an event are the same then that event may not be altered in time; it is termed a *critical event*. Any alteration to the timing of a critical event will alter the over-all duration of the project. Other events, for example, events 3, 5 and 6, have earliest and latest times that are not the same. For these events there is some scope for choice of when the event should actually occur. Providing an event occurs at a time somewhere between its earliest and latest times the over-all project duration will not be altered. Thus far, then, the analysis of the network has provided valuable information on which events are critical ones having a very direct effect on the completion of the project and which events are not critical to the completion time.

It is also possible to deduce similar information about critical and non-critical activities. A *critical activity* is defined as an activity with critical events at each end whose time difference is equal to the duration of the activity. On figure 4.7 activity 2-4 is a critical activity because both event 2 and event 4 are critical events and the duration of activity 2-4 of 5 time units is equal to the difference (11-6) of the times of the critical events. Other critical activities on figure 4.7 are activities 1-2, 4-7, 7-8 and 8-9. It is possible for an activity to have critical events at each end and not to be a critical activity if the difference in event times

www.EngineeringBooksPDF.com

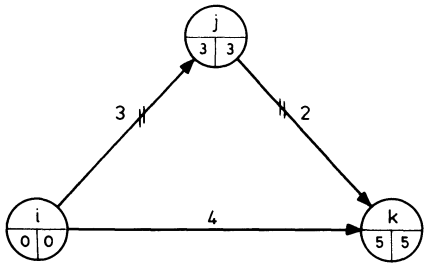


Figure 4.8 Critical and non-critical activities

exceeds the activity duration. Figure 4.8 demonstrates this. Activities *i–j* and *j–k* are critical activities but activity *i–k* is not critical. On figure 4.7 the critical activities have been marked by means of short diagonal lines. All the critical activities and critical events lie on a single path which traverses the network from the start to the end events. A path such as this composed of only critical events and critical activities is called a *critical path*. Summing the activity durations along a critical path gives the shortest possible completion time for the project. Clearly any alteration to an event time or an activity duration on a critical path must alter the project completion time. It is essentially this property of a critical path that makes it and its constituent events and activities critical. As has already been noted, the critical path is that path along which the sum of activity durations is greatest. Figure 4.7 has a single critical path. Generally, networks may have more than one critical path but the sum of the activity durations along each critical path in a network must be the same. Owing to the importance of the critical path, this network analysis approach is usually referred to as the *critical path method* (CPM).

Activities and events that do not lie on a critical path are by definition not critical. It is possible to alter the timing of non-critical events and adjust the durations of non-critical activities without affecting the duration of the project. Non-critical events have different earliest and latest event times and the difference between these is usually termed the *slack* of an event. The slack in an event is closely similar to the idea of slack in a constraint as described in the previous chapter. It is a measure of the degree of flexibility in the timing of the event. Critical events have no slack and are, therefore, analogous to tight constraints. These analogies with the linear programming concepts described in chapter 3 are not merely fortuitous. They are examined further later in this chapter. Conventionally *non-critical events* have slack and the corresponding term describing any flexibility in the durations of *non-critical activities* is *float.*

An examination of the example network of figure 4.7 reveals that there are different classes of activity floats each imposing different conditions upon the construction operations. Consider activity 6–7 which has a duration of 1 time unit. If its preceding event occurs at its latest event time, 14, and its succeeding

www.EngineeringBooksPDF.com

event occurs at its earliest event time, 16, then the minimum duration available for this activity is the difference of these times, that is, 2 units. The actual activity duration, however, is only 1 unit; therefore the activity 6-7 is said to have an *independent float* of (2-1) units, that is, 1 unit. This means that activity 6-7 could commence 1 unit of time late, or finish 1 time unit early or be rearranged to take 2 time units instead of 1, without in any way affecting the event times of the network. Similarly, activity 2-5 has an independent float given by

$$\begin{aligned}\text{independent float, } 2\text{-}5 &= E_5 - L_2 - t_{25} \\ &= 11 - 6 - 3 \\ &= 2\end{aligned}$$

Thus two extra units of time are available for activity 2-5 in addition to its estimated duration of 3 units. This activity can be rearranged to make maximum beneficial use of this extra time without in any way affecting the event times of the network.

Returning to activity 6-7, another form of float can be identified. If both preceding and succeeding events start at their earliest times, the time available for activity 6-7 is $E_7 - E_6 = 3$ units. Subtracting the activity duration of 1 unit gives a *free float* of 2 units. It has already been shown that an independent float of 1 unit is available to this activity and that use of this makes no demands upon the rest of the network. Now it has been shown that more float is available, that is, a free float of 2 units, which can be used if required. The penalty to be paid, however, for using up free float rather than independent float is that it is essential for the preceding event to take place at its earliest time. This, therefore makes some restrictions on all activities leading up to the preceding event in order that it shall occur at its earliest time. A final form of float, for which the network of figure 4.7 has no good example, is known as the *total float* and is given by taking the earliest time of the preceding event from the latest time of the succeeding event and subtracting the activity duration. Total float makes maximum use of any slack in the events at both ends of the activity; it therefore represents the very maximum flexibility available for an activity. To use up the total float of an activity, however, imposes restrictions on both preceding and succeeding activities in the network since preceding and succeeding events must be arranged to occur at their earliest and latest times, respectively.

The float in the non-critical activities of a network is a very important feature since it represents flexibility in a project. Float is an acceptable way of soaking up the unforeseen occurrences and delays which happen in any project. It also offers a framework within which resources may be allocated to make the project run as smoothly and efficiently as possible. In the context of resource allocation it is important to know how the float is distributed among independent, free and total float. Independent float is certainly the easiest to use for resource allocation purposes since it has no effect on the network as a whole. The use of free or total float requires careful planning of preceding and succeeding activities.

www.EngineeringBooksPDF.com

4.1.2.5 The Network Analysis Table

The network diagram of figure 4.7 displays the analysis of a project in pictorial form. It is often useful to reduce this information to a tabular form which shows all information relating to each activity. The analysis table is a conventional means of doing this. Table 4.1 shows the analysis table for the network of figure 4.7. First of all every activity is listed in ascending order of the preceding and succeeding events. The durations of activities are then added. In construction management the major interest lies in activities rather than events, so the next four columns list the start and finish times for each activity. In each case an earliest and latest time is quoted. Note that these are not earliest and latest event times but earliest and latest activity start and finish times.

Table 4.1 Analysis table for the network of figure 4.7

Activity	Duration	Start		Finish		Float			
		Early	Late	Early	Late	Total	Free	Independent	
1-2	6	0	0	6	6	0	0	0	Critical
1-3	5	0	2	5	7	2	1	1	
2-3	0	6	7	6	7	1	0	0	
2-4	5	6	6	11	11	0	0	0	Critical
2-5	3	6	9	9	12	3	2	2	
2-6	6	6	8	12	14	2	1	1	
3-6	7	6	7	13	14	1	0	0	
4-5	0	11	12	11	12	1	0	0	
4-7	5	11	11	16	16	0	0	0	Critical
5-7	4	11	12	15	16	1	1	0	
6-7	1	13	15	14	16	2	2	1	
6-8	4	13	14	17	18	1	1	0	
7-8	2	16	16	18	18	0	0	0	Critical
8-9	5	18	18	23	23	0	0	0	Critical

Earliest start time is the earliest possible moment at which an activity may start. It is equal to the earliest event time of the preceding event.
Latest start time is the latest possible moment at which an activity may start. It is equal to the latest event time of the succeeding event minus the activity duration.
Earliest finish time is the earliest possible moment at which an activity may finish. It is given by the earliest event time of the preceding event plus the activity duration.
Latest finish time is the latest possible moment at which an activity may finish. It is equal to the latest event time of the succeeding event.

www.EngineeringBooksPDF.com

Using these definitions, the four activity start and finish times may be entered for each activity. The next three columns list the three different types of float associated with each activity. These values are calculated using the definitions of float already given. Note that occasionally an activity may appear to have a negative independent float. In such cases it is usual to take the independent float as zero. In figure 4.7 activity 3–6 appears to have a negative independent float. A final column merely notes which activities are critical. These are easily identified by the fact that no float of any type is associated with them.

4.1.2.6 Uncertainty in Activity Durations

As was noted in section 4.1.2.2 a crucial stage in the network analysis process described above is that of assigning durations to all activities. The network analysis depends very much upon the estimates made for 'average, likely durations' of activities. If any durations differ appreciably from those originally estimated it is possible for the critical path to change and to invalidate the entire network analysis. It is very difficult, however, to be absolutely precise in estimating activity durations. Some degree of uncertainty is inherent in any prediction. Can the network analysis process be modified to permit this uncertainty to be recognised and planned for?

A network analysis approach known as *PERT* (Program Evaluation Review Technique) attempts to incorporate uncertainty in its network analyses. This chapter would be incomplete without some reference to PERT but for a fuller description other books such as those by Pilcher, Battersby or Moder and Phillips should be consulted. In the PERT approach an activity is not represented by a single duration but rather by a frequency distribution of durations. For any particular activity the frequency distribution represents the spread of durations which have occurred for similar activities in the past, ranging from very short durations which are not achieved very often, through most likely durations which are often achieved to the rarely occurring very long duration. The *β-distribution* is assumed to fit the form of activity frequency distributions, although this assumption can only be very tenuously justified. With the assumption of a β-distribution for all activity durations, however, the planner is required to estimate three likely durations for each activity. These are

a *pessimistic* estimate, a, of the duration of the activity if everything goes badly;
a *most likely* estimate, b, of the duration; and
an *optimistic* estimate, c, of the duration of the activity if everything goes perfectly.

These three duration estimates are then used to calculate an *expected value* for the activity duration, d, from the relationship

$$d = (a + 4b + c)/6 \tag{4.1}$$

www.EngineeringBooksPDF.com

which results from the β-distribution assumption. The variance of the distribution is given by

$$v = \left(\frac{a - c}{6}\right)^2 \tag{4.2}$$

The quantities d and v are then used to analyse the network in place of the 'average, likely durations' previously used. If another assumption is made that all the activity durations are statistically independent of one another, (also a questionable assumption), then the central limit theorem can be invoked to calculate a distribution of project completion times. The project completion time will be a normal distribution whose mean is the sum of values of d for all activities on a critical path, and whose variance is the sum of the variances v for all activities on the same critical path.

Since PERT represents activity durations by frequency distributions rather than by single deterministic quantities, the nature of the entire analysis process is fundamentally altered. Instead of a single project completion time PERT provides a frequency distribution of completion times. This enables interesting questions to be asked such as: 'What is the probability that the project will be finished in x weeks?' and 'What is the probability for $(x + 1)$ weeks?. Quantities such as activity floats take on an entirely new complexion when cast into the probability domain. Indeed the entire network analysis, resource allocation and scheduling must now be carried in probabilistic terms. This certainly complicates the analysis and as a result PERT is much less commonly used in practice than the deterministic critical path analysis described in previous sections. On the credit side, however, PERT does provide direct measures of the uncertainty of the entire construction process whereas the deterministic critical path analysis pretends, quite falsely, that no uncertainty exists.

4.2 LINEAR PROGRAMMING AND CONSTRUCTION PLANNING NETWORKS

Section 4.1 described how construction planning networks are drawn and analysed. The methods outlined are frequently used in practice, are well-suited to hand calculation and can be easily understood in practical terms. They are also very simple to program for computer solution. They are, however, essentially of an *ad hoc* nature, designed for ease of use specifically for planning networks. They give little insight into the fundamental linearity of such networks and are of little value outside their specific application. This section describes other ways of solving planning network problems; methods which may be inferior in computational efficiency to those of the previous section but which bring out the basic concepts of linearity which are common to many network problems.

www.EngineeringBooksPDF.com

4.2.1 The Linearity of Activity Durations and Event Times

Consider an activity, i–j, in a network with i the preceding event and j the succeeding event. Let T_i and T_j be event times for i and j and let t_{ij} be the activity duration. Here $\boldsymbol{T}$ and $\boldsymbol{t}$ are generalised variables which may represent earliest, latest or actual times, estimated or actual durations, etc., as necessary. In order for the activity to be performed and completed between the event times it is necessary that the difference between the event times should be at least as long as the activity duration, that is

$$T_j - T_i \geqslant t_{ij} \tag{4.3}$$

Inequality 4.3 must be satisfied by every activity in a network so, for a network with N activities, N inequalities of the form of 4.3 may be written, that is

$$T_j - T_i \geqslant t_{ij} \text{ for all activities } i\text{–}j \tag{4.4}$$

The set of inequalities 4.4 represents conditions which event times and activity durations must satisfy if the network is to be *feasible*, that is, the project is capable of completion.

Inequality 4.3, and hence also the set of inequalities 4.4, is *linear* in both $\boldsymbol{T}$ and $\boldsymbol{t}$. If activity durations are specified and it is required to find event times $\boldsymbol{T}$ which correspond to a feasible network they must satisfy relationships 4.4 which are linear functions of $\boldsymbol{T}$. Conversely if the event times $\boldsymbol{T}$ are known and suitable activity durations $\boldsymbol{t}$ must be found then relationships 4.4 which are linear in $\boldsymbol{t}$ must again be satisfied. Thus the two most important elements of a planning network – the event times and activity durations – are related by simple linear functions.

The set of inequalities 4.4 must be satisfied by all event times and activity durations. They, therefore, represent a set of constraints on the feasibility of a network. Consider, now, how they can be used in the analysis of a network.

4.2.2 Finding Event Times by Linear Programming

Suppose that for a particular project a network has been drawn and logically numbered and that activity durations t_{ij} are all known. Suppose it is required to find earliest event times for all events. Inequalities 4.4 can be rewritten for earliest event times simply by substituting $\boldsymbol{E}$ for $\boldsymbol{T}$, that is

$$E_j - E_i \geqslant t_{ij} \text{ for all activities } i\text{–}j \tag{4.5}$$

For a feasible network all earliest event times must satisfy inequalities 4.5. The definition of an earliest event time given in section 4.1.2.2 implies that each earliest event must occur at the minimum feasible time after the start event, $E_1 = 0$. Thus the sum of the earliest event times must also be a minimum. This leads directly to a linear programming problem for finding earliest event times

www.EngineeringBooksPDF.com

$$\left.\begin{array}{l}\text{Minimise } \sum_{k=1}^{n} E_k \text{ over all events} \\ \text{subject to } E_j - E_i \geqslant t_{ij} \text{ for all activities } i\text{-}j \\ \text{with } E_1 = 0\end{array}\right\} \quad (4.6)$$

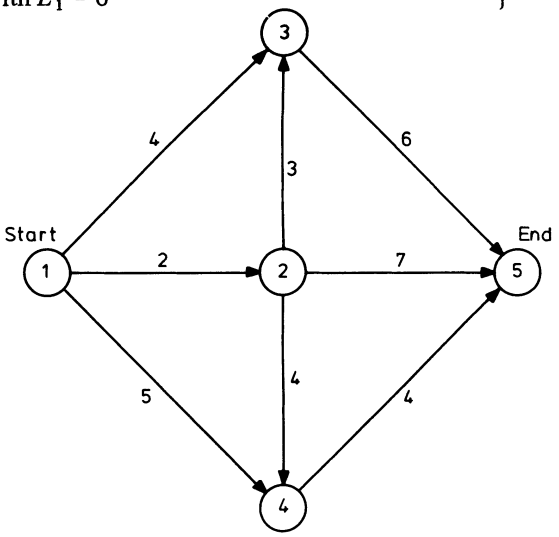


Figure 4.9

Figure 4.9 shows a network with eight activities and five events. Using it as a numerical example of problem 4.6, the earliest event times will be found by solving the following linear programming problem

$$\left.\begin{array}{l}\text{Minimise } E_2 + E_3 + E_4 + E_5 \\ \text{subject to} \quad E_2 \geqslant 2 \\ \qquad\qquad E_3 \geqslant 4 \\ \qquad\qquad E_4 \geqslant 5 \\ \qquad E_3 - E_2 \geqslant 3 \\ \qquad E_4 - E_2 \geqslant 4 \\ \qquad E_5 - E_2 \geqslant 7 \\ \qquad E_5 - E_3 \geqslant 6 \\ \qquad E_5 - E_4 \geqslant 4\end{array}\right\} \quad (4.7)$$

The simplex method as described in chapter 3 may be used to solve this problem. The first stage is to introduce slack variables to convert all inequality constraints into equalities. Let the slack variables be S_1 for the first constraint to S_8 for the last, A typical constraint is then

$$\text{that is} \quad \begin{array}{l} E_j - E_i - S_k = t_{ij} \\ S_k = E_j - E_i - t_{ij} \end{array} \quad (4.8)$$

www.EngineeringBooksPDF.com

The right-hand side of equation 4.8 is simply an algebraic form of the definition of the free float of activity i–j. Thus the slack variables in the simplex method solution of problem 4.7 are identical with the free float of the activities in the network. It is left to the reader to obtain the following solution to problem 4.7

$$\left.\begin{array}{ll} E_2 = 2 & S_1 = S_4 = S_5 = S_7 = 0 \\ E_3 = 5 & S_2 = 1 \\ E_4 = 6 & S_3 = 1 \\ E_5 = 11 & S_6 = 2 \\ & S_8 = 1 \end{array}\right\} \tag{4.9}$$

The linear programming problem 4.6, therefore, finds earliest times for all events, a minimum completion time for the project and free floats for all activities. Another linear programming problem can be formulated very similarly to problem 4.6 to yield latest event times. Instead of minimising the sum of the earliest event times with $E_1 = 0$ as in problem 4.6, this time the sum of the latest event times is maximised subject to the requirement that $L_n = E_n$ where n is the end event number. Network feasibility is ensured by substituting $\boldsymbol{L}$ for $\boldsymbol{T}$ in inequalities 4.4. This yields the problem

$$\left.\begin{array}{l} \text{Maximise } \displaystyle\sum_{k=1}^{n} L_k \text{ over all events} \\ \text{subject to } L_j - L_i \geqslant t_{ij} \text{ for all activities } i\text{–}j \\ \text{with } L_n = E_n \end{array}\right\} \tag{4.10}$$

The numerical equivalent of problem 4.10 for the network of figure 4.9 is

$$\left.\begin{array}{l} \text{Maximise } L_1 + L_2 + L_3 + L_4 \\ \text{subject to } L_2 - L_1 \geqslant 2 \\ \qquad L_3 - L_1 \geqslant 4 \\ \qquad L_4 - L_1 \geqslant 5 \\ \qquad L_3 - L_2 \geqslant 3 \\ \qquad L_4 - L_2 \geqslant 4 \\ \qquad L_2 \leqslant 4 \\ \qquad L_3 \leqslant 5 \\ \qquad L_4 \leqslant 7 \end{array}\right\} \tag{4.11}$$

and the latest event times given by the solution of this problem are

$$\left.\begin{array}{ll} L_1 = 0 & L_3 = 5 \\ L_2 = 2 & L_4 = 7 \end{array}\right\} \tag{4.12}$$

A hand analysis of figure 4.9 confirms that the linear programming results given in 4.9 and 4.12 are correct. Using these results the few remaining elements of the analysis table may be completed.

www.EngineeringBooksPDF.com

4.2.3 Time-Cost Optimisation of Activity Durations

The difficulty of specifying for each activity in a network an 'average, likely duration' has been mentioned earlier in this chapter. As stated in section 4.1.2.2, the duration of an activity depends very much on the resources allocated for its completion and hence upon the money spent on that activity. As more money is spent, more resources of labour and plant are bought and an activity duration consequently reduces. One way of resolving the difficulty of specifying activity durations is to select durations which minimise the total cost of the project.

Examining this minimum total cost approach further, consider a typical activity i-j. Assume that the relationship between time and cost for the activity is of the form

$$C_{ij} = K_{ij} - \alpha_{ij}\, t_{ij} \tag{4.13}$$

where

C_{ij} = total cost of activity i-j
t_{ij} = duration of activity i-j
K_{ij} = constant representing cost elements of i-j which are time-invariant
α_{ij} = constant representing the cost of reducing the duration of activity i-j by 1 time unit

Equation 4.13 is a linear relationship between cost and time for the activity and the negative sign ensures that costs reduce as time increases and vice versa. If all activities in a network can be represented by equation 4.13,,with appropriate values for K_{ij} and α_{ij}, the total cost of the entire project is then the sum of all N activity costs

$$\begin{aligned}\text{Total cost} &= \sum_{ij}^{N} (K_{ij} - \alpha_{ij}\, t_{ij}) \\ &= \sum_{ij}^{N} K_{ij} - \sum_{ij}^{N} \alpha_{ij}\, t_{ij} \\ &= K - \sum_{ij}^{N} \alpha_{ij}\, t_{ij}\end{aligned} \tag{4.14}$$

In equation 4.14 the constant K represents total time-invariant costs for the project. If the total cost in equation 4.14 is to be minimised this can be achieved by maximising the variable part. Several constraints must be imposed giving the linear programming problem

www.EngineeringBooksPDF.com

$$\left.\begin{aligned}
&\text{Maximise} \sum_{ij}^{N} \alpha_{ij}\, t_{ij} \\
&\text{subject to } T_j - T_i - t_{ij} \geqslant 0 \qquad \text{for all activities} \qquad (4.15a)\\
&\qquad t_{ij}(\min) \leqslant t_{ij} \leqslant t_{ij}(\max) \quad \text{for all activities} \qquad (4.15b)\\
&\qquad T_n \leqslant T' \qquad (4.15c)\\
&\qquad T_1 = 0 \qquad (4.15d)
\end{aligned}\right\}$$

Constraints 4.15a ensure feasibility of the network. Event times T are also variables whose values must be found. Constraints 4.15b impose a lower and an upper time limit on the duration of each activity; $t_{ij}(\min)$ and $t_{ij}(\max)$ must be specified for each activity. Equality 4.15d simply states the convention that the time of the start event must be zero. Constraint 4.15c imposes a maximum duration for the project, T′, which must be specified. Solution of this linear programming problem may be carried out by the simplex method and the resulting cost-optimal activity durations can then be used for further detailed network analysis.

Once again it has been shown that linear programming can be used effectively within the network analysis process, further strengthening previous conclusions about the fundamental linearity of construction planning networks. Problem 4.15 exploits the linear time-cost relationship for activities. This linear relationship is commonly found among very many activities. Sometimes, however, specific types of activities cannot be represented by equation 4.13. For example some activities may have discrete-valued time-cost relationships; if an expensive piece of plant is supplied there will be a discrete jump in both cost and time rather than a linear continuous function. Some activities have non-linear time-cost equations. When the time-cost relationships are not linear other specialised forms of problem 4.15 must be solved by different methods (see the book by Moder and Phillips for examples). Even in these cases, however, the problem constraints remain those of problem 4.15 demonstrating the strong linearity of network analysis.

4.3 RESOURCE ALLOCATION AND PROJECT CONTROL

4.3.1 Resource Allocation

Once the network has been drawn and analysed for a construction project, resources must be allocated to all the activities in order that they can be completed within the estimated times available and as efficiently as possible. The resources needed for any activity may be classified under five headings – men, plant, materials, money and time. The nature of each activity determines exactly how much material is needed for its completion. Allocation of materials consists of ensuring that all materials to be used are available when needed in the correct amounts – no more, no less. This requires careful control of the

www.EngineeringBooksPDF.com

ordering and storing processes and a good inventory system. Money is an essential resource but not one which, in the strictest sense of the word, can be allocated. Money itself does not construct anything, it can only be used to buy other resources. Its presence or absence, therefore, controls what allocations of other resources, particularly men and plant, may be made. Similarly, time is a resource that can only be very indirectly controlled through different allocations of men and plant among the activities of a project.

Of the five main resources only two – men and plant – are available to the planner for direct allocation in varying quantities among the activities. Men and plant may be thought of as independent variables in resource allocation problems. Money and time are essentially dependent variables whose values determine efficient allocations of men and plant. Typically resource allocation problems consist of determining how men and plant should be assigned to the various project activities so that one or more of various conditions is met. Frequently used conditions are those of minimum project cost or duration, fixed limits upon cost or duration and maintaining a constant pool of labour (resource levelling). At different stages of a construction project different resource allocation problems must be solved. At the start of a project of long duration it is usual to concentrate upon finding resource allocations which minimise costs with project duration as a constraint. Towards the end of a project, as cumulative delays and crises push the likely completion date beyond that originally estimated, it becomes more important to allocate resources so that the project duration is minimised.

Earlier in this chapter the fundamental linearity of construction planning networks has been demonstrated. In chapter 3, section 3.7 noted that problems of resource allocation can often be posed as linear programming problems. Consequently it might be expected that resource allocation problems in construction planning would be solved by LP. In fact, network resource allocation problems are fundamentally linear but in practice LP is very rarely used to solve such problems. There are several reasons for this.

Firstly, if men are to be allocated to activities by LP an integer solution must be obtained since fractions of men cannot be allocated. This calls for special LP algorithms for solving integer or mixed-integer problems. Secondly, construction work is often done by gangs or crews of men. In a crew comprising a number of men each man has his own skills and specialities. Any allocation of extra men to an activity, with the expectation that the duration of that activity will reduce, is complicated by the fact that this reduction may only occur if the new mix of skills is appropriate to that activity. In theory it is possible to incorporate this 'compatibility' requirement in a linear programming model but the model would be complicated, of little practical value and hard to solve. Very similar difficulties arise in allocating plant. As a result formal linear models are very rarely used. Instead a number of very specific methods have been developed for construction planning resource allocation problems. These methods are simple to use, suitable for hand calculation and they combine elements of linearity with the construction

www.EngineeringBooksPDF.com

planner's own expertise and knowledge of allocations which are likely to be 'compatible'.

There is a limit to the extent to which the basic mathematical nature of construction planning networks can be exploited. That limit is reached at the resource allocation stage. At the network drawing and analysis stages all projects obey the same rules of linearity and computational methods based on this are invaluable in network analysis. At the resource allocation stage, however, so much depends on the nature of the activities and resources that there is very little similarity among projects. Mathematical and numerical aids at this stage are of much less value to a construction planner than a wide experience of actual construction projects and how they are built. For this reason no study is made here of analytical methods for resource allocation problems in construction networks. Details of the simple methods referred to above can be found in construction management books such as that by Moder and Phillips.

4.3.2 Project Control

In practice, critical path planning is a continuous process throughout the construction work. Before construction work commences very detailed planning must be done. The complete network of activities should be drawn and analysed in detail. Resource requirements should be estimated and provisionally allocated. Estimates of the money required at all stages must be made. Should any of these preliminary analyses be unsatisfactory in any sense the work must be replanned to remove any possible delays or bottlenecks. Several different network analyses may have been made before a final one is decided on which satisfies all criteria. This initial planning and replanning is invaluable as it shows up many problems and shortcomings in a plan and allows them to be solved before work starts. At the start of construction work a complete potentially problem free plan should exist.

No planning process is perfect, however, and the work of the planning team is by no means over when construction commences. As has been described earlier the planning process relies on *estimates* of activity durations, costs, resources, etc. Once construction work starts data must be collected about actual durations, times, costs, etc., for comparison against estimated values. It frequently happens that delays in some activities or cumulative small differences between actual and estimated times cause the critical path to change as the work progresses. The planning team must respond with a modified plan for the subsequent activities and detailed forecasts of money requirements, completion dates, etc. Towards the end of a project as the projected completion date draws nearer the planning group remains busy. Heavy financial penalties are sometimes exacted from a contractor who fails to meet a completion date. The threat of such penalties, together with the pressure of accumulated delays and discrepancies which have occurred throughout construction, usually requires intense replanning of the final stages of a project so that it is completed at as small an additional cost as

www.EngineeringBooksPDF.com

possible. Even after the project is finished there is work for a planning group to do. Experience gained during the project should be consolidated to allow better estimates to be made for the next project. This means refining of unit costs, estimates of durations of common activities, etc.

The critical path method and all its associated techniques have a considerable effect on the smooth running of a construction project. Network analysis has brought order to the planning and control of construction work. The larger the project the more important is the work of the planning team. Although the planning network itself is only a very small part of the wide range of techniques available to the construction planner it provides two essential ingredients of the planning process. The first of these is an immediate pictorial view of the way the different activities and construction processes fit together into a whole project. This mental picture of the project is most important to any planner since it allows him to concentrate on the details of construction while retaining a broad picture of the project in its entirety. The second is that the planning network provides a framework for trying out different construction techniques and policies on the drawing board rather than in practice. The effects on time and money of innovative construction techniques may be assessed by network analysis before they are tried out on a project. This predictive capacity of networks is very valuable and is not limited to construction planning networks, as the rest of this chapter shows.

4.4 GENERALISED NETWORK PROBLEMS

The three previous sections of this chapter have dealt with construction planning networks. Problems arising from these networks occur again in this book from time to time but detailed study of construction network analysis *per se* is pursued no further. The interested reader will doubtless seek further information from the many books available on this specific problem. As far as this book is concerned, intense concentration upon the minutiae of techniques for solving the specific problems of construction planning networks can sometimes obscure the fact that such networks are specific examples of a much wider class of problems. Many other equally important network concepts and problems exist. The remainder of this chapter presents a brief introduction to the more general field of networks. It is hoped that this will generate a wider appreciation of the importance of networks and will set specific cases such as construction planning networks in proper context.

Many books have been written on the mathematical and practical aspects of graph and network theory. Most of them adopt a scientific approach in first examining networks in an abstract mathematical way, building up a mass of general theory, theorems, lemmas, etc., which is then used to particularise the study to specific examples and problems. The scope of this book does not allow the luxury of this expansive treatment of networks. The rest of this chapter gives only a flavour of the subject and attempts to proceed backwards from the

www.EngineeringBooksPDF.com

specific problems towards the more general case. The intention here is to introduce the subject, arouse curiosity and pose questions rather than to answer them in detail. Anyone whose interest is stimulated should consult the more detailed books such as those by Minieka, Ford and Fulkerson and Wilson.

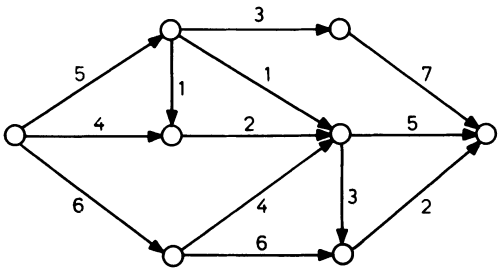


Figure 4.10

Consider the network shown in figure 4.10. The reader who has recently studied the three previous sections of this chapter will most probably interpret figure 4.10 as a construction planning network. The network represents the logical interrelationships among activities with the arrows representing activities against which durations are marked. The nodes represent events and the network has a single start event and a single end event. This interpretation is perfectly valid; a complete critical path network analysis could be carried out upon figure 4.10.

This interpretation, however, is not the only one possible. There is nothing in figure 4.10 which defines it as a construction planning network. An equally valid interpretation is that it represents an idealised road map. The nodes represent towns or road junctions and the arrows represent roads joining them. Permissible directions of travel are indicated by the directions of the arrows and the number against each arrow is the length of that road in miles between nodes. With this interpretation it would be possible to analyse this network and determine, for example, the shortest route between the two towns on the extreme left and the extreme right of the figure. Of course, in order to perform this analysis, methods different from those for construction planning would have to be used.

Another equally valid interpretation of figure 4.10 is that it represents a water purification process. Grossly polluted water passes through the network from the left-hand node and arrives as purified water at the extreme right-hand node. Each arrow represents a different partial purification process with the number against each arrow representing the maximum flow capacity of that process in millions of gallons per day. The directions of the arrows indicate the sequence of processes from left to right. With this interpretation an appropriate question which might be asked is: 'What is the maximum volumetric

www.EngineeringBooksPDF.com

throughput of the entire purification system?'. In order to answer this question suitable analysis methods must be applied. Clearly they will not be the same methods as are used for construction planning networks or travel routing.

These three interpretations (and many more possible interpretations) all apply to exactly the same network, figure 4.10. The network has not changed but in each case the rules for interpreting it, analysing it, and calculating information from it have changed. It is therefore possible to view critical path network analysis as described earlier merely as a particular set of rules that may be applied in a specific instance to a much wider, more general network representation. The network is a common core of a wide variety of problems of which construction planning is one. A detailed study of construction planning networks leads to a good understanding of construction planning. The same amount of study applied to basic networks leads to an appreciation of a much wider range of practically useful, seemingly disparate problems. The study of networks is, therefore, a very important part of any serious approach to civil engineering systems which tries to identify and exploit commonality among different problems.

4.4.1 Definitions

Much of the terminology of graphs and networks is in a state of mild confusion and differs from book to book. It is, therefore, necessary to define some terms used in this book.

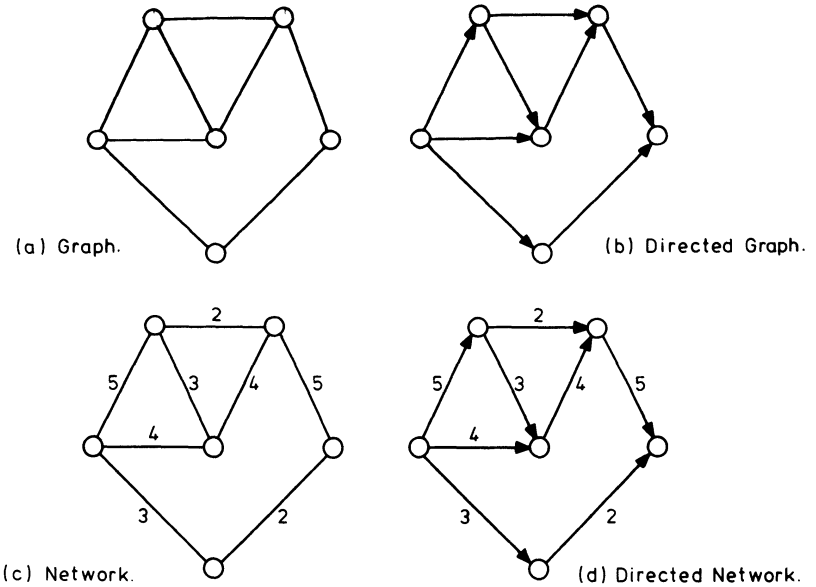


Figure 4.11 Graphs and networks

www.EngineeringBooksPDF.com

Figure 4.11 shows the differences between graphs and networks. Figure 4.11a shows a *graph* that is the simplest form consisting of a set of *vertices* (the circles or nodes) and a set of *arcs* (the lines joining the vertices). In mathematical terms a graph such as figure 4.11a can be defined as a set of numbered vertices – in this case the set would comprise six numbers – and a set of ordered pairs of vertices representing the arcs – in this case eight pairs. Figure 4.11b shows a *directed graph* that is simply a graph in which a direction is associated with each arc. Notationally the direction can be indicated by the order of the pair of vertices for each arc. The first vertex of any pair represents the tail of the directional arrow and the second represents the head. Figure 4.11c shows a *network* that is defined as a graph with one or more numbers associated with each arc, and figure 4.11d shows a *directed network* that is a graph with both numbers and a direction associated with each arc. Unfortunately in the literature it is common to find the word graph or network applied to all four cases of figure 4.11 quite indiscriminately.

Figure 4.10 is a directed network and the three interpretations of it suggested above are essentially concerned with directed networks. Figure 4.10 has a *single source* and a *single sink*. A source is a vertex that has no arc directed into it and a sink is a vertex that has no arc directed away from it. It is possible to have directed networks with multiple sources and sinks although the specific case of construction planning networks requires that there shall be only one of each. In the following section some problems of directed networks are examined.

4.5 DIRECTED NETWORKS

4.5.1 Path Problems

One problem which arises frequently in directed networks is that of finding paths through the network which exhibit special properties. For example, if figure 4.10 represents a road network the *shortest path* from source to sink is obviously of interest. If figure 4.10 represents a construction planning network, the critical path is of interest and, as has been noted earlier, the critical path is also the *longest path* through the network. This section examines these path problems in a more general way.

The numbers associated with the arcs of a directed network may represent many different physical quantities depending upon the particular interpretation of the network. For simplicity these numbers will be referred to as the *lengths* of the arcs irrespective of whether they represent physical lengths, times, costs, weights, etc. Thus l_{ij} is the length of arc i–j. The *length of a path* is defined as the sum of the lengths of the individual arcs comprising the path. Let the vertices be numbered $k = 1, \ldots, n$ and let D_k be a number associated with vertex k, $k = 1, \ldots, n$ which denotes the length of a path as yet unspecified from the source where $D_1 = 0$ to vertex k.

www.EngineeringBooksPDF.com

For a construction planning network the l_{ij}s represent activity durations and the D_ks represent event times. It has already been shown that the critical path is the longest duration path in a planning network and that E_n is the length of this path. Therefore, all the methods and results which have been derived for construction planning networks in sections 4.1 and 4.2 may be directly expressed in terms of the general notation $\boldsymbol{D}, \boldsymbol{l}$ and may be used to determine longest paths through a directed network such as figure 4.10. Specifically, the basic network feasibility rule for longest paths becomes

$$D_j - D_i \geqslant l_{ij} \text{ for all arcs } i\text{-}j \tag{4.16}$$

which is the general equivalent of 4.4. The earliest event time algorithm can be used to calculate E_n for a construction planning network where E_n is the length of the critical path. Thus, changing the notation to D_ks and l_{ij}s this algorithm can be generally used to calculate D_n, the length of the longest path through a more general network such as figure 4.10. The linear programming aspects of construction planning networks also transfer directly to the more general case of directed networks so problem 4.6 with Es and ts changed to Ds and ls can also be used to find the length of the longest path through a network, locate the critical lengths and hence locate the longest path.

What about shortest paths? What changes are necessary in order to calculate shortest rather than longest paths in a directed network? Firstly, examine the feasibility rule 4.16. Rewriting rule 4.16 as

$$D_j \geqslant D_i + l_{ij}$$

implies that the number D_j to be associated with the vertex at the head of an arc must not be less than the greatest path length of all paths directed into vertex j. It is essentially a longest path inequality. To convert this to a shortest path rule, the smallest rather than the greatest incoming path length should be chosen. Thus for shortest paths the feasibility rule becomes

$$D_j \leqslant D_i + l_{ij}$$

or

$$D_j - D_i \leqslant l_{ij} \text{ for all arcs } i\text{-}j. \tag{4.17}$$

that is, for shortest paths, the sign of the inequality must be changed in rule 4.16. By similar reasoning it can be shown that to change the longest path algorithm (the general form of the earliest event time algorithm) to a shortest path algorithm all that is required is that the maximisation process within that algorithm should be changed to minimisation. Similarly the linear programming general form of problem 4.6 that finds longest paths is changed to a shortest path problem by reversing the sign of the inequalities in the constraints and maximising the objective function rather than minimising it.

www.EngineeringBooksPDF.com

These results may be summarised as follows

Feasibility rule

$$D_j - D_i \left\{\begin{matrix}\geq\\ \leq\end{matrix}\right\} l_{ij} \text{ for } \left\{\begin{matrix}\text{longest}\\ \text{shortest}\end{matrix}\right\} \text{ paths} \tag{4.18}$$

Path algorithm

$$\left.\begin{array}{l}\text{Step 1} \quad D_1 = 0\\ \text{Step 2} \quad \text{For } j = 2, \ldots, n\\ \qquad D_j = \left\{\begin{matrix}\text{maximum}\\ \text{minimum}\end{matrix}\right\} \{D_i + l_{ij}\} \text{ for } \left\{\begin{matrix}\text{longest}\\ \text{shortest}\end{matrix}\right\} \text{ paths}\end{array}\right\} \tag{4.19}$$

with the extremisation extending over all paths terminating at j.

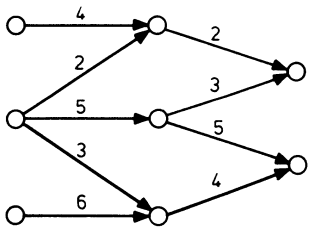
Linear programming path problem

$$\left.\begin{array}{l}\text{For } \left\{\begin{matrix}\text{longest}\\ \text{shortest}\end{matrix}\right\} \text{ paths, } \left\{\begin{matrix}\text{minimise}\\ \text{maximise}\end{matrix}\right\} \sum_{k=1}^{n} D_k\\ \text{subject to } D_j - D_i \left\{\begin{matrix}\geq\\ \leq\end{matrix}\right\} l_{ij} \text{ for all arcs } i\text{-}j\\ \text{with } D_1 = 0\end{array}\right\} \tag{4.20}$$

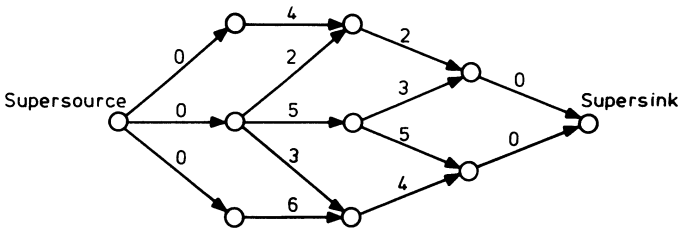
These summarised results show that the methods developed earlier for construction planning networks can, with a little thought, be extended to be applicable to a much wider class of path problems in directed networks. It is useful to consider how general this extension really is. Are the methods above applicable to all directed networks or are there perhaps some exceptions? In fact there are; the above results were obtained by generalising results obtained for construction planning networks and so they are limited by the limitations of that type of network. One requirement for construction planning networks is that the networks must have a single source and a single sink. Therefore, the methods presented above are also essentially suitable for single source/sink networks. The more general case of multiple sources and sinks, however, can be accommodated by some preparatory work. Any multiple source, multiple sink network may be converted to a single source, single sink network by adding a *supersource* and a *supersink*. A supersource is an additional preceding vertex linked by arcs to each of the multiple sources. Similarly, each of the multiple sinks is linked by an additional arc to a single extra succeeding sink vertex. All the additional arcs are given a length of zero. Thus the additional supersource and supersink allow the methods of this section to be used on multiple source/sink networks. Figure 4.12 demonstrates a supersource and a supersink added to a network.

A more fundamental problem which is encountered in trying to show the generality of these path algorithms is that of loops or circuits as they are sometimes called. Figure 4.6 shows that loops are specifically prohibited from

www.EngineeringBooksPDF.com



a) Multiple source/sink directed network.



b) Expanded network with supersource and supersink.

Figure 4.12

construction planning networks because they are illogical there. Their illogicality, however, does not extend to other interpretations of directed networks. An idealised map of a one-way street system can be represented as a directed network and loops would be quite logical and permissible in such a system. Clearly a directed network without loops is a particular case of a general directed network in which loops may be present. The concept of a longest path, however, is meaningless in a looped system because a path may be infinitely lengthened by circulating many times around a loop. This argument does not apply to shortest paths except in the impractical case of negative arc lengths and so it must be concluded that in the general case of directed networks only shortest path problems can have significance. The existence of longest paths is confined to a small sub-set of the general case, a subset in which loops do not occur. The methods of this section obviously cannot be used for longest path problems in looped systems – the problem is not well defined. Also they cannot be used for shortest path problems in looped networks. This is because a loop violates the feasibility rule 4.18 and hence invalidates algorithm 4.19 and problem 4.20. Loops plainly present difficulties to path-finding algorithms and the methods described here are only applicable to loopless networks. General algorithms for directed networks with loops are available but are not examined here. Brief reference is made to some other interesting path problems in section 4.6 on undirected networks.

www.EngineeringBooksPDF.com

4.5.2 Flow Problems

4.5.2.1 Linear Programming Maximum Flow Problem

In section 4.4, figure 4.10, a directed network, was interpreted as a flow problem – find the maximum value of the total flow which can be passed through the network of figure 4.10 if the number associated with each directed arc represents the flow capacity of that arc. This problem is completely different from the previous path problems; here all paths may contribute to the maximum flow. Although the method which will be derived in this section is different from those above it still, however, embodies and exploits the linear aspects of directed networks.

In the maximum flow problem two quantities are associated with each directed arc. The first of these for a typical arc i–j is the known *maximum flow capacity*, c_{ij}, which on figure 4.10 is the number written beside each arc. The second quantity, which is not shown on figure 4.10, is the unknown *actual flow* in an arc which for arc i–j can be called x_{ij}. For a network with N arcs these definitions immediately lead to N inequalities which must be satisfied and which state that the actual flow in an arc must not exceed that arc's capacity, that is

$$x_{ij} \leqslant c_{ij} \text{ for all } N \text{ arcs } i\text{–}j \tag{4.21}$$

As the problem is concerned with flows in arcs which converge and diverge at the vertices it is essential that flow equilibrium is maintained. The only places where external flows enter or leave the network are the source and the sink, thus at each other vertex equilibrium is maintained by requiring that the sum of the flows on arcs directed into the vertex must be equal to the sum of the flows on arcs directed out of the vertex, that is

$$\sum_{(\text{IN})} x_{ij} - \sum_{(\text{OUT})} x_{ji} = 0 \text{ at vertex } j,\ j = 2, \ldots, n-1 \tag{4.22}$$

At the source, flow is assumed to enter the network from outside and this input flow must equal the sum of the flows on all arcs directed away from the source. Similarly, at the sink the sum of the flows in all arcs directed into the sink must equal the output from the network. The input and output must, of course, be equal and since the problem requires that the flow through the system is maximised this gives an objective function

$$\text{Maximise } F = \Sigma\, x_{1j} \equiv \Sigma\, x_{in} \tag{4.23}$$

Bringing together this objective function 4.23, the flow equilibrium constraints 4.22 and the arc capacity inequalities 4.21, gives a linear programming maximum flow problem as

$$\left.\begin{aligned} &\text{Maximise } F = \Sigma\, x_{1j} \equiv \Sigma\, x_{in} \\ &\text{subject to } \sum_{(\text{IN})} x_{ij} - \sum_{(\text{OUT})} x_{ji} = 0 \text{ vertices } j = 2, \ldots, n-1 \\ &\qquad 0 \leqslant x_{ij} \leqslant c_{ij} \text{ for all arcs } i\text{–}j \end{aligned}\right\} \tag{4.24}$$

www.EngineeringBooksPDF.com

In problem 4.24 either of the alternative objective functions may be used as they must have the same value. If the directed network of figure 4.9 is used as an example the linear programming maximum flow problem is

$$\left.\begin{aligned}
\text{Maximise } & F = x_{12} + x_{13} + x_{14} \text{ (or } F = x_{25} + x_{35} + x_{45}\text{)} \\
\text{subject to } & x_{12} - x_{23} - x_{24} - x_{25} = 0 \\
& x_{13} + x_{23} - x_{35} = 0 \\
& x_{14} + x_{24} - x_{45} = 0 \\
& x_{12} \leqslant 2 \\
& x_{13} \leqslant 4 \\
& x_{14} \leqslant 5 \\
& x_{23} \leqslant 3 \\
& x_{24} \leqslant 4 \\
& x_{25} \leqslant 7 \\
& x_{35} \leqslant 6 \\
& x_{45} \leqslant 4 \\
& \text{All } x_{ij} \geqslant 0
\end{aligned}\right\} \quad (4.25)$$

Problem 4.25 may be solved by the simplex method of chapter 3 to yield a maximum flow of $F = 10$ units.

4.5.2.2 A Simpler Maximum Flow Method

There are methods much simpler than linear programming for finding maximum flows through directed networks; the problem was posed here as one of linear programming to stress once again the basic linearity of this class of problems. Several algorithms for different flow problems are based on the concept of examining possible paths through the network. Figure 4.13 shows how this concept works on the network of figure 4.9.

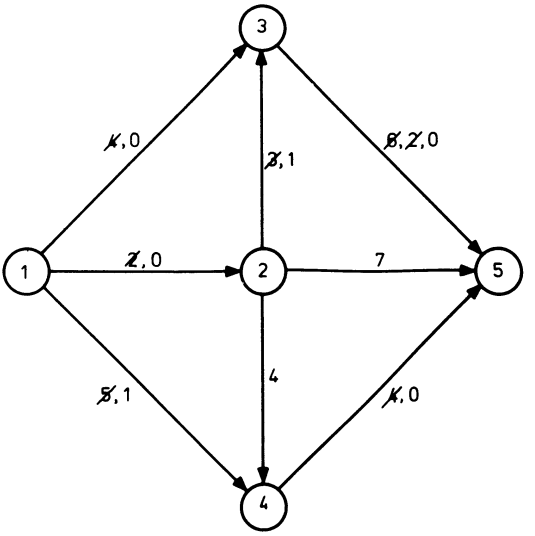
Consider any path from source to sink on figure 4.9. For example, select the path consisting of arcs 1-3, 3-5. Arc 1-3 has a capacity of 4 units of flow and the capacity of arc 3-5 is 6 units. Clearly the maximum amount of flow which can be passed along the path 1-3-5 is restricted by the capacities of the arcs in that path to 4 units. Thus the capacity of a path is given by

$$\text{path capacity} = \text{minimum } \{\text{arc capacities in the path}\} \quad (4.26)$$

Assign a flow of 4 units to path 1-3-5, noting this by the side of figure 4.13. Since 4 units of flow are assigned to path 1-3-5 the capacities of the arcs in that path are effectively reduced by 4 units. For example arc 3-5 has a total capacity of 6 units but since it now carries 4 units of flow its remaining effective capacity is now only 2 units. Similarly arc 1-3 has a new effective capacity of zero units – it is full. These new effective capacities replace the initial capacity values on figure 4.13 which are crossed out.

Now select another path from source to sink. For example select path 1-2-3-5. The present capacities of the arcs in this path, arcs 1-2, 2-3 and 3-5

www.EngineeringBooksPDF.com



Flow	Path
4	1-3-5
2	1-2-3-5
4	1-4-5
10	

Figure 4.13 Maximum flow solution of the network of figure 4.9

are 2, 3 and 2 units respectively. (Note that the present effective capacity of arc 3-5 is used here rather than its initial capacity since this arc already carries some flow). Using equation 4.26 give the capacity of path 1-2-3-5 as

$$\text{path capacity} = \text{minimum } \{2,3,2\}$$
$$= 2$$

This capacity indicates that 2 units of flow can be assigned to path 1-2-3-5. This flow is assigned and noted separately beside figure 4.13, along with the previously assigned flow. As a result of this assignment of flow, new effective capacities can be calculated for the arcs in path 1-2-3-5. These effective capacities are 0, 1 and 0 units for arcs 1-2, 2-3, 3-5, respectively.

A further path is selected, path 1-4-5. Its capacity is

$$\text{path capacity} = \text{minimum } \{5, 4\}$$
$$= 4$$

This flow of 4 units is assigned and noted separately. Arcs in the path 1-4-5 have their effective capacities reduced by 4 units to 1 for arc 1-4 and 0 for arc 4-5. Figure 4.13 shows the state of the network at this stage of the solution. The next step is to select another path through the network from source to sink. No useful path, however, can be found. A path must start at the source but if arcs 1-3 or 1-2 form parts of a path no flow can be passed along such a path because the effective capacities of those arcs are now zero. If arc 1-4 is chosen, the only path to the sink includes arc 4-5 with an effective capacity of zero. This path has already been used. It is tempting to try path 1-4-2-5 because it does not have any arc in it with zero effective capacity. It cannot be used,

www.EngineeringBooksPDF.com

however, because it would require flow to be passed from vertex 4 to vertex 2 against the specified direction of that arc. Therefore, as no useable path remains it is concluded that no more flow can be passed through the network. Totalling the flows noted separately beside figure 4.13 gives a total of $F = 10$ which is, therefore, the maximum throughput of the network.

This maximum flow algorithm is based upon examining all possible paths from source to sink and filling them all with capacity flows. However, it is not as straightforward as it seems, as it depends upon the order in which paths are chosen. For example, using figure 4.13 again, if path 1-2-4-5 is chosen as the second path instead of 1-2-3-5, 2 units of flow can be passed down it by virtue of equation 4.26. Then, choosing 1-4-5 as the only remaining path enables a further 2 units of flow to be passed since arc 4-5 now has a reduced capacity of only 2 units. No further path seems to be available yet the total flow passed through the network is only 8 units instead of 10 as shown in figure 4.13. A further path is available, however. Path 1-4-2-5 can be used despite the fact that this appears to violate the directional sense of arc 2-4. The arrow on arc 2-4 means that the *net* flow must always be in the direction shown, but since there is already a flow of 2 units in the positive direction 2-4 it is possible to visualise a reverse flow in the direction 4-2. Provided that the positive flow always exceeds the negative flow, the resultant or net flow will always be in the specified arrow direction. Thus path 1-4-2-5 can carry 2 units of flow by virtue of the reverse capacity of arc 4-2 and the maximum flow through the network is then 10 units as before. This solution implies zero flow in arc 2-4. Note that in the original example of figure 4.13 no reverse flow was permissible on arc 2-4 because there was no positive flow in the sense of the arrow.

The possibility of reverse flows complicates the simple maximum flow algorithm and renders it applicable only to relatively simple, small networks as a hand method. For larger, complicated networks more rigorous methods are needed. Some very good algorithms exist (see the book by Minieka) which use the positive and negative flow concepts in a more rigorous fashion.

4.5.2.3 Minimum Cost Flows

Another flow problem in directed networks which has important practical relevance is the minimum cost flow problem. Given a directed network with capacities for all arcs, suppose that it is required to pass a specified flow through the network. If each arc passes flow at a different cost then the total cost of passing the specified flow through the network will depend on the paths chosen. The problem consists of finding the cheapest way of passing the required flow.

Let each directed arc of the network have associated with it a known capacity c_{ij} and an unknown flow x_{ij} as before. Also let a_{ij} be the cost of

www.EngineeringBooksPDF.com

sending one unit of flow down arc i–j of the network. Suppose that it is required to pass a total flow V through the network from source to sink at minimum cost. With these definitions the minimum cost flow problem can be expressed as

$$\left.\begin{array}{lll} \text{Minimise } \Sigma a_{ij} x_{ij} & & \text{over all arcs } i\text{–}j \\ \text{subject to } \underset{(\text{IN})}{\Sigma} x_{ij} - \underset{(\text{OUT})}{\Sigma} x_{ji} = 0 & & \text{at vertices } j=2, \ldots, n-1 \\ \Sigma x1_j = \mathrm{V} & & \text{at source} \\ \Sigma x_{in} = V & & \text{at sink} \\ 0 \leqslant x_{ij} \leqslant c_{ij} & & \text{all arcs } i\text{–}j \end{array}\right\} \quad (4.27)$$

Problem 4.27 is again a linear programming problem. The objective function represents the total cost of passing flow down all arcs. The constraints at all internal vertices ensure equilibrium of flow as in the maximum flow problem. The two constraints for the source and sink ensure that the required total flow, V, enters and leaves the network. Only one of this pair of constraints is necessary. The final constraints ensure that flows are non-negative and do not exceed the arc capacities.

Special purpose algorithms for solving problem 4.27 are available in the literature (see the book by Minieka) and are quicker than the LP simplex method. The reader should by now be aware, however, of the wide applicability of linear programming which lies at the core of a wide range of problems. As with path problems, flow problems in directed networks with multiple sources and sinks may be solved using the methods described above after first adding a supersource and a supersink. Capacities of the additional arcs should be set to infinity and flow costs to zero. Loops presents no special difficulties. The ideas explored here are quite general and may be employed in other flow problems of equal interest which have not been described here.

4.5.3 Civil Engineering Applications of Directed Networks

Construction planning is perhaps the best known application of directed networks in civil engineering but there are many other important uses. In construction planning the network is used as a framework for analysis and project control. It is a useful way of representing diagrammatically and numerically the interrelationships among different activities and it permits the effects of different planning policies and decisions to be evaluated in advance of any action. Applications of directed networks outside construction planning use networks in a similar way, as a framework for representing and evaluating the effects of many trial policies. The major applications, therefore, lie in the planning and operation stages of a project. Although undirected networks are sometimes used for specific planning problems, directed networks are more common. In an engineering sense

www.EngineeringBooksPDF.com

arcs in a network usually have an implied sense of direction resulting from the nature of the project they represent. The aim of the planning process is to decide upon numbers and quantities to associate with each arc and so this naturally leads to the use of directed networks.

As an example consider a typical planning problem concerning the sewage treatment facilities of a small town which is expected to expand significantly in population. What should be done about sewage treatment? Is the existing plant big enough? If it is, could it operate efficiently with an increased population and still permit possible maintenance shutdowns or failures of some of the treatment processes? If it isn't big enough, on which of the existing processes should money be spent in order to increase the capacity? Which new processes should be introduced? How much will it all cost and how should the modified plant be operated? These are typical planning questions which must be asked and answered in a positive numerate way. The use of a directed network greatly simplifies the way in which many of the answers are obtained.

First of all the existing treatment facilities should be represented in directed network form such as figure 4.10 showing the links between different processes. Then capacities and costs of all the processes should be evaluated and added to the network. The maximum throughput of the existing plant can then be calculated using the maximum flow algorithm described earlier, and compared with estimates of likely future throughput resulting from an increased population. Arcs of the network may be deleted to represent process failures or maintenance shutdowns and new estimates of throughput may be made. Studies such as these determine whether the existing plant is large enough and whether safety margins against partial failures are adequate. If the plant is not large enough then further examination of the network will show up those arcs which have spare capacity and those which are flow bottlenecks. An inexpensive expansion programme might spend money on increasing the capacity of the bottlenecks to soak up the spare capacity elsewhere. The network allows the effect of such a policy on the plant throughput to be evaluated and compared with requirements. If this is still not sufficient then extra treatment processes must be built. These can be represented as extra arcs in the network with extra costs and capacities. The network can then be used to study the effects of different new arcs in different positions. Finally, for any expanded system, performance can be estimated in less than maximum flow conditions and minimum cost flow routing used to determine possible operating policies.

Essentially the network is a numerical model upon which a wide variety of planning experiments may be carried out. The algorithms which will be used in any planning problem depend very much on what planning questions are important to the over-all project which the network represents. Sometimes flow algorithms are needed, sometimes path algorithms, sometimes others. Usually a wide variety of methods are used. The end result of this process should ideally be a complete dossier of alternative policies which may be pursued. Each policy should have been evaluated in detail under possible adverse operating

www.EngineeringBooksPDF.com

situations and the costs of each policy should be known. The final decision-making body, whatever it may be, then has only to choose from the alternatives and is aware of the consequences of its decision. The directed network is, therefore, much more than a mathematical device: it is a comprehensive planning tool of great versatility.

A similar use for directed networks is found in transport planning. As cities expand, their traffic problems grow more acute. Many cities have tried to keep traffic moving by introducing complicated one-way street systems (directed networks). This removes the freedom of choice to the traveller which is offered by two-way streets but effectively increases the flows which can be carried. The directed network is a fairly obvious way of representing a street system and it allows many planning problems to be examined and solved numerically rather than practically. Many different alternative one-way street systems can be examined to provide practical solutions to traffic planning problems. Advanced algorithms have been developed from general network concepts, especially for traffic planning, so that different policies may be tried out well in advance of any implementation.

As a final example consider the planning associated with the terminal facilities of a large airport. At some existing airports the arrival within a short space of time of two wide-bodied jets each with up to 400 passengers causes the terminal facilities to seethe chaotically resulting in long delays to the individual traveller. Yet other airports manage to handle upwards of 100 000 passengers per day during the peak period with very few delays. The good or poor performance of airport terminal facilities is largely a result of the planning which went into them and the use of network ideas is vital to good airport planning. Typically, incoming and outgoing passengers are segregated and handled by separate systems (directed networks). For example, consider the passengers arriving on an incoming international flight. Some will be international transit passengers for foreign destinations who must be streamed off into transit lounges before they join their onwards flights. Others will be arriving foreign nationals requiring immigration checks. These must be streamed off from arriving nationals who require simpler passport checks. Both streams must then be reunited in baggage claim areas before they pass through customs. Of course the baggage itself must already have been unloaded and transit and terminating baggage sorted out. After passing through customs some passengers will require onward flights to national destinations; they must join the departure network. Others require transport away from the airport by rail, bus, car. Others require waiting areas, meeting areas, banks, hotel and flight reservations, etc.

For the departure network a completely separate system is necessary – check-in points, waiting areas, passport control, security checks, duty-free areas, lounges, etc. The arriving and departing passenger networks are, therefore, separate directed graphs which may share some common facilities (catering, customs, etc). In order to handle 100 000 passengers per day there is little scope for reliance on 'lucky' design. Everything must operate faultlessly or chaos will

www.EngineeringBooksPDF.com

ensue. This means that the initial design of the terminal facilities must be such that all facilities (arcs on the network) are capable of handling the anticipated flows. The designer must manipulate expected passenger flows, volumes, capacities, times, etc., so that under estimated peak conditions steady flow will be maintained. From a specific civil engineering viewpoint the directed networks concepts of this chapter are used in planning a new airport to make very basic decisions about how large all the terminal areas need to be, how they should be laid out in relation to one another, what special facilities should be incorporated (moving walkways, baggage handling facilities, etc.), how much the whole project will cost and what performance can be expected from it. Thus for a new airport the use of network concepts is vital to the initial planning stages, feasibility studies, and to the macro-design of the major elements. Once again the mathematical, algorithmic side of directed networks is perhaps less important than the over-all network approach which relates all the elements together in a comprehensible way and allows many different planning ideas to be tried out at a very early stage in the project.

The final section of this chapter is concerned with undirected networks and graphs. In practice most civil engineering applications of networks turn out to use directed graph concepts. There are, however, some specific problems that arise from time to time and can only be represented in undirected network or graph form. The next section introduces some of these problems which have civil engineering applications.

4.6 UNDIRECTED NETWORKS AND GRAPHS

Sections 4.4 and 4.5 have shown that the basic ideas used to analyse construction planning networks can be modified and generalised to become suitable for use in solving other directed network problems. Some difficulties were encountered, however, in section 4.5.1 in trying to generalise some results for path problems. These difficulties increase if a further extrapolation of results is attempted from directed networks to undirected networks and graphs. To change a directed network into an undirected one merely requires the removal of all directional restrictions upon the arcs. This gives freedom of choice about the direction in which any arc is traversed and hence creates a freer, less restrictive type of problem. The change, however, also alters the whole character of the problems associated with undirected networks.

In directed networks, problems centred around a source, a sink and the paths and flows between them. Recalling the definition of a source and a sink given in section 4.4.1 it immediately follows that, in a network without specified arc directions, the concept of a source and a sink loses precision – any vertex may be arbitrarily chosen as a 'source' and any vertex can become a 'sink'. Freedom of choice about the direction in which an arc is traversed also implies that looping is inherently possible in most undirected networks. Hence longest path

www.EngineeringBooksPDF.com

problems cease to have significance. Consequently, the problem of finding the longest or shortest path through a directed network from source to sink has, as its closest analogue in undirected networks, the problem of finding a shortest path between any pair of vertices.

The derivation of an algorithm for finding shortest paths in undirected networks will not be presented here. As has already been stated most practical applications of networks tend to use directed rather than undirected network concepts. Nevertheless a full range of algorithms for undirected networks is available in the literature. It is more useful here to describe briefy some problems of undirected networks which do not arise in directed networks but are of practical value.

4.6.1 The Postman Problem

The postman problem is such a well-known problem in graph and network theory that it is usually referred to by this title, but of course, it has much wider applications than the title suggests. Consider the undirected network of figure 4.14. Suppose it represents a network of streets. A postman picks up letters at the post office situated at one of the vertices. The letters have to be delivered to premises on the arcs of the network. The postman wishes to travel down all arcs of the network delivering letters and arrive back at his starting vertex having delivered all the letters. What route must the postman take if he wishes to minimise the total distance travelled?

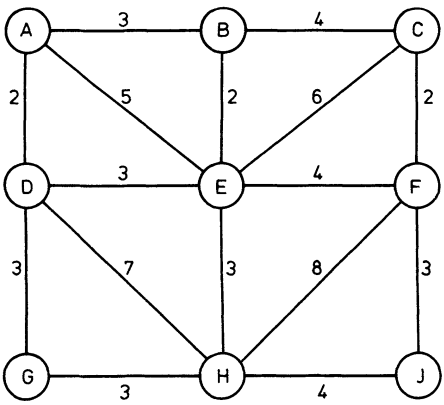


Figure 4.14 An undirected network

The absolute minimum travel distance in figure 4.14 must be equal to the sum of the lengths of all the arcs. To achieve this absolute minimum distance each arc must be covered once only. Does a route through the network exist which covers each arc only once and arrives back at the starting point? The

www.EngineeringBooksPDF.com

antiquity of the postman problem may be judged from the fact that such a route, if it exists, is called an *Euler tour* after the Swiss mathematician Leonhard Euler (1707–83) who studied the problem. If an Euler tour exists it must by definition be an optimal route for the postman. If an Euler tour does not exist then the postman must resign himself to traversing at least one arc twice. His problem then becomes one of finding a route which minimises the sum of the lengths of the arcs traversed twice.

The key to solving the postman problem lies in the vertices. In an Euler tour the postman must arrive at a vertex along one arc and leave by another. Thus a vertex with an even number of arcs may be part of an Euler tour but a vertex with an odd number of arcs may not. (For example, at a vertex with three arcs two may be used by the postman arriving and leaving but if he later arrives along the third arc he may only leave along an arc which he has already covered.) Thus a necessary condition for a network to contain an Euler tour is that all vertices must have an even number of incident arcs. If any vertex has an odd number an Euler tour is not possible.

Many methods have been proposed for solving the postman problem and the reader is referred to specialist literature for details (for example, Minieka). The problem is included here because it has practical applications. In a city, solid waste collection from domestic, commercial and industrial premises by large collection vehicles must be organised and planned so that the areas allocated to each team are similar. Then, within each area, routes must be worked out so that all streets are covered as efficiently as possible, that is, as in the postman problem. Other similar uses of the postman problem include routing of street-sweeping and spraying vehicles, winter gritting and salting of roads and the routine maintenance and inspection of street lighting and other public services.

4.6.2 The Salesman Problem

A somewhat similar problem to the postman problem for undirected networks is known as the salesman problem and can be posed as follows. Let the undirected network of figure 4.14 represent a system of towns and highways. A salesman starts at his home town (any one of the vertices) and wishes to make calls upon clients in all other towns (all other vertices of the network) before returning home. Find the route which passes through all vertices of figure 4.14 and has the minimum total length.

In the postman problem all arcs have to be traversed so, necessarily, all vertices will also be visited at least once. In the salesman problem it is not necessary to cover all the arcs but all vertices must be visited. It is clearly a path problem since a shortest path must be found but the 'source' and 'sink' of the path are the same vertex. Parallels exist with the Euler tour of the postman problem; in the salesman problem a path which calls at each vertex once only is called a *Hamiltonian circuit* after the Irish mathematician Sir William Hamilton

www.EngineeringBooksPDF.com

who first studied the problem in 1859. Good algorithms for the salesman problem exploit aspects of hamiltonian circuits to develop shortest paths.

As with the postman problem, no specific algorithms are investigated here. Nevertheless the salesman problem has practical applications. Apart from the obvious direct application to routing of salesmen it is used in the distribution and delivery of goods, regular inspection of engineering works, maintenance of pumping and gauging stations for water supply systems, etc. The salesman problem also leads naturally into the final topic of this section.

4.6.3 Spanning Trees

A *tree* is a sub-graph (part of a larger graph) which is a connected set of arcs and vertices containing no loops. A *spanning tree* is a tree which includes every vertex of the graph. Figure 4.15 shows various trees and spanning trees of the network of figure 4.14. In figure 4.15, a, b and c are trees, d, e and f are spanning trees because they include all vertices. Clearly a network such as figure 4.14 contains very many trees and spanning trees. Trees and spanning trees are found in all forms of graphs and networks both directed and undirected.

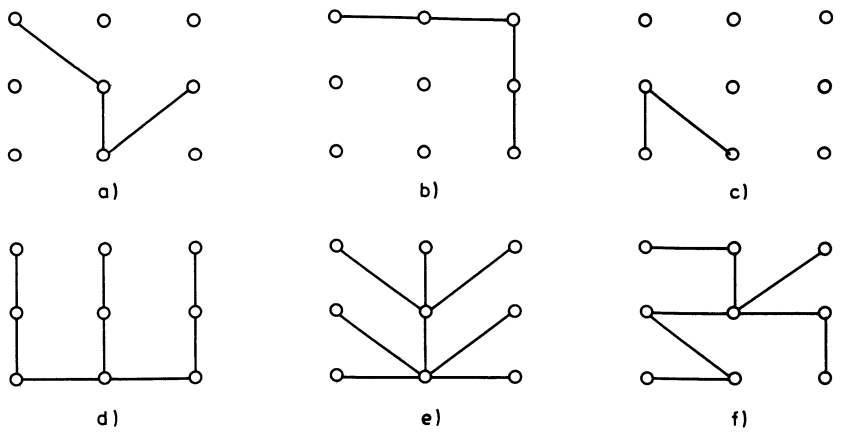


Figure 4.15 Trees (a, b, c) and spanning trees (d, e, f) of figure 4.14

Each of the arcs of the network of figure 4.14 has a length associated with it so each spanning tree of that network may have a different total length. Many practical applications of trees centre around finding a spanning tree which has a maximum or a minimum total length. If the vertices of a network represent towns which must be linked together by a water distribution pipeline or by new roads or by telephone or electric power lines, etc., then the *minimum spanning tree* represents the cheapest way of linking all the towns together if the arc lengths represent costs. It is this idea of the cheapest way of linking many points

www.EngineeringBooksPDF.com

together which forms the basis of many practical applications of minimum spanning trees. A very simple spanning tree algorithm will be described here which locates a spanning tree of a network. It is followed by a simple extension which allows a minimum or maximum spanning tree to be found.

4.6.3.1 A Spanning Tree Algorithm

The algorithm examines arcs of a network and sorts them into arcs which form a spanning tree and arcs not in the tree. To do this each arc is allocated a colour, blue or red, and the vertices of an arc are consigned to one of a number of stores. This use of colouring and stores is merely a device for sorting and classifying the arcs and vertices and is a convenient way of representing a computational process in which information is assigned to a particular region of the computer memory.

Step 1: All arcs are initially uncoloured and all stores are empty.
Step 2: Select any arc. Colour it blue and place both its end vertices in an empty store.
Step 3: Select any uncoloured arc. If no uncoloured arc exists then go to step 5. The uncoloured arc must satisfy one of four conditions
(a) Neither end vertex is in a store. Colour the arc blue and place both its end vertices in an empty store.
(b) One end vertex is in a store, the other end vertex is not in a store. Colour the arc blue and place the unstored end vertex in the same store as the other end.
(c) Both end vertices are in the same store. Colour the arc red and repeat step 3.
(d) Each end vertex is in a different store. Colour the arc blue and combine the contents of the two stores into a single store leaving the remaining store empty.
Step 4: Examine each store. If any store contains all the vertices then stop, the blue arcs now form a spanning tree. Otherwise, repeat step 3.
Step 5: Examine each store. If any store contains all the vertices then stop, the blue arcs now form a spanning tree. If no store contains all the vertices then stop, no spanning tree exists.

4.6.3.2 Maximum and Minimum Spanning Tree Algorithms

The spanning tree algorithm may be very simply modified to calculate minimum and maximum spanning trees. To find a minimum spanning tree the arcs must be examined in ascending order of arc length starting with the smallest length arc. To find a maximum spanning tree the arcs are examined in descending order starting with the longest length arc. For both minimum and maximum spanning trees if two or more arcs have the same length they may be examined in any order.

www.EngineeringBooksPDF.com

Table 4.2 Operation of minimum spanning tree algorithm on figure 4.14

Arc	Length	Colour	Stores		
			1	2	3
-			empty	empty	empty
A-D	2	blue	A,D	empty	empty
B-E	2	blue	A,D	B,E	empty
C-F	2	blue	A,D	B,E	C,F
A-B	3	blue	A,B,D,E	empty	C,F
D-E	3	red	A,B,D,E	empty	C,F
D-G	3	blue	A,B,D,E,G	empty	C,F
G-H	3	blue	A,B,D,E,G,H	empty	C,F
E-H	3	red	A,B,D,E,G,H	empty	C,F
F-J	3	blue	A,B,D,E,G,H	empty	C,F,J
B-C	4	blue	A,B,C,D,E,F,G,H,J	empty	empty
E-F	4				
H-J	4				
A-E	5				
C-E	6				
D-H	7				
H-F	8				

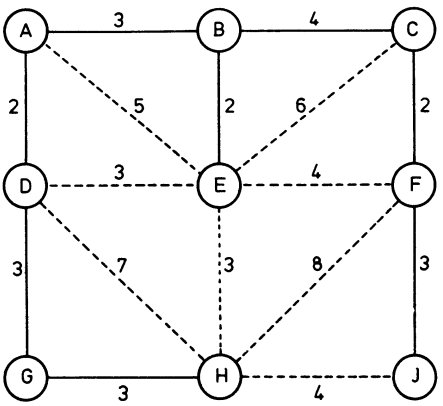


Figure 4.16 A minimum spanning tree of figure 4.14

www.EngineeringBooksPDF.com

As an example, consider figure 4.14 and find a minimum spanning tree of that network. Table 4.2 shows the calculations. The sixteen arcs are listed in ascending order of length. The next column is for the colour allocated to each arc and the next three columns are stores. Three stores are needed for this problem but the number required is not usually known in advance. The spanning tree algorithm of section 4.6.3.1 is then applied to each arc in turn and is self-explanatory. After arc B–C has been examined the algorithm terminates because store 1 contains all nine vertices of the network. The blue-coloured arcs now form a minimum spanning tree of the network and its total length is 22 units. Figure 4.16 shows the minimum spanning tree.

SUMMARY

This chapter has described one of the major methods which can be used for planning and controlling the actual construction of civil engineering projects – critical path networks. It has shown how a network representing the activities of a project may be drawn and analysed to determine which of the construction tasks are critical to the project's completion time and which tasks have some flexibility. The problems of resource allocation and the use of the network to maintain control of the construction work have been discussed. The fundamental linearity of construction planning networks was demonstrated and several problems of network analysis were formulated as linear programming problems thus linking this chapter with the previous one. From a systems viewpoint this fundamental linearity is most important as it confirms the very wide applicability of linear programming which can be seen as a basic element of the systems tool kit.

Chapter 4 then continued by examining construction planning networks in a much wider context as merely a specific (though very important) example of general networks. General features of directed networks were examined and the fundamental linearity of path and flow problems was demonstrated. It was shown that the concepts and ideas of directed networks are of great value in the planning of many different kinds of civil engineering projects. The network diagram of arcs and vertices is an ideal way of representing the complicated interactions and interrelationships of any project in a clear pictorial fashion. The network also allows many different planning ideas or policies to be tried out theoretically and their effects to be assessed. Not all practical problems can be represented as directed networks; some problems require undirected networks and graphs. Typical problems were examined. The chapter has shown that the concept of a network is a very basic one which can be used throughout the whole range of civil engineering planning, design, construction and operation problems. The network is a systematic unifying concept which allows many widely different practical problems to be examined and solved by a number of very simple methods such as linear programming and other linearly-based methods. Occasionally, as was noted in this chapter, the linearity appears to

www.EngineeringBooksPDF.com

break down in the face of discrete-valued, discontinuous variables and functions. The next chapter shows that, even for these problems, network ideas can be used to provide solutions. The network, therefore, is confirmed as another major component of the systems philosophy.

BIBLIOGRAPHY

Battersby, A., *Network Analysis for Planning and Scheduling*, second edition (Macmillan, London, 1967)

Ford, L. R., and Fulkerson, D. R., *Flows in Networks*, (Princeton University Press, 1962)

Minieka, E., *Optimization Algorithms for Networks and Graphs*, (Marcel Dekker, New York, 1978)

Moder, J. J., and Phillips, C. R., *Project Management with CPM and PERT* (Van Nostrand Reinhold, New York, 1970)

Pilcher, R., *Principles of Construction Management*, second edition (McGraw-Hill, London, 1976)

Wilson, R., *Introduction to Graphy Theory* (Oliver & Boyd, Edinburgh, 1972)

EXERCISES

4.1 A construction company must carry out the following operations in order to complete a small project.

Activity	Duration (weeks)	Interrelationships
A	2	Independent of all other activities
B	5	Independent of all other activities
C	7	Dependent on completion of A
D	3	Dependent on completion of C
E	6	Dependent on completion of A and B
F	9	Dependent on completion of B
G	1	Dependent on completion of B
H	7	Dependent on completion of D
I	5	Dependent on completion of C, E, F and G
J	4	Dependent on completion of G
K	8	Dependent on completion of G

Construct a network diagram for the project and then use it to establish

(1) the minimum project duration
(2) the critical path
(3) the total, free and independent floats for each activity.

www.EngineeringBooksPDF.com

4.2 In the sewerage construction network of figure 4.3, trenching (T) requires 3 days per length excavated, pipe-laying (L) 2 days per length, jointing (J) 2 days, backfilling (B) 4 days and reinstating (R) 3 days. How long will the complete project take?

When all the lengths of trench have been excavated the excavator may be converted to assist in any backfilling which remains to be done. One day is required to convert the excavator and each length of backfilling then takes only 2 days instead of 4. What effect does this have on the completion time of the work?

What is the effect on the project completion time if only one specialist laying and jointing crew is available and laying and jointing can only be performed by this crew?

4.3 The network shown in figure 4.7 represents a construction project with activity durations quoted in months. Shortly after starting construction work a steel fabrication sub-contractor warns that he will only be able to fulfil his order for steelwork after some delay. The steelwork is required for activities 5-7, 6-7 and 6-8. The sub-contractor offers two alternatives. He can supply either the steelwork for activities 6-7 and 6-8 at the end of month 15 and that for activity 5-7 one month later, or the steelwork for activity 5-7 at the end of month 15 followed one month later by the rest of the order. Which alternative should be requested from him?

4.4 Pessimistic, realistic and optimistic estimates of activity durations are shown below for the construction planning network shown in figure 4.4. Calculate expected activity durations, expected earliest and latest event times and an expected earliest completion time for the project. Hence find the expected critical path. What is the variance of the expected project completion time?

Activity	Estimated duration		
	Pessimistic	Realistic	Optimistic
A	10	6	4
B	7	5	3
C	4	3	2
D	7	5	4
E	10	6	4
F	10	7	5
G	8	4	3
H	8	5	3
J	3	1	1
K	6	4	3
L	4	2	1
M	10	5	2

www.EngineeringBooksPDF.com

4.5 Find earliest and latest times for all events in the construction network shown in figure 4.9 by solving two linear programming problems. Use the following set of activity durations.

Activity	Duration	Activity	Duration
1-2	2	2-4	4
1-3	6	2-5	8
1-4	7	3-5	4
2-3	3	4-5	5

Check your answers using the simpler event time algorithms.

4.6 Find the lengths of the longest and shortest paths through the directed network shown in figure 4.10 if the number by each arc represents its length.

4.7 Find the maximum flow that can be passed from source to sink through the directed network of figure 4.10 if the number by each arc is its flow capacity.

4.8 Find the cheapest way of passing 7 units of flow from source to sink through the network shown in figure 4.9. The number beside each arc is its flow capacity. Assume that for each arc the cost of passing 1 unit of flow is proportional to the reciprocal of the arc capacity. The proportionality coefficient for all arcs is 100.

4.9 Find the length of a maximum spanning tree of the network of figure 4.14.

4.10 A water authority wishes to record levels at six gauging stations, A to F, and monitor these levels at a central control point, P. Level sensing and recording devices have been installed and it remains to link these back to the control point by means of land lines. Approximate costs have been calculated for land lines joining all pairs of points, A to F and P, and are as shown below.

	P	A	B	C	D	E	F
P	0	12	37	25	9	53	18
A		0	14	19	44	36	28
B			0	22	16	13	36
C				0	38	32	44
D					0	20	51
E						0	15
F							0

What is the cost of the cheapest system of land lines which links all the gauging stations to the central control point P?

www.EngineeringBooksPDF.com

5 SERIAL SYSTEMS AND DYNAMIC PROGRAMMING

There are many ways in which a problem may be categorised according to its major features – linear, non-linear, continuous, discrete, stochastic, etc. One important category of problems which cuts across all the conventional category boundaries consists of problems which exhibit serial or sequential features. This chapter examines the nature of sequential systems and demonstrates how problems can be solved by a method known as dynamic programming (DP).

Dynamic programming is best thought of as a solution philosophy rather than a numerical technique. It does not have a standard algebraic or algorithmic form as does, for example, the simplex method of linear programming. Each problem must be studied carefully and manipulated into a serial form before it can be solved. Since so much preparatory work is required DP is possibly harder to comprehend immediately than some other solution methods. Nevertheless the effort necessary to understand DP fully is worth making because DP has some very positive advantages; it can be a very rapid and efficient solution technique, it is not troubled by discrete-valued or integer-valued variables which can be troublesome to many other methods and DP can handle discontinuous functions or tabular values of functions as easily as continuous ones. For practical engineering planning and design purposes these are very important attributes.

Chapter 5 starts by examining in detail a critical path network problem and uses this example to establish the precise nature of a serial system. A solution to this example is found using the shortest path algorithm for directed networks from the previous chapter. This algorithm when applied to a serial system develops quite naturally into the DP method which is described in detail. Having established the DP method, several other practical examples from civil engineering are presented and solved in detail, each example highlighting a specific feature of the method.

EXAMPLE 5.1 – A CRITICAL PATH PROBLEM

Towards the end of a large construction project the single critical path consists

www.EngineeringBooksPDF.com

of five activities which have still to be completed. These activities A, B, C, D and E have estimated durations of 7, 6, 5, 8 and 6 weeks respectively, thus the estimated time needed to complete the project is 32 weeks. Cumulative delays, however, have been encountered in the earlier stages of the project and the agreed completion date is only 29 weeks away. The contractor wishes to reduce the length of the critical path by 3 weeks in order to meet the agreed deadline. To achieve this reduction extra plant and labour must be supplied at extra cost. How should the contractor replan the critical activities so as to save 3 weeks at least extra cost to himself?

In order to examine and solve this problem a couple of simplifying assumptions are made. Firstly it is assumed that the critical activities and the critical path will not alter if activity durations are altered. Any non-critical paths consist of activities with sufficient float not to become critical and affect the problem. Secondly, it is assumed that each critical activity can only be reduced by an integer number of weeks. Both these assumptions simplify the problem so as to exhibit the principles of the solution approach more clearly but are not essential to the solution process. Once the solution method is fully understood it becomes obvious that the assumptions may be relaxed or removed without any loss of applicability of the solution method.

This example has been chosen to illustrate a sequential or serial system because chapter 4 has already demonstrated the sequential nature of the activities in a critical path. Activity A must be performed before activity B which must in turn precede activity C, etc. This example will be solved using some network ideas also taken from chapter 4. The contractor's problem is that he has to save a total of 3 weeks by the time activity E is completed. There are very many possible ways in which a total saving of 3 weeks may be achieved over five activities; all 3 weeks could be saved in one activity, or the savings could be made a week or 2 weeks at a time over several activities; the possibilities are many. The solution method used here consists of constructing a directed network which represents all possible ways of saving 3 weeks and then using the shortest path algorithm of chapter 4 to find that path through the network which has the least total cost.

With this solution approach in mind consider activity A. Before activity A starts, zero time has been saved. When activity A is finished four possibilities exist – zero time has been saved, 1 week has been saved, 2 weeks have been saved or 3 weeks have been saved. The cost of saving zero time in activity A is zero. Let the costs of saving 1, 2 or 3 weeks in activity A be C_{A1}, C_{A2}, C_{A3}, respectively. Figure 5.1 shows activity A with its preceding and succeeding events and below it is a directed network which represents the time saving possibilities for activity A. A single source vertex represents the zero time saved before activity A starts and four vertices arranged in a column represent the four possibilities which can exist when activity A is finished. Directed arcs link the source to each of the other four vertices and each arc represents a possible time-

www.EngineeringBooksPDF.com

saving policy for activity A. The cost of each policy is placed beside the appropriate arc.

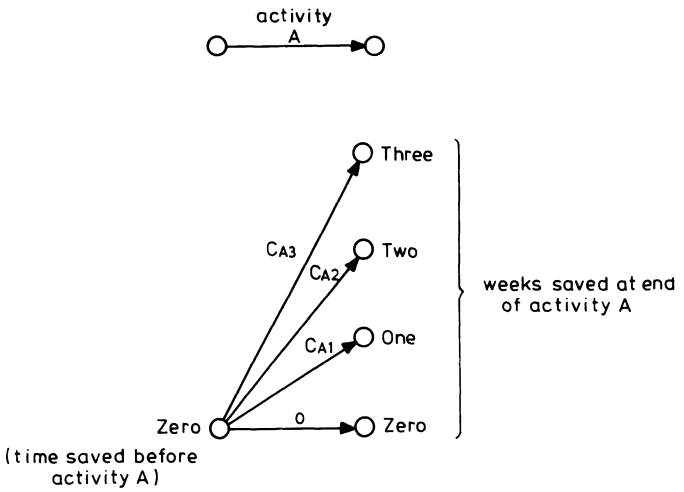


Figure 5.1 Possible time savings in activity A

Now consider activity B. Since activity B follows activity A sequentially, the four possibilities of time saved at the end of A must represent four possible times saved before the start of B. When activity B is completed, four possibilities exist again for the cumulative time saved so far – it must be zero, 1, 2 or 3 weeks. A new column of four vertices represents this. Figure 5.2 shows figure 5.1 expanded to include activity B. For the sake of clarity the directional arrows have been omitted; all arcs must be traversed from left to right. Also all arc costs have been omitted. The arcs in figure 5.2 which link the two columns of vertices represent all the possible time-saving policies for activity B. For example, if 3 weeks have been saved by the end of activity A then only one possibility exists for activity B – no further time need be saved. Thus only one arc leads out of the vertex labelled 'Three' after activity A, and this arc terminates at the vertex labelled 'Three' after activity B. The cost to be associated with this arc is zero since no time is saved in activity B by this policy. As a second example, if 1 week has been saved by the end of activity A this vertex on figure 5.2 must be connected to three succeeding vertices. Possibilities exist in this case of saving zero, 1 or 2 weeks in activity B in addition to the 1 week already saved. Thus three arcs that leave the vertex labelled 'One' after activity A can be drawn. The costs to be associated with each of these arcs are C_{B2}, C_{B1} and zero corresponding to the costs of saving 2, 1 and zero weeks in activity B respectively. These costs are

www.EngineeringBooksPDF.com

associated with the respective arcs which join the vertex labelled 'One' after activity A to the vertices labelled 'Three', 'Two' and 'One' after activity B.

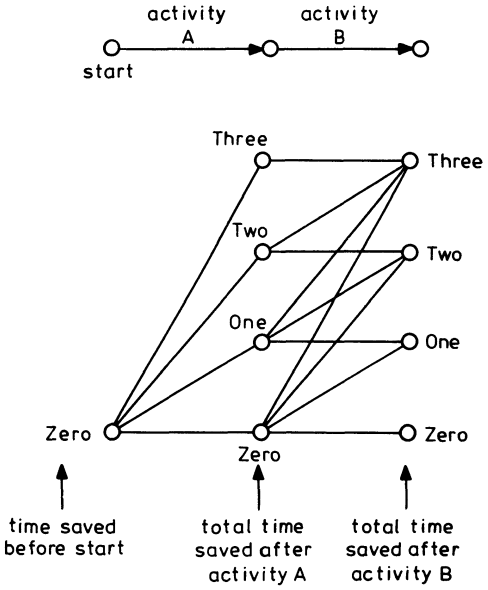


Figure 5.2 Possible time savings after two activities

The network of possible policies can be further extended by adding activity C to those already examined. Figure 5.3 shows this. Four possibilities exist for the cumulative time saved after activity B has been finished – zero, 1, 2 or 3 weeks may have been saved. The same possibilities must also exist after activity C is completed so a new column of four vertices represents these possible states after activity C. The arcs linking the last two columns of vertices represent possible ways of saving time in activity C and they link together the vertices in the last two columns entirely logically. The reasoning is exactly the same as was used in figure 5.2 and the examples quoted for that apply to figure 5.3 also. Each of the new arcs in figure 5.3 must be costed according to the number of weeks saved in activity C which it represents. These costs would be zero, C_{C1}, C_{C2} and C_{C3}. Once again all arc directions and costs have been omitted for clarity.

The network of figure 5.3 can be extended for activity D in exactly the same way as above. For the final activity E, four possibilities of total time saved after activity D (that is, before activity E) will exist – zero, 1, 2 or 3 weeks saved. When activity E is completed only one possibility must exist – 3 weeks total must have been saved. Thus only one vertex exists after activity E; a single sink. Arcs link each of the possible states which might exist after D to this single sink

www.EngineeringBooksPDF.com

and each arc corresponds to a time which must be saved in activity E. Each arc is costed according to the time saved, C_{E3}, C_{E2}, C_{E1} and zero.

Figure 5.4 shows the complete network of possible time-saving policies for the critical path of five activities. Directional arrows are omitted but each arc

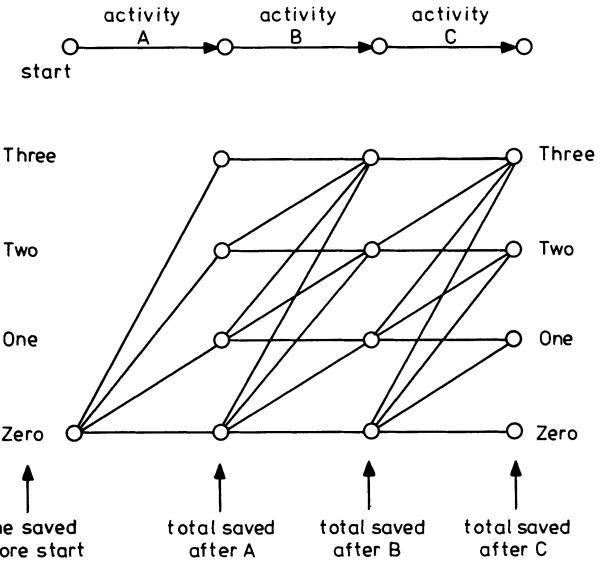


Figure 5.3 Possible time saving policies for three activities

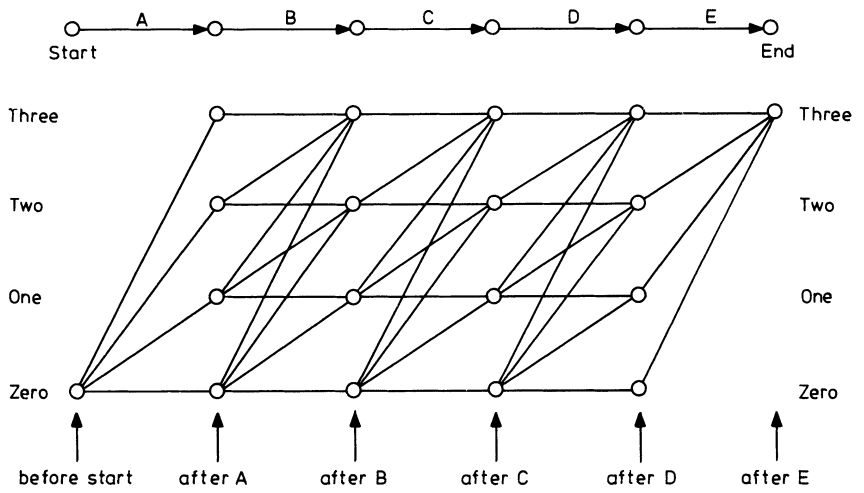


Figure 5.4 Complete network of possible time-saving policies

www.EngineeringBooksPDF.com

must be traversed from left to right only. Each arc has a known cost associated with it. This directed network has a single source and a single sink and has been so constructed that any path from source to sink will save a total of 3 weeks over the five activities in the critical path. Clearly there are very many possible paths and each path will in general have a different total cost. Chapter 4, however, has already described a number of ways of rapidly finding the shortest path (in this case the least cost path) through a directed network such as figure 5.4. The shortest path algorithm described by relationships 4.19 in section 4.5.1 is perhaps the most rapid method of finding the least cost path. It is worth examining the operation of this algorithm on figure 5.4 because it leads directly to a powerful general solution approach to all serial systems.

Firstly, the algorithm assigns a total cost of zero to the source vertex. It then examines each of the vertices in the column labelled 'after A' and to each vertex in that column it assigns a cost equal to the cost associated with the arc linking that vertex to the source. The vertices labelled 'Three', 'Two', 'One' and 'Zero' in the 'after A' column have costs of C_{A3}, C_{A2}, C_{A1} and zero given to them respectively. The algorithm then moves on to the vertices in the column labelled 'after B' and to each vertex in that column it assigns a value equal to the minimum cumulative cost along all paths from the source which enter that vertex. For example, consider the vertex labelled 'Three' in the 'after B' column. Four arcs enter that vertex from vertices in the 'after A' column. Costs of travelling from the source to each of those four 'after A' vertices have just been calculated and to each of these costs is added the cost associated with the arc linking that vertex to the candidate vertex 'Three' in the 'after B' column. Thus there are four cumulative total costs at this vertex each corresponding to a distinct path. The algorithm chooses the least of the four costs and gives it as a least cost to that vertex. In fact the algorithm gives a cost of

$$\text{Minimum} \left\{ \begin{array}{l} C_{A3} + 0 \\ C_{A2} + C_{B1} \\ C_{A1} + C_{B2} \\ 0 \quad + C_{B3} \end{array} \right\} \tag{5.1}$$

to vertex 'Three' in the 'after B' column. A similar process allocates to each of the other 'after B' vertices a cost equal to the cheapest cumulative cost on any path entering the vertex. Having dealt with the 'after B' vertices the algorithm moves on to vertices in the 'after C' column assigning a minimum cumulative cost to each vertex, then to the 'after D' vertices and finally to the sink vertex. For this last stage the algorithm will already have assigned to each vertex in the 'after D' column a cost equal to the least total cost along any path from the source which reaches that vertex. To each of these cumulative costs 'after D' is added the cost of traversing the arc linking that vertex to the sink and the algorithm selects the lowest of these four new cumulative costs. Thus the over-all least cost of saving 3 weeks over the five activities is found.

www.EngineeringBooksPDF.com

Having found the least cost it is simple to trace out the path from source to sink which gives this cost. The path is traced out in a backwards direction. The least cost was given by one of four possibilities at the sink node. The least cost possibility includes one of the final four arcs of the network which can, therefore, be found and marked. The vertex at the tail of that arc will have a cumulative cost associated with it and the last element of that cost corresponds to one of the arcs between the 'after C' vertices and the critical 'after D' vertex. By examining the arcs terminating at the critical 'after D' vertex the arc in the least cost path can again be found and marked. The tail end of that arc locates a critical vertex in the 'after C' column and a further trace back isolates a critical arc between 'after B' and 'after C'. Eventually this process isolates the complete least cost path. Once the path is known the time-saving policy to which it corresponds can be determined.

This network approach to example 5.1 allows a solution to be found very easily. Some characteristics of the network approach should be noted as they are fundamental to serial systems. First of all, the problem itself is sequential in nature. Activity A must start the process, activity B cannot commence until A is complete, etc. Secondly, the network of possible solutions was constructed sequentially following the same sequence as the activities. Figure 5.1 representing possible policies for the first stage, activity A, was drawn first. Activity B, the second stage of the sequence, was then examined and figure 5.2 extended figure 5.1 for activity B. Activity C, the third stage, extended the network to give figure 5.3, etc. Thirdly, the solution algorithm when applied to the network also operated in a sequential fashion. It initially examined the paths between the source vertex and vertices in the 'after A' column, then it moved on to examine activity B, looking at the arcs between all 'after A' and 'after B' vertices, etc. From these three observations it is concluded that a problem involving a sequential process of several stages may be solved in a sequential fashion stage by stage. This is a very different approach from that used in solving linear programming problems, for example, where a single large problem is set up and solved. A sequential system may be solved as a sequence of smaller interconnected problems.

The network approach has been useful in solving example 5.1. In fact it has far more general applications than this example. It is a fundamental concept in solving serial systems by dynamic programming. In order to use it in other examples, however, some tidying up and generalisation is necessary.

5.2 GENERALISATION OF THE NETWORK APPROACH

In example 5.1 it has been shown that the series of activities A to E in a critical path form a five-stage sequential system. Indeed it is convenient to refer to the activities by the stage numbers, thus activity A becomes stage 1, activity B stage 2, etc. In the network solution of figure 5.4 the stages correspond to the arcs. Five sets of arcs are present with the columns of vertices separating the stages.

www.EngineeringBooksPDF.com

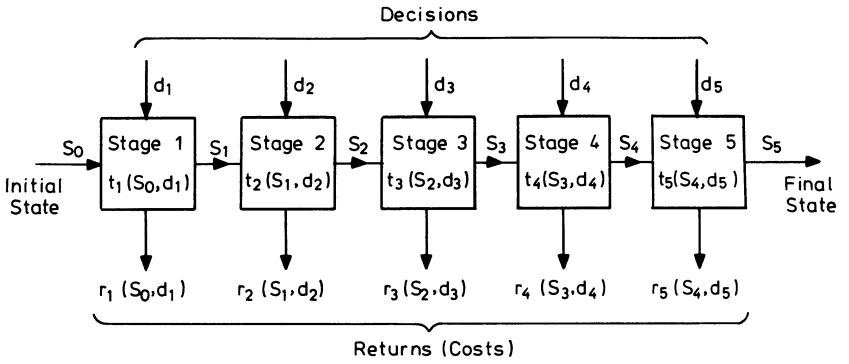


Figure 5.5 A general serial system (five stages)

Figure 5.5 shows a diagram of a five-stage serial system. Each stage is represented by a box and the stages are connected together in sequence by a *state variable* which passes through the entire system. The state variable is a variable or parameter that alters in value as it passes through the stages. Initially the state variable, S in figure 5.5, has the value S_0. As it passes through the first stage some *decisions* are made, d_1, which cause the value of the state variable to change to S_1, the *output state* from stage 1. The change in value of the state is represented by the *transition function* $t_1(S_0,d_1)$ so that

$$S_1 = t_1(S_0,d_1)$$

that is, the value of the output state from stage 1, S_1, is a function of the value the input state to stage 1, S_0, and the decisions made in stage 1. As a result of the decisions made in stage 1 and the change in value of the state variable, some costs or returns are generated by stage 1. These are represented by $r_1(S_0,d_1)$, that is, the stage 1 return is a function of the stage 1 input state and decisions.

S_1, the output state from stage 1, now becomes the *input state* to stage 2. In stage 2 decisions d_2 are made which alter the value of S from S_1 to S_2 by means of the transition function t_2

$$S_2 = t_2(S_1,d_2)$$

These stage 2 decisions and alterations to the value of S again generate returns for stage 2 given by $r_2(S_1,d_2)$. S_2 now enters stage 3 in which this process continues. Eventually stage 5 is reached and the state variable emerges from stage 5 with a final value of S_5.

To illustrate this with example 5.1 the five stages are clearly the activities, the decisions $d_1, \ldots, d_5$ are the number of weeks to be saved in each of the five stages, and the stage returns $r_1, \ldots, r_5$ are the costs of the savings in each of the five stages. The parameter which links everything together is the state variable, S and in example 5.1 S is the total number of weeks saved. Thus before stage 1 starts, S has the value S_0 = zero, since no time has yet been saved. After stage 1,

www.EngineeringBooksPDF.com

however, when a decision d_1 has been made on the number of weeks to be saved in stage 1, the value of S changes to S_1, where S_1 is given by

$$S_1 = t_1(S_0,d_1) = S_0 + d_1$$

This value of S_1, the total number of weeks saved after the end of stage 1 now becomes an input state to stage 2. A decision d_2 is made on the number of weeks to be saved in activity B (stage 2) and this alters S_1 to S_2 by virtue of the transition

$$S_2 = t_2(S_1,d_2) = S_1 + d_2$$

This process of modifying the state variable continues through all the stages until after the fifth stage when it has the value S_5. For example 5.1 a total of 3 weeks must be saved so the value of the final state must be $S_5 = 3$. Thus the example fits the serial system model of figure 5.5 exactly.

In order to generalise a solution method, attention is redirected to the network of possibilities in figure 5.4. Analogies will be drawn between this network and figure 5.5. It has already been noted that the five sets of arcs correspond to the possible policies available at each stage. The five sets of arcs are separated by the columns of vertices which reference to figure 5.5 suggests must represent the state variable S. In fact, the column of vertices labelled 'after A' represent the four possible numbers of weeks which could have been saved after activity A (stage 1) is finished. Each vertex in the 'after A' column, therefore represents a possible value of the state variable S_1. Each vertex in the 'after B' column represents a possible value of the state variable S_2. The source vertex labelled 'Zero' 'before start' clearly refers to the single value S_0 = zero and the sink vertex labelled 'Three' 'after E' refers to the requirement that by the end of the project 3 weeks must be saved, that is, $S_5 = 3$.

This demonstrates that the network of possibilities approach to the solution of a serial problem is equivalent to considering *all possibilities of values of the state variable after each stage*. Now, examining the operation of the shortest path solution method, it is seen that stage 1 is considered first. The value of the initial state is known ($S_0 = 0$). The solution method considers *all possible values of the output state* from stage 1. For each possible output state it examines the cost involved in achieving that output state value from the input state. The cheapest cost of achieving each possible output state value is associated with that particular value of S_1 and stored. The four possible output states from stage 1 are $S_1 = 0, 1, 2$ or 3. Since $S_0 = 0$, then $S_1 = 0$ is achieved by deciding to save zero weeks in stage 1, that is, $d_1 = 0$. The cost of this policy is zero so the cost associated with $S_1 = 0$ is $C'_{S_1=0}$ = zero, in which the superscript $'$ indicates a cumulative total cost associated with a particular value of an output state (in this case $S_1 = 0$). $S_1 = 1$ is achieved from $S_0 = 0$ by the decision $d_1 = 1$ and the cost of this is C_{A1}. Thus the cost associated with $S_1 = 1$ is $C'_{S_1=1} = C_{A1}$. Similar costs are evaluated for the possible output states $S_1 = 2$ and $S_1 = 3$. Note that in the above calculations only stage 1 has been examined.

Now stage 2 is examined. This time there are four possible input states to

www.EngineeringBooksPDF.com

stage two ($S_1 = 0, 1, 2$ or 3). There are also four possible values of the output state from stage 2, $S_2 = 0, 1, 2$ or 3. Consider each possible value of S_2 in turn, for example $S_2 = 3$. There are four possible ways of achieving $S_2 = 3$, one from each of the input states $S_1 = 0, 1, 2, 3$. Evaluate the *cumulative total cost* of each of the four ways of achieving $S_2 = 3$ and select the cheapest. Thus $C'_{S_2=3}$ is given by

$$C'_{S_2=3} = \text{minimum} \left\{ \begin{array}{l} C'_{S_1=3} + 0 \\ C'_{S_1=2} + C_{B1} \\ C'_{S_1=1} + C_{B2} \\ C'_{S_1=0} + C_{B3} \end{array} \right\} \tag{5.2}$$

Similar minimum cumulative costs are evaluated for $C'_{S_2=2}$, $C'_{S_2=1}$ and $C'_{S_2=0}$. In order to evaluate equation 5.2, which is the same as relationship 5.1, it should be noted that values for all the nodes C'_{S_1} are already known. They were evaluated in the first stage of the solution process. The only new calculations needed to evaluate equation 5.2 are in those for the various second stage costs. Thus, as was noted for stage 1, in order to calculate cumulative stage 2 costs only the stage 2 decisions and states are required.

Having calculated least cumulative costs C'_{S_2} for all possible values of state S_2, the solution moves on to the next stage, stage 3. The four possible S_2 values become input states to stage 3. Four possible output state values are defined, $S_3 = 0, 1, 2, 3$ and minimum cumulative total costs of achieving each of these values are calculated. The solution then proceeds stage by stage until stage 5 which has only one output state value $S_5 = 3$.

The solution process described above is, of course, identical to that described earlier for the network approach. In order to use the method above, however, the network is not required. The new method is described in terms of the general sequential system, figure 5.5, and is therefore a generally applicable method for all serial problems.

5.2.1 Dynamic Programming and Bellman's Principle of Optimality

The solution process for serial problems described in the previous section has been given the name *dynamic programming* (DP) by its originators. Unlike other mathematical programming methods such as linear programming, dynamic programming does not have a simple, rigorous mathematical formulation, nor can the DP calculations be easily programmed into a general computer package. Dynamic programming is a solution philosophy for serial systems rather than a rigorously detailed mathematical technique. This is why it has been introduced in this chapter by way of an example and why it can sometimes be harder to understand than other more explicit methods.

Dynamic programming is a way of solving problems which have a serial representation. Figure 5.5 shows precisely the required serial form. The stages must

www.EngineeringBooksPDF.com

be sequential, without loops and must be distinctly separate. Each stage must have its own decision variable or variables and these must not appear in any other stage. The stages must be linked together by a state variable or parameter whose value is governed by stage transition functions of the form shown. For the *i*th stage in a serial system the transition must be of the form

$$S_i = t_i(S_{i-1}, d_i) \tag{5.3}$$

that is, only S_{i-1} and d_i must appear in the transition function. Stage returns or costs must similarly consist of relationships of the form, for the *i*th stage, $r_i(S_{i-1}, d_i)$. Figure 5.5 shows all these requirements precisely.

Given that a problem has this serial form, the dynamic programming solution process is as described previously. Each stage is examined sequentially starting with the first, and a set of discrete output state values is postulated. For each discrete output state, cumulative total costs or returns associated with achieving that state value are examined and the minimum (or maximum if the problem requires it) value is selected from the possible candidate values. This process proceeds iteratively through the sequence of stages. This cumulative approach for possible output states is the essence of the DP method and can be stated thus

$$C'_{S_i} = \min_{d_i}(\text{or max})\ \{C'_{S_{i-1}} + r_i(S_{i-1}, d_i)\} \tag{5.4}$$

that is, the cumulative cost C' to be associated with a particular output state from the *i*th stage is equal to the least value (or greatest value if the problem is one of maximisation) obtained by adding the cumulative cost associated with each possible input state, $C'_{S_{i-1}}$, to the appropriate *i*th stage return $r_i(S_{i-1}, d_i)$. Relationship 5.4 is sometimes referred to as the *dynamic programming recurrence relationship*.

The topic of serial decision-making and sequential processes was pioneered in the 1950s by an American named Richard Bellman whose various books on the subject should be consulted for more detailed study. Bellman's group invented the DP method and gave it its name. As has already been mentioned DP is a solution philosophy rather than a mathematical method. Bellman himself stated the principle upon which DP is based and this has come to be known as *Bellman's Principle of Optimality*. It states: 'An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision'.

Bellman's principle means that at any stage in a serial system such as figure 5.5 the only information needed to determine an optimal solution for the remaining stages of the system is the input information to the present stage. Thus if figure 5.5 were entered at stage 3 instead of stage 1 it would only be necessary to know the values of S_2 and C'_{S_2} in order to find an optimum solution for stages 3, 4 and 5. It would be necessary to know how S_2 or C'_{S_2} were determined, or any details of earlier stages – the optimal policy for future stages does not depend upon how the present stage was reached.

A good understanding of DP is perhaps best gained by studying examples and

www.EngineeringBooksPDF.com

extracting the principles from them. Before considering further civil engineering examples of DP it is useful to round off example 5.1 in a numerical sense.

5.2.2 Numerical Solution of Example 5.1

Suppose the contractor has calculated the costs of saving 1, 2 or 3 weeks in each of the activities A, B, C, D and E. Table 5.1 shows these costs in units of £100.

Table 5.1 Cost of time savings for example 5.1

	Cost of time saved in activity				
Time saved (weeks)	A	B	C	D	E
0	0	0	0	0	0
1	3	2	4	3	4
2	7	6	6	5	8
3	12	10	9	8	12

In order to find the cheapest way of saving 3 weeks in total the solution proceeds as follows. Numerical calculations are given first with notes afterwards.

Stage One

$S_0 = 0; S_1 = 0, 1, 2, 3; S_1 = S_0 + d_1; C'_{S_0} = 0$

$$C'_{S_1=0} = 0$$
$$C'_{S_1=1} = 3$$
$$C'_{S_1=2} = 7$$
$$C'_{S_1=3} = 12$$

Stage Two

$S_1 = 0, 1, 2, 3; S_2 = 0, 1, 2, 3; S_2 = S_1 + d_2$

$$C'_{S_2=0} = 0$$
$$C'_{S_2=1} = \min \begin{Bmatrix} \textcircled{0+2} \\ 3+0 \end{Bmatrix} = \textcircled{2}$$
$$C'_{S_2=2} = \min \begin{Bmatrix} 0+6 \\ 3+2 \\ 7+0 \end{Bmatrix} = 5$$
$$C'_{S_3=3} = \min \begin{Bmatrix} 0+10 \\ 3+6 \\ 7+2 \\ 12+0 \end{Bmatrix} = 9$$

www.EngineeringBooksPDF.com

Stage Three

$S_2 = 0, 1, 2, 3; S_3 = 0, 1, 2, 3; S_3 = S_2 + d_3$

$$C'_{S_3=0} = 0$$

$$C'_{S_3=1} = \min \left\{ \begin{matrix} 0+4 \\ \textcircled{2+0} \end{matrix} \right\} = \textcircled{2}$$

$$C'_{S_3=2} = \min \left\{ \begin{matrix} 0+6 \\ 2+4 \\ 5+0 \end{matrix} \right\} = 5$$

$$C'_{S_3=3} = \min \left\{ \begin{matrix} 0+9 \\ 2+6 \\ 5+4 \\ 9+0 \end{matrix} \right\} = 8$$

Stage Four

$S_3 = 0, 1, 2, 3; S_4 = 0, 1, 2, 3; S_4 = S_3 + d_4$

$$C'_{S_4=0} = 0$$

$$C'_{S_4=1} = \min \left\{ \begin{matrix} 0+3 \\ 2+0 \end{matrix} \right\} = 2$$

$$C'_{S_4=2} = \min \left\{ \begin{matrix} 0+5 \\ 2+3 \\ 5+0 \end{matrix} \right\} = 5$$

$$C'_{S_4=3} = \min \left\{ \begin{matrix} 0+8 \\ \textcircled{2+5} \\ 5+3 \\ 8+0 \end{matrix} \right\} = \textcircled{7}$$

Stage Five

$S_4 = 0, 1, 2, 3; S_5 = 3; S_5 = S_4 + d_5$

$$C'_{S_5=3} = \min \left\{ \begin{matrix} 0+12 \\ 2+8 \\ 5+4 \\ \textcircled{7+0} \end{matrix} \right\} = 7$$

Least cost of saving 3 weeks = £700. Decisions are $d_5 = 0; d_4 = 2; d_3 = 0; d_2 = 1; d_1 = 0$, that is, save 1 week in activity B, 2 weeks in activity D.

Each stage is examined sequentially starting with stage one. Possible input and output states are written down together with the form of the transition

www.EngineeringBooksPDF.com

function for that stage. For the first stage an initial cost $C'_{S_0} = 0$ is specified. The recurrence relationship 5.4 is then used to calculate least cumulative costs for each possible output state. The encircled figures are used in the traceback of the optimal policy and are not encircled in the initial calculations. Eventually the least cumulative cost at stage five is found, that is, $C'_{S_5} = 7$ units = £700.

The traceback process locates the optimal policy. The value $C'_{S_5} = 7$ results from the encircled 7 + 0 at stage five. The zero there corresponds to $d_5 = 0$ and the 7 corresponds to $C'_{S_4} = 7$. This value $C'_{S_4} = 7$ leads to the corresponding value at stage four being circled. $C'_{S_4} = 7$ is given by the encircled 2 + 5 in stage four. The 5 comes from $d_4 = 2$ and the 2 from $C'_{S_3} = 2$. Thus in stage three $C'_{S_3} = 2$ is encircled. This is derived from 2 + 0 signifying $d_3 = 0$, $C'_{S_2} = 2$. $C'_{S_2} = 2$ comes from the encircled 0 + 2 in stage two, corresponding to $d_2 = 1$, $C'_{S_1} = 0$, that is, $d_1 = 0$. Thus values of the stage decisions and hence the optimal policy are found.

5.2.3 Comments on Example 5.1

At the start of the example, two assumptions were made. One of these was that each activity could only be reduced in duration by an integer number of weeks. It should now be fairly obvious why this assumption was made. Dynamic programming consists of examining different possible ways of achieving discrete output state values in each stage and the assumption ensured that discrete (in fact integer) values of the state were available. Although it is sometimes possible to solve DP problems for continuous variables (see section 5.6.4), this form of DP has very limited applications and for most practical problems the requirement that the state variable should be discrete-valued remains essential. The choice of only four output state values for this example was made, however, only to simplify the description of the solution. It would be perfectly reasonable to assume that time reductions could be considered in units of an integer number of days instead of weeks. This would give more, closer-spaced state variable values. Much more calculation would be required to solve the problem and a slightly lower total cost may be expected.

This highlights one aspect of the nature of DP solutions. If the state variable is truly a continuous variable, discrete DP will never find the exact optimum solution. By considering only a number of discrete values of the state only an approximate solution will be found. If the problem is one of minimisation the DP solution will be an upper bound to the exact solution, if maximising is involved the DP solution will be a lower bound. Sometimes it is important to know how close the bound is to the true solution but in the vast majority of engineering planning and management problems it is of little importance. For instance, in example 5.1 it is not usually practicable for a contractor to make profitable use of time savings of less than a day so although time is strictly a continuous variable its division into discrete one-day intervals still provides a wholly acceptable practical solution to the problem.

www.EngineeringBooksPDF.com

The second assumption made at the start of example 5.1 was that the five-activity critical path remained critical as the activity durations were changed. This assumption was made in order not to clutter the description of the solution with unnecessary complications. In practice, however, it is quite possible that as critical activities are retimed, other originally non-critical paths may become critical. In order to remove this second assumption and make the problem really practical it is necessary to study branched serial systems. This is done later in section 5.6.1 of this chapter.

One big advantage which DP possesses is that it makes practically no demands upon the forms of the transition or return functions, in contrast to, say, linear programming in which only linear functions of the variables are permitted. In dynamic programming the stage returns $r_i(S_{i-1}, d_i)$ and transitions $t_i(S_{i-1}, d_i)$ can be linear or non-linear, continuous or discrete-valued or indeed may have any form. The only requirement is that given a value of the decision d_i and the input state S_{i-1} at the ith stage it must be possible to calculate the return r_i and the output state S_i. In the numerical example of section 5.2.2, table 5.1 gives numerical values of costs (r_i) for different decisions (d_i) at several stages (i). The DP method is not concerned with how these costs are evaluated or what the functional relationships are; it merely asks that they exist. Similarly with the transitions, DP is not concerned with precisely how state S_i is related to S_{i-1} and d_i; it merely requires that some process exists whereby S_i can be calculated from given values of S_{i-1} and d_i. In fact, different return and transition functions can be used in different stages of a serial system.

In chapters 3 and 4 it was seen that many problems of resource allocation are linear when viewed with a wide perspective but when they are studied in closer detail the linearity dissolves into discreteness and disappears. As money is invested in an activity the activity duration decreases. This may be represented by a linear relationship which is useful for some problems of overall planning. For more detailed planning and resource allocation, however, the money may only be invested in discrete amounts to pay for integer numbers of men and plant. At a detailed level, therefore, the continuous linear relationship becomes a set of discrete points which may perhaps fall approximately on a straight line but only the points are achieveable; the gaps between them are not. In chapter 4 it was seen that this detailed discreteness destroyed the usefulness of LP in detailed resource allocation. Dynamic programming, however, requires discrete values in order to find a solution; it thrives on discreteness and does not force variables to fit any algebraic form. Consequently, DP is becoming recognised as a very powerful method for engineering planning and management. Its biggest disadvantage is that it requires a serial form such as figure 5.5. It is not always possible to cast practical problems in this form, but whenever there is even a hint that seriality may exist in a practical problem it should be seized upon and exploited by dynamic programming. The rest of this chapter is devoted to examples of civil engineering problems which have been cast into a serial form and solved by DP.

www.EngineeringBooksPDF.com

EXAMPLE 5.3 – ALLOCATING A TOWER CRANE

A contractor has work on 4 separate sites A, B, C and D and is considering hiring a large tower crane for 6 months to help in the construction work. Because of the limited mobility of the crane it can only be moved from site to site at the end of each month. The crane is initially at the hiring yard, Y, and must be returned there after 6 months. The costs of moving the crane between the yard and each of the sites and the costs of moves between sites have been estimated and are shown in table 5.2. The contractor has estimated the benefits to him in cash terms of using the crane on each of the 4 sites in each of the 6 months. These benefits are shown in Table 5.3.

The hirer has quoted a cost of £2000 for hire of the crane for 6 months, the contractor to pay all moving costs. How should the contractor allocate the crane among his sites to maximise his total returns over the 6 months?

Table 5.2 Costs (in £100 units) of moving the crane between locations

	Y	A	B	C	D
Y	0	2	1	3	4
A	2	0	4	2	1
B	1	4	0	6	2
C	3	2	6	0	4
D	4	1	2	4	0

Table 5.3 Monthly benefits from use of crane on each site (in £100 units)

		Month					
		1	2	3	4	5	6
Site	A	3	4	7	6	8	2
	B	9	8	3	2	0	0
	C	8	4	6	7	2	1
	D	7	8	8	2	4	6

This problem can be represented in serial form (figure 5.5) but with seven stages. Each stage consists of moving the crane from its present location to a new location and, except for the last stage when it must return to the yard, using it there for 1 month. The decision that must be taken in each stage consists of choosing which location to move the crane to (or of course, leaving it where it is). The state variable in this problem is not a numerical one, it is the location of the crane, A, B, C, D or Y. The initial state and final state must be Y, possible state values at other stages are A, B, C or D. The stage transition functions are also non-numerical. The input state to any stage is the present location of the

www.EngineeringBooksPDF.com

crane, the decision taken is where to move it to, and the output state is the new location. Thus, clearly $S_i = t_i(S_{i-1}, d_i)$. The return function at each stage $r_i(S_{i-1}, d_i)$ consists of the benefit from use of the crane in its new location for that month minus the cost of moving it there from its previous location. Thus the DP solution proceeds as below.

Stage One

S_0 = Y; S_1 = A, B, C, D

$$C'_{S_1=A} = 3 - 2 = 1$$
$$C'_{S_1=B} = 9 - 1 = (8)$$
$$C'_{S_1=C} = 8 - 3 = 5$$
$$C'_{S_1=D} = 7 - 4 = 3$$

Stage Two

S_1 = A, B, C, D; S_2 = A, B, C, D

$$C'_{S_2=A} = \max \begin{Bmatrix} 1+4-0 \\ 8+4-4 \\ 5+4-2 \\ 3+4-1 \end{Bmatrix} = \max \begin{Bmatrix} 5 \\ 8 \\ 7 \\ 6 \end{Bmatrix} = 8$$

$$C'_{S_2=B} = \max \begin{Bmatrix} 1+8-4 \\ (8+8-0) \\ 5+8-6 \\ 3+8-2 \end{Bmatrix} = \max \begin{Bmatrix} 5 \\ (16) \\ 7 \\ 9 \end{Bmatrix} = (16)$$

$$C'_{S_2=C} = \max \begin{Bmatrix} 1+4-2 \\ 8+4-6 \\ 5+4-0 \\ 3+4-4 \end{Bmatrix} = \max \begin{Bmatrix} 3 \\ 6 \\ 9 \\ 3 \end{Bmatrix} = 9$$

$$C'_{S_2=D} = \max \begin{Bmatrix} 1+8-1 \\ (8+8-2) \\ 5+8-4 \\ 3+8-0 \end{Bmatrix} = \max \begin{Bmatrix} 8 \\ (14) \\ 9 \\ 11 \end{Bmatrix} = (14)$$

Stage Three

S_2 = A, B, C, D; S_3 = A, B, C, D

$$C'_{S_3=A} = \max \begin{Bmatrix} 8+7-0 \\ 16+7-4 \\ 9+7-2 \\ 14+7-1 \end{Bmatrix} = \max \begin{Bmatrix} 15 \\ 19 \\ 14 \\ 20 \end{Bmatrix} = 20$$

www.EngineeringBooksPDF.com

$$C'_{S_3=B} = \max \begin{Bmatrix} 8+3-4 \\ 16+3-0 \\ 9+3-6 \\ 14+3-2 \end{Bmatrix} = \max \begin{Bmatrix} 7 \\ 19 \\ 6 \\ 15 \end{Bmatrix} = 19$$

$$C'_{S_3=C} = \max \begin{Bmatrix} 8+6-2 \\ 16+6-6 \\ 9+6-0 \\ 14+6-4 \end{Bmatrix} = \max \begin{Bmatrix} 12 \\ 16 \\ 15 \\ 16 \end{Bmatrix} = 16$$

$$C'_{S_3=D} = \max \begin{Bmatrix} 8+8-1 \\ (16+8-2) \\ 9+8-4 \\ (14+8-0) \end{Bmatrix} = \max \begin{Bmatrix} 15 \\ (22) \\ 13 \\ (22) \end{Bmatrix} = (22)$$

Stage Four

S_3 = A, B, C, D; S_4 = A, B, C, D

$$C'_{S_4=A} = \max \begin{Bmatrix} 20+6-0 \\ 19+6-4 \\ 16+6-2 \\ (22+6-1) \end{Bmatrix} = \max \begin{Bmatrix} 26 \\ 21 \\ 20 \\ (27) \end{Bmatrix} = (27)$$

$$C'_{S_4=B} = \max \begin{Bmatrix} 20+2-4 \\ 19+2-0 \\ 16+2-6 \\ 22+2-2 \end{Bmatrix} = \max \begin{Bmatrix} 18 \\ 21 \\ 12 \\ 22 \end{Bmatrix} = 22$$

$$C'_{S_4=C} = \max \begin{Bmatrix} 20+7-2 \\ 19+7-6 \\ 16+7-0 \\ 22+7-4 \end{Bmatrix} = \max \begin{Bmatrix} 25 \\ 20 \\ 23 \\ 25 \end{Bmatrix} = 25$$

$$C'_{S_4=D} = \max \begin{Bmatrix} 20+2-1 \\ 19+2-2 \\ 16+2-4 \\ 22+2-0 \end{Bmatrix} = \max \begin{Bmatrix} 21 \\ 19 \\ 14 \\ 24 \end{Bmatrix} = 24$$

Stage Five

S_4 = A, B, C, D; S_5 = A, B, C, D

$$C'_{S_5=A} = \max \begin{Bmatrix} (27+8-0) \\ 22+8-4 \\ 25+8-2 \\ 24+8-1 \end{Bmatrix} = \max \begin{Bmatrix} (35) \\ 26 \\ 31 \\ 31 \end{Bmatrix} = (35)$$

www.EngineeringBooksPDF.com

$$C'_{S_5=B} = \max \begin{Bmatrix} 27+0-4 \\ 22+0-0 \\ 25+0-6 \\ 24+0-2 \end{Bmatrix} = \max \begin{Bmatrix} 23 \\ 22 \\ 19 \\ 22 \end{Bmatrix} = 23$$

$$C'_{S_5=C} = \max \begin{Bmatrix} 27+2-2 \\ 22+2-6 \\ 25+2-0 \\ 24+2-4 \end{Bmatrix} = \max \begin{Bmatrix} 27 \\ 18 \\ 27 \\ 22 \end{Bmatrix} = 27$$

$$C'_{S_5=D} = \max \begin{Bmatrix} 27+4-1 \\ 22+4-2 \\ 25+4-4 \\ 24+4-0 \end{Bmatrix} = \max \begin{Bmatrix} 30 \\ 24 \\ 25 \\ 28 \end{Bmatrix} = 30$$

Stage Six

S_5 = A, B, C, D; S_6 = A, B, C, D

$$C'_{S_6=A} = \max \begin{Bmatrix} 35+2-0 \\ 23+2-4 \\ 27+2-2 \\ 30+2-1 \end{Bmatrix} = \max \begin{Bmatrix} 37 \\ 21 \\ 27 \\ 31 \end{Bmatrix} = 37$$

$$C'_{S_6=B} = \max \begin{Bmatrix} 35+0-4 \\ 23+0-0 \\ 27+0-6 \\ 30+0-2 \end{Bmatrix} = \max \begin{Bmatrix} 31 \\ 23 \\ 21 \\ 28 \end{Bmatrix} = 31$$

$$C'_{S_6=C} = \max \begin{Bmatrix} 35+1-2 \\ 23+1-6 \\ 27+1-0 \\ 30+1-4 \end{Bmatrix} = \max \begin{Bmatrix} 34 \\ 18 \\ 28 \\ 27 \end{Bmatrix} = 34$$

$$C'_{S_6=D} = \max \begin{Bmatrix} \textcircled{35+6-1} \\ 23+6-2 \\ 27+6-4 \\ 30+6-0 \end{Bmatrix} = \max \begin{Bmatrix} \textcircled{40} \\ 27 \\ 29 \\ 36 \end{Bmatrix} = \textcircled{40}$$

Stage Seven

S_6 = A, B, C, D; S_7 = Y

$$C'_{S_7=Y} = \max \begin{Bmatrix} 37-2 \\ 31-1 \\ 34-3 \\ \textcircled{40-4} \end{Bmatrix} = \max \begin{Bmatrix} 35 \\ 30 \\ 31 \\ \textcircled{36} \end{Bmatrix} = \textcircled{36}$$

www.EngineeringBooksPDF.com

Maximum total return = £3600. Less £2000 hire cost = £1600 total profit. Policy traceback: $d_7 = D \rightarrow Y$; $d_6 = A \rightarrow D$; $d_5 = A \rightarrow A$; $d_4 = D \rightarrow A$; $d_3 = D \rightarrow D$; $d_2 = B \rightarrow D$; $d_1 = Y \rightarrow B$. Thus optimal allocation of crane is

1st month – site B
2nd month – site D
3rd month – site D
4th month – site A
5th month – site A
6th month – site D

An alternative policy shown in the traceback by dotted circles has $d_3 = B \rightarrow D$; $d_2 = B \rightarrow B$, that is, the crane spends month 2 at site B. The profit from both policies is the same.

5.3.1 Comments on Example 5.3

This example demonstrates that dynamic programming can be used to solve serial systems in which the decision and state variables do not have numerical values. None of the functions involved in this solution has an algebraic form. This shows the great versatility of the DP solution philosophy for serial systems.

This example and the first one share one common feature. In both problems an initial state and a final state, S_0 and S_N, were specified and known. Many serial systems are such that either or both of the initial and final states is not known. This presents no fundamental difficulties for the DP solution method as the next example shows.

EXAMPLE 5.4 – A PURIFICATION PROCESS

A large manufacturing company wishes to discharge its industrial waste water into a river. The waste water contains a very high level of a toxic substance, the precise level of this impurity varying widely from day to day depending upon industrial activities. The water authority has insisted that if the waste water is to be discharged into the river it must first be purified and has specified a maximum permissible level for the toxic substance in the discharge to the river.

The company has built a three-stage purification plant to detoxify the waste water. It is able to measure the impurity level in the waste as it enters the first stage, I_0, and it wishes to operate the three-stage plant as cheaply as possible, so that a final, known impurity level, I_3, is achieved. Each of the three stages is different and in each it is possible to vary the levels of impurities removed. The cost of operating each stage depends on the impurity level in the water entering that stage and on the purification achieved in that stage. How can an operating system for the purification plant be devised?

Let d_1, d_2 and d_3 be the amounts of impurity removed by the first, second and third stages of the process respectively. The ideal way of operating the system

www.EngineeringBooksPDF.com

would be to measure the initial impurity level, I_0, and, depending upon that value, to select values for d_1, d_2 and d_3 such that the known value of I_3 is achieved and the sum of the stage costs is minimised. This is clearly a serial problem which can be represented by figure 5.6. I_0 is a high initial impurity level.

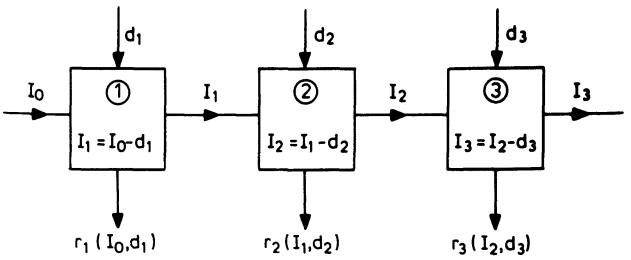


Figure 5.6 Serial representation of the purification process

The value of this state variable I decreases through the stages until a known final effluent impurity level, I_3, is achieved. The stage transition functions t_1, t_2 and t_3 must therefore be of the form

$$I_i = I_{i-1} - d_i \tag{5.5}$$

for the ith stage. In order to evaluate an optimal operation policy the stage costs must be calculated. Since each stage represents a different purification process the cost functions will probably have different forms for different stages. Thus, the general form

$$r_i(I_{i-1}, d_i) \tag{5.6}$$

will have different functional relationships for different i. These relationships may already be known from the specifications of the stage processes themselves. If they are not known they may be evaluated in tabular form. Each stage is operated separately and the operational cost is measured for different combinations of input impurity level and purification process. In this way tables of operating costs can be drawn up for each stage.

The solution of this problem by dynamic programming is complicated by the fact that although a value is known for the final state, I_3, the initial state, I_0, may have many different values. It would be quite possible to specify a set of discrete possible values of I_0 selected to cover the range and for each of these to solve a complete DP problem with known initial and final values. If this is done then for each problem the initial value I_0 is known and the cumulative calculations of the DP method are done for a range of possible discrete values of the output state, I, from each stage. The traceback routine locates an optimal operating policy for the three stages for that particular value of I_0. This solution is then repeated for different values of I_0 thus building up a variety of optimal operation policies for different initial impurity levels.

www.EngineeringBooksPDF.com

To solve the problem many times like this is inefficient and a much simpler method can be used. Instead of solving the problem in a forwards direction starting with stage one and ending with stage three, the whole process may be inverted for solution purposes. If this is done I_3 becomes a known initial value of the state variable. I_3 is input to stage three and I_2 becomes the output from stage three and the input to stage two. I_1 is the output from stage two and the input to stage one. The output from stage one is I_0, the original initial impurity level, which may have many different values. Thus the direction of the solution process through the stages is reversed. In order to reverse the direction of solution in this way the stage transitions and costs must be rewritten. Conventionally the transitions and returns are expressed as functions of the input state and decisions. Equation 5.5 gives the transition at the ith stage in a forward solution method. For a backwards solution it is transposed to become

$$I_{i-1} = I_i + d_i \tag{5.7}$$

The ith stage return function 5.6 must also be modified to the general form

$$r_i(I_i, d_i) \tag{5.8}$$

This means an algebraic manipulation if the return functions are algebraic, or, if they are tabular, they must be retabulated to give stage costs for different combinations of original output impurity level and purification decision.

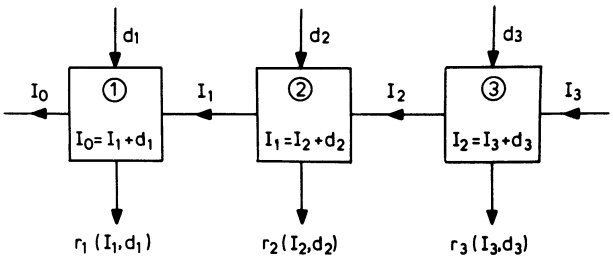


Figure 5.7 Reversed form of the serial system of figure 5.6

Having rewritten the transitions and returns figure 5.7 shows the reversed equivalent of figure 5.6 ready for solution. The solution proceeds normally. The value of I_3 is known. A range of possible values is specified for I_2 and the cost of achieving each of them from I_3 is evaluated. A range of possible values is selected for I_1 and the different ways of achieving I_1 from the range of values of I_2 are examined. The least cumulative cost route is selected for each discrete value of I_1. The process continues with stage one, specifying a range of possible values of I_0, and for each value a least cumulative cost is calculated. The traceback routine is now carried out *from each discrete value of* I_0 to I_3. The different policies resulting from these tracebacks each correspond to the least cost policy for

www.EngineeringBooksPDF.com

achieving the known I_3 from the particular value of I_0, that is, the same policies which would have resulted from solving the original problem in a forwards direction many times.

For this purification problem with an unknown initial state and a known final state, reversal of the direction of solution results in a serial problem which is solved once only with multiple tracebacks. Solving in a forwards direction would have required the problem to be solved many times with a traceback each time. The reversed problem is therefore much less expensive in time and effort to solve.

5.4.1 Comments on Example 5.4

The example demonstrates the advantages which can sometimes accrue by reversing the direction in which a serial problem is solved. It is inherent in the nature of serial problems of the general form of figure 5.5 that they can all be solved either forwards or backwards. Exactly the same optimal policies will result whichever direction is chosen. Naturally, the choice of direction depends upon the specific problem. If a serial problem has a single-valued initial or final state it is usually more efficient if the problem is solved in such a direction as to make the single value become an initial state.

In this chapter the dynamic programming solution method has been introduced, derived and explained in a forwards sense. This forward solution approach evolved naturally from an examination of network analogies in example 5.1. It is also natural for very many engineering problems. Most of the specialist books devoted solely to DP, however, such as those listed in the bibliography, treat backwards solutions as standard. The reason for this is that some of the mathematical proofs for specialist topics in DP turn out to be simpler in a backwards notation. The principle remains true, however, that serial problems can be solved either forwards or backwards and will yield the same optimal solutions.

Returning to the purification example itself the reversed solution identifies a set of optimal operating policies for each of the stages, each set corresponding to a particular discrete value of the initial impurity level I_0. In a practical solution it may be necessary to consider a large number of discrete values of I_0 at close spacings in order to model the complete range of variations. In the statement of the problem it was assumed that the initial impurity level I_0 varied widely 'from day to day'. This implies that I_0 may have a wide range of possible values but will keep any particular value sufficiently long to warrant a discrete change in the operation policy. For many processes this assumption is not valid. Many processes, in chemical engineering particularly, are similar in over-all concept to the purification process described above and in many of them the value of the initial state I_0 varies continuously with time. Such processes require that the operational policies of the separate stages must also be continuously controlled. Special methods are needed to determine control policies for time-varying pro-

www.EngineeringBooksPDF.com

cesses and they are beyond the scope of this book. This topic comes under the general heading of *optimal control*. In civil engineering very few problems that require continuous, time-varying optimal control arise so it will not be considered here.

The final example of this chapter is chosen from the design phase of a project. It is a problem in which neither the initial nor final state value is known.

EXAMPLE 5.5 – DRAINAGE DESIGN

This example, like the last one, is non-numerical. Its purpose is to show how dynamic programming can be used to improve a traditional manual design method and to produce least cost designs.

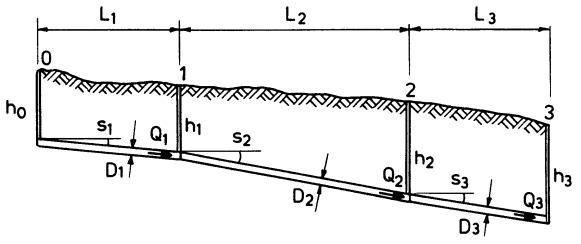


Figure 5.8 Typical drainage run (three pipes)

Figure 5.8 shows a cross-section of a run of underground sewerage. Foul waste or storm water (or both) enters the pipes at many points along individual lengths. The pipes slope downwards towards an outfall and carry away the waste water by unsurcharged gravity flow. Manholes are placed along the pipe run for inspection and maintenance purposes. In this example it is assumed that manhole positions and hence the plan lengths of all pipes are known. This is often the case in urban areas in which the arrangement of streets and buildings predetermines where manholes may be placed. Changes in the slopes of the pipes or in their diameters are only permitted at manholes. To protect the pipes from damage by surface loads the top of any pipe must not be closer to the surface than a prescribed minimum cover distance.

The designer's task in designing a pipe run such as that shown in figure 5.8 is to select a pipe diameter, slope and depth for each pipe between consecutive manholes. If the pipe run is a long one with many manholes there will be very many design decisions to be made and an injudicious choice for any one of them could result in a design which is needlessly expensive to construct. Instead of trying to design the pipe system as a complete entity designers usually split it

www.EngineeringBooksPDF.com

up into separate pipes each of which is then designed separately. Thus the complete design is built up as a sum of many separate designs.

The way this works is as follows. The designer starts at the extreme upstream end of the pipe run and designs the length of pipe between the first two manholes. He estimates the likely maximum flow Q_1 which this length of pipe will be required to carry. He then selects a minimum slope and depth for the pipe so that it lies as close to the surface as is permissible. Having selected a slope and depth he then designs the pipe size. The pipe diameter is found by trying several possible sizes starting with the smallest and increasing the diameter until one is found which, when flowing full, will pass the design flow Q_1 at a velocity somewhere between known minimum and maximum permissible values. This completes the design of the first pipe. The second pipe is considered next. This second pipe is placed as close to the ground surface as is practicable with its upstream end aligned with the downstream end of the first pipe. The maximum likely flow Q_2 is estimated and the pipe diameter is selected as before, that is, the smallest diameter which will pass Q_2 at an acceptable velocity. The third pipe length and all subsequent lengths are designed in this fashion until the complete design is achieved.

The implied logic of this method is that by always placing pipes as close to the surface as possible, trench depths for laying the pipes, and hence earthworks costs, will be as small as possible. Also manhole depths, and hence manhole costs, will be as small as possible. By always choosing the smallest feasible pipe size, pipe costs should also be minimised. Hence logic implies that, since each individual pipe element is optimally designed for least construction cost, the whole pipe system must also have been optimally designed.

Unfortunately the logic is false. This design method almost always results in a design which is non-optimal. It is usually possible to find a different design which has a cheaper cost of construction. Why is this so?

The logic outlined above is based on the assumption that the minimum value of a sum of several terms is equal to the sum of the minimum values of the terms. Mathematically, the assumption is that

$$\min\left\{\sum_{i=1}^{N} t_i\right\} = \sum_{i=1}^{N} \min \{t_i\}$$

In mathematical terms this equation is only true if each of the t_i, $i = 1, \ldots, N$ is completely independent. For the logic of the drainage design method described above to be valid the design of each element must be completely independent of all other elements. This is not so because the upstream end of each pipe is dependent for its position upon the downstream end of the preceding pipe. Thus the design of each pipe length depends upon the design of the preceding pipe. This design process is one in which

www.EngineeringBooksPDF.com

$$\min \left\{ \sum_{i=1}^{N} t_i \right\} \neq \sum_{i=1}^{N} \min \{t_i\}$$

It is also questionable whether this design method produces a least cost design even for an individual element. The method first chooses the depth and slope of the pipe and then chooses the smallest pipe size which is feasible *for that depth and slope*. Circumstances can occur such that if the slope of the pipe is slightly increased (thus incurring slightly increased earthworks costs) a smaller pipe diameter becomes feasible (resulting in reduced pipe costs). The sum of these two costs may favour the increased slope design as being optimal rather than the original standard design.

It is widely believed by practising drainage designers that this long-established, traditional design method produces the cheapest possible design for the whole system. This somewhat lengthy examination of that method was necessary in order to show that this belief is unfounded. It also helps to explain how DP can be used to produce truly optimal drainage designs of least cost of construction.

5.5.1 Dynamic Programming and Drainage Design

The traditional design method described above for underground drainage is a sequential process. The complete system design is built up element by element starting at the extreme upstream end and finishing at the outfall. Viewed in the perspective of the previous examples of this chapter this serial process appears to be ideal for the application of dynamic programming. The manual design method fails to produce the cheapest design for the complete system because it designs each element once only so that that element is as cheap as possible. This does not permit the possibility that an element may be designed to cost slightly more than is strictly necessary if this design results in even cheaper designs for other elements further down the pipe run. Dynamic programming allows this possibility by considering not one but several possible designs for each element and combining the sequential element designs together in such a way that cumulative total costs are minimised.

Figure 5.9a shows the pipe run of figure 5.8 split up into a sequence of elements. An element forms a stage in the DP process. Each element consists of an upstream manhole, a length of pipe between consecutive manhole positions, and the enclosed volume of earth which must be removed and refilled to lay the pipe. If each element in figure 5.9a forms a DP stage then the state variable which links the stages together must be the depth below the known ground level of the ends of each pipe length, h. The fact that h links the elements together thus destroying their complete independence has already been noted above. h_0 is the initial state, h_1 is the output state from stage 1 and is also the input state to stage 2, etc. The DP solution process requires that a range of possible discrete values is specified for the state variable at each stage. In this problem

www.EngineeringBooksPDF.com

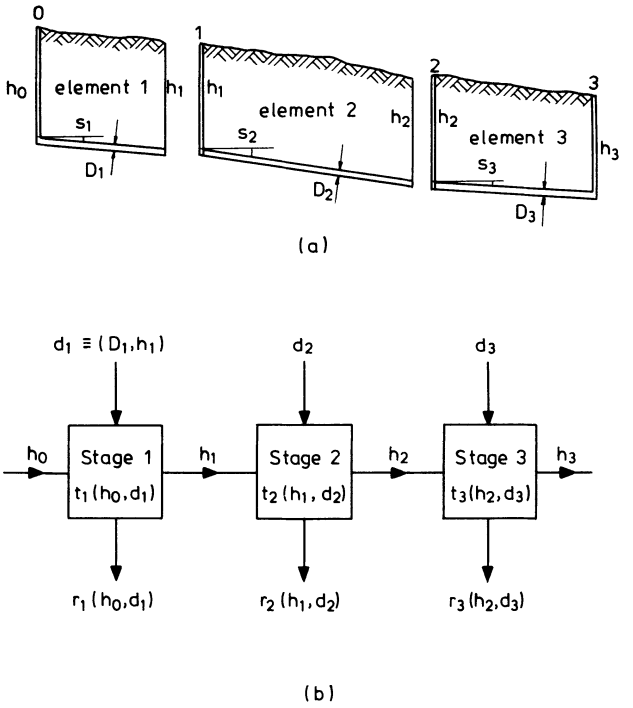


Figure 5.9 Serial representation of drainage design

the initial state h_0 is not known in value, nor is the value of the final state h_N, so, for each of these a range of values must be prescribed. The decision variables in each stage are the pipe diameter and slope but since the slope is defined by the depths of the two ends of the pipe and the known ground levels it is convenient to use pipe diameter and downstream end depth as decision variables, that is, D_i and h_i for stage i. The stage return is actually the cost of construction of the element and over the whole N-stage system the sum of the element costs must be minimised. Figure 5.9b shows the design problem in serial form ready for solution by DP.

The solution starts with stage 1 for which possible discrete values of h_0 and h_1 are postulated. For a particular output state h_1 pipes connecting back to each discrete depth h_0 are examined. Any of these connections which violate the minimum cover requirement or have too small a downward slope are eliminated. The design flow Q_1 is estimated for each remaining connection and then the smallest pipe diameter D_1 is selected for each possible connection which passes Q_1 at an acceptable velocity. Each of these designs is then costed and the smallest cost is associated with that particular value of h_1. This element design and costing process is repeated for each discrete value of h_1.

www.EngineeringBooksPDF.com

Moving to stage 2 a set of possible output states h_2 is defined. For each value of h_2 possible connections back to every value of h_1 are examined and infeasible or impracticable connections eliminated. Smallest technologically feasible diameters D_2 are selected for the remaining connections and each stage 2 design is costed. Cumulative costs of pipe systems ending at each h_2 value are found by adding the stage 2 costs to the costs associated with the appropriate h_1 value, and for each h_2 the least cumulative cost is selected. The remaining stages are dealt with in a similar fashion until the Nth stage is reached. After the Nth stage there will be a number of possible final state values, h_N, and a cumulative least cost associated with each. To each of these costs is added the cost of the final outfall manhole of a depth equal to that particular h_N. Of these final cumulative costs the least is selected. This gives the cheapest cost of the entire system and the optimal value of the final state, h_N. The traceback process then isolates values for all the other optimal depths and pipe diameters. An optimal value of h_0 is also found by the traceback. Thus the complete least cost design for the pipe run is found.

5.5.2 Comments on Example 5.5

The example shows how DP can solve serial problems for which neither an initial nor a final state value is known. A range of possible values is chosen for each and the solution process automatically finds optimal values. This problem is fundamentally different from the previous example, the purification process, in which an optimal solution (policy) was required for *each of the very many possible* initial state values. Here only the solution corresponding to the *optimal values* of multi-valued initial and final states was needed.

In the non-optimal manual design method only one design is made for each element. In the optimal DP solution each element requires many different designs to be made and costed. It therefore requires computational assistance and is not suitable for manual design. DP does not lend itself to the concept of a general computer package suitable for solving any DP problem; the problems themselves are too varied to fit into a precise computer program for general use. Drainage, however, is a ubiquitous feature of civil engineering projects of all types and special purpose DP computer programs specifically for least cost drainage design are now available. This example has demonstrated the fundamental principles of the solution method used in these programs but has been slightly simplified. There are several aspects of the programs which have been glossed over here. Such aspects include the different methods of estimating maximum design flows, the way in which element designs are tested for feasibility, the way they are costed, etc. All these factors have been studied and successfully incorporated into fully practical design programs. The interested reader should consult the technical literature (for example papers by Templeman and Walters) for further details.

The design of many complex civil engineering projects is very often ac-

www.EngineeringBooksPDF.com

complished by splitting up the complete project into many different elements which are designed separately and sequentially. Drainage design is a good example of this and there are many others. Whenever a sequence exists in a design process there is always a possibility that a true serial system can be found suitable for exploitation and optimum design using DP. A tall multi-storey building can be viewed as a sequence of storeys each representing a stage in a serial design process; a varying depth beam can be designed as a sequence of discrete sections. The best alignment for a road can be found by considering it as a sequence of short road elements linked together and each with many possible alignments. In all these design problems the existence of a sequential process is the essential element.

Example 5.5 considered a single pipe run. A drainage system usually consists of many such pipe runs connected together to form a network. This raises the question of whether and how DP can be used to design complete systems composed of several branches which diverge or converge although each branch is essentially a serial process. Example 5.1, the critical path problem, also raised this question and to solve that problem it was assumed that the critical path was in no way affected by any other activities or paths in the network. The next section deals with branched serial systems, answers these questions and also takes a broad look at the efficiency of dynamic programming in solving sequential problems.

5.6 FURTHER ASPECTS OF DYNAMIC PROGRAMMING

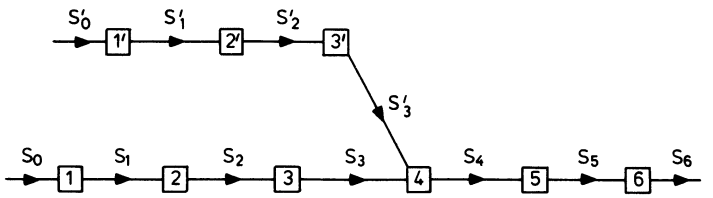
5.6.1 Branched Serial Systems

All the examples of this chapter have been concerned with a single sequence of stages. Some serial systems exhibit *branches*, that is, they are composed of several sequences of stages which converge or diverge. Figure 5.10 shows convergent and divergent branched serial systems. How does the existence of branches affect the DP solution?

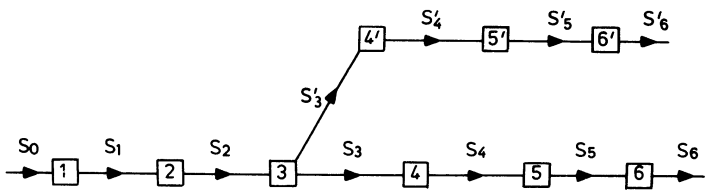
Example 5.4 showed that DP can be used as either a forwards or a backwards solution method and so, to be precise, there is no difference between convergent or divergent branches. Figure 5.10b can be obtained from figure 5.10a by reversing the direction of flow and renumbering the stages in a backwards sense.

Consider the convergent branched system of figure 5.10a. The figure shows a single sequence of six stages which is joined at stage 4 by another single sequence of three stages denoted by the $'$ superscripts. A forwards solution process may be used on the first three stages of the lower sequence resulting in cumulative total returns at stage 3 for each of several discrete values of the output state S_3. The upper sequence is now solved separately using a forwards solution process on the first three stages resulting in a set of cumulative total returns for the upper sequence associated with several discrete values of output state S'_3. At stage 4 the two branches merge. Thus stage 4 has two input states, S_3 and S'_3 but one output

www.EngineeringBooksPDF.com



a) A convergent branched serial system



b) A divergent branched serial system

Figure 5.10 Branched serial systems

state S_4. There are therefore two transition functions, one for each branch but both involving the same stage 4 decision variable d_4, that is, $t_4(S_3, d_4)$ and $t'_4(S'_3, d_4)$. These transition functions describe how the two incoming state variables are merged into the output state variable by the stage 4 decisions. There are also two return functions at stage 4, one for each branch, $r_4(S_3, d_4)$ and $r'_4(S'_3, d_4)$. The dynamic programming recurrence relationship, equation 5.4, at stage 4 has the form

$$C'_{S_4} = \min_{d_4} \text{(or max)} \{C'_{S_3} + C'_{S'_3} + r_4(S_3, d_4) + r'_4(S'_3, d_4)\} \qquad (5.9)$$

The relationship states that the decision variable d_4 must be selected so as to minimise (or maximise if the problem requires maximisation) the sum of the cumulative total returns at the previous stage on both branches and the two returns at stage 4. The recurrence relationship 5.9 for stage 4 enables a cumulative total return to be assigned to each possible value of output state S_4. Stages 5 and 6 are now solved normally in a forwards direction. After stage 6 an optimal total return for the complete branched system can be found and an optimal policy is isolated by the traceback routine. When the traceback reaches stage 4 it branches as a result of the two transition and return functions there and each

www.EngineeringBooksPDF.com

branch is traced back separately. The complete solution can, therefore, be found.

The essence of handling convergent branches is, therefore, that each branch is solved separately until the stage at which branches converge. At that stage separate transition and return functions for each converging branch describe how the output state and cumulative returns are formed. After the merging stage the solution proceeds normally.

Diverging serial systems such as figure 5.10b are most easily solved as convergent systems after reversal of the flow direction. It is much more complicated to solve the system of figure 5.10b in the direction shown. The difficulties arise at stage 3 because the decision variable there controls two output states S_3 and S_3'. Thus possible values for S_3 and S_3' must be linked in pairs, each pair corresponding to a particular S_2 and d_3. Solutions of the two diverged branches are also linked by virtue of the linking of their initial states S_3 and S_3'. Not only must the state variable be split into two components at stage 3, the cumulative returns must also be split. This too raises difficulties. Hence, wherever possible, divergent branched systems should be reversed and solved as convergent systems which are much easier to handle. Specialist texts such as those listed in the bibliography should be consulted for a detailed treatment of branching in serial systems.

It is possible to have a serial system with a diverging branch which later converges onto the sequence. Clearly this sort of circumstance could arise in the expanded critical path problem, example 5.1. This presents new difficulties, the resolution of which is beyond the scope of this book. Nevertheless they can be resolved. As far as drainage design is concerned (example 5.5) it is common practice to permit only converging branches in a drainage system. On the rare occasions that diverging branches do occur they are usually of a very simple nature and soluble by reversal. Thus the general DP-based computer programs for drainage design mentioned earlier are fully able to design complicated multi-branched systems.

5.6.2 Efficiency of the DP Method

A comment sometimes heard about dynamic programming is that it seems to be a complicated type of enumeration method. Would not some simpler form of enumeration be just as efficient? In order to provide an answer to this question and to examine the efficiency of the DP method it is necessary to distinguish between explicit and implicit enumeration methods. An *explicit enumeration* method is one which identifies and evaluates all possible solutions before choosing the optimal one. An *implicit enumeration* method is one which identifies and evaluates only some of the possible solutions, the remainder not being evaluated because they are known not to contain the optimal solution. Both are what may be termed complete enumeration methods in the sense that all possible solutions are considered. In explicit enumeration all possible solutions are directly or explicitly evaluated. In implicit enumeration the evaluation of

www.EngineeringBooksPDF.com

some possible solutions is not done directly but is only implied. Dynamic programming is an implicit enumeration method and is consequently much more efficient than any explicit enumeration technique. The following example clarifies this statement.

Consider an N-stage serial system with one state variable which has ten possible values at each stage. Suppose that neither an initial nor a final value is known. S_0, the initial state, will therefore have ten values and S_1, the output from stage 1 will also have ten values. There are therefore 10^2 different policies for connecting S_0 to S_1. Similarly there are 10^2 ways of connecting the ten S_1 values to the ten S_2 values. Therefore to connect S_0 to S_2 must involve $10^2 \times 10^2$ possibilities, that is, 10^4 policies for achieving S_2 from S_0. To reach S_3 from S_2 needs another 10^2 possibilities so to reach S_3 from S_0 requires 10^6 policies. Continuing this sequence to the Nth stage it follows that there will be a total of 10^{2N} different possible policies for achieving S_N from S_0. Explicit enumeration must evaluate all of them before selecting the optimal policy. If the system is composed of ten stages explicit enumeration will involve examining every one of 10^{20} different policies: a very large number. To give an idea of how big this number is imagine a very fast computer which can evaluate one thousand possible policies per second. To enumerate explicitly all the possible policies of this ten stage system with ten values of the state variable would take the computer considerably in excess of 4×10^9 *years*!

In comparison, the dynamic programming solution evaluates the 10^2 possibilities at stage 1 and selects the ten best. It evaluates another 10^2 possible ways of linking S_1 to S_2 and selects the ten cumulatively best. At stage 3 another 10^2 policies are examined and so on until the Nth stage. The total number of policies evaluated in the DP method is therefore $N \times 10^2$. For a ten-stage system as above, this number is 10^3 policies which would take the fast computer around one *second*! If this time is doubled to allow time for the selection of the optimal cumulative values at each stage and for the traceback of the optimal policy, it still demonstrates conclusively that DP is far more efficient than explicit enumeration.

The reason for this high efficiency of DP lies in the selection of the ten cumulatively best policies at each stage. One hundred policies are evaluated and ninety discarded at each stage because, as a consequence of Bellman's principle of optimality, those ninety *cannot possibly contribute to a policy which is optimal for the serial system*. There is therefore no need to evaluate policies in later stages which contain any of the ninety rejected stage policies because they could not possibly be optimal. Their evaluation and rejection is implied and is not explicitly done. DP is consequently an implicit enumeration method. This means that the solution found by DP will always be the correct one and the same solution which would have been found by explicit enumeration of all possible policies. Dynamic programming is therefore an implicit enumeration method which is much more efficient than any explicit enumeration method. Indeed it is so efficient as a solution technique that any new problem should first of all be

www.EngineeringBooksPDF.com

inspected to see if any serial process is involved and, if so, the problem should preferentially be solved by DP.

Further advantages which DP has over other methods can also be restated here. DP is able to solve many different types of linear or non-linear, continuous or discrete problems and it makes no demands for a particular algebraic form of a problem. It can handle discontinuous functions, discrete-valued functions and tabular values with ease. Its only restriction is that the problem must have a sequential form as in figure 5.5. Although many practically important problems can be expressed in serial form, many cannot.

5.6.3 Multiple State Variables

All the examples of DP presented in this chapter have used a single state variable. It is often found that practical problems can only be expressed in a serial form suitable for DP solution if two or more state variables are used. Independent multiple state variables are permitted in the DP method which remains unaltered. If two state variables are used however, one with m discrete possible values at each stage and the other with n discrete values, then the process of evaluating cumulatively optimal returns for each discrete output state must be applied at each of the $m \times n$ discrete values, that is, at each of the possible combinations of m with n. If three state variables are used with ℓ, m and n discrete values, then all $\ell \times m \times n$ possible output states must be investigated. As the number of state variables increases, clearly the amount of calculation involved increases much faster. This imposes a practical limit to the number of state variables which can conveniently be used in a DP problem.

The efficiency example of section 5.6.2 had ten stages and a single state variable with ten discrete values. The fast computer evaluating 1000 policies per second would need 1 second to solve the problem. If two state variables are used instead of one and each has ten discrete values the computing time required becomes 100s. If three ten-valued states are used this time leaps to 10^4 s, that is, approaching 3 h. Four state variables would require nearly 12 days' continuous computation. Thus the number of state variables governs whether a problem can be solved by DP within practicable time limits. Problems with two state variables are solved fairly regularly; three state variable problems are much less frequently solved.

5.6.4 Continuous Dynamic Programming

Occasionally it is possible to solve DP problems continuously rather than as discrete-valued problems. Consider a problem in which the transition and return functions are algebraic in form. The transition function 5.3 may be rewritten to express S_{i-1} in terms of S_i and d_i, that is

$$S_i = t_i(S_{i-1}, d_i) \text{ becomes } S_{i-1} = t_i'(S_i, d_i) \tag{5.10}$$

www.EngineeringBooksPDF.com

If this relationship for S_{i-1} is now substituted into the DP recurrence relationship 5.4 that relationship becomes

$$C'_{S_i} = \min_{d_i} \{C'_{S_{i-1}} + r_i(t'_i[S_i, d_i], d_i)\} \tag{5.11}$$

For the first stage in the DP solution process C'_{S_0} is conventionally equal to zero and 5.11 becomes

$$C'_{S_1} = \min_{d_1} \{r_1(t'_1[S_1, d_1], d_1)\} \tag{5.12}$$

In discrete DP a set of possible values of S_1 would be selected and for each of these values d_1 would be chosen so as to minimise the first stage return r_1. The resulting minimum value C'_{S_1} would be associated with that specific value of S_1. This procedure is represented by equation 5.12. Sometimes, however, the function r_1 in equation 5.12 has a convenient algebraic form which permits the minimisation over d_1 to be performed analytically giving the optimal d_1^* as an algebraic function of S_1. Thus the values of d_1^* and C'_{S_1} can be found as continuous functions of the output state S_1, that is

$$d_1^* = d_1(S_1) \text{ and } C'_{S_1} = C'_{S_1}(S_1) \tag{5.13}$$

The second stage of the solution concerns the recurrence relationship 5.11 in the form

$$C'_{S_2} = \min_{d_2} \{C'_{S_1} + r_2(t'_2[S_2, d_2], d_2)\}$$

Substituting 5.13 for C'_{S_1}, this becomes

$$C'_{S_2} = \min_{d_2} \{C'_{S_1}(S_1) + r_2(t'_2[S_2, d_2], d_2)\} \tag{5.14}$$

The transposed transition function 5.10 for the second stage is

$$S_1 = t'_2(S_2, d_2)$$

which when substituted into 5.14 gives

$$C'_{S_2} = \min_{d_2} \{C'_{S_1}(t'_2[S_2, d_2]) + r_2(t'_2[S_2, d_2], d_2)\} \tag{5.15}$$

The cumulative returns to be minimised over d_2 in 5.15 are here represented by an algebraic function of S_2 and d_2 only. Thus the minimisation of this function with respect to d_2 can again be done analytically. This yields

$$d_2^* = d_2(S_2) \text{ and } C'_{S_2} = C'_{S_2}(S_2) \tag{5.16}$$

The third and subsequent stages are handled similarly, the objective being for the ith stage to set up a recurrence relationship involving only S_i and d_i. This function is then minimised with respect to d_i giving d_i^* and C'_{S_i} as functions of S_i. The entire sequential problem can thus be solved continuously and analytically.

For continuous DP it is necessary that the transitions and returns are all

www.EngineeringBooksPDF.com

easily manipulated functions. For practical engineering problems this is rarely the case and, except for some minor, very simple problems, there are very few practical uses for continuous DP. It is mentioned here without a numerical example to illustrate the way in which algebraic manipulations can be made to the transition and return functions and the recurrence relationship. The general lack of practical civil engineering applications of continuous dynamic programming renders further study of it unnecessary.

SUMMARY

This chapter has examined problems which exhibit serial properties. In such problems decisions are made in a sequential fashion rather than simultaneously. Although these sequential problems can theoretically be solved by complete enumeration the method known as dynamic programming is considerably more rapid and efficient. Several practical civil engineering problems have been used to establish the nature of a sequential problem and some network concepts from chapter 4 were used to develop the dynamic programming recurrence relationship and to highlight the over-all concept of the DP solution method.

Once the first example had established the DP method, some more examples of a practical nature were used to show its versatility and to demonstrate some of its additional features. In particular these showed the ability of DP to handle non-numerical variables and relationships (example 5.3), reversal of the direction of solution (example 5.4) and solution of problems in which neither an initial nor a final state value is known (example 5.5). Finally, a brief introduction to branched serial problems was given, followed by a demonstration of the very high solution efficiency of DP compared with complete enumeration.

The high efficiency of DP suggests that whenever a problem gives even a hint of seriality, an attempt should be made to isolate a sequential decision-making process within the problem and to solve it by dynamic programming.

BIBLIOGRAPHY

Bellman, R., *Dynamic Programming* (Princeton University Press, 1957)

Bellman, R., and Dreyfus, S., *Applied Dynamic Programming* (Princeton University Press, 1962)

Larson, R. E., and Casti, J. L., *Principles of Dynamic Programming. Part 1: Basic Analytic and Computational Methods* (Marcel Dekker, New York, 1978)

Nemhauser, G. L., *Introduction to Dynamic Programming* (Wiley, New York, 1966)

Templeman, A. B., and Walters, G. A., Optimal design of stormwater drainage networks for roads, *Proc., Instn Civ. Engrs,* Part 2, **67**, (1979) 573-87

www.EngineeringBooksPDF.com

Walters, G. A., and Templeman, A. B., Non-optimal dynamic programming algorithms in the design of minimum cost drainage systems, *Engng Optimization*, **4**, No 3 (1979) 81-90

EXERCISES

5.1 A contractor has agreed to complete a building project in 30 weeks with a penalty paid to the client of £13 000 per week for every week after week 30 that the project remains unfinished.

In preparing a network analysis for the project a critical path has been identified composed of four consecutive activities, A, B, C and D with durations of 8, 12, 7 and 7 weeks respectively. By employing more resources at extra cost each of the four activity durations may be reduced as indicated below.

Activity A: Costs of reductions of time by 1, 2, 3 or 4 weeks are £15 000, £22 000, £33 000 and £44 000, respectively.

Activity B: Cost, c, of achieving a reduction of r weeks is given by $c = £4000r\,(2 + r/2)$.

Activity C: Cost, c, of achieving a reduction of r weeks is given by $c = £(10\,000 + 2000r + 1500r^2)$.

Activity D: Cost is £20 000 plus £5000 per week reduction.

Assuming that the four activities remain critical and that reductions in duration must always be made in integer weeks, what policy should the contractor adopt in order to minimise his extra expenditure and how much will it cost him?

5.2 A bus company wishes to increase its profits from a bus route by adding buses over a three year period. Each year it may add 0, 1, 2 or 3 buses. The profit p_i to be expected in year i from adding x_i buses in that year is given by

$$p_i = (2i - 1)\left(2x_i + \frac{3i}{1 + x_i}\right)$$

This profit will continue to be made in each subsequent year. How many buses should be added to the route in each of the three years so as to maximise the total profits? If a total of only five buses may be added over the three years how shall this be done and what will the total profit be?

5.3 Ten men are available for labouring work on a construction site and they must be allocated among three simultaneously occurring jobs. The benefit accruing from each job is dependent upon the number of men allocated to that

www.EngineeringBooksPDF.com

job in the following way. If x_1, x_2 and x_3 are the numbers of men allocated to jobs 1, 2 and 3, respectively, the benefits, r_1, r_2 and r_3 are

$$r_1 = 2x_1 + \frac{x_1^2}{4} - \frac{x_1^3}{18}$$

$$r_2 = 3x_2 - \frac{x_2^2}{4}$$

$$r_3 = 2.0 + x_3 + \frac{1}{x_3} - \frac{9}{x_3^2}$$

Each job must have a minimum of two men allocated to it. Use a DP approach to determine how many men should be allocated to each job so that the total benefit from all three jobs is maximised.

If two men report sick how should the remaining eight be allocated so as to maximise total benefits?

5.4 Problem 3.18 in chapter 3 is an integer LP problem. Solve it by dynamic programming using two state variables, one for each constraint.

5.5 The production of precast concrete piles must be scheduled over five periods. Production during each period is restricted to complete piles and the maximum number which can be produced per period is 4. The table shows total production costs (in units of £100) for the different numbers of piles that may be produced in each period. In addition to these production costs there is a storage cost of £100 per pile per period stored (that is, 2 piles produced during period 3 must be stored for periods 4 and 5 at a total storage cost of £400).

Number Produced	Period				
	1	2	3	4	5
0	2	2	3	5	3
1	3	4	4	6	8
2	7	6	8	8	10
3	10	11	13	17	15
4	11	12	14	21	18

Find the optimal number of piles to be produced during each period so that the total costs are minimised if the total production requirement is (a) 18 piles; (b) 15 piles; (c) 13 piles; (d) 10 piles.

5.6 Figure 5.11 shows the network of activities which remain to be completed in a building project with activity durations given in weeks. The contracted completion date is at the end of week 29. To replan the activity durations so that the contract is completed by this time at least extra cost the contractor has

www.EngineeringBooksPDF.com

calculated the extra costs of reducing each activity duration by 1 or 2 weeks. The table shows these costs in units of £100.

Time Saved (weeks)	Activity								
	3 - 4	4 - 5	1 - 2	2 - 5	5 - 9	9 - 10	6 - 7	7 - 8	8 - 9
1	3	1	2	3	4	6	5	4	8
2	7	5	4	8	8	15	9	8	10

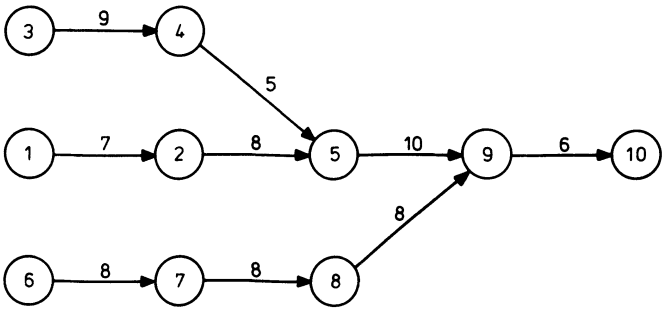


Figure 5.11

How should activity durations be altered to complete the project on time at least extra cost? Activities 3 - 4, 1 - 2 and 6 - 7 start simultaneously at the start of week 1.

www.EngineeringBooksPDF.com

6 SYSTEMATIC DESIGN AND NON-LINEAR PROBLEMS

This chapter and chapters 7 and 8 are concerned with non-linear systems. In civil engineering by far the largest source of non-linear decision-making problems is the design phase of a project. Since almost all design problems are non-linear, it is therefore logical that in this book the examples chosen to introduce and illustrate non-linear solution methods should be design examples. This chapter begins by showing that economical and efficient designs result from a formal, systematic approach to design and that mathematical optimisation affords a very convenient framework for systematic design.

Several simple design problems are then examined from an optimisation viewpoint and are shown to be composed of non-linear functional relationships. A graphical visualisation of a two-variable non-linear optimisation problem is examined and some fundamental characteristics of non-linear problems are established from this. A comparison with a graphical two-variable LP problem allows basic differences in possible solution philosophies to be found. The chapter ends by introducing ideas of convexity and local optima and leads naturally into the two following chapters which deal with specific solution methods for unconstrained and constrained non-linear problems.

6.1 SYSTEMATIC DESIGN

A major part of any civil engineering project is the design of the works themselves. The design phase of a project may be viewed as having two parts; firstly macro-design followed by micro-design. Macro-design consists of determining the numbers, layout, over-all proportions and sizes of all the major items which are to be built. When the macro-design is completed, micro-design consists of the very detailed design of each of the items within the limits established by the macro-design.

Engineering design is always a process of 'optimisation', to use that word in its loosest sense. Good designers strive to produce the best possible designs of which they are capable. In macro-design the major items must be sized and

www.EngineeringBooksPDF.com

arranged to complement each other and to achieve a functional harmony so that the project as a whole will fulfil its task efficiently and without unnecessary expense. Similarly in micro-design it is good practice to produce detailed designs for all the elements so that each one fulfils its purpose as cheaply as possible. An engineer who consistently prepares designs that are unnecessarily expensive to build does not find favour within the profession. Thus the guiding principle of design is that of always trying to produce the best possible design, that is, optimisation.

Design is fundamentally a process of *synthesis*. The designer must act in a creative capacity using his imagination, experience and technological skills to generate a complete, feasible, practical and efficient design where nothing existed previously. In contrast, the formal education of most civil engineers consists largely of *analysis*. Given an existing design, most graduate and many undergraduate students of civil engineering would feel able to analyse it and predict its behaviour in minute detail. This formal training in analysis, though essential to an understanding of how civil engineering works behave, does not train minds to think in a creative, design-oriented manner. Thus the process of synthesis of a design is often a very arduous and taxing one for a young engineer. The many codes of practice are of little value here. They too are essentially analytical in nature being useful in checking out an existing design and ensuring that no aspect of it violates currently accepted good practice. Codes of practice provide limits or boundaries against which an existing design may be compared but they are only rarely useful in the creation or synthesis process.

To synthesise a design an engineer relies very much upon his own and other engineers' experience and expertise in judging what constitutes a good design. The accumulation of this design experience is part of the long process of design education and it cannot be formally taught in a complete sense. There are ways, however, in which this learning process can be assisted and accelerated and this chapter deals with one of these ways. The process of analysis of a given design is very formalised and logical. A particular numerical model of the design is selected and is manipulated to yield the desired performance results. Cannot some similar mathematical model be used to synthesise a design rather than to analyse it? In chapter 1 the idea of a systematic approach to problem solving and decision making was introduced. In the subsequent chapters this systematic approach was applied to many problems arising in the construction, planning and operation phases of a project. In this chapter, it is applied to problems of design synthesis.

The first step in the systematic approach of chapter 1 is to determine what decisions must be made and to assign a mathematical variable to each decision. To synthesise a design the decisions which must be made and hence the mathematical variables which must be assigned are the physical quantities of the design; the numbers, sizes, dimensions and geometry of the components of the design. The second step in the systematic approach asks what restrictions are imposed upon the variables by the design problem and a mathematical model of

www.EngineeringBooksPDF.com

these relationships must be constructed. This second step is often a very difficult and complicated one in design problems. The restrictions can originate from a variety of sources, such as

(1) the mechanical behaviour of the design
(2) the properties of the materials used in the design
(3) limits imposed by codes of practice
(4) limits imposed by the boundaries of the design
(5) limits imposed by the availability of materials
(6) limits imposed by aesthetic considerations of the design
(7) limits imposed by the envisaged construction methods.

This list is by no means complete. To follow the systematic approach completely each restriction must be examined in detail and expressed mathematically as a function or functions of the design variables. In some cases this is not difficult. The mechanical laws governing the behaviour of the design are usually well-known as are the limiting values of the properties of the materials chosen for the design. Relevant codes of practice can be consulted to provide additional restrictions and the known boundary conditions for the design should also provide limits upon over-all dimensions, etc. Thus the first four items on the above list usually lend themselves well to mathematical modelling.

The last three items and other considerations such as the designer's own preferences, are not easily quantified. For example, the aesthetic aspects of a bridge or a block of offices cannot be expressed as a mathematical function of the design variables (sizes, dimensions, etc.). For many centuries architects and artists have studied the mathematical aspects of aesthetics without reaching any general quantifiable conclusions. Thus it must be concluded that in most design synthesis problems it is impossible to express in mathematical terms *all* the restrictions upon the feasibility of a design. The objective technological restrictions can be modelled mathematically but the subjective restrictions cannot. Since the list of mathematical restrictions forms the constraints on the feasibility of a design the absence of any constraints of a subjective or aesthetic origin means that any design which results from the synthesis process will be technologically feasible but may be subjectively unsatisfactory.

Once the mathematical forms of all the technological constraints is known, the third step of the systematic approach requires that some criterion representing the excellence of the design must be chosen and a mathematical objective function written to express this criterion as a function of all the design variables. Depending on the particular design problem a particular criterion will be appropriate and may be a technological one, for example, maximum stiffness or minimum depth, or an economic one, for example, minimum cost. Whatever criterion is adopted it must be expressed as a function of the design variables as completely as possible. As was noted above for the constraints, completeness cannot often be achieved in design problems. This applies to the objective function also, particularly if cost is used as the criterion. It is easy to formulate

www.EngineeringBooksPDF.com

the costs of the various materials used by the design; it is also easy to include the costs of essential extra materials which will be used in building the design such as formwork and shuttering for concrete, but it can be hard to formulate other cost elements. Labour, plant and all construction costs should ideally be included but rarely can be. If the particular design forms part of a much larger over-all design then it will have effects upon the design costs of other elements. Thus the cost objective function, like the constraints, can seldom be absolutely complete. The best that can usually be done is that a cost function is formulated which is representative in a general way of the likely cost of the design.

Having examined the design problem in this systematic way and having formulated a mathematical problem comprising an objective function and constraints, the problem must be solved to yield optimum values for all the variables. Chapters 7 and 8 describe solution methods. The resulting optimum values of the design variables then represent the synthesised design – an initial design to be analysed and modified in detail until it becomes a final design. One point must be stressed. This systematic synthesis approach is aimed only at synthesising an *initial design* which is technologically feasible and maximises or minimises some selected criterion. Only very rarely does it result in a final design. Since the problem constraints are incomplete technologically and subjectively and because the objective function is not usually complete, the optimal solutions are not complete either. The synthesised design may not look good and it may be awkward to build but it should be a technologically efficient starting point for a process of iterative analysis and modification which ends with a finalised design. When there is no background experience or design expertise this systematic approach pursued to its conclusion provides an excellent means of clearing the first design hurdle and obtaining a feasible and efficient initial design. The objective of systematic design and the use of optimisation methods in design is not in any way to *replace* design experience. The object is to *supplement* it. A good designer balances the technological aspects of a design against very many subjective, unquantifiable factors. The systematic approach cannot perform this balance; it can only make the best possible decisions upon the technological aspects. It is, therefore, to be expected that so-called optimum designs will require further analysis and perhaps considerable modification to strike the right balance between technological and subjective aspects. This final balance will always require the experience of a good designer.

This systematic approach to design has much to commend it even if it is not pursued to its final optimisation conclusion. It provides a framework for studying design problems The formal problem structure of an objective function and constraints guides the mind towards thinking about the design. What objective function should be used? Is the most important feature of a good design its cost or its weight or its stiffness or what other performance index is important? How is this objective function formed in terms of the design variables? Which ones have a major effect and which ones are of minor

www.EngineeringBooksPDF.com

importance? How is a feasible design constrained? Which major modes of failure must be prevented? What materials should be used and what properties are likely to be critical? What limits are imposed by the rest of the design?

The formalism of optimisation requires that very detailed thought be given to the objective function and to each constraint. It requires that all possible constraints from a variety of sources be examined in detail. This concentrates the attention of the designer on the design itself and forces him to think about how it may behave, how it may fail, whether it will fit together, etc. With some design problems there is no need to continue the systematic approach to its final stage. The very process of detailed systematic thought about the design and its constraints has itself generated a much deeper appreciation of the design problem which enables a good initial design to be made without further recourse to optimisation. Thus the systematic approach and the overall framework of optimisation provide a logical way of thinking about design and of synthesising an initial design. The following examples show how it is applied to some very simple design problems and also demonstrate the fundamental non-linearity of optimum design problems.

6.2 SIMPLE DESIGN EXAMPLES

6.2.1 Beam Design

Design a simply supported elastic beam of rectangular cross-section and known span, L, to carry a live uniformly distributed line load of w kN/m. The material of the beam weighs ρ kN/m^3.

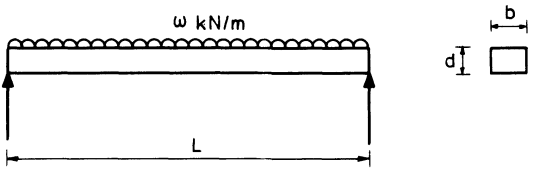


Figure 6.1 Simply supported beam

The beam span is known and its cross-section is known to be rectangular. Therefore the designer must select values for the breadth, b, and depth, d, of the beam to specify a design. b and d are therefore the design variables. An acceptable or feasible design for a beam such as this must satisfy several technological requirements. It must not deflect excessively and the maximum bending and shearing stresses induced in the beam must also not be excessive. Considering firstly the deflection of the beam, a maximum permissible deflection of the beam due to live load can usually be found from an appropriate code of practice. Typically, for steel beams the maximum permissible deflection of the beam, δ',

www.EngineeringBooksPDF.com

is related to the span, for example, $\delta' = L/360$. For this beam the maximum deflection, δ, will occur in the middle of the beam and is given by

$$\delta = \frac{5wL^4}{384EI}$$

in which E and I are the elastic modulus of the beam material and the second moment of area (moment of inertia) of the beam cross-section. For a rectangular cross-section

$$I = \frac{bd^3}{12}$$

thus the deflection requirement to be satisfied for this beam is

$$\left(\frac{5wL^4}{32E}\right) b^{-1} d^{-3} \leqslant \delta' \tag{6.1}$$

The items within brackets in inequality 6.1 are known constants and δ' is also known. Inequality 6.1 is non-linear in the design variables.

Maximum permissible bending stresses, σ', can usually be found in codes of practice and the design must be such that the maximum bending stresses in the beam caused by live load plus the beam's self-weight do not exceed σ'. The maximum bending moment on the beam is

$$M = (w + \rho bd)\, L^2/8$$

and the maximum bending stress constraint then becomes

$$(w + \rho bd) \frac{L^2}{8} \times \frac{\mathrm{d}}{2} \times \frac{12}{bd^3} \leqslant \sigma'$$

that is

$$\left(\frac{3wL^2}{4}\right) b^{-1} d^{-2} + \left(\frac{3\rho L^2}{4}\right) d^{-1} \leqslant \sigma' \tag{6.2}$$

This is another non-linear function of the design variables b and d.

The requirement that shearing stresses are acceptable may take a variety of forms. If the average shearing stress is not to exceed some code permissible value τ'_{av} this leads to the relationship

$$(w + \rho bd) \frac{L}{2bd} \leqslant \tau'_{av}$$

that is

$$\left(\frac{wL}{2}\right) b^{-1} d^{-1} + \left(\frac{\rho L}{2}\right) \leqslant \tau'_{av} \tag{6.3}$$

www.EngineeringBooksPDF.com

If the maximum value of the shearing stress is to be limited by some code value τ'_{max} this leads to

$$(w + \rho bd)\frac{L}{2} \times \frac{bd}{2} \times \frac{d}{4b} \times \frac{12}{bd^3} \leqslant \tau'_{max}$$

that is

$$\left(\frac{3wL}{4}\right)b^{-1}d^{-1} + \left(\frac{3\rho L}{4}\right) \leqslant \tau'_{max} \tag{6.4}$$

Both 6.3 and 6.4 are non-linear inequalities. In addition to the above technological constraints upon a feasible design there may also be limits placed upon the over-all dimensions or upon the relative proportions of the design. Thus one or more of the following constraints 6.5 may be required

$$\left.\begin{array}{lll} b'_{min} & \leqslant b & \leqslant b'_{max} \\ d'_{min} & \leqslant d & \leqslant d'_{max} \\ (bd^{-1})'_{min} & \leqslant bd^{-1} & \leqslant (bd^{-1})'_{max} \end{array}\right\} \tag{6.5}$$

In 6.5 as in all the previous relationships the $'$ superscript denotes a known constant.

A feasible design is represented by values of b and d which do not violate any of the constraint inequalities 6.1 to 6.5 inclusive. Many possible feasible designs exist and in order to choose the 'best' of these designs some criterion of efficiency or economy may be used. A very simple and often used criterion is that the cost of the beam must be as small as possible. For a beam material such as steel the cost of the structure is roughly proportional to the volume of steel used. Thus a suitable objective function for this problem might be the beam volume which is to be minimised, that is

$$C = Lbd \tag{6.6}$$

The entire design problem then consists of finding values for b and d which minimise equation 6.6 and do not violate any of the constraint inequalities 6.1 to 6.5. The functions involved in this problem have various non-linear combinations of the variables and so this design problem is classed as one of *non-linear programming* (NLP). A non-linear programming problem is an optimisation problem in which any one or more of the functions in either the objective or the constraints is non-linear.

This example of a very simple beam design demonstrates the fundamental non-linearity of most structural design problems. If the very simplest type of simply supported beam gives rise to a non-linear design problem it is to be expected that the more complicated beam systems which arise commonly throughout structural design will also be non-linear. Such is the case. To state that all structural design problems are non-linear is not a very large exaggeration. The non-linearity arises because the design itself usually consists of selecting

www.EngineeringBooksPDF.com

values for design variables which all have dimensions of length, that is, both b and d have a dimensionality of $(\text{length})^{+1}$. If the structural mechanics or behaviour of the design also only involved functions of dimensionality $(\text{length})^{+1}$ the design problem would probably not be non-linear. The constraint functions, however, involve things such as area which has dimension $(\text{length})^{+2}$, section modulus which has dimension $(\text{length})^{+3}$, moment of inertia dimension $(\text{length})^{+4}$. Non-linear expressions must therefore arise when the dimensionality of the constraint functions or the objective function is different from the dimensionality of the variables themselves. The next simple example also demonstrates this fundamental non-linearity of design problems.

6.2.2 Pipe Design

Design a length of circular sewer pipe of known slope θ' so that it will pass a full-bore unsurcharged gravity flow of not less than Q' at a velocity somewhere between known limits $V'_{\min}$ and $V'_{\max}$.

As the pipe length and slope are known, the design consists of selecting a value for the pipe diameter D. For a pipe of diameter D and slope θ' a reasonable approximation to the full-bore unsurcharged flow in the pipe, Q, is given by

$$Q = c_1 \theta'^{1/2} D^{8/3}$$

Thus if the pipe must pass a flow of not less than the known Q' the flow constraint becomes

$$(c_1 \theta'^{1/2})\, D^{8/3} \geqslant Q' \tag{6.7}$$

in which c_1 is a known constant. The velocity of this full-bore flow, V, is equal to the flow, Q, divided by the cross-sectional area of the pipe. Thus the velocity constraints can be written as

$$V'_{\min} \leqslant \left(\frac{4c_1\, \theta'^{1/2}}{\pi}\right) D^{2/3} \leqslant V'_{\max} \tag{6.8}$$

The cost of sewerage pipes is actually a non-linear function of the pipe diameter D but the non-linearity is not high and often a linear cost approximation similar to that approximation made in example 2.3 of chapter 2 can be used. Thus if the cost of the pipe is proportional to D the minimum cost design problem becomes that of minimising D subject to satisfaction of the constraints 6.7 and 6.8.

This simple design problem is again a non-linear one. It must necessarily be non-linear because the dimensionalities of the flow and velocity constraints are different and both constraints are different in dimensionality from the design variable. It is not possible to reformulate the problem in terms of a new design

www.EngineeringBooksPDF.com

variable so that the non-linearity disappears; it is a fundamental feature of the problem. All hydraulic design problems share this non-linearity which is unavoidable whenever flows, velocities, pressure heads, etc., are concerned in the constraints on the feasibility of a design.

6.2.3 Storage Tank Macro-design

A tank for the storage of bulk liquids must be designed. The tank must be a horizontal circular cylinder of diameter D and length L with hemispherical ends as shown in figure 6.2. The tank has a raft foundation. The cylindrical walls of the tank cost £C_1 per unit area to construct and the hemispherical ends £C_2 per unit area. The cost of the raft foundation is given by £C_3DL^2, that is, a cost per unit volume of the raft multiplied by the plan area, DL, of the raft multiplied by the raft thickness which is proportional to L. The tank must be able to hold V' m^3 of liquid. Find the best dimensions for the tank so that total construction costs will be minimised.

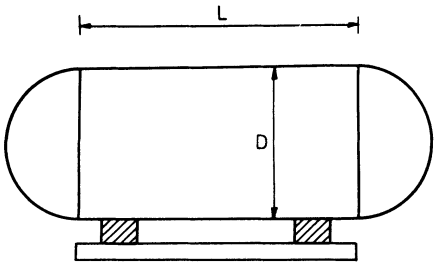


Figure 6.2 Cylindrical storage tank

This example is not a detailed technologically constrained design problem but is one of determining the most economical configuration for the tank. There is only one constraint, that the volume of the tank shall be V' m^3. This is an equality constraint which can be used to eliminate one of the variables.

The total cost of the tank will be

$$C = C_1\pi DL + C_2\pi D^2 + C_3DL^2 \tag{6.9}$$

This cost must be minimised over the design variables D and L. It is clearly a non-linear cost function. The volume of the tank is

$$\frac{\pi}{4}D^2L + \frac{\pi}{6}D^3$$

Thus the constraint becomes

$$\frac{\pi}{4}D^2L + \frac{\pi}{6}D^3 = V' \tag{6.10}$$

www.EngineeringBooksPDF.com

The optimum design problem is then to find values of D and L that minimise equation 6.9 and satisfy the constraint 6.10.

The variable L may be eliminated by rearranging 6.10 and substituting into 6.9. Thus from 6.10.

$$L = \frac{4V'}{\pi D^2} - \frac{2D}{3} \tag{6.11}$$

Substituting 6.11 into 6.9 gives

$$C = C_1 \pi D\left(\frac{4V'}{\pi D^2} - \frac{2D}{3}\right) + C_2 \pi D^2 + C_3 D\left(\frac{4V'}{\pi D^2} - \frac{2D}{3}\right)^2$$

and the design problem is then

$$\underset{D}{\text{Minimise}}\ C = C_1 \pi D\left(\frac{4V'}{\pi D^2} - \frac{2D}{3}\right) + C_2 \pi D^2 + C_3 D\left(\frac{4V'}{\pi D^2} - \frac{2D}{3}\right)^2 \tag{6.12}$$

Problem 6.12 is clearly non-linear in the variable D and there are no constraints. Although there are no technological or behavioural aspects to this problem to create non-linearity, it arises nevertheless because of the fundamental differences in dimensionality among the design variables (lengths), the costs (areas and volumes) and the single constraint (volume). The lack of any constraints in problem 6.12 highlights a basic difference between linear and non-linear programming problems. In LP problems constraints are essential otherwise no solution exists. In NLP problems constraints are not essential.

6.3 FEATURES OF NON-LINEAR PROGRAMMING PROBLEMS

The three simple design problems examined in section 6.2 show why almost all design problems are non-linear. It is now necessary to leave the practical examples and to begin to examine the mathematical nature of non-linear problems and methods of solving them.

The nature of linear programming problems was examined in section 2.4 of chapter 2. Problem 2.26 gives a mathematical statement of the necessary form of an LP problem and this form is very specific. In contrast to this, non-linear programming problems are very unspecific. The title non-linear programming is itself a negation rather than a positive assertion. It implies that non-linear programming includes everything not specifically classified as linear programming. It is therefore not particularly useful to try to write down an all-embracing mathematical statement of the necessary form of an NLP problem. Such problems have too wide a variety for this to have any value. To study NLP problems and their solution in depth it is first necessary to identify and classify problems into different types so that each type may be studied separately. In this book several

www.EngineeringBooksPDF.com

commonly occurring types of NLP problems will be examined but the scope of non-linear programming is so large that many more problems must of necessity be ignored.

Figure 2.8 of chapter 2 shows graphically a two-variable linear programming problem. A graphical representation of a two-variable non-linear programming problem is shown in figure 6.3. The constraint boundaries are curved lines instead of straight ones and the objective function is an irregularly curved surface rather than a plane of constant slope. It is not possible to show graphically the very great variety of possible NLP problem configurations. For a problem to be classed as one of non-linear programming only one of the functions involved need be non-linear. Thus figure 6.3 might have shown linear constraints and a non-linear objective function or a linear objective function and some non-linear constraints. Actually it shows all non-linear functions.

The constraints in figure 6.3 have all been conventionally shaded on the infeasible side and they separate the problem space into a feasible and an infeasible region. Contours of the objective function are shown with values of the function indicated against each contour. Figure 6.3 can be interpreted as the problem of minimising the non-linear objective function $f(x_1, x_2)$ over variables x_1 and x_2 subject to non-violation of the set of four non-linear inequality constraints $g_j(x_1, x_2), j = 1, \ldots, 4$. Inspection of figure 6.3 shows the point $X_1 = (x_1^*, x_2^*)$ to be the solution of this problem, that is, a ball placed on the curved surface of the objective function within the feasible region would roll down the surface and come to rest at the lowest point, X_1. At this point

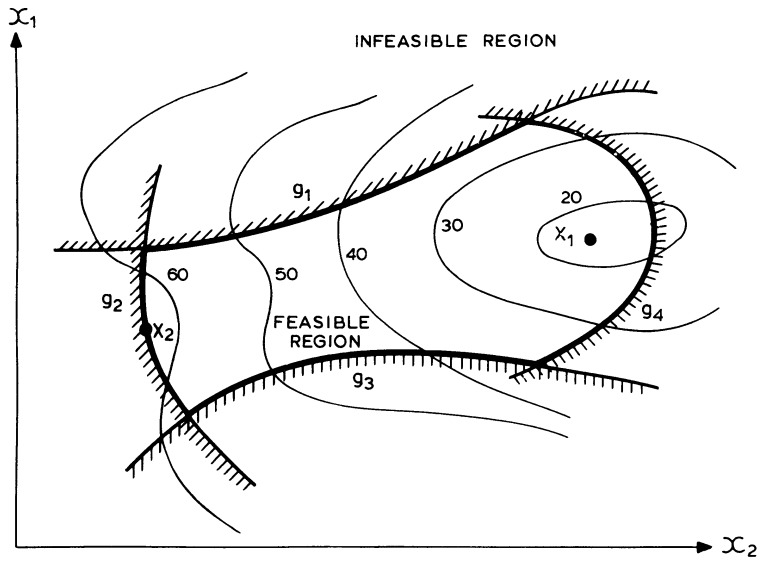


Figure 6.3 Graphical representation of a two-variable NLP problem

www.EngineeringBooksPDF.com

none of the constraints is active. The solution is an internal optimum of the objective function and is totally unaffected by the existence of constraints. This result is in total contrast to the case of a linear programming problem in which the solution must always lie at a constraint vertex.

Figure 6.3 also permits the examination of another NLP problem; that of maximising the objective function subject to non-violation of the four constraints. In this problem the highest point on the objective function surface within the feasible region is sought. By inspection this is the point X_2. This time the solution point is not an internal unconstrained one but lies on the constraint boundary g_2. At the solution one constraint, g_2, is active but the others are slack.

Contrasting figures 2.8 and 6.3 it is evident that the natures of the solutions of LP and NLP problems are fundamentally different. The solution of an LP problem *must* be at a constraint vertex; that for an NLP problem *may* be at a constraint vertex or may be on a constraint boundary or may be unconstrained. Solution methods for NLP problems must, therefore, be fundamentally different from those for LP problems which are based on examining constraint vertices. There are two major sub-divisions within the general topic of non-linear programming. These are *unconstrained NLP* which is examined in detail in chapter 7 and *constrained NLP* to which chapter 8 is devoted. This sub-division is very convenient for classifying and studying different solution methods but it has difficulties when viewed from an engineering problem-solving viewpoint. It assumes that for any problem it will be known *ab initio* whether the solution will be unconstrained or constrained. Often this information is not knowr in advance. In figure 6.3 the solution of the minimisation problem is unconstrainec but it needs only a small movement of constraint g_4 towards the left so that point X_1 becomes infeasible for the solution to become constrained by g_4. For many problems it is not known which constraints will be active or indeed whether any of them will be. It is, therefore, necessary for a solution method for constrained NLP problems to be able to locate an internal unconstrained optimum solution if a particular problem has one. This means that constrained NLP problems are technically more difficult and take longer to solve than unconstrained ones.

One common feature of NLP problems which causes considerable difficulty is the existence of multiple optima. This difficulty does not arise in LP problems because any simplex-type method always locates a feasible solution point at which the objective function has its lowest value. In general NLP problems several solution points may exist each with a different value of the objective function. There is no universal method of finding the best of these multiple optima. It is convenient to examine this difficulty in terms of minimisation but the same difficulty exists for maximisation problems.

For demonstration purposes consider a very simple NLP problem, that of minimising an unconstrained function of one variable, $f(x)$. Figures 6.4a and b show the graphs of two possible functions $f(x)$. In figure 6.4a the minimum of $f(x)$ is at point A but in figure 6.4b the function $f(x)$ has two minima at points B and C. Both B and C in figure 6.4b satisfy the necessary and sufficient

www.EngineeringBooksPDF.com

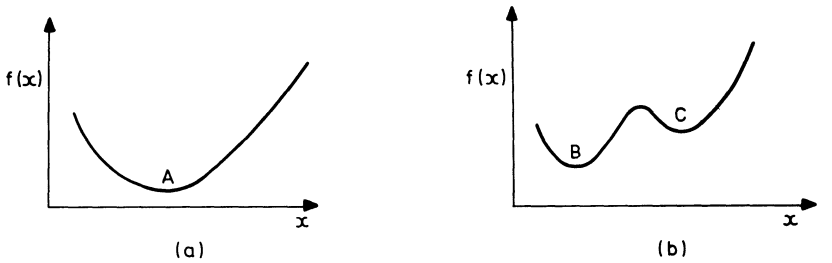


Figure 6.4 Non-linear functions may have more than one minimum

conditions for a minimum, that is

$$\frac{df}{dx} = 0 \text{ and } \frac{d^2f}{dx^2} > 0 \tag{6.13}$$

and so both points are true minima of $f(x)$ within particular ranges of the variable x. Both are called *local minima* but point B, having a lower value of f than point C is called the *global minimum* of $f(x)$. Ideally an NLP solution method should be able to locate a global minimum of a function but practically it turns out to be very hard to do this. All the popular NLP methods are essentially designed to seek out one optimum, check that it is a true optimum, (for example, check relationships 6.13), and terminate. If the function happens to be like that in figure 6.4a the solution will terminate having found a global optimum at point A. If the function $f(x)$ happens to be like figure 6.4b the solution may locate either a solution at B or a solution at C. At both possible points the checks will indicate a true optimum and the method will terminate. Only chance determines whether B or C has been found. *There is no known method which guarantees to find a global optimum of a general NLP problem.* The most that can be guaranteed is that a local optimum will be found which may or may not also be the global optimum.

This guarantee is rather unsatisfactory to anyone who wishes to use NLP methods to solve practical problems. It must be accepted that for many problems the solution obtained will certainly be better than any other point in the immediate vicinity of the solution but may not be as good as some other solution far distant from the point found. Nevertheless, weak though this guarantee is, it is the only one available for general NLP problems. Of course if the problem solver happens to know that the objective function or problem is like figure 6.4a with only one minimum then this strengthens the guarantee; if only one optimum is known to exist it must be a global one. Consequently methods have been devised for examining whether a general problem will have only one optimum. For problems that pass these tests a global optimum can be guaranteed. For problems that fail and cannot be shown to have only one optimum the weak guarantee of finding a local optimum, which may or may not be global, must be accepted. Research

www.EngineeringBooksPDF.com

continues into methods for locating global optima in problems with multiple solutions. Methods suggested centre around determining how many local optima exist (a very difficult problem for general functions), locating all of them (equally difficult), and comparing their objective function values. Success has been very limited and even for the particular problem types where success has been achieved the solution methods are complicated and laborious.

One way of at least partially checking the globality of a solution is to solve the problem several times, each time starting from a different starting point. If all starting points lead to the same solution then its global optimality may be strongly suspected though not guaranteed. If different starting points lead to different solutions the one with the best function value can be selected. This approach to global optimality checking is rather crude and very time-consuming but where local optima are suspected it does provide a partial check.

Returning to figure 6.4 it is clearly worth examining the tests that can be made to determine whether a function $f(x)$ has only one minimum. The graph of $f(x)$ in figure 6.4a is *unimodal* because its curvature (second derivative) has the same sign over the whole function. In figure 6.4b $f(x)$ is not unimodal because the sign of its curvature changes several times. This idea of unimodality is further refined in the concept of *convexity* (positive unimodality) and *concavity* (negative unimodality).

A function of a single variable $f(x)$ is a *convex function* if, for any pair of values of x, say x' and x''

$$f[\lambda x'' + (1 - \lambda) x'] \leqslant \lambda f(x'') + (1 - \lambda) f(x') \tag{6.14}$$

for all values of λ such that $0 \leqslant \lambda \leqslant 1$. It is a *strictly convex function* if $\leqslant$ can be replaced by $<$. It is a *concave function* if $\leqslant$ is replaced by $\geqslant$ and a *strictly concave function* if $\leqslant$ is replaced by $>$.

This definition of convexity is explained graphically in figure 6.5. The left-hand side of the inequality 6.14 represents the function $f(x)$ between any pair of points x' and x'' and the right-hand side describes the line segment (dotted in figure 6.5) which joins x' and x''. If the function $f(x)$ never lies above the line segment, as in figure 6.5a, the function is convex. If the function never lies below the line segment, as in figure 6.5b, the function is concave. Figure 6.5c is similar to the graph of $f(x)$ in figure 6.4b and has two minima. Applying the above definition to it and selecting points x_1' and x_1'', figure 6.5c shows that $f(x)$ always lies below the straight line joining $f(x_1')$ to $f(x_1'')$. The definition requires, however, that for convexity this should be so for any pair of points. If x_2' and x_2'' are chosen it is seen that $f(x)$ lies sometimes above and sometimes below the straight line joining $f(x_2')$ and $f(x_2'')$. Thus in figure 6.5c the function $f(x)$ is neither convex nor concave. Figure 6.5d shows a linear function $f(x)$ and it is clear that it satisfies the definitions of both convexity and concavity since the function lies on the straight line segment. A linear function is, therefore, both convex and concave.

www.EngineeringBooksPDF.com

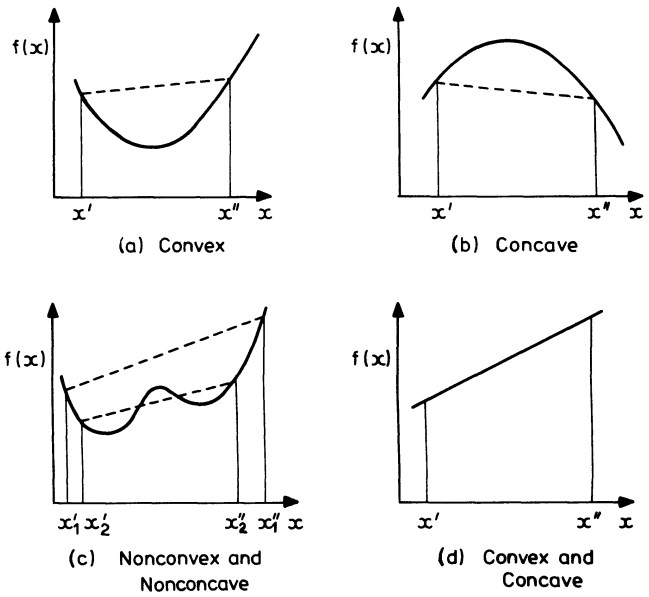


Figure 6.5 Definitions of convex and concave functions

Figures 6.5a and 6.5b show that a global optimum will be found if a convex function is minimised and if a concave function is maximised. Convex minimisation and concave maximisation are entirely equivalent. This result is intuitively obvious; in chapter 2 it was shown that maximisation of a function $f(x)$ was equivalent to minimising the function $-f(x)$. If $f(x)$ is a concave function $-f(x)$ will be a convex function.

In theory the concept of convexity could be used to test any function which might be encountered in unconstrained optimisation. In practice, however, important though the concept of convexity is, it does not provide an effective function-testing procedure. Figure 6.5c demonstrates why. It is not sufficient merely to choose two values of x which span the range of the function and to check function values and line segment values between the points. In order to be certain, all pairs of points must be tested, Also, to apply the test to a particular pair of points, values of the function $f(x)$ and of the line segment must be evaluated at all points between the end points. Since function values are evaluated they themselves will show whether local minima exist and so the convexity test is unnecessary. To test numerically for convexity is a far more laborious process than optimising the function. Convexity was never intended to be used as a numerical test. The great importance of convexity is that it has been used theoretically to determine classes of functions which are known to be convex. Many types of functions are now known to be convex and consequently whenever they must be minimised a global solution can be

www.EngineeringBooksPDF.com

guaranteed. If a particular function to be optimised does not fall into any class of known convex or concave functions then the weak guarantee of local optimality should be accepted because it is too laborious to prove convexity numerically.

The definitions of convexity and concavity given above were stated for a function of one variable x. They remain the same and are equally valid for a function of N variables, $f(x_i)$, $i = 1, \ldots, N$. In N-dimensions, convexity still requires that the straight line segment joining any two points (defined by values of the N variables) lies above or on the N-dimensional function $f(x_i)$.

The existence of multiple optima has been explained thus far in terms of an unconstrained non-linear function. This enabled the idea of convexity to be developed simply. It is in constrained NLP problems, however, that multiple optima are most often found and are most difficult to deal with. Figure 6.6 shows the graph of a two-variable non-linear function $f(x_1, x_2)$ and four non-linear constraints. If $f(x_1, x_2)$ is to be maximised within the feasible region defined by the constraints, five local maxima exist at A, B, C, D and E. Four of these are at constraint vertices but point B is not. If $f(x_1, x_2)$ is to be minimised feasibly, two local minima exist at F and G which are not at constraint vertices. Thus this simple two-variable problem has many possible solution points.

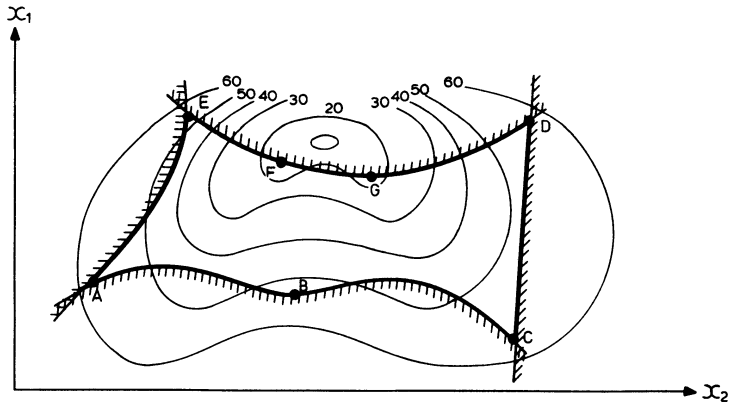


Figure 6.6 Multiple optima in a constrained non-linear problem

For unconstrained optimisation the convexity of an objective function determined whether it would have only one minimum. The idea of convexity can be extended to constrained optimisation. A constrained NLP problem typically has M non-linear constraints in N variables which can be expressed in the form

$$g_j(x_1, \ldots, x_N) \left\{ \begin{matrix} \leq \\ \text{or} \\ \geq \end{matrix} \right\} 1 \quad j = 1, \ldots, M$$

www.EngineeringBooksPDF.com

These constraints form a *convex set* when

(1) $g_j(x_1, \ldots, x_N)$ is a convex function and the inequality has a $\leqslant$ sign
(2) $g_j(x_1, \ldots, x_N)$ is a concave function and the inequality has a $\geqslant$ sign.

These definitions determine whether the constraints of a non-linear problem form a convex set. If any constraint does not satisfy these rules the entire constraint set is deemed to be a non-convex set. Essentially these rules determine whether all the boundaries of the feasible region 'bulge outwards'. If they do, the constraints form a convex set. If any boundary 'bulges inwards' the constraints form a non-convex set. Figure 6.7 demonstrates this.

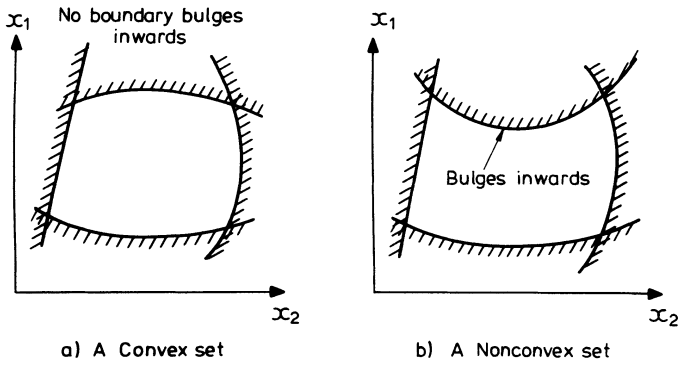


Figure 6.7 Convex and non-convex sets

In figure 6.7a the constraints form a convex set because the boundaries bulge outwards. The linear constraint does not bulge either way but it has already been shown by figure 6.5d that linear functions are both concave and convex. In figure 6.7b the constraint set is non-convex because one of the constraint boundaries bulges inwards into the feasible region. The constraint sets in figures 6.3 and 6.6 are both non-convex.

The importance of a convex set of constraints is that a global optimum can be guaranteed if a convex function is minimised or a concave function maximised over the convex set of constraints. If the constraint set is non-convex or the objective function does not satisfy these conditions, local optima may exist (as in figure 6.6) and no global optimum can be guaranteed. Figure 6.3 also fails to meet these conditions yet, fortuitously, it has only global optima. Only very slight adjustments to the constraints in figure 6.3 would produce several local optima. In linear programming all constraints are linear and they therefore form a convex set. The linear objective function may be thought of as convex for minimisation problems and concave for maximisation. Therefore, linear programming problems satisfy all convexity conditions and a global optimum can be guaranteed.

www.EngineeringBooksPDF.com

6.4 ENGINEERING AND MATHEMATICAL VIEWPOINTS ON NON-LINEAR OPTIMISATION

Section 6.3 has demonstrated some of the difficulties which local optima present to the solution of NLP problems. From a mathematical viewpoint the general NLP problem has very many different forms and, for many of these, global optimality cannot be assured. Research continues into developing globally optimal non-convex solution methods but at present these methods are of limited applicability and are laborious. Nevertheless most NLP algorithm writers nurture the ultimate, though far-distant, goal of developing a general, globally optimal solution method for all non-linear problems. Realising the short-term impossibility of achieving this goal many mathematicians have concentrated upon achievable goals particularly in the area of convex problems. Many methods have been proposed for solving non-linear problems with linear constraints, convex constraint sets, quadratic functions, etc. The aim this work is to produce rapid and efficient solution methods for particular classes of problems which will, if the problem is convex, locate a global optimum as quickly as possible.

The engineering approach to non-linear problems contrasts with this. Engineering design problems are usually non-linear and each physical design problem creates a new and different non-linear programming problem. An engineer cannot pick and choose the appropriate form for a design problem. The mathematical form of a design problem is determined by the physical requirements and mechanical behaviour of the object being designed. Usually very many widely different designs can be created for a single project and this suggests that many locally optimum designs may be possible. Many design problems are, therefore, inherently non-convex.

To the engineering designer it is often frustrating to see so many mathematical methods available for solving small, convex NLP problems when, in contrast, all engineering design problems seem to be large and non-convex. It sometimes seems as if algorithm writers inhabit an unreal world in which problems are all conveniently structured and convex. It is hoped that section 6.3 has demonstrated why, at the present time, the general NLP problem cannot be solved. The mathematical nature of non-convex problems is not fully understood and methods for finding global optima are consequently not available. Frustrating though this may be, it must be accepted. The NLP methods which are available are usually efficient at locating a global optimum of a convex problem and so it can be intuitively expected that they will also be able to locate at least a local optimum of a non-convex problem fairly efficiently. Taking a wider perspective, in engineering problems it is not usually of such paramount importance that a globally optimal solution be found. As was mentioned in section 6.1 the mathematical problem of design synthesis is rarely a complete problem. It seldom synthesises a final design but rather produces an initial design for further modification and analysis. It is consequently not usually important that a globally

www.EngineeringBooksPDF.com

optimum rather than a locally optimum design is synthesised since modifications to it are inevitable. The fact that a locally optimal design is better than any other design in its immediate vicinity is sufficient to warrant searching for a locally optimal design.

The next two chapters develop several solution techniques for non-linear optimisation problems. This chapter has shown that unconstrained optimisation is conceptually simpler than constrained non-linear programming. Chapter 7 is therefore devoted to solution methods for unconstrained problems.

SUMMARY

A systematic mathematical modelling approach may be used for decision-making in many aspects of the planning, construction, operation and management of civil engineering projects. A similar systematic method can be applied to problems of the technological design of the works themselves. Design is fundamentally a problem of synthesis and the framework afforded by mathematical optimisation is an excellent aid to design. This chapter has shown how different technological design problems can be mathematically modelled for synthesis purposes as optimisation problems. These problems necessarily involve non-linear functions of the design variables and consequently require non-linear programming methods to solve them.

The nature of non-linear programming problems was examined and was found to be far more varied and complex than that of linear programming problems. Two major classes of problems – those with constraints and those without – were identified for further study. Superficial examination of both classes revealed the possibility that many local optima may exist and that in general the goal of finding a global optimum solution can only be guaranteed in the case of convex problems. This is less than ideal for solving engineering design problems, many of which are non-convex, but nevertheless it must be accepted that frequently only a locally optimum solution will be found. The chapter examined the nature of convex functions and convex programming problems but showed that convexity tests can be laborious and give no indication of how many local optima will exist. The idea of convexity is, however, very important as it underlies most of the algorithms studied in the next two chapters.

www.EngineeringBooksPDF.com

7 NON-LINEAR UNCONSTRAINED OPTIMISATION METHODS

Non-linear optimisation methods can be broadly classified into two groups: methods for solving unconstrained problems and those for solving constrained problems. This chapter introduces unconstrained methods. Although many technological design problems are constrained, within civil engineering generally there are many uses for unconstrained non-linear optimisation methods. A major reason for a detailed examination of unconstrained optimisation is to allow the principles of solution methods to be demonstrated simply. Many of these principles are used in solution methods for unconstrained problems described in the next chapter. Indeed, one of the most widely used methods for solving constrained optimisation problems, the penalty function method, expresses the constrained problem as a sequence of unconstrained problems which are solved by the methods of this chapter.

This chapter starts by examining the classical differential method of finding the maximum or minimum of a function of several variables and discusses why the method fails to solve many types of problem. Numerical search methods are then described starting with methods that make only function evaluations. Methods suitable for use when first derivative information is available are then examined and finally methods using actual or approximated second derivatives are described. The relative efficiencies of the methods are compared. Finally, ways of selecting methods appropriate to different problem characteristics are explored.

7.1 THE CLASSICAL DIFFERENTIAL METHOD

Suppose it is required to find the least value of a function of a single variable, that is

$$\text{Minimise } f(x) \text{ over variable } x$$

The methods described in this chapter are equally applicable to finding greatest values as well as least values. For simplicity only minimisation will be

www.EngineeringBooksPDF.com

described in detail. Necessary and sufficient conditions for a minimum are

$$\left.\begin{aligned} &\frac{df}{dx} = 0 \\ &\text{and} \\ &\frac{d^2f}{dx^2} > 0 \end{aligned}\right\} \tag{7.1}$$

These conditions require that the slope or gradient of the function is zero at a possible minimum point and that the function is locally convex in the region of the minimum. These conditions can form the basis of a solution procedure for the minimisation problem. The procedure consists of setting up the equation represented by the first of the conditions 7.1 and finding all possible solutions of the equation. At each of the candidate solution points the sign of the second derivative should be checked. Only those points with strictly positive second derivatives are retained. The remainder will include points with negative second derivatives corresponding to maxima, or with zero second derivatives corresponding to points of inflexion, and are discarded. Having retained all solutions satisfying conditions 7.1 these are local minima. The value of the function f should be evaulated at each of them and the point with the least value of f is the global minimum.

This method may be extended to problems involving a function of several variables. Consider the problem

$$\text{Minimise } f(x_1, x_2, \ldots, x_N) \text{ over variables } x_i, i = 1, \ldots, N$$

The first of the conditions 7.1 represents stationarity of the function. When a function has N variables a stationary point must be stationary with respect to all variables simultaneously. The N-variable equivalent of the first of the conditions 7.1 is then

$$\frac{\partial f}{\partial x_i} = 0 \qquad i = 1, \ldots, N$$

Notationally, each of the first partial derivatives $\partial f/\partial x_i, i = 1, \ldots, N$ can be represented by $g_i, i = 1, \ldots, N$. The g_i then form a column vector $\boldsymbol{g}$ and the above stationarity condition may be written as

$$\boldsymbol{g} \equiv \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_N \end{bmatrix} \equiv \begin{bmatrix} \dfrac{\partial f}{\partial x_1} \\ \dfrac{\partial f}{\partial x_2} \\ \vdots \\ \dfrac{\partial f}{\partial x_N} \end{bmatrix} = 0 \tag{7.2}$$

www.EngineeringBooksPDF.com

All possible solutions of the N simultaneous equations $\boldsymbol{g} = 0$ must be found. The second of the conditions (7.1) ensures that the stationary point is a minimum. For a problem in N variables this condition is more complicated. Instead of a simple check on the sign of a single second derivative, the sufficiency check for a minimum in N variables is that the matrix of second partial derivatives, called the *hessian matrix*, $\boldsymbol{H}$, is positive definite. For a maximum $\boldsymbol{H}$ must be negative definite. Notationally

$$\boldsymbol{H} \equiv \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \cdots & \frac{\partial g_1}{\partial x_N} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \cdots & \frac{\partial g_2}{\partial x_N} \\ \vdots & \vdots & & \vdots \\ \frac{\partial g_N}{\partial x_1} & \frac{\partial g_N}{\partial x_2} & \cdots & \frac{\partial g_N}{\partial x_N} \end{bmatrix} \tag{7.3}$$

$$\equiv \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_N} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_N} \\ \vdots & \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_N \partial x_1} & \frac{\partial^2 f}{\partial x_N \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_N^2} \end{bmatrix}$$

One way of testing for positive definiteness of $\boldsymbol{H}$ is to evaluate the determinants of various sub-matrices of $\boldsymbol{H}$. Defining

$$d_1 = \left| \frac{\partial g_1}{\partial x_1} \right| \qquad d_3 = \begin{vmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \frac{\partial g_1}{\partial x_3} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \frac{\partial g_2}{\partial x_3} \\ \frac{\partial g_3}{\partial x_1} & \frac{\partial g_3}{\partial x_2} & \frac{\partial g_3}{\partial x_3} \end{vmatrix}$$

$$d_2 = \begin{vmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} \end{vmatrix} \tag{7.4}$$

$$d_4 = \text{etc.}$$

www.EngineeringBooksPDF.com

the matrix $\boldsymbol{H}$ is positive definite if all the determinants $d_1, d_2, d_3, d_4, \ldots, d_N$ are positive. If the problem has only a few variables this check is simple but it becomes a major computational task if N is large.

Having obtained all possible solutions of $\boldsymbol{g} = 0$ and having checked the positive definiteness of $\boldsymbol{H}$ at each candidate solution, the local minima are found and any other stationary points cast aside. Evaluation of f at each local minimum allows the global minimum to be found.

7.1.1 Difficulties Arising with the Differential Method

The differential method sometimes solves problems satisfactorily and quickly. More usually, however, it does not yield results. There are many reasons for this and it is useful to examine some of them because the numerical methods described in this chapter were developed to circumvent the failures of the differential method.

For notational simplicity a function of many variables $f(x_1, x_2, \ldots, x_N)$ will be written as $f(\boldsymbol{x})$ where $\boldsymbol{x}$ is a vector with N components x_i, $i = 1, \ldots, N$. Consider the problem of minimising $f(\boldsymbol{x})$. The first step of the classical method is to differentiate $f(\boldsymbol{x})$ and to set up the equations $\boldsymbol{g} = 0$ (equations 7.2). $f(\boldsymbol{x})$ must, therefore, be analytically differentiable. Also, both $f(\boldsymbol{x})$ and its first derivatives $\boldsymbol{g}(\boldsymbol{x})$ must be continuous functions of a closed algebraic form in order to set up $\boldsymbol{g} = 0$. Clearly the differential method cannot be applied to functions that cannot be differentiated algebraically nor to discontinuous functions. There are very many problems in civil engineering that exhibit these awkward features. None of the variables in $\boldsymbol{x}$ must be integer or discrete valued because the first differentials will not then exist. This rules out problems involving the optimisation over numbers of men or sizes of pipes, rolled steel sections, etc., and it also rules out functions $f(\boldsymbol{x})$ representing, for example, excavation where different unit costs are applicable over specific ranges of depths of excavation. Thus the requirement of continuous, differentiable functions is a fairly severe restriction upon the applicability of the method to many engineering problems.

There is an even more severe difficulty arising in solution of the equations $\boldsymbol{g} = 0$ even if $f(\boldsymbol{x})$ and $\boldsymbol{g}(\boldsymbol{x})$ are continuous and differentiable. Since $f(\boldsymbol{x})$ is a general non-linear function its first derivatives $\boldsymbol{g}(\boldsymbol{x})$ are also generally non-linear. The equations $\boldsymbol{g} = 0$ are then a set of N simultaneous non-linear equations in N unknowns. The solution of this set of equations can be very difficult because of their non-linear nature. The differential method strictly requires that all possible solutions be found. For non-linear equations it is not usually possible to determine how many solutions exist; even to locate one solution may be difficult and can usually only be done numerically rather than analytically.

For the above reasons the classical differential method is only really applicable to relatively simple problems involving a few variables. Most engineering problems do not satisfy the stringent demands of the method and cannot be solved in this way. Problems, however, do exist and solutions must be found.

www.EngineeringBooksPDF.com

Consequently a range of different methods have been developed to solve general unconstrained optimisation problems. The best feature of the classical method is its theoretical completeness. In principle it locates all stationary points and, therefore, enables a global optimum to be found but in practice it does not often work. The alternative methods described in this chapter are successful because they do not aim to find a complete global optimum; they are designed to find a local optimum only. The biggest disappointment with the classical method is that it fails to solve a large proportion of problems. A prime requirement for an alternative method is, therefore, robustness; it must produce a local optimum for any problem whatever the form of the objective function $f(\boldsymbol{x})$. It is natural to expect that in some problems a lot of information will be available about the nature of $f(\boldsymbol{x})$, that is, values of the function and its first and second derivatives may be readily calculable. In other problems, however, little information about $f(\boldsymbol{x})$ may be available, that is, values of $f(\boldsymbol{x})$ might only be obtainable after lengthy computations. If information about $f(\boldsymbol{x})$ is hard to obtain, a local minimum will be harder to find than if a lot of information were known. Methods for unconstrained optimisation can be classified into groups according to how much is known about $f(\boldsymbol{x})$. These classes are

(1) Zeroth-order methods: methods which use only values of the function $f(\boldsymbol{x})$. Given numerical values of the components of $\boldsymbol{x}$, a value of $f(\boldsymbol{x})$ must be calculable.

(2) First-order methods: methods which use function values and first derivative values. Given a numerical $\boldsymbol{x}$, values of $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ must be calculable.

(3) Second-order methods: methods that rely on the availability of function values and values of first and second derivatives.

There is a certain amount of overlap among these classes. For instance, a first-order method sometimes estimates approximate second derivatives from first derivative values. In this chapter, however, each of the three classes will be examined separately.

7.2 ZEROTH-ORDER METHODS

Zeroth-order methods use only values of the function $f(\boldsymbol{x})$ and do not require the function to be continuous or differentiable. Consequently they are applicable to all unconstrained optimisation problems and are particularly valuable as the only methods applicable to solving the awkward problems with discontinuities. The specific requirement of a zeroth-order method is that it must be possible to evaluate in some way the value of $f(\boldsymbol{x})$ given numerical values for the components x_i, $i = 1, \ldots, N$, of $\boldsymbol{x}$. Zeroth-order methods are consequently numerical search methods. Each numerical evaluation of $f(\boldsymbol{x})$ at a given $\boldsymbol{x}$ is called a *trial* and the objective of a zeroth-order method is to locate the point $\boldsymbol{x}^*$ at which $f(\boldsymbol{x})$ achieves a minimum value using as few trials as possible.

www.EngineeringBooksPDF.com

7.2.1 Single-variable Problems. Minimisation along a Line

It is convenient to start examining zeroth-order methods by studying problems involving a function $f(x)$ of only one variable, x. As will be shown later in this chapter many of the search methods used for optimising functions of many variables, $f(\boldsymbol{x})$, actually solve the problems as a sequence of single-variable minimisations. Thus single-variable minimisation is a crucial aspect of all non-linear optimisation methods and deserves careful study.

The problem examined here is that of finding a value for the variable x that gives a general function $f(x)$ a minimum value. In order to understand the difficulties of solving such a problem numerically it is helpful to imagine that values of $f(x)$ are provided by a computer program, details of which are unknown. If a value of x is input to the program the output will be a value of $f(x)$. How can the least value of $f(x)$ be found with the fewest number of input trial values of x?

7.2.1.1 Grid Search and Random Search

Faced with a completely unknown function $f(x)$ an idea which would probably occur to many people is that the computer should evaluate $f(x)$ at a series of equally spaced values of x and so locate very approximately the position of the minimum. Unfortunately, nothing is known about $f(x)$. Its minimum may be located anywhere between $x = -\infty$ and $x = +\infty$, therefore the initial spacing of the values of x can only be guessed. Suppose, however, that ten values equally spaced over a large finite range are chosen and $f(x)$ is evaluated at each and that at one particular value, x', f is lower in value than at any other point. Figure 7.1 shows this. Logically a minimum of $f(x)$ must now lie somewhere between the two values of x immediately adjacent to x'. Ten further trials are

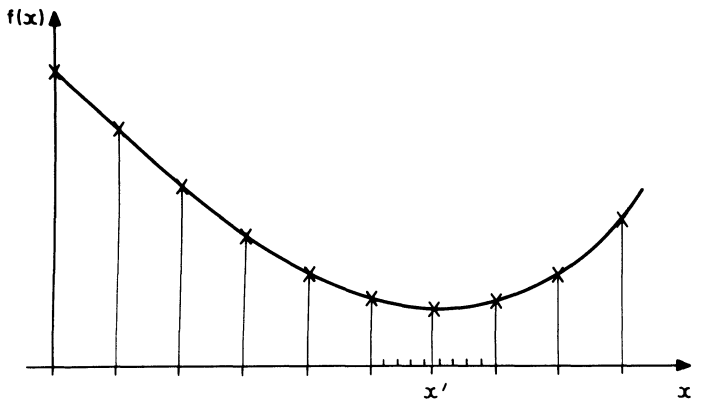


Figure 7.1 Grid search on a function of one variable

www.EngineeringBooksPDF.com

then made at equally spaced values of x between these two limiting values and the lowest value of $f(x)$ is again found. This process of sub-division of the interval of x which brackets the minimum continues until the minimum of $f(x)$ and its corresponding x^* are found to any desired degree of accuracy. This method is known as *grid search* and it is a most inefficient solution method. As described above, seven out of every ten trials made are discarded and only three are retained. This is a very high proportion of useless trials. The disadvantages of grid search become even more apparent if the method is applied to functions $f(x)$ of more than one variable. For a two-variable problem, the range of each variable must be divided up into, say, ten intervals and the function evaluated at each point on the ten by ten grid which covers the function, that is, 100 trial evaluations. Of these only 9 are retained to bracket the minimum and 91 discarded. For a problem in three variables 1000 trials (10 x 10 x 10) must be made and only 27 retained. Thus the grid search method is highly inefficient in its use of trials.

Another possible search method is that of *random search.* Instead of dividing the range of variable x into an equally spaced grid it is possible to generate a completely random set of values of x and evaluate $f(x)$ at each random point. In practice it is found that the total number of trials needed to locate the minimum to the same degree of accuracy as the grid search method is somewhat larger than the number required by grid search, that is, far too many.

Both grid search and random search are essentially crude and inefficient methods. Too much information is evaluated and cast aside unused. A far better approach is to make evaluations of $f(x)$ singly and to use each evaluation to determine where to place the next one so that the value of $f(x)$ will decrease.

7.2.1.2 Sequential Search. Bracketing the Minimum

In the grid search method the minimum was bracketed by the two values of x immediately above and below x' and the interval between these values was then successively reduced. A more efficient way of bracketing the minimum is shown in figure 7.2a.

Using a superscript notation to denote sequential values of the variable x, a starting value of variable x is chosen, x^0, and a step-length, s. The function f is evaluated at x^0 and then at $x^1 = x^0 + s$. If $f(x^0) > f(x^0 + s)$ as shown in figure 7.2a, new trials are made, one at a time, at $x^2 = x^0 + 2s$, $x^3 = x^0 + 3s$, $x^4 = x^0 + 4s$, . . . , checking the value of $f(x)$ after each trial until a function increase over the previous trial is obtained. The last three trials, x^4, x^5 and x^6 in figure 7.2a then bracket the minimum. If after the first two trials $f(x^0) < f(x^0 + s)$ the new trials are made at $x^0 - s$, $x^0 - 2s$, $x^0 - 3s$, . . . , again checking the value of $f(x)$ after each trial, continuing with new trials if $f(x)$ continues to decrease and stopping when a function increase is found. Again, the minimum is bracketed by the final three trials.

In this method the first two trials determine on which side of the arbitrarily chosen starting point the minimum lies. Subsequent trials exploit this knowledge

www.EngineeringBooksPDF.com

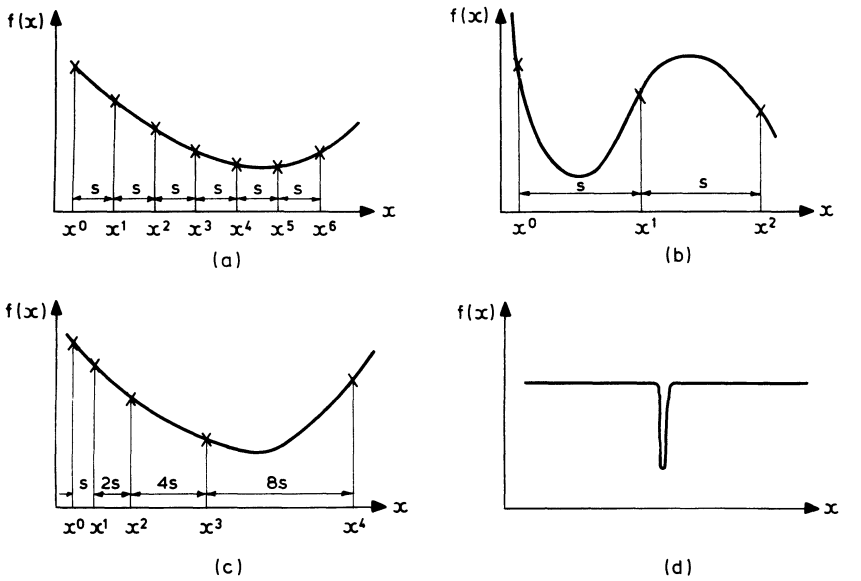


Figure 7.2 Bracketing the minimum in a one-variable search

and are placed so that $f(x)$ should always decrease in value until a minimum is bracketed. The choice of a step-length s is important. If s is too large the minimum may be overstepped and subsequent trial results misinterpreted as in figure 7.2b. If s is too small very many trials may be needed to bracket a minimum far distant from the starting point. To try to overcome these potential difficulties methods using a variable step-length are usually used. Instead of keeping s constant throughout the search s starts small and increases in magnitude at each trial if the function values continue to decrease steadily. Typically s may be doubled at each new trial giving a sequence of trials as shown in figure 7.2c. Another method alters the step-length s in accordance with the Fibonacci series (see 7.2.1.4). In this scheme successive step-lengths are made proportional to the numbers of the Fibonacci series, 1, 1, 2, 3, 5, 8, 13, 21, 34, ... This has advantages if the same Fibonacci series is later used to locate the minimum accurately within the bracketed limits (see the example in section 7.2.1.5). Many other variable step-length schemes are possible. They all have the same goal: to bracket the minimum as quickly and within as small an interval as possible using the fewest numbers of trials, yet to avoid missing it completely. None of them is entirely foolproof: the function shown in figure 7.2d would probably defeat all discrete search methods except for a very lucky positioning of trials. Sequential bracketing methods, however, are far more efficient than grid or random search methods and are among the best available for use in minimising a function of a single variable. Fortunately, there appear to be very few, if any, practical problems involving functions like figure 7.2d.

www.EngineeringBooksPDF.com

7.2.1.3 Sequential Search. Locating the Minimum

The bracketing methods of the previous section are such that the last three trials define an interval of the variable x within which a minimum of the function must occur. The optimum value, x^*, of the variable x is located such that $a \leqslant x^* \leqslant b$ where a and b are the values, not necessarily respectively, of the last and the last-but-two trial values of x. For a constant step-length search the interval $(b - a)$ will be of length $2s$. If a variable step-length search has been used it may be much larger than this. This section is concerned with placing further trials within this interval to locate the minimum more precisely. An exact optimum can never be located because the term 'exact' does not have a precise meaning for a continuous-valued quantity. There is always a tolerance or error-margin around any numerical solution. For example the 'exact' solution $x^* = 3.792$ means that x^* lies somewhere in the interval $3.7915 \leqslant x^* \leqslant 3.7925$, that is x^* has been located to lie within an interval of length 0.001. Locating the minimum of the function is then the same as reducing the size of the initial interval $(b - a)$ to some desired small size ϵ within which the minimum is known to lie.

It is possible to reduce the initial interval $(b - a)$ by using the bracketing methods once again, starting at one end of the interval with a much reduced step-length s. This process will bracket the minimum within a much reduced interval which can be further reduced using the same method and a still smaller value for the step length. This method certainly works but uses very many more trials to achieve the same interval reduction than do the methods now to be described. The key concept of these more efficient methods is that new trials are used to reduce the initial interval by being so placed as to identify regions of the interval in which the minimum *cannot* lie. The interval is reduced by eliminating unwanted portions.

The bracketing method operates in such a way that the last and the last-but-two trials define an initial interval and the last-but-one trial lies somewhere in that interval with a function value less than either the preceding or the subsequent trial. The last three trials define a region of $f(x)$ which has the appearance of unimodality. Actually it is possible for $f(x)$ not to be unimodal despite these appearances. Nevertheless the following methods assume that the function $f(x)$ is unimodal within the bracketed interval.

The three existing trials do not permit the interval to be reduced. Figure 7.3 shows that, depending on the shape of the function, the minimum of $f(x)$ may lie on either side of the last-but-one trial, x^2. Neither of the intervals $(x^2 - x^1)$ or $(x^3 - x^2)$ may be eliminated on logical grounds. If, however, another trial x^4 is made to one side of x^2 this trial should identify a region within which x^* cannot lie. There are four possible outcomes of the trial x^4 depending where it is placed.

(1) $x^4 > x^2$ and $f(x^4) > f(x^2)$. This is a point A on figure 7.3. The function described is $f_1(x)$. Values $f(x^1), f(x^2)$ and $f(x^4)$ define a unimodal region which must contain a minimum $f(x^*)$. The minimum definitely cannot lie between x^4 and x^3 so the initial interval can be reduced by an amount $(x^3 - x^4)$.

www.EngineeringBooksPDF.com

(2) $x^4 > x^2$ and $f(x^4) < f(x^2)$, that is, point B on figure 7.3. $f_2(x)$ is the function described. This time $f(x^2), f(x^4)$ and $f(x^3)$ define a unimodal region. The minimum cannot lie between x^1 and x^2 so the initial interval can be reduced by an amount $(x^2 - x^1)$.

(3) $x^4 < x^2$ and $f(x^4) > f(x^2)$. This is point C on figure 7.3 describing $f_2(x)$. A unimodal region is defined by $f(x^4), f(x^2)$ and $f(x^3)$. The minimum cannot lie between x^1 and x^4 so the interval can be reduced by $(x^4 - x^1)$.

(4) $x^4 < x^2$ and $f(x^4) < f(x^2)$. This is point D on figure 7.3 describing $f_1(x)$. $f(x^1), f(x^4)$ and $f(x^2)$ define a unimodal region. The minimum cannot lie between x^2 and x^3 so the interval can be reduced by $(x^3 - x^2)$.

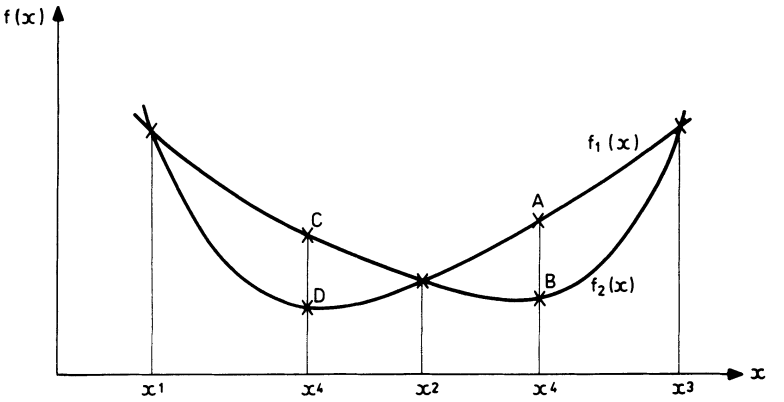


Figure 7.3 Possible outcomes of trial x^4 on two functions $f_1(x)$ and $f_2(x)$

Each outcome of the trial x^4 enables a portion of the initial interval $(x^3 - x^1)$ to be eliminated because unimodality precludes the minimum from being located in the eliminated portion. Having eliminated some region as a result of trial x^4, the new reduced interval has trials at each end and at some intermediate point and these trials describe a unimodal region. A further trial x^5 can be placed within the new reduced interval and the possible outcomes of that trial allow a further portion to be eliminated. This process of trial-elimination-trial-elimination continues until the length of the unimodal interval is less than some prescribed tolerance ϵ. This method is efficient because every trial reduces the interval within which x^* lies. It raises the interesting question of how the trials should be placed so that an initial interval $(b - a)$ is reduced to a final interval of ϵ using the smallest number of new trials.

This question still awaits an answer. Since the shape of the function $f(x)$ is unknown during the search process, except for values at previous trial points, the outcome of any new trial cannot be predicted. There is a chance element in all purely numerical search strategies. Consequently trials cannot be placed so as to achieve maximum reduction of a present interval because the result of a trial,

www.EngineeringBooksPDF.com

on which the size of any reduction depends, is not known until after the trial is made. No answer to the question as posed above is possible. However, it may be posed in a way that does permit an answer. If a pessimistic view is taken of the likely outcome of all trials, that is, that the outcome of any trial will achieve only the smallest of the interval reductions which might be achieved by possibilities (1) to (4), a best-pessimistic strategy can be worked out. This approach of trying to find the best policy assuming that the worst will always happen is often used in optimisation and is referred to as a *minimax strategy*. Use of a minimax strategy to place trials within an interval to locate a minimum yields the Fibonacci search method.

7.2.1.4 Fibonacci Search Method

The Fibonacci method is perhaps best understood if the problem it solves is posed in the form of a question: 'How many trials, n, are needed to reduce an interval of length L within which x^* lies to an interval of length ϵ also containing x^* assuming pessimistic trial outcomes?'. The answer to this question may be deduced graphically and inductively in a reversed sense, that is, by answering the question: 'What is the largest interval L that can always be reduced to ϵ by a specified number of trials, n, assuming pessimistic outcomes?'.

Figure 7.4a shows an interval L_1 with existing trials at each end, x^0 and x^{00}. The function $f(x)$ within L_1 is assumed to be unimodal and to contain x^*.

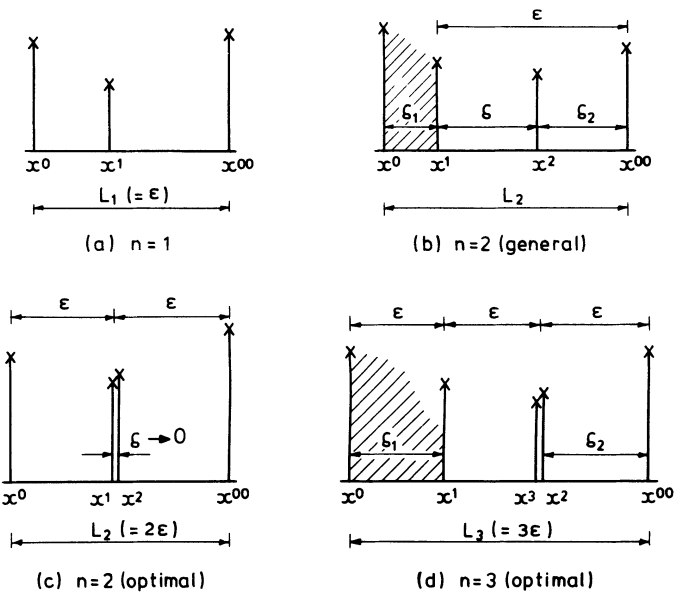


Figure 7.4 Maximising an initial interval L which can be reduced to ϵ

www.EngineeringBooksPDF.com

Suppose only one extra trial may be made within the interval L_1, $n = 1$. What is the maximum size of L_1 such that x^* lies within an interval ϵ after one trial? The answer to this is clearly $L_1 = \epsilon$. It has been shown earlier that *two* internal trials are needed if L_1 is to be reduced by elimination. One trial x^1 is insufficient to permit any reduction so L_1 must equal the final interval ϵ. By a similar argument the maximum interval which can be reduced to ϵ after zero further trials is $L_0 = \epsilon$.

Figure 7.4b shows an interval L_2 also with existing trials at each end. Suppose $n = 2$, that is, two more trials may be made within the interval. How big can L_2 be so that x^* is found within ϵ after two more trials? Suppose the trials are made at x^1 and x^2 with $x^2 > x^1$ as shown in figure 7.4b. There are two possible outcomes of these trials: $f(x^2) > f(x^1)$ or $f(x^2) < f(x^1)$. If the outcome is $f(x^2) > f(x^1)$, the minimum must lie somewhere between x^0 and x^2 with the interval (x^2, x^{00}) being eliminated. If the outcome is $f(x^2) < f(x^1)$, the minimum must lie between x^1 and x^{00} with the interval (x^0, x^1) being eliminated. The pessimistic policy requires that the smaller of the intervals (x^2, x^{00}) and (x^0, x^1) is the one which must be eliminated. Suppose (x^0, x^1) is the smaller and is of length δ_1 as in figure 7.4b. Interval (x^2, x^{00}) is of length δ_2 which is larger than δ_1. The pessimistic policy then eliminates δ_1 and leaves the interval (x^1, x^{00}) within which x^* lies. As no more trials are possible this must clearly equal the final interval ϵ. Thus the initial interval L_2 must be of length $\delta_1 + \epsilon$. For L_2 to be as large as possible δ_1 must be made as large as possible. As already noted, however, δ_1 is the smaller of the two possible eliminated intervals δ_1 and δ_2. δ_1 obviously cannot be increased such that it becomes greater than δ_2 because it would not then be the pessimistic outcome of the two trials. Hence the greatest value δ_1 can have is when it becomes equal to δ_2. Thus L_2 will be greatest when $\delta_1 = \delta_2$ and the trials x^1 and x^2 are symmetrically placed within L_2.

Since $\delta_1 = \delta_2$ the geometry of figure 7.4(b) now requires that when either δ_1 or δ_2 has been eliminated as a result of the two trials, the final intervals remaining must be of the same size and equal to ϵ. The following relationships can be established from the geometry of figure 7.4b.

$$L_2 = \delta_1 + \epsilon$$
$$\delta_1 + \delta = \epsilon = \delta_2 + \delta$$

which can be used to eliminate δ_1 giving

$$L_2 = 2\epsilon - \delta \tag{7.5}$$

The length interval L_2 as given by equation 7.5 increases as δ, the spacing of the trials x^1 and x^2, decreases. Thus

$$L_2 \to 2\epsilon \text{ as } \delta \to 0$$

The maximum interval L_2 that can be reduced to ϵ by two trials is therefore $L_2 = 2\epsilon$. The two trials x^1 and x^2 should be placed as close together as possible

www.EngineeringBooksPDF.com

(theoretically coincident) at the centre of the interval L. Figure 7.4c shows this. Essentially the two almost coincident trials determine whether $f(x)$ slopes downwards to the left or to the right at the centre of the interval and hence determine in which semi-interval of length ϵ x^* lies. Figure 7.4c accords with intuition and has been proved above to be the optimal placement for two trials.

Figure 7.4d shows an interval L_3 within which three trials are to be placed, $n = 3$. What is the largest interval L_3 which can be reduced to ϵ with three trials? Again the trials at the ends of L_3 are assumed to have been made already. A logical proof can be derived in a similar way to the $n = 2$ case but is not given here in full. Briefly, however, when the first two trials x^1 and x^2 are made either δ_1 or δ_2 in figure 7.4d must be eliminated. A minimax policy again requires that δ_1 and δ_2 must be of the same size and so x^1 and x^2 must be symmetrically placed within L_3. Irrespective of whether δ_1 and δ_2 is eliminated the remaining interval will have one internal trial within it (either x^1 or x^2 depending on whether δ_2 or δ_1 has been eliminated). One further trial may be made. Recalling the $n = 2$ case it can be deduced that the intermediate trial should be in the centre of the remaining interval so that the third trial can be placed very close to it. This reasoning leads to the conclusion that $L_3 = 3\epsilon$ with trials x^1 and x^2 spaced so as to trisect L_3, trial x^3 is very close to either x^1 or x^2 depending on previous outcomes.

Solutions for $n = 4, 5$, etc., follow a similar pattern of using the placings of the solution for the previous value of n. It can be deduced that $L_4 = 5\epsilon, L_5 = 8\epsilon$, etc. If these results are listed a pattern becomes evident

$$\left.\begin{aligned} L_0 &= \epsilon \\ L_1 &= \epsilon \\ L_2 &= 2\epsilon = L_1 + L_0 \\ L_3 &= 3\epsilon = L_2 + L_1 \\ L_4 &= 5\epsilon = L_3 + L_2 \\ L_5 &= 8\epsilon = L_4 + L_3 \end{aligned}\right\} \qquad (7.6)$$

Hence one can deduce a general expression for L_n, the largest interval which can be reduced to ϵ in n trials.

with
$$\left.\begin{aligned} L_n &= L_{n-1} + L_{n-2} \\ L_0 &= L_1 = \epsilon \end{aligned}\right\} \qquad (7.7)$$

The coefficients of ϵ in equations 7.6 and 7.7 form a sequence 1, 1, 2, 3, 5, 8, 13, 21, 34, . . . , each number in the sequence being the sum of the two previous numbers. This sequence is known as the *Fibonacci series* being named after the thirteenth-century Italian who investigated it (not in a non-linear optimisation context). The sequential numerical search procedure in which it plays a part has consequently been called the Fibonacci search method.

The Fibonacci search method uses relationship 7.7 and the knowledge of where trials must be placed described above in a reverse sense. Figure 7.5 shows how the method is used. Starting with a large interval L_n within which x^* lies

www.EngineeringBooksPDF.com

it first estimates the number of trials which need to be made, n, to reduce L_n to a specified interval size ϵ within which x^* lies. The ratio L_n/ϵ gives a number usually much greater than 1. The Fibonacci sequence 1, 1, 2, 3, 5, 8, 13, 21, . . . is examined to find that number in the sequence which is equal to or just greater than L_n/ϵ. Thus if $L_n/\epsilon = 10.66$, for example, the Fibonacci number selected would be 13 which is the seventh number in the sequence. The first number, however, is defined as L_0. The seventh number will therefore be L_6, thus $n = 6$. Six further trials are, therefore, needed to find x^* within ϵ starting with an interval of 10.66ϵ. The complete interval L_n is made proportional to the seventh Fibonacci number, 13, that is, thirteen equal sub-intervals are defined each of size

$$L' = \frac{L_n}{13} = \frac{10.66\epsilon}{13} = 0.82\epsilon \tag{7.8}$$

The first two trials of the six are now placed at spacings within L_n proportional to the two previous Fibonacci numbers. In this example the two Fibonacci numbers which precede 13 are 5 and 8; thus trials are placed at

$$x^1 = 5L' = 4.10\epsilon$$
$$x^2 = 8L' = 6.56\epsilon$$

Note that these trials are symmetrical within L_n. The results of these trials determine which of the two equal end intervals in L_n is to be eliminated. Suppose that $f(x^1) > f(x^2)$ as shown on figure 7.5. Then the left-hand interval (x^0, x^1) of length $5L'$ would be eliminated leaving an interval of length $8L'$ within which x^* lies. There is already a trial (x^2) in this interval at a distance $(8 - 5)\,L' = 3L'$ from x^1. The next trial, x^3, is therefore placed symmetrically within the reduced interval at a distance $3L'$ from the right-hand end. The situation now is that the reduced interval is of length $8L'$ and has trials x^2 and x^3 within it at spacings $3L'$ from each end. The function values at x^2 and x^3 are evaluated. Suppose this time that $f(x^3) > f(x^2)$. The right-hand end of the $8L'$ interval can be eliminated leaving a new reduced interval of length $(8 - 3)\,L' = 5L'$ between trials x^1 and x^3. Furthermore there is a trial within the interval at x^2 a distance $3L'$ from the left-hand end. The next trial, x_4^4, is placed symmetrically at a distance $3L'$ from the right-hand end of the interval. Further elimination takes place as a result of this new trial. Figure 7.5 shows $f(x^4) < f(x^2)$; thus the interval $(x^2, x^3) = 2L'$ can be eliminated. The reduced interval is now of length $3L'$ between trials x^1 and x^2 and there is an internal trial at x^4, distance L' from the right-hand end of the reduced interval. The fifth trial x^5 is placed symmetrically with x^4, that is, at a spacing L' from the left-hand end of the interval. In the diagram $f(x^5) < f(x^4)$, the interval (x^4, x^2) is thus eliminated. The reduced interval is now of length $2L'$ between x^1 and x^4 and has a trial x^5 in its centre. The final trial x^6 is placed as close to x^5 as possible. Figure 7.5 shows $f(x^6) < f(x^5)$ eliminating the interval (x^1, x^5) and locating x^* somewhere within the interval (x^5, x^4) of length L'. Numerically the size of L' is

www.EngineeringBooksPDF.com

given by equation 7.8 as 0.82ϵ. Thus the minimum has been located to within an interval of slightly less than ϵ after six trials.

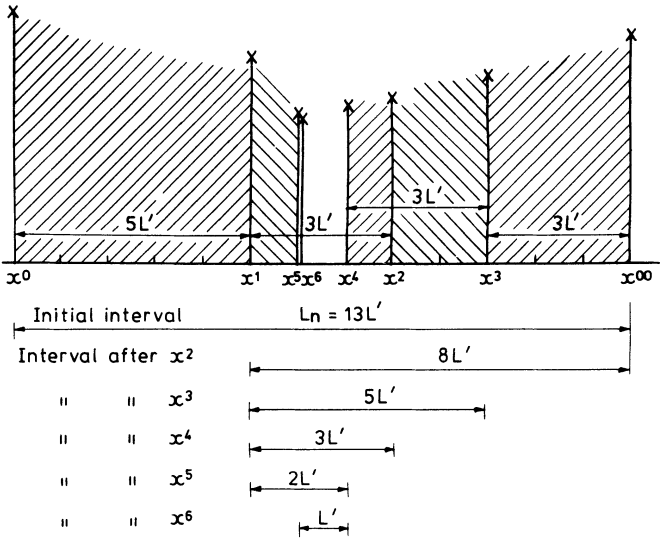


Figure 7.5 Operation of the Fibonacci search method

The above example shows that although the Fibonacci method is superficially a complicated one it is actually easy to apply. The essence of it is that the number of trials needed, n is first found by examining the ratio of the initial interval to the desired final interval. The first two trials are then placed at spacings proportional to the two Fibonacci numbers below L_n/ϵ. The remainder of the search is then automatic. After each elimination the next trial is always placed symmetrically to the existing trial in the reduced interval unless it is the last trial in which case it is placed as close as possible to the existing trial in the interval. The method may be easily programmed for computer use. Other methods have been proposed and used for locating the minimum which are slightly easier to understand and program than the Fibonacci method but they all require more trials to reduce a given interval L to ϵ. No mention has yet been made of the unlikely occurrence of two internal trials x^i and x^j yielding exactly equal function values, namely, $f(x^i) = f(x^j)$. This is unlikely for a real-valued function f because equality must strictly exist to an infinite number of decimal places. It does, however, sometimes arise within the limits of computer accuracy that two trials are equal. For a unimodal function this means that the minimum must lie between x^i and x^j and *two* end intervals may, therefore, be eliminated instead of one. Theoretically this reduces the interval much more quickly than the normal case of unequal trial outcomes

www.EngineeringBooksPDF.com

and should be exploited. Practically, however, it complicates the method as it interrupts the natural placing of subsequent trials and is best not exploited. A simple rule can be incorporated in the method so that, if trials are exactly equal, one of the intervals (say the left-hand one) is eliminated. The use of the Fibonacci sequence in bracketing the minimum was mentioned in section 7.2.1.2. The advantage of this is that once the minimum is bracketed the internal trial (last-but-one) will be correctly placed within the bracketed interval to start the Fibonacci search method to reduce the interval. The search then starts by placing the next trial symmetrically with respect to the existing one. The bracketed interval can then be reduced to an interval of size s, the initial step length, using the same number of trials minus two as was needed to bracket the minimum.

The Fibonacci method is the most efficient and rigorous method for finding the minimum of a unimodal function of one variable if function values only can be evaluated. If derivative values of the function can be found other methods described later are more efficient. If the function is not unimodal no firm statement can be made about the efficiency or rigour of the Fibonacci method or indeed of any other method. The great advantage of the Fibonacci method is its robustness; it will always find a local minimum if presented with three trials describing a locally unimodal region of the function. It assumes nothing more than it calculates about a function and, for this reason, it can sometimes be rather slow. If some assumptions about the function $f(x)$ can be made other methods described below can be more efficient than the Fibonacci method although they achieve this extra efficiency at the expense of rigour and robustness.

7.2.1.5 A Fibonacci Search Example

Find a minimum of the function

$$f = x^4 - 4x^3 - 6x^2 - 16x + 4$$

within an interval of length 0.1 starting from the point $x = 0$.

The minimum must first be bracketed. A basic step-length of $s = 0.1$ is selected equal to the final desired interval length and is subsequently increased according to the Fibonacci sequence. Trials are made sequentially as shown in table 7.1 until a function increase is recorded.

Table 7.1 shows the minimum of $f(x)$ to be bracketed by trials 7 and 9. The length of this interval is equal to the sum of the step lengths contained therein, that is, (13 + 21) x 0.1. Trial 8 is correctly placed within this interval for reduction of the interval by a Fibonacci search. Table 7.2 lists the new trials made in reducing this interval of length 3.4 to one of length 0.1 within which the minimum lies. As nine trials were used to bracket the minimum a further seven (9 - 2) will be needed. The first three columns list the existing trials at the left-hand end of the interval (low +ve x), within the interval, and at the right-hand end of the interval. The fourth column lists the new trial.

www.EngineeringBooksPDF.com

Table 7.1 Bracketing the minimum by a Fibonacci sequence

Trial no.	Position (x)	Step-length (factored by 0.1)	$f(x)$
1	0	–	4.0000
2	0.1	1	2.3361
3	0.2	1	0.5296
4	0.4	2	- 3.5904
5	0.7	3	- 11.2719
6	1.2	5	- 28.6874
7	2.0	8	- 68.0000
8	3.3	13	-139.2959
9	5.4	21	- 36.9104

Table 7.2 Interval reduction by a Fibonacci sequence

	LH Trial	Internal Trial	RH Trial	New Trial
Trial no.	7	8	9	10
Spacing	–	13 x 0.1 from LH	–	13 x 0.1 from RH
x	2.0	3.3	5.4	4.1
$f(x)$	-68.0000	-139.2959	-36.9104	-155.5679
Trial no.	8	10	9	11
Spacing	–	8 x 0.1 from LH	–	8 x 0.1 from RH
x	3.3	4.1	5.4	4.6
$f(x)$	-139.2959	-155.5679	-36.9104	-138.1584
Trial no.	8	10	11	12
Spacing	–	8 x 0.1 from LH	–	8 x 0.1 from RH
x	3.3	4.1	4.6	3.8
$f(x)$	-139.2959	-155.5679	-138.1584	-154.4144
Trial no.	12	10	11	13
Spacing	–	3 x 0.1 from LH	–	3 x 0.1 from RH
x	3.8	4.1	4.6	4.3
$f(x)$	-154.4144	-155.5679	-138.1584	-151.8879
Trial no.	12	10	13	14
Spacing	–	3 x 0.1 from LH	–	3 x 0.1 from RH
x	3.8	4.1	4.3	4.0
$f(x)$	-154.4144	-155.5679	-151.8879	-156.0000
Trial no.	12	14	10	15
Spacing	–	2 x 0.1 from LH	–	2 x 0.1 from RH
x	3.8	4.0	4.1	3.9
$f(x)$	-154.4144	-156.0000	-155.5679	-155.5919
Trial no.	15	14	10	
Spacing	–	central	–	
x	3.9	4.0	4.1	
$f(x)$	-155.5919	-156.0000	-155.5679	

www.EngineeringBooksPDF.com

The last set of results in table 7.2 shows that trial 14 is centrally spaced between trials 15 and 10, the whole interval lying between $x = 3.9$ and $x = 4.1$. The final trial, 16, should be placed close to trial 14 to determine within which interval of length 0.1 the minimum of f lies. Selecting $x = 4.0001$ the value of $f(x)$ is -155.9999 showing that x^* must lie between 3.9 and 4.0001. This is the solution of the problem: $3.9 \leqslant x^* \leqslant 4.0001$ and $f^*(x^*) \leqslant -156.0000$. In this particular example, however, had the final trial been placed at $x = 3.9999$ instead of $x = 4.0001$ an identical value of $f(x) = -155.9999$ would have been obtained. This implies the problem solution to be $3.9999 \leqslant x^* \leqslant 4.1, f^*(x^*) \leqslant -156.0000$. The reason for this rather odd result is that the function $f(x)$ has its minimum at $x = 4$ and trial 14 is located exactly at the minimising value of x^*. This is a purely fortuitous occurrence which cannot be fully exploited by the Fibonacci search method.

7.2.1.6 Curve-fitting Methods

Suppose a function $f(x)$ of a single variable has been evaluated at three points x^1, x^2 and x^3 and values of $f(x^1), f(x^2)$ and $f(x^3)$ are known. A general quadratic curve

$$\hat{f}(x) = ax^2 + bx + c \tag{7.9}$$

may be fitted so that it passes through the three known points of $f(x)$. The three unknown coefficients a, b and c are found by solving the three simultaneous equations

$$\left.\begin{aligned} f(x^1) &= a(x^1)^2 + b(x^1) + c \\ f(x^2) &= a(x^2)^2 + b(x^2) + c \\ f(x^3) &= a(x^3)^2 + b(x^3) + c \end{aligned}\right\} \tag{7.10}$$

Instead of trying to search numerically for the minimum of $f(x)$ a minimum of the fitted function $\hat{f}(x)$ can be found very easily. If $\hat{f}(x)$ is a reasonable approximation to the shape of $f(x)$, the minimising value of x calculated from $\hat{f}(x)$ will be close to the true x^* which minimises $f(x)$.

Conditions for a minimum of $\hat{f}(x)$ are that its first derivative is zero and its second derivative is positive. Thus

$$\frac{d\hat{f}}{dx} = 2ax + b = 0 \tag{7.11a}$$

$$\frac{d^2\hat{f}}{dx^2} = 2a \tag{7.11b}$$

If a is positive equation 7.11b indicates a minimum which, from equation 7.11a occurs at

$$\hat{x^*} = -b/2a \tag{7.12}$$

www.EngineeringBooksPDF.com

This value of $x^{\hat{*}}$ minimises $\hat{f}(x)$, the approximating quadratic function, and is therefore a good position at which $f(x)$ should next be evaluated. Trial x^4 is therefore made at $x = -b/2a$ and $f(x^4)$ is evaluated. One of the three original trials can now be discarded and replaced by the new trial at x^4. A new quadratic curve of the form of equation 7.9 can be fitted to these three points, new values of a, b and c calculated as before and the minimising point $x^{\hat{*}}$ of this new quadratic found from equation 7.12. This gives a new trial value, x^5, at which the actual function $f(x)$ is next evaluated. This process of successive quadratic fits continues until the sequence of new trial points converges on a small region which must contain the minimum of $f(x)$.

If used wisely this quadratic fitting method can be extremely quick and efficient at locating a minimum and far more efficient than Fibonacci. It is, however, an approximate method based upon the assumption that $\hat{f}(x)$ can represent $f(x)$. If used indiscriminately where this assumption cannot be justified it can often fail to converge and may sometimes diverge. Fortunately, some general rules of thumb can be derived for determining when its use is justified. These will now be examined.

It is possible to show that if the three trials necessary to fit a quadratic are such that the function value at the internal trial point is less than the function values at the two end points then the coefficient a will always be positive and, furthermore, the minimum of the quadratic will be located within the interval of the two end trials. These conditions always obtain if the bracketing method of section 7.2.1.2 is used. Figure 7.6a shows this. The three trials at x^1, x^2 and x^3 suggest that $f(x)$ has a minimum within the interval (x^1, x^3). The dotted quadratic curve shown also has an internal minimum. In this situation where the quadratic function *interpolates* within an interval, the quadratic fitting method, while not entirely foolproof, generally converges very rapidly and is recommended. Figure 7.6b shows three trials that appear to describe a function having a minimum outside the interval they cover. The quadratic approximation also has a minimum well outside the interval. In this situation where the quadratic *extrapolates*, particularly if the amount of extrapolation is very large, the quadratic fitting method is likely to converge slowly if it converges at all. Consequently any extrapolation should be treated with suspicion unless it is small and succeeding trials should return to interpolation. Figure 7.6c demonstrates another practical precaution that can improve the convergence of the method. Three trials have been made at x^1, x^2 and x^3 and the dotted quadratic fitted. The minimum has been located at x^4 and a new trial made here. The actual value of $f(x^4)$ is less than that of $\hat{f}(x^4)$. It is now necessary to discard one of the first three trials and replace it with the new trial at x^4 to fit a new quadratic. Which one should be discarded? A logical choice might be to discard the trial at x^3 since it is obviously the furthest point from the latest predicted position of the optimum at x^4. x^3, however, should not be discarded because the next quadratic would be fitted through the trials at x^1, x^2 and x^4 and must inevitably extrapolate to locate a minimum. Instead, the trial at x^1 should be discarded

www.EngineeringBooksPDF.com

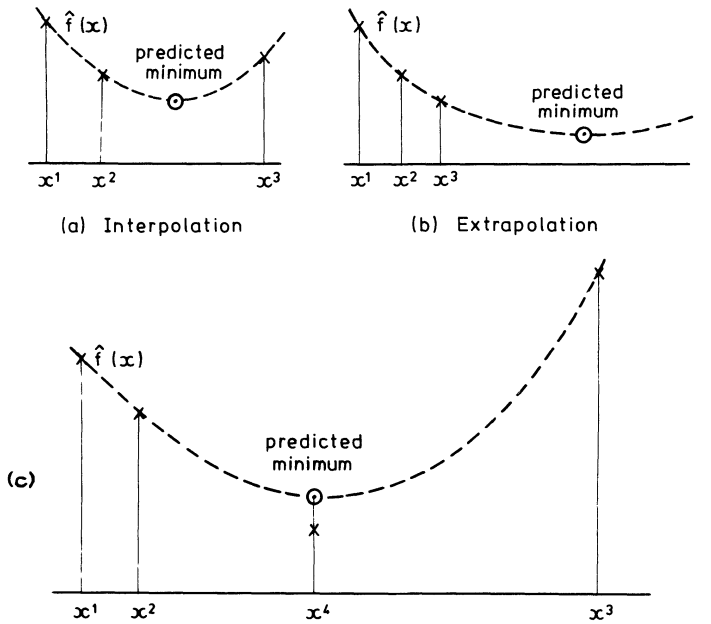


Figure 7.6 Fitting a quadratic function $\hat{f}(x)$ to trials

and an interpolating quadratic fitted through the trials at x^2, x^4 and x^3. This precaution is easy to implement computationally and increases the chance and rate of convergence. It is quite possible for the quadratic function $\hat{f}(x)$ to display a maximum rather than a minimum. As equation 7.11b shows, this would be indicated by a negative value of a. Naturally any quadratic-fitting procedure should include a check upon the positivity of a if a minimum is sought.

The success and efficiency of quadratic-fitting depends upon how well the quadratic approximates to $f(x)$. A higher-order polynomial such as a cubic equation is likely to approximate to a general function better than a simple quadratic. It is quite possible to use a cubic-fitting method although there are some added difficulties. The general cubic function is

$$\hat{f}(x) = ax^3 + bx^2 + cx + d \tag{7.13}$$

with four coefficients a, b, c and d. Four trials instead of three as before are thus needed to determine these cofficients and calculation of the coefficients will be lengthier. The cubic function 7.13 will have a minimum at

$$\hat{x^*} = \frac{-b + \sqrt{(b^2 - 3ac)}}{3a}$$

provided that

$$b^2 > 3ac \tag{7.14}$$

www.EngineeringBooksPDF.com

As with quadratic-fitting the cubic approximation is best used when it interpolates within an interval rather than when it extrapolates. An internal minimum will be estimated if the function values at both internal trial points are each less than those at the nearest end trial of the interval. One particular difficulty experienced with cubics is that sometimes the coefficient a can become very small particularly near the optimum. As a occurs in the denominator of $x^{\hat{*}}$ in equation 7.14 this tends to cause $x^{\hat{*}}$ to be inaccurately estimated. There are ways of overcoming this but they require extra computational effort. Generally cubic-fitting is a more complicated and trouble-prone process than quadratic-fitting but is slightly more efficient in operation.

Curve-fitting methods used in conjunction with the bracketing methods of section 7.2.1.2 are very popular and efficient. Generally they considerably outperform more rigorous methods such as that of Fibonacci. It must be remembered, however, that they do sacrifice some rigour in favour of approximations which may not be valid. On really awkwardly shaped or discontinuous functions they can fail to converge when the more pedestrian Fibonacci method locates a minimum without difficulty.

7.2.2 Problems with More than One Variable

The grid search and random search methods described for a function of a single variable $f(x)$ in section 7.2.1.1 can be directly applied to functions of several variables $f(\boldsymbol{x})$ although with considerably reduced efficiency as has been shown. The sequential search methods for single variable functions do not directly extend to more than one variable. They were devised for minimisation along a line (the x-co-ordinate). Multi-variable functions are represented as surfaces rather than lines. (Strictly only a function of two variables can be represented as a surface, functions of more than two variables are described by *hypersurfaces*). Since methods for minimising along a line are efficient, many methods available for minimising functions of several variables solve the problem as a sequence of line minimisations. Section 7.2.2.1 examines such methods for problems in which function values only can be evaluated, that is, zeroth-order methods for multi-variable functions.

7.2.2.1 Sequential Line-minimisation Methods

Given the many possible methods of minimising along a line easily and reliably it is not surprising that multi-variable functions $f(\boldsymbol{x})$ are usually minimised by a sequence of line minimisations. A starting point is first chosen consisting of values for the variables x_i, $i = 1, \ldots, N$ which comprise $\boldsymbol{x}$. $(N - 1)$ of the variable values are selected to remain constant while the value of one variable, x_j, is allowed to vary. A minimum of $f(\boldsymbol{x})$ with respect to x_j is then sought by any of the one-variable minimisation methods. Having found this minimum another variable, x_k is chosen to be varied while all remaining variables including x_j are

www.EngineeringBooksPDF.com

held constant at their present values. $f(\boldsymbol{x})$ is minimised over x_k, another line minimisation. This process of allowing one variable at a time to vary and minimising $f(\boldsymbol{x})$ with respect to it continues until the value of $f(\boldsymbol{x})$ cannot be reduced further by altering any of the variables. This then represents the minimum of $f(\boldsymbol{x})$ and the solution of the problem.

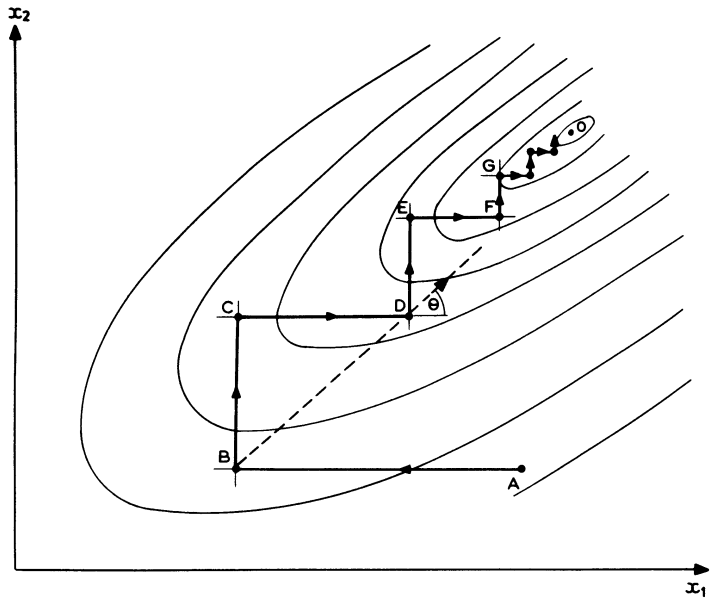


Figure 7.7 Sequential line minimisations to minimise $f(x_1, x_2)$

Figure 7.7 depicts this sequential method applied to a function of two variables. Starting from point A, x_2 is held constant and x_1 allowed to vary. A minimum of $f(x_1, x_2)$ is sought along a line through A parallel to the x_1 axis and is found at B. x_1 is then held constant and x_2 allowed to alter. A minimum is sought along a line through B parallel to the x_2 axis and is found at C. From C another line minimisation is performed over variable x_1 again holding x_2 constant. D is the minimum along this line. Subsequent line searches over one variable at a time are DE, EF, FG, etc., and they clearly are zigzagging towards the minimum of $f(x_1, x_2)$ at point O.

Several insights may be gleaned from the example of figure 7.7. Firstly, at the starting point A the choice of which variable to alter initially is purely arbitrary because at this starting point no information about the shape of $f(\boldsymbol{x})$ is available. To alter variable x_1 first turned out to be a very bad choice; $f(\boldsymbol{x})$ actually reduces in a direction *away from the minimum.* Had variable x_2 been chosen to vary at A a line search over x_2 would have found a line minimum closer to the true minimum of $f(\boldsymbol{x})$. Thus the performance of the method can be adversely affected by the purely arbitrary choice of the order in which variables

www.EngineeringBooksPDF.com

are altered. In a problem with many variables this difficulty assumes an even greater relevance than is shown in this example. Secondly, the zigzagging nature of the search is not very efficient or desirable. The minimum at O is not approached cleanly but by a very large number of line minimisations each shorter than the last. If O must be located to within a very tight tolerance it is clear that many line searches will be required and the time required for the complete set of searches may be very long. A third insight, related to the second, may be gained by realising that the zigzagging is partially caused by the shape of the function $f(\boldsymbol{x})$ in relation to its co-ordinate directions. The contours of $f(\boldsymbol{x})$ shown describe a long narrow bowl-shaped valley the long major axis of which is inclined at an angle of about 40° to the x_1 axis. If one of the axes x_1 or x_2 had coincided with the long axis of the valley a line minimisation along that axis would have located the minimum at O cleanly without zigzagging. In the example of figure 7.7 the choice of the co-ordinate directions x_1 and x_2 as orthogonal directions in which to perform line searches was, therefore, an unfortunate choice. This raises two questions: 'Can a better set of directions be found?' 'How may the search along them be carried out?'. The answer to the first of these questions can be seen in figure 7.7. The first line minimisation establishes point B the first line minimum. Point A is merely an arbitrary starting point not a line minimum. From point B two line searches are carried out sequentially establishing point D. It is clear that the line joining B to D lies in a direction much closer to the axis of the valley than either of the co-ordinate axes. From point D it would be much more profitable to conduct the next line search in the dotted direction shown (BD extended) rather than in the co-ordinate direction DE. A line minimum much closer to the true minimum of $f(\boldsymbol{x})$ at O would be found. This direction is often called the *pattern direction.* The pattern direction is the direction formed by joining the first and last points of any complete cycle of N line searches in an N-variable problem.

Line minimisation along a direction which is not a co-ordinate direction turns out to be very simple. In the case of figure 7.7 a new variable λ is defined to represent distance along the dotted direction from D. Point D represents $\lambda = 0$ and the minimisation is directed at finding that value of λ which minimises $f(x_1, x_2)$ along the dotted direction from D. Geometrically, any value of λ enables the values of x_1 and x_2 corresponding to that point to be found from

$$\left.\begin{aligned} x_1 &= x_1^{\mathrm{D}} + \lambda \cos\theta \\ x_2 &= x_2^{\mathrm{D}} + \lambda \sin\theta \\ \tan\theta &= \left(\frac{x_2^{\mathrm{D}} - x_2^{\mathrm{B}}}{x_1^{\mathrm{D}} - x_1^{\mathrm{B}}}\right) \end{aligned}\right\} \qquad (7.15)$$

Thus for each trial value of λ corresponding values of x_1 and x_2 can be found from equations 7.15 and $f(x_1, x_2)$ may be evaluated. For problems in more

www.EngineeringBooksPDF.com

than two variables equations 7.15 take the form of equation 7.16

$$x_i = x_i^D + \frac{\lambda(x_i^D - x_i^B)}{\left[\sum_{j=1}^{N} (x_j^D - x_j^B)^2\right]^{1/2}} \qquad i = 1, \ldots, N \tag{7.16}$$

in which B and D are the start and finish points of a complete cycle of N line minimisations in the N co-ordinate directions.

Having carried out a minimisation along the pattern direction, several possibilities exist for the choice of directions for further line searches. One possibility is to return to the original co-ordinate directions, carry out another cycle of N sequential line minimisations and use the start and finish points of this cycle to establish a new pattern direction. Another possibility, which usually turns out to be more efficient, is to use the pattern direction and the search along it as the first of a new cycle of N line minimisations with the remaining $(N - 1)$ directions being chosen orthogonal to the pattern direction. This is equivalent to rotating the co-ordinate system so that one co-ordinate axis lines up with the pattern direction. Even better possibilities exist for a choice of search directions but in order to understand these a little better it is necessary to examine why the pattern direction is a good one for it is not mere chance which makes this so.

Represented graphically the contours of a two-variable *quadratic* function $f(x_1, x_2)$ are concentric ellipses with a function minimum at the centre of the ellipses. Figure 7.7, therefore, shows a function which is quadratic (or very nearly so). Consider any pair of parallel lines in figure 7.7 which intersect the ellipses, for example, AB and CD. B and D are points at which the two parallel lines are tangential to the concentric ellipses. It is fairly simple to prove geometrically that the line BD, the pattern direction on figure 7.7, joining the contact points of any pair of parallel tangents to concentric ellipses passes through the centre of the ellipses. The direction BD is then said to be *conjugate* to directions AB and CD. In figure 7.7 direction BD should pass through O, as should direction CE which is conjugate to BC and DE. Thus in a two-variable quadratic problem such as figure 7.7 the pattern direction BD is a conjugate direction and must pass through the function minimum. It is this conjugacy property of the pattern direction which makes it an excellent direction for searching along.

7.2.2.2 Conjugate Direction Methods

Figure 7.8 shows how conjugacy can be used in a search procedure. The contours of a two-variable function $f(x_1, x_2)$ have been deliberately chosen to be non-elliptical to demonstrate how the method works on non-quadratic functions. A quadratic function would have required far fewer searches. Point A is an

www.EngineeringBooksPDF.com

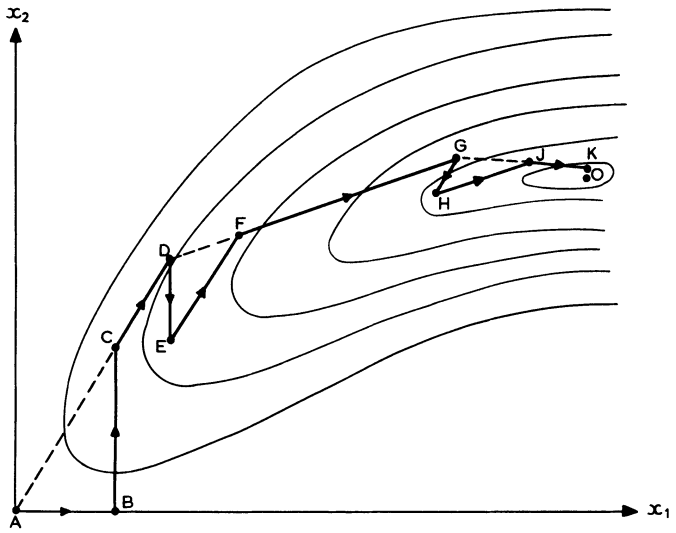


Figure 7.8 Powell's method

arbitrarily chosen starting point and AB and BC are successive line minimisations carried out in the x_1 and x_2 directions. These two searches in a two-variable problem define a pattern direction AC. The next line minimisation is made in this pattern direction from C locating a line minimum at D. At point D a new cycle of searches begins but instead of using the orthogonal directions which were used in the first cycle (x_1 and x_2) one of those directions is discarded (in this example x_1) and is replaced by the pattern direction. Thus for the search cycle from D the next line search is made in the x_2 direction locating point E, followed by a search EF in a direction parallel to the pattern direction CD. Thus F is located. Note that the method has created a pair of parallel tangents CD and EF with contact points D and F. The direction DF is, therefore, a conjugate direction and hence a potentially good search direction. Had the function been quadratic DF would have passed through the function minimum. The method continues with a search in the direction DF from F, locating a line minimum at G. At G another cycle begins. This time the other original direction x_2 is discarded and replaced by the direction FG. Thus the first search GH is made parallel to the first pattern direction (CD) and locates point H. The second search is parallel to the second pattern direction (the conjugate direction FG) and locates point J. Note again that parallel tangents FG and HJ have been created with the pattern direction GJ being conjugate to them. This new conjugate direction GJ is now searched along from J and a line minimum located at K. In the example of figure 7.8, K is close to the true function minimum at O and no further iterations are shown. If further searches from K are required it is usual to return to the original co-ordinate directions x_1 and x_2 and repeat the sequence of searches as before until a solution point is reached.

www.EngineeringBooksPDF.com

The important thing to note about the example of figure 7.8 is how a series of parallel tangents is created so that the conjugate directions defined by their contact points may be used as potentially good search directions. On this non-quadratic example the search follows the changing orientation of the axis of the valley described by the contours extremely well with very little zig-zagging. Thus, although the use of conjugacy can only be proved to be effective on quadratic functions, it can turn out to be very efficient on non-quadratic general functions.

Figure 7.8 is a two-variable example. The method and all the properties of conjugacy may be extended to N-variable problems. It is unfortunate that no graphical representation can be given for higher-dimensional problems because there are several subtle points which cannot otherwise be demonstrated easily. In N dimensions the contours of $f(x)$ are not two-dimensional ellipses but N-dimensional hyperellipsoids; the two parallel tangents become two parallel tangent hyperplanes. In figure 7.8 it was relatively easy to define two parallel tangent directions and search along each for a line minimum. In N dimensions two parallel tangent hyperplanes must be created and a minimum of the function on each hyperplane found. This itself requires a multi-dimensional search and so the N-dimensional problem is more complicated than at first appears from a two-dimensional example. The method described in figure 7.8, is known as *Powell's method* after its originator and may be applied to N-variable problems as follows.

From an arbitrary starting point a sequence of line searches in each of the N co-ordinate directions is carried out followed by a search in the pattern direction. Another cycle of line searches is then carried out with the last search direction being discarded and replaced by a search in the previous pattern direction. From this cycle of N searches a new pattern direction is established and searched along. This process is repeated, each time replacing the last of the original co-ordinate directions by the previous pattern direction. Eventually all the original co-ordinate directions will have been replaced and all the N search directions in the cycle will be conjugate directions. A final pattern direction search is then made which, if the function is quadratic, should locate the minimum. If the function is not quadratic and a minimum is not found the method continues by restoring the N co-ordinate directions and the whole search sequence is repeated until a minimum is found.

This method is perhaps the simplest of those methods which attempt to use conjugacy. Many modifications and improvements have been suggested including some by Powell, the originator. A further investigation of conjugacy is not attempted in this book because of the highly mathematical nature of the proofs associated with it. The specialist texts listed in the bibliography should be consulted for further study. Conjugacy is a very important concept in the development of very efficient numerical optimisation methods. It is possible to prove that an N-variable quadratic function will be minimised after at most $(N^2 + 1)$ line searches in a full conjugate direction method. For a non-quadratic

www.EngineeringBooksPDF.com

function experience has shown that methods such as Powell's using conjugacy are among the most efficient.

7.2.2.3 The Simplex Method

A refreshingly different method for finding the minimum of $f(x)$ using only function values is the *simplex method* shown in figure 7.9. The name simplex method is most unfortunate and misleading because the method has absolutely nothing to do with the simplex method used in linear programming. The method was devised by Spendley, Hext and Himsworth and later developed by Nelder and Mead. A *simplex* is defined as a pattern of $N + 1$ points (for an N-variable function) which do not all lie in the same N-dimensional hyperplane. For a two-variable problem a simplex has three points arranged in a triangle, for a three-variable problem the four points define a tetrahedron.

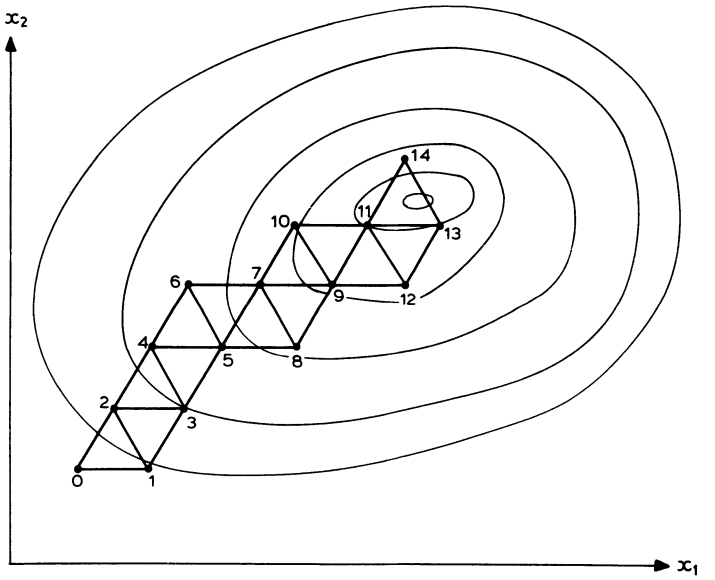


Figure 7.9 The simplex method

Figure 7.9 shows contours of a two-variable function $f(x_1, x_2)$. An initial simplex is chosen with vertices 0, 1 and 2. In this example the simplex is an equilateral triangle although the method to be described does not require the simplex to be a regular shape. Having specified an initial simplex the value of $f(x_1, x_2)$ is calculated at each vertex. The vertex with the highest function value of the three is then 'reflected' in the side joining the other two vertices, that is, in figure 7.9 the initial simplex is toppled over about the side 1-2 so that vertex 0 at which f has the highest value takes up position 3. The function is evaluated at vertex 3

www.EngineeringBooksPDF.com

and now the function values at points 1, 2 and 3 in this second simplex are compared. Point 1 now has the highest value so is reflected in side 2–3 giving a third simplex with vertices 2, 3 and 4. This process continues and the triangular simplex topples its way over the function towards the function minimum. Each new simplex requires only one additional function evaluation at the new vertex for comparison with values at the existing vertices. In the two-variable problem, figure 7.9, the reflection process is simple to understand. In N dimensions there are $N + 1$ vertices and reflection consists of finding the vertex with the highest function value, locating the centroid of the remaining N vertices and establishing the new vertex an equal distance from the centroid on the line joining the highest vertex to the centroid but on the opposite side of the centroid.

In figure 7.9 the reflection process ends with the two simplexes 11, 12, 13 and 11, 14, 13 endlessly reflecting each other. At this stage some new rules can be introduced in order to make further progress. Typically the second-highest vertex might be reflected instead of the highest. It is clear from figure 7.9, however, that near the optimum the size of the simplex is too large to find the minimum accurately. Various rules and procedures have been suggested for shrinking the simplex and for altering its shape. These rules can be invoked once oscillatory or cyclical behaviour occurs and eventually a minimum will be found.

The simplex method is purely a numerical one which assumes nothing about the function and for which no convergence proofs exist. It has the advantages that it is easy to program for computer use, does not require line searches and is suitable for solving the most discontinuous of problems. Experience with it shows it to be a robust method for getting to the region of the minimum although it sometimes fails to find an exact minimum. This need not deter engineering users for whom an exact minimum often has little meaning. The efficiency of the method depends upon the shape, size and orientation of the initial simplex. Nevertheless it is a simple and very useful method.

7.3 FIRST-ORDER METHODS

Section 7.2 examined methods for minimising $f(\boldsymbol{x})$ that use only values of the function. Often values of the first partial derivatives can be calculated in addition to function values. This implies that much more is known about the nature of $f(\boldsymbol{x})$ and so it is to be expected that methods for locating a minimum can be devised which are more efficient than zeroth-order methods. Such is the case. The methods described in this section require that exact values of all first derivatives of $f(\boldsymbol{x}) \equiv f(x_1, x_2, \ldots, x_N)$ can be calculated rapidly. These derivatives are

$$g_j(\boldsymbol{x}) \equiv \frac{\partial f(\boldsymbol{x})}{\partial x_j} \qquad j = 1, \ldots, N$$

An immediate query arises concerning numerical derivatives. It is possible to evaluate the derivative $\mathrm{d}f/\mathrm{d}x$ of a function $f(x)$ of one variable by making two

www.EngineeringBooksPDF.com

trial function evaluations very close together at x and $x + \Delta x$. Then an approximate derivative value is given by

$$\frac{df}{dx} \approx \frac{f(x + \Delta x) - f(x)}{\Delta x} \tag{7.17}$$

The limit of the right-hand side of equation 7.17 as $\Delta x \to \infty$ is the exact value of the first derivative. Are derivatives calculated by numerical approximations such as equation 7.17 suitable for use in first-order methods?

This question cannot be answered with certainty but the balance is perhaps tilted slightly towards 'No'. The difficulties arise in an accuracy context. For the numerical derivative to be accurate Δx must be very small and the function values, therefore, tend to be similar in magnitude. This leads to their difference being contaminated by numerical inexactitudes in computation and the result is an inaccurate derivative value. If Δx is made larger so as to increase the difference in function values and to reduce computational inaccuracy, the right-hand side of equation 7.17 no longer approximates a derivative. More accurate approximate derivatives can be obtained from the central difference formula

$$\frac{df}{dx} \approx \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x} \tag{7.18}$$

but this requires an extra trial and is thus less efficient. Various suggestions have been made for improving the accuracy and efficiency of numerical derivative evaluations but none is really recommended.

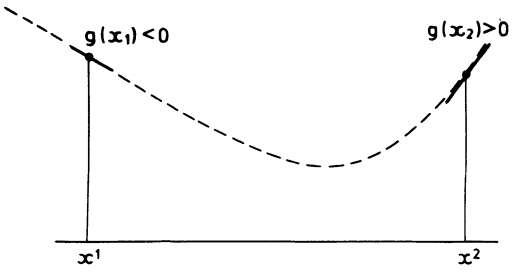
Also, in order to estimate the N first derivative values numerically at a point in an N-variable problem $N + 1$ function evaluations must be made. None of these $N + 1$ trials directly reduces the value of $f(\boldsymbol{x})$ whereas in a good zeroth-order method such as that of Powell considerable progress can be made towards the minimum in $N + 1$ trials. The answer to the question of numerical derivatives is, therefore, a delicate balance involving considerations of accuracy and whether trials should be used to evaluate derivative values or to reduce the function value. A good zeroth-order method such as Powell's method has the advantage of robustness and in engineering problems this is very valuable. A robust method which usually provides a solution is preferable to a temperamental method which can fail to produce a result because of computer inaccuracies and round-off errors. The balance is therefore perhaps tilted against the use of numerically approximated derivative values.

7.3.1 One-variable Problems. Line Minimisation

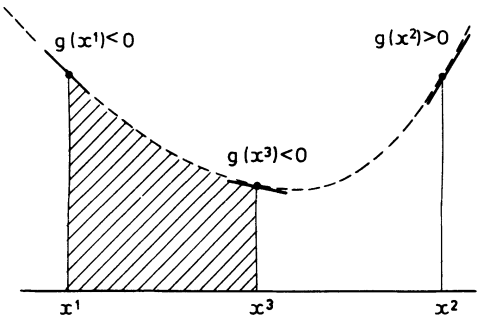
As with zeroth-order methods it is instructive to begin by examining line minimisation problems and then use these methods to solve multiple-variable problems. For a single-variable function $f(x)$ the first derivative $df/dx \equiv g(x)$ represents the slope or gradient of the function. A negative value of $g(x)$ means that the function $f(x)$ decreases in values as x increases; a positive value means an increase in $f(x)$

www.EngineeringBooksPDF.com

with x increasing. As figure 7.10a shows, a minimum of $f(x)$ can be bracketed between two trials x^1 and x^2 where $x^2 > x^1$ if $g(x^1) < 0$ and $g(x^2) > 0$. Bracketing methods very similar to those used for zeroth-order problems can be used. A starting value of x is chosen, x^0, and a step-length, s. The gradient $g(x^0)$ is calculated. If $g(x^0) < 0$ a new trial is placed at $x^1 = x^0 + s$. If $g(x^0) > 0$ the trial x^1 is placed at $x^0 - s$. At the new trial point $g(x^1)$ is calculated and, if it is of the same sign as the previous trial, further trials are placed sequentially at $x^2, x^3, \ldots$, in the same co-ordinate direction (x increasing or decreasing) as before. If at some trial value x^k the gradients $g(x^k)$ and $g(x^{k-1})$ have different signs, the bracketing method is terminated. A minimum of $f(x)$ then lies somewhere between x^{k-1} and x^k. Note that in this method function values need not be calculated. Only gradient values are used. The final two trials bracket the minimum whereas in the similar zeroth-order method the final three trials were needed to bracket the minimum. Variable step-lengths can be used as before. As will be shown shortly the Fibonacci series is not optimal for line searches using gradients. Doubling the step-length after each trial is preferred. Bracketing using gradient values is marginally more efficient than using zeroth-order function values but just as many failure possibilities exist. A minimum may be stepped over just as easily.



(a) Bracketing a minimum.



(b) Interval reduction to locate a minimum

Figure 7.10 Line minimisations using gradient values

www.EngineeringBooksPDF.com

Having located an interval within which x^* lies, as seen in figure 7.10a, it is necessary to reduce this interval to find x^* more accurately. The best way of doing this using only gradient values is much simpler than the Fibonacci method for function values. Interval reduction methods using function values require two trials within an initial interval in order to identify which portion may be eliminated. If gradient values are available only one internal trial is needed to eliminate that portion of the interval within which x^* does not lie. Figure 7.10b shows an initial interval and a trial at the mid-point of the interval. The gradients at each end of the interval are necessarily of opposite sign. The sign of $g(x^3)$ must, therefore, correspond to that of one of the end points which in this example is $g(x^1)$. The minimum must lie between x^3 and x^2 so the interval (x^1, x^3) is eliminated. Clearly a minimax policy of making the best out of the worst possible outcome requires that the internal trial x^3 should always be placed in the middle of any existing interval (x^1, x^2). Computationally this makes for a very simple algorithm and is more efficient in all ways than the Fibonacci method for function values. Again only gradient values are required. Even though function values exist they are not used. Furthermore there is no way in which they can be used together with gradient values to produce a more efficient method. Occasionally in the bracketing or interval reduction a trial may be placed at which $g(x)$ is arbitrarily small or even zero. If the function $f(x)$ is known to be convex this must represent the minimum of the function. If, however, $f(x)$ is not known to be convex it may be multi-modal and the point may possibly represent a local maximum or a point of inflexion. Unless $f(x)$ is known to be unimodal a trial unexpectedly resulting in $g(x) = 0$ should be treated with suspicion. New trials placed each side of it will show whether or not it is a minimum.

Curve-fitting methods are simple to use with gradient values. The quadratic equation, $\hat{f}(x)$, given by equation 7.9 requires three pieces of information in order to find values for the coefficients. Its derivative is

$$\frac{d\hat{f}}{dx} = 2ax + b$$

so two gradient values plus one function value or one gradient value plus two function values could be used to find a, b and c. Various combinations of gradient and function values can be used to determine values of the coefficients a, b, c and d in the cubic function of equation 7.13. There are no special advantages to be gained by using gradient values in curve-fitting methods. Again such methods should be used to interpolate a minimum rather than to extrapolate. Both the cubic and the quadratic will interpolate within an interval if the gradients at each end are of opposite sign.

7.3.2 Problems with More than One Variable

Because gradient-based line searches are marginally more efficient than function-based line searches all the zeroth-order methods using a sequence of line

www.EngineeringBooksPDF.com

minimisations may be improved in efficiency by substituting gradient-based line searches. Other minor improvements may be made by using gradient information. For example, at the minimising point of a line search in an N-variable problem it is logical to choose as the next variable to be searched over, that one corresponding to the largest value of the gradients $\partial f/\partial x_1, \partial f/\partial x_2, \ldots,$ at that point. Clearly, the greater the gradient the greater the expected reduction in f will be over a unit distance. Figure 7.11 shows contours of $f(x_1, x_2)$ and a starting point A. If $|(\partial f/\partial x_1)_A| > |(\partial f/\partial x_2)_A|$ variable x_1 is chosen as the first variable for a line search, variable x_2 being chosen if $|(\partial f/\partial x_1)_A| < |(\partial f/\partial x_2)_A|$. Figure 7.11 shows, however, that neither of the co-ordinate directions is a particularly good one for a line search. Neither of them will give very large reductions in the values of $f(x_1, x_2)$. A far better direction for a line search would be the direction through B from point A shown on figure 7.11. This search direction is normal to the contour at A and is, therefore, the direction in which the function decreases fastest. It is called the *steepest gradient direction.* This direction can be found in terms of the known first derivatives at point A which is information not readily available to zeroth-order methods. The concept of a steepest gradient is fundamental to first-order methods. It is exploited by many first-order methods and leads to unconstrained minimisation methods which are generally much more efficient than zeroth-order methods.

7.3.2.1 The Steepest Gradient Method

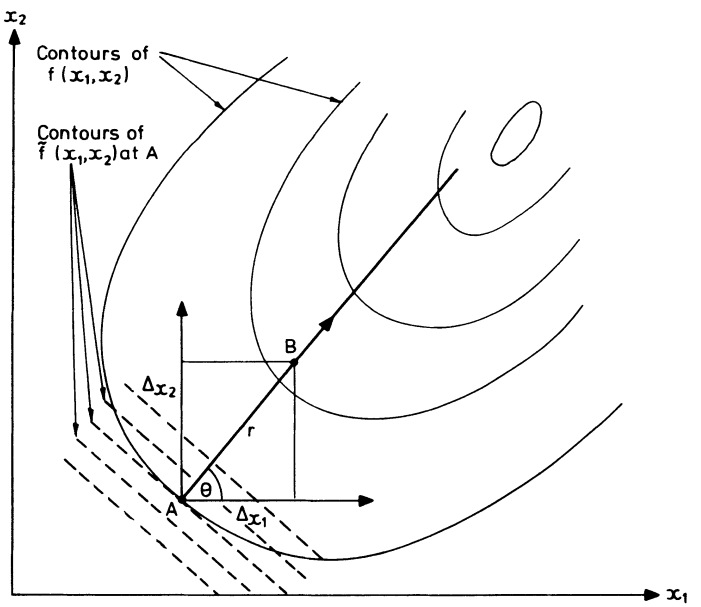


Figure 7.11 The steepest gradient direction

www.EngineeringBooksPDF.com

On figure 7.11 let the angle which the steepest gradient direction in the x_1x_2 plane makes with the x_1 direction be θ. Consider a point B, distance r in the x_1x_2 plane along the steepest gradient direction from A. The co-ordinates of A in the x_1x_2 plane are (x_1^A, x_2^A) and from figure 7.11 the co-ordinates of B must be

$$(x_1^B, x_2^B) = (x_1^A + \Delta x_1, x_2^A + \Delta x_2)$$

Also, from the geometry of figure 7.11 it is evident that

$$\left.\begin{aligned} \Delta x_1 &= r\cos\theta \\ \Delta x_2 &= r\sin\theta \end{aligned}\right\} \tag{7.19}$$

Thus

$$(x_1^B, x_2^B) = (x_1^A + r\cos\theta, x_2^A + r\sin\theta) \tag{7.20}$$

To find the co-ordinates of B the angle θ must be found.

The steepest gradient direction AB is known to be normal to the contour of $f(x_1, x_2)$ at point A in the x_1x_2 plane. More correctly, since the contour is a curved line, the direction AB is normal to the tangent to the contour at A, as shown on figure 7.11. If, at A, the curved nature of the surface of $f(x_1, x_2)$ is replaced by a tangent plane to the surface, $\tilde{f}(x_1, x_2)$, the linear contours of which are shown as broken lines on figure 7.11, the steepest gradient will be directly down the tangent plane normal to all the contours. This is the way in which θ is found. A function $f(x_1, x_2)$ is locally replaced by its tangent plane $\tilde{f}(x_1, x_2)$ and the steepest gradient direction θ is found from the tangent plane equation.

The equation of the tangent plane $\tilde{f}(x_1, x_2)$ to $f(x_1, x_2)$ at point A is given by a Taylor series expansion of $f(x_1, x_2)$ truncated after the linear terms. Thus

$$\tilde{f}(x_1, x_2) = f(x_1^A, x_2^A) + \left(\frac{\partial f}{\partial x_1}\right)_A \Delta x_1 + \left(\frac{\partial f}{\partial x_2}\right)_A \Delta x_2 \tag{7.21}$$

Substitution of Δx_1 and Δx_2 from equation 7.19 into equation 7.21 gives the value of $\tilde{f}$ at point B.

$$\tilde{f}(x_1^B, x_2^B) = f(x_1^A, x_2^A) + \left(\frac{\partial f}{\partial x_1}\right)_A r\cos\theta + \left(\frac{\partial f}{\partial x_2}\right)_A r\sin\theta \tag{7.22}$$

For any specified value of r angle θ must be such that $\tilde{f}(x_1^B, x_2^B)$ is a minimum, that is, point B must be on the direction in which $\tilde{f}$ decreases by the greatest amount. A minimum of $\tilde{f}(x_1^B, x_2^B)$ occurs when its derivative with respect to θ is zero. Thus

$$\frac{d}{d\theta}\ \tilde{f}(x_1^B, x_2^B) = 0 = \left(\frac{\partial f}{\partial x_1}\right)_A (-r\sin\theta) + \left(\frac{\partial f}{\partial x_2}\right)_A (r\cos\theta)$$

Hence

$$\tan\theta = \frac{(\partial f/\partial x_2)_A}{(\partial f/\partial x_1)_A} \tag{7.23}$$

www.EngineeringBooksPDF.com

Equation 7.23 is the equation sought which gives θ, the steepest gradient direction in the x_1x_2 plane, as a function of the known first derivatives at A. This relationship for θ may now be substituted into equations 7.20 and 7.22 to give the co-ordinates of point B and the value of the tangent plane function $\tilde{f}$ at B. Point B is

$$(x_1^B, x_2^B) = (x_1^A + \Delta x_1, x_2^A + \Delta x_2)$$

where

$$\Delta x_1 = \frac{r(\partial f/\partial x_1)_A}{\sqrt{[(\partial f/\partial x_1)_A^2 + (\partial f/\partial x_2)_A^2]}} \tag{7.24}$$

and

$$\Delta x_2 = \frac{r(\partial f/\partial x_2)_A}{\sqrt{[(\partial f/\partial x_1)_A^2 + (\partial f/\partial x_2)_A^2]}}$$

and the value of $\tilde{f}$ at B is

$$\tilde{f}(x_1^B, x_2^B) = f(x_1^A, x_2^A) + r\sqrt{[(\partial f/\partial x_1)_A^2 + (\partial f/\partial x_2)_A^2]} \tag{7.25}$$

Given some starting point A the function value and gradients at A can be evaluated. Selection of a value of r allows Δx_1 and Δx_2 to be calculated thus generating the co-ordinates of a new point B in the steepest gradient direction from A. A new trial can, therefore, be made at B. By choosing different values for r a line minimisation of $f(x_1, x_2)$ can be carried out in the steepest gradient direction.

The steepest gradient direction is the direction in which $f(x_1, x_2)$ increases or decreases the fastest. The tangent plane approximation in equation 7.25 shows that $\tilde{f}(x_1, x_2)$ and hence $f(x_1, x_2)$ will decrease if negative values are chosen for r and increase if positive r are chosen. Also, from equation 7.25 the slope of the linearised function $\tilde{f}(x_1, x_2)$ in the steepest gradient direction is

$$\frac{\partial}{\partial r}\tilde{f}(x_1, x_2) = \sqrt{[(\partial f/\partial x_1)_A^2 + (\partial f/\partial x_2)_A^2]} \tag{7.26}$$

This gradient value can be useful if a curve-fitting method is used to locate a line minimum in the steepest gradient direction.

The above results have been derived for a two-variable function but can be extended to functions of N variables to give the general relationships in equations 7.27 to 7.30.

$$\Delta x_i = \frac{r(\partial f/\partial x_i)_A}{\left[\sum_{j=1}^{N} (\partial f/\partial x_j)_A^2\right]^{1/2}} \quad i = 1, \ldots, N \tag{7.27}$$

www.EngineeringBooksPDF.com

$$x^B = x^A + \Delta x \tag{7.28}$$

$$\tilde{f}(x^B) = f(x^A) + r\left[\sum_{i=1}^{N} (\partial f/\partial x_i)_A^2\right]^{1/2} \tag{7.29}$$

$$\frac{\partial}{\partial r}\tilde{f}(x^B) = \left[\sum_{i=1}^{N} (\partial f/\partial x_i)_A^2\right]^{1/2} \tag{7.30}$$

Examining equations 7.27 and 7.28 a little further it is noted that the denominator of equation 7.27 will have the same constant value for all $i = 1, \ldots, N$. It is, therefore, possible to combine this with the length parameter r, which for minimisation purposes must be negative, and to express equation 7.27 in the form

$$\Delta x_i = c(-\partial f/\partial x_i)_A \qquad i = 1, \ldots, N \tag{7.31}$$

where the positive coefficient $c = -r/[\sum_{j=1}^{N} (\partial f/\partial x_j)_A^2]^{1/2}$. The right-hand side of equation 7.31 has the form of a positive length parameter c times a direction vector $(-\partial f/\partial x_i)_A$. The displacement vector Δx is then of the form

$$\Delta x = c(-g_A) \equiv c\begin{bmatrix} -g_1 \\ -g_2 \\ \cdot \\ \cdot \\ \cdot \\ -g_N \end{bmatrix}_A \tag{7.32}$$

where subscript A denotes values evaluated at point A and g is defined by equation 7.2.

7.3.2.2 A Steepest Gradient Example

Consider minimising the function

$$f = x_1^6 + x_2^6 + 2x_1^2x_3^4 + 4x_2^4x_3^2 + 10x_1^2x_2^2x_3^2$$

by the steepest gradient method using the starting point (1, 1, 1).

Firstly, values of the function and its derivatives are found at the starting point. These are

$$f(1, 1, 1) = 18$$

$$\frac{\partial f}{\partial x_1} = 6x_1^5 + 4x_1x_3^4 + 20x_1x_2^2x_3^2 = 30$$

$$\frac{\partial f}{\partial x_2} = 6x_2^5 + 16x_3^2x_2^3 + 20x_1^2x_2x_3^2 = 42$$

$$\frac{\partial f}{\partial x_3} = 8x_1^2x_3^3 + 8x_2^4x_3 + 20x_1^2x_2^2x_3 = 36$$

www.EngineeringBooksPDF.com

The value of $\partial f/\partial r$ at the starting point is then found from equation 7.30

$$\left[\sum_{i=1}^{3} (\partial f/\partial x_i)^2\right]^{1/2} = \sqrt{(30^2 + 42^2 + 36^2)} = 69.929$$

Using equation 7.27 values for Δx_i, $i = 1, 2, 3$, are

$$\Delta x_1 = r(30/62.929) = 0.47673r$$
$$\Delta x_2 = r(42/62.929) = 0.66742r$$
$$\Delta x_3 = r(36/62.929) = 0.57208r$$

Since $f(x_1, x_2, x_3)$ must be minimised r must be chosen to be negative. Co-ordinates of points on the steepest gradient direction through the starting point are then parametrically dependent upon r and are

$$(1 + 0.47673r, 1 + 0.66742r, 1 + 0.57208r) \tag{7.33}$$

Values of r may now be chosen and new trials made corresponding to each value of r with the object of locating a line minimum. Any method may be used for this, for example, Fibonacci, curve-fitting, etc. Suppose a quadratic curve-fitting method is to be used. The quadratic equation in r is

$$\hat{f}(r) = ar^2 + br + c \tag{7.34}$$

Two pieces of information are already known. At the starting point (1, 1, 1) where $r = 0$ it is known that the function value is 18 and the gradient of the steepest slope is 62.929. In order to find the coefficients a, b and c in the quadratic one more piece of information must be generated. One more trial will achieve this. Select any negative value of r, say $r = -1$. From equation 7.33 the co-ordinates corresponding to this point are (0.52327, 0.33258, 0.42792). The value of $f(x_1, x_2, x_3)$ at this point is calculated to be 0.10466. The third piece of information needed to find a, b and c is then that $\hat{f} = 0.10466$ at $r = -1$. The values of a, b and c are then calculated to be

$$a = 45.034;\ b = 62.929;\ c = 18$$

The value of r^* which minimises $\hat{f}(r)$ is given by

$$r^* = -b/2a = -62.929/90.068 = -0.69868$$

Substituting this value of r* back into equation 7.34 gives the minimum value of the fitted quadratic as $\hat{f}^* = -3.9837$. The actual function $f(x_1, x_2, x_3)$ is now evaluated at the same point. Substituting r^* into equation 7.33 gives the x-co-ordinates of the point as (0.66692, 0.53369, 0.60031). Evaluation of $f(x_1, x_2, x_3)$ here yields $f = 0.80010$.

Examining these results it is seen that the predicted (quadratic) function minimum of $\hat{f} = -3.9837$ and the actual function value $f = 0.80010$ are con-

www.EngineeringBooksPDF.com

siderably different. The first quadratic fit was not a particularly accurate approximation to the function $f(x_1, x_2, x_3)$ in the chosen direction. Another quadratic fit may, therefore, be made to locate a better line minimum. This time the gradient information is discarded and the three function values are used to fit a new quadratic

$$\bar{f}(r) = a'r^2 + b'r + c'$$

Values of a', b' and c' are found from the conditions

(1) $\bar{f} = 18$ at $r = 0$
(2) $\bar{f} = 0.80010$ at $r = -0.69868$
(3) $\bar{f} = 0.10466$ at $r = -1$

Values of a', b' and c' are 22.30974, 40.20508 and 18, respectively, and the new estimate of r^* is at $r = -0.90107$. Evaluating $\bar{f}$ and f here yields $\bar{f} = -0.11371$ and $f = 0.21941$. These values are much closer than those resulting from the first quadratic fit. The value of f at the internal, supposedly optimal, point, however, is still higher than at the end point of the interval where $f = 0.10466$. Thus more quadratic fits are suggested to locate the line minimum more closely. Eventually a minimising value of r will be found. This point is then used to calculate derivatives of $f(x_1, x_2, x_3)$ and find a new steepest gradient direction. Thus the search procedure eventually locates the minimum of $f(x_1, x_2, x_3)$.

Often, using quadratic or cubic curve-fitting for line searches along a steepest gradient direction, an accurate iterative search is not carried out. Only one quadratic or cubic is fitted and its minimising point estimated. In the above example this is the point (0.66692, 0.53369, 0.60031) at which the value of f is 0.80010. Instead of fitting more quadratics to locate the minimum more precisely this point is instead used as a starting point for another steepest gradient calculation and search. There are several good reasons for doing this. In the numerical example this point shows a considerable reduction in value of f from 18 to 0.80010. The next quadratic fit, however, only achieves a marginal reduction in f to 0.21941. This supports general numerical experience which shows that using only one quadratic fit per search direction is more efficient than using several iterative quadratic fits. There are good theoretical reasons also. If the function $f(\boldsymbol{x})$ is a quadratic, then the minimising point of a single quadratic fit in the steepest gradient direction can be shown to be an accurate line minimum. Further iterations are unnecessary. Many functions $f(\boldsymbol{x})$ which are non-quadratic functions are approximated very well by quadratics in the region of the optimum in which case again only one quadratic fit per steepest gradient direction is necessary to locate an accurate line minimum. In regions of a general function far from the optimum region it may be argued that it is wasteful to use extra trials to locate a line minimum accurately if that point is nowhere near the true function minimum. The trials are best used in getting closer to the function minimum by means of new steepest gradient searches.

www.EngineeringBooksPDF.com

7.3.2.3 Performance of Steepest Gradient Searches

The solution of a multi-variable minimisation problem by a sequence of steepest gradient line minimisations is generally faster than the solution resulting from minimising along a sequence of co-ordinate directions although perhaps not as fast as might be expected. The reason for this is that zigzagging still occurs with steepest gradient searches. Figure 7.7 depicts this. Although figure 7.7 was constructed to demonstrate a zeroth-order search method each of the directions BC, CD, DE, EF, etc., is in fact a steepest gradient direction. If a line search along a co-ordinate direction locates a line minimum, say B, then that search direction AB must naturally be a tangent to the contour at B. In two dimensions the next variable to be searched over defines a direction BC, normal to AB which is, therefore, normal to the contour tangent at B and so must be the steepest gradient direction at B. Thus for a two-variable problem a sequence of alternating co-ordinate line searches is exactly equivalent to a sequence of steepest gradient searches and their efficiencies are consequently similar. In problems with more than two variables the pattern of directions is not the same and the steepest gradient method is more efficient. The zigzag behaviour, however, still occurs unless the search direction is aligned with the axis of any valley present in $f(\boldsymbol{x})$. The use of pattern directions (BD in figure 7.7) is, therefore, just as important in steepest gradient methods as in zeroth-order methods.

The most efficient zeroth-order method described was Powell's method shown in figure 7.8 which uses conjugacy to speed the convergence of the sequence of line searches. The exploitation of conjugacy is made much easier by the existence of gradient information. In Powell's method on an N-variable problem it was noted that a cycle of N sequential line searches using function values was needed to establish each new conjugate direction. Using gradient information it turns out that a new conjugate direction may be established after each steepest gradient search. Thus considerably fewer line searches are required in gradient-based conjugate methods than in function-based methods. Methods using conjugacy are among the most efficient of first-order methods. There are several such conjugacy-based first-order methods but probably the best of all of them is the method devised by Fletcher and Reeves. This method starts by carrying out a steepest gradient minimisation from a starting point. For each subsequent line minimisation a new steepest gradient direction is calculated but it is not searched along. Instead it is first modified by the accumulated knowledge of the conjugacy of all previous search directions and this modified direction is searched along. The directions of search are merely stated below since a derivation and explanation of the method is beyond the scope of this book. The interested reader should consult the specialised texts listed in the bibliography of this chapter for further details.

Equations 7.32 show that the steepest gradient at any point has a direction vector $-\boldsymbol{g}$ in the case of minimisation where the components of $\boldsymbol{g}$ are the N first derivatives of the N-variable function $f(\boldsymbol{x})$ evaluated at the point. The

www.EngineeringBooksPDF.com

conjugate gradient method of Fletcher and Reeves consists of carrying out line minimisations along a series of directions defined by N-vectors $\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3$, etc., where the subscripts number the sequence of new directions. The search directions are defined by equations 7.35.

$$\left.\begin{aligned} \boldsymbol{p}_1 &= -\boldsymbol{g}_1 \\ \boldsymbol{p}_2 &= -\boldsymbol{g}_2 + \frac{(\boldsymbol{g}_2)^T \boldsymbol{g}_2}{(\boldsymbol{g}_1)^T \boldsymbol{g}_1} \cdot \boldsymbol{p}_1 \\ &\vdots \\ \boldsymbol{p}_{i+1} &= -\boldsymbol{g}_{i+1} + \frac{(\boldsymbol{g}_{i+1})^T \boldsymbol{g}_{i+1}}{(\boldsymbol{g}_i)^T \boldsymbol{g}_i} \cdot \boldsymbol{p}_i \end{aligned}\right\} \qquad (7.35)$$

In equations 7.35 the direction vectors $-\boldsymbol{g}_1, -\boldsymbol{g}_2, -\boldsymbol{g}_3, \ldots$, are the steepest descent direction vectors for each new search in the sequence. The first search direction $\boldsymbol{p}_1$ is thus the steepest gradient direction, $-\boldsymbol{g}_1$, at the starting point. This search locates a line minimising point at which a new search direction is calculated. This search direction is made up of the new steepest gradient direction at the previous line minimising point, $-\boldsymbol{g}_2$, plus a multiple of the previous search direction $\boldsymbol{p}_1$. Each new search direction has a similar form.

Since the Fletcher and Reeves method constructs a new conjugate direction after each line minimisation it follows that the method can be guaranteed to find the minimum of a quadratic function $f(\boldsymbol{x})$ in N variables after $N + 1$ line minimisations. This is obviously much faster than Powell's method which uses only function values and requires $N^2 + 1$ line searches to minimise a quadratic. On a non-quadratic function no guarantees of convergence can be given but in general the Fletcher–Reeves method is efficient and robust. It is usual to use it in cycles of $N + 1$ searches on non-quadratic functions returning to the steepest gradient direction $\boldsymbol{p}_1$ after each cycle.

7.4 SECOND-ORDER METHODS

First-order methods are usually much more efficient that zeroth-order methods because more information in the form of gradient values is available to be used in a search procedure. Logically it may be expected that if still more information about $f(\boldsymbol{x})$ is available in the form of second derivatives representing curvatures and twists of the f hypersurface, methods of even higher efficiency may be devised. This is true but not completely so as will be shown. There are some very efficient second-order methods which on relatively small problems considerably outperform the best first-order methods. As the number of variables increases, however, the computational effort involved in merely calculating function values and all the first and second derivatives for use in a second-order method also increases and this can impair the over-all efficiency of the methods.

Second-order methods are based on strategies different from zeroth-order and

www.EngineeringBooksPDF.com

first-order methods. Zeroth-order methods seek to reduce directly the value of $f(\boldsymbol{x})$ as rapidly as possible. First-order methods also try directly to do this by monitoring values of slopes. Second-order methods are different because instead of trying to decrease $f(\boldsymbol{x})$ directly they centre around trying to satisfy the conditions for a minimum of $f(\boldsymbol{x})$, that is

$$\left.\begin{aligned} &\boldsymbol{g}(\boldsymbol{x}) = 0 \\ &\text{and} \\ &\boldsymbol{H} > 0 \end{aligned}\right\} \tag{7.36}$$

The second derivative information is therefore used to try to find a solution of the equations $\boldsymbol{g} = 0$ at which $\boldsymbol{H}$ is positive definite. The simplest and best known method of doing this is the long-established Newton–Raphson method.

7.4.1 The Newton–Raphson Method

Consider the problem of minimising a function of one variable $f(x)$ if function values and values of first and second derivatives can be evaluated. A necessary condition for a minimum of $f(x)$ is that

$$g(x) \equiv \mathrm{d}f/\mathrm{d}x = 0$$

Suppose that the first derivative $g(x)$ has been evaluated at some point x^{A} and is non-zero. Suppose that at some point x^{B} an unknown distance Δx from x^{A} the first derivative $g(x)$ is zero. Then it is necessary to find Δx such that

$$g(x^{\mathrm{B}}) = g(x^{\mathrm{A}} + \Delta x) = 0 \tag{7.37}$$

The function $g(x^{\mathrm{A}} + \Delta x)$ may be expanded as a Taylor series at point A giving

$$g(x^{\mathrm{A}} + \Delta x) = g(x^{\mathrm{A}}) + \left(\frac{\mathrm{d}g}{\mathrm{d}x}\right)_{\mathrm{A}} \Delta x + \text{higher order terms}$$

If the higher order terms are ignored and the expansion substituted in equation 7.37 it is found that

$$g(x^{\mathrm{B}}) \approx g(x^{\mathrm{A}}) + \left(\frac{\mathrm{d}g}{\mathrm{d}x}\right)_{\mathrm{A}} \Delta x = 0$$

and from this the value of Δx emerges as

$$\Delta x = -g(x^{\mathrm{A}})/(\mathrm{d}g/\mathrm{d}x)_{\mathrm{A}} \tag{7.38}$$

In terms of the derivatives of $f(x)$ this is equivalent to

$$\left.\begin{aligned} &\Delta x = -(\mathrm{d}f/\mathrm{d}x)_{\mathrm{A}}/(\mathrm{d}^2 f/\mathrm{d}x^2)_{\mathrm{A}} \\ &\text{and} \\ &x^{\mathrm{B}} = x^{\mathrm{A}} + \Delta x \end{aligned}\right\} \tag{7.39}$$

www.EngineeringBooksPDF.com

To use equation 7.39 to search for a minimum of $f(x)$ requires a starting point x^A at which the first and second derivatives of f are evaluated. From these derivatives equation 7.39 allows Δx to be calculated and a new point x^B found. If $f(x)$ is a quadratic x^B should be a stationary point of $f(x)$. The first derivative at x^B, $(df/dx)_B$ should be zero and the sign of the second derivative at x^B, $(d^2f/dx^2)_B$, determines whether $f(x^B)$ is a minimum, a maximum, or a point of inflexion. If $f(x)$ is not a quadratic then $(df/dx)_B$ will not be zero but this point is now a new starting point at which equation 7.39 may be used again to calculate a new Δx and a new point x^C. Eventually a point will be reached at which $df/dx = 0$ and $d^2f/dx^2 > 0$.

This method may be extended to minimise functions $f(\boldsymbol{x})$ of N variables. A starting point $\boldsymbol{x}^A$ is chosen and all first derivatives and second derivatives evaluated there. Generally, all the first derivatives will not be zero. At a candidate minimum point, $\boldsymbol{x}^B$, distance $\Delta\boldsymbol{x}$ from $\boldsymbol{x}^A$ all the first derivatives must be zero. Thus the N-dimensional equivalent of equation 7.37 is

$$\boldsymbol{g}(\boldsymbol{x}^B) = \boldsymbol{g}(\boldsymbol{x}^A + \Delta\boldsymbol{x}) = 0 \tag{7.40}$$

Using a truncated Taylor series expansion of $\boldsymbol{g}(\boldsymbol{x}^A + \Delta\boldsymbol{x})$ the equation 7.40 becomes

$$\boldsymbol{g}(\boldsymbol{x}^B) \approx \boldsymbol{g}(\boldsymbol{x}^A) + \boldsymbol{H}_A \cdot \Delta\boldsymbol{x} = 0$$

$\Delta\boldsymbol{x}$ is therefore found by solving the set of linear equations

$$\boldsymbol{H}_A \cdot \Delta\boldsymbol{x} = -\boldsymbol{g}(\boldsymbol{x}^A)$$

This gives the iterative scheme

$$\left.\begin{aligned} \Delta\boldsymbol{x} &= -\boldsymbol{H}_A^{-1} \cdot \boldsymbol{g}(\boldsymbol{x}^A) \\ \boldsymbol{x}^B &= \boldsymbol{x}^A + \Delta\boldsymbol{x} \end{aligned}\right\} \tag{7.41}$$

Equation 7.41 is the N-dimensional equivalent of equation 7.39. In equation 7.41 all the quantities $\boldsymbol{x}, \boldsymbol{g}$ and $\Delta\boldsymbol{x}$ are N-component column vectors and $\boldsymbol{H}$ is the $N \times N$ hessian matrix of second derivatives of $f(\boldsymbol{x})$. $\boldsymbol{H}^{-1}$ is, of course, the inverse matrix of $\boldsymbol{H}$, but is never directly calculated. The use of equation 7.41 as a search procedure is straightforward. At the starting point $\boldsymbol{x}^A$ all first derivatives $\boldsymbol{g}(\boldsymbol{x}^A)$ and all second derivatives in $\boldsymbol{H}_A$ are calculated (that is, $\partial^2 f/\partial x_i \partial x_j$ for all $i = 1, \ldots, N$, and $j = 1, \ldots, N$). The components of $\Delta\boldsymbol{x}$ are found by solving the linear equations $\boldsymbol{H}_A \cdot \Delta\boldsymbol{x} = -\boldsymbol{g}(\boldsymbol{x}^A)$ and the co-ordinates of point B are then found. The process continues until eventually all components of $\boldsymbol{g}(\boldsymbol{x})$ are zero (or infinitesimal) and $\boldsymbol{H}$ is positive definite. This then represents a minimising point of $f(\boldsymbol{x})$.

7.4.1.1 A Newton-Raphson Example

The example chosen is that of section 7.3.2.2, already studied in a steepest

www.EngineeringBooksPDF.com

gradient context. Minimise the function

$$f = x_1^6 + x_2^6 + 2x_1^2x_3^4 + 4x_2^4x_3^2 + 10x_1^2x_2^2x_3^2$$

using (1, 1, 1) as a starting point.

Values of the function and all first and second derivatives at (1, 1, 1), point A, are

$$f = 18$$

$$\frac{\partial f}{\partial x_1} = 6x_1^5 + 4x_1x_3^4 + 20x_1x_2^2x_3^2 = 30$$

$$\frac{\partial f}{\partial x_2} = 6x_2^5 + 16x_2^3x_3^2 + 20x_1^2x_2x_3^2 = 42$$

$$\frac{\partial f}{\partial x_3} = 8x_1^2x_3^3 + 8x_2^4x_3 + 20x_1^2x_2^2x_3 = 36$$

$$\frac{\partial^2 f}{\partial x_1^2} = 30x_1^4 + 4x_3^4 + 20x_2^2x_3^2 = 54$$

$$\frac{\partial^2 f}{\partial x_1\,\partial x_2} = \frac{\partial^2 f}{\partial x_2\,\partial x_1} = 40x_1x_2x_3^2 = 40$$

$$\frac{\partial^2 f}{\partial x_1\,\partial x_3} = \frac{\partial^2 f}{\partial x_3\,\partial x_1} = 16x_1x_3^3 + 40x_1x_2^2x_3 = 56$$

$$\frac{\partial^2 f}{\partial x_2^2} = 30x_2^4 + 48x_2^2x_3^2 + 20x_1^2x_3^2 = 98$$

$$\frac{\partial^2 f}{\partial x_2\,\partial x_3} = \frac{\partial^2 f}{\partial x_3\,\partial x_2} = 32x_2^3x_3 + 40x_1^2x_2 = 72$$

$$\frac{\partial^2 f}{\partial x_3^2} = 24x_1^2x_3^2 + 8x_2^4 + 20x_1^2x_2^2 = 52$$

Thus $\boldsymbol{g}$ and $\boldsymbol{H}$ at the starting point (1, 1, 1) are

$$\boldsymbol{g}_A = \begin{bmatrix} 30 \\ 42 \\ 36 \end{bmatrix} \qquad \boldsymbol{H}_A = \begin{bmatrix} 54 & 40 & 56 \\ 40 & 98 & 72 \\ 56 & 72 & 52 \end{bmatrix}$$

and the values of $\Delta\boldsymbol{x}$ are found from

$$\begin{bmatrix} 54 & 40 & 56 \\ 40 & 98 & 72 \\ 56 & 72 & 52 \end{bmatrix} \cdot \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \end{bmatrix} = \begin{bmatrix} -30 \\ -42 \\ -36 \end{bmatrix} \quad ; \quad \Delta\boldsymbol{x} = \begin{bmatrix} -0.2 \\ -0.2 \\ -0.2 \end{bmatrix}$$

The predicted minimum of f is then at B, the point

$$(1 - 0.2, 1 - 0.2, 1 - 0.2) = (0.8, 0.8, 0,8)$$

www.EngineeringBooksPDF.com

and the value of f here is $f = 4.7185$. Using equation 7.4 to check the positive definiteness of $\boldsymbol{H}$ it is found that d_1 and d_2 are positive, but d_3 is negative, thus $\boldsymbol{H}$ is not positive definite at the starting point.

A second iteration starts by calculating all the derivatives at point B, (0.8, 0.8, 0.8)

$$\frac{\partial f}{\partial x_1} = 9.8304; \quad \frac{\partial f}{\partial x_2} = 13.7626; \quad \frac{\partial f}{\partial x_3} = 11.7965$$

$$\frac{\partial^2 f}{\partial x_1^{\,2}} = 22.1184; \quad \frac{\partial^2 f}{\partial x_2^{\,2}} = 40.1408; \quad \frac{\partial^2 f}{\partial x_3^{\,2}} = 21.2992$$

$$\frac{\partial^2 f}{\partial x_1\,\partial x_2} = 16.3840; \quad \frac{\partial^2 f}{\partial x_1\,\partial x_3} = 22.9376; \quad \frac{\partial^2 f}{\partial x_2\,\partial x_3} = 29.4912$$

Thus $\boldsymbol{g}$ and $\boldsymbol{H}$ at point B are

$$\boldsymbol{g}_B = \begin{bmatrix} 9.8304 \\ 13.7626 \\ 11.7965 \end{bmatrix}; \qquad \boldsymbol{H}_B = \begin{bmatrix} 22.1184 & 16.3840 & 22.9376 \\ 16.3840 & 40.1408 & 29.4912 \\ 22.9376 & 29.4912 & 21.2992 \end{bmatrix}$$

and $\Delta \boldsymbol{x}$ is found from $\boldsymbol{H}_B \cdot \Delta \boldsymbol{x} = -\boldsymbol{g}(\boldsymbol{x}^B)$ to be

$$\Delta \boldsymbol{x} = \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \end{bmatrix} = \begin{bmatrix} -0.16 \\ -0.16 \\ -0.16 \end{bmatrix}$$

This gives a predicted minimum at point C, (0.64, 0.64, 0.64), at which $f = 1.2369$. Further iterations are required.

7.4.1.2 Modifications to the Newton-Raphson Method

The Newton-Raphson method as described is, for various reasons, not a very efficient or robust method. The main reason for its lack of efficiency lies in the truncation of the Taylor series which is made after the linear terms. It is assumed that higher order terms are negligible whereas in general problems they may be significant. It is clear from the previous numerical example that at the starting point (1, 1, 1) the third and fourth partial derivatives are very large. Consequently, by ignoring these the method does not make a very good predicition of the location of the function minimum.

One useful modification which improves the basic Newton-Raphson method is to use the predicted location not as a predicted point but rather as defining a useful direction of search. A search over a line parameter λ is made along the direction joining the starting point A to the point B predicted by the Newton-Raphson method. Thus equations 7.41 become

$$\left.\begin{aligned} \Delta \boldsymbol{x} &= -\boldsymbol{H}_A^{-1} \cdot \boldsymbol{g}(\boldsymbol{x}^A) \\ \boldsymbol{x}^B &= \boldsymbol{x}^A + \lambda^* \Delta \boldsymbol{x} \end{aligned}\right\} \qquad (7.42)$$

www.EngineeringBooksPDF.com

A numerical search is then made to locate λ^*, the value of λ which minimises $f(x)$ along the direction defined by $\Delta \boldsymbol{x}$.

If this modification is incorporated in the numerical example of the previous section the first iteration produces the same $\Delta \boldsymbol{x}$ that is,

$$\Delta \boldsymbol{x} = \begin{bmatrix} -0.2 \\ -0.2 \\ -0.2 \end{bmatrix}$$

but the point B is now given by the co-ordinates

$$(1 - 0.2\lambda^*, 1 - 0.2\lambda^*, 1 - 0.2\lambda^*)$$

and a numerical line minimisation is carried out over the variable λ to find λ^*. If this is done it is found that $\lambda^* = 5$ and point B turns out to be the co-ordinate origin (0, 0, 0). At this point $f = 0$ and all the first and second derivatives are zero and this is in fact the solution of the problem. Thus for this problem the modification to the basic Newton-Raphson method is strikingly successful. (Actually, the function f is somewhat peculiar in that the matrix $\boldsymbol{H}$ is indefinite throughout the search and is singular at the optimum. The obvious minimising solution, therefore, does not satisfy the sufficiency check for a minimum.)

Another difficulty often encountered with the basic Newton-Raphson method is its failure to converge. In the example of section 7.4.1.1 it can be shown that an infinite number of iterations will be required to locate the solution at (0, 0, 0) since each consecutive Δx_i, $i = 1, 2, 3$ will be 0.8 times the previous value. Frequently divergence is encountered. The reason for this is that unless $\boldsymbol{H}$ is positive definite convergence to a minimum cannot be guaranteed. If $\boldsymbol{H}$ is not positive definite the sequence of predictions may lead to a maximum of f or to a saddle point (the multi-variable equivalent of a point of inflexion). The modified method which carries out a line search along the predicted direction instead of merely moving to a predicted point is obviously helpful in overcoming this difficulty. The line search can identify whether the function is increasing in value and search in the reverse direction until a line minimum is found. It is still possible, however, for the modified Newton-Raphson method to come to a halt at a saddle point where $\boldsymbol{g} = 0$ and $\boldsymbol{H}$ is not positive definite. (Note that at such a point, because $\boldsymbol{g} = 0$ a steepest gradient search could not be used to find a better point.)

A factor which affects the robustness of both the basic and modified Newton-Raphson methods is that the matrix $\boldsymbol{H}$ can sometimes become singular during iterations. When $\boldsymbol{H}$ is singular the iterative schemes in equations 7.41 and 7.42 do not permit $\Delta \boldsymbol{x}$ to be calculated. This very often happens on general functions where at some starting point $\boldsymbol{H}$ is not positive definite but as searches proceed $\boldsymbol{H}$ should later become positive definite. The changeover in definiteness of $\boldsymbol{H}$ is associated with the determinant of $\boldsymbol{H}$ changing from negative to positive and at the point at which $|\boldsymbol{H}| = 0$ the matrix $\boldsymbol{H}$ will be singular. One way of overcoming this singularity problem in a search is that whenever $\boldsymbol{H} \cdot \Delta \boldsymbol{x} = -\boldsymbol{g}$ cannot be

www.EngineeringBooksPDF.com

solved a steepest gradient line minimisation is carried out instead of using the Newton-Raphson method. This usually restarts the solution process. Once a region of $f(\boldsymbol{x})$ is encountered in which $\boldsymbol{H}$ is positive definite the basic or modified Newton-Raphson methods will locate a minimum quite rapidly for most problems.

Although the basic Newton-Raphson method has several disadvantages as outlined above it is possible by quite simple modifications and improvements to overcome them. The modified method can be made quite robust and very efficient for solving unconstrained problems involving a few variables. A fundamental barrier to its effective use on larger problems is that of the number of non-productive calculations which must be made. At each starting point for a basic or modified Newton-Raphson iteration on an N-variable problem N first derivatives and $N(N+1)/2$ second derivatives must be calculated to set up $\boldsymbol{g}$ and $\boldsymbol{H}$. If N is large a significant amount of time is required to calculate this information. Considerably more time is required to solve the equations $\boldsymbol{H} \cdot \Delta\boldsymbol{x} = -\boldsymbol{g}$ to yield $\Delta\boldsymbol{x}$. Even using the most sophisticated and streamlined numerical methods this evaluation of $\Delta\boldsymbol{x}$ requires something of the order of N^3 separate calculations. Consequently, for problems other than those having only a few variables it is usually found that the gains in general efficiency and speed of convergence offered by the use of second-order information are more than negated by the time spent calculating and manipulating the extra information.

7.4.2 Quasi-Newton Methods

Most of the difficulties and disadvantages of the Newton-Raphson method are connected with the matrix $\boldsymbol{H}$: it can be time-consuming to construct and manipulate, it may be singular, it may not be positive definite. A group of methods known as *quasi-Newton methods* remove these difficulties by not using the matrix $\boldsymbol{H}$. Instead a positive definite matrix $\boldsymbol{J}$, and its inverse $\boldsymbol{J}^{-1}$, is used which approximates $\boldsymbol{H}$ more and more closely as iterations proceed until $\boldsymbol{J}^{-1}$ and $\boldsymbol{H}^{-1}$ become identical at the optimum. The name quasi-Newton is given to these methods because they resemble in algebraic form the Newton and modified Newton expressions for $\Delta\boldsymbol{x}$ very strongly. $\Delta\boldsymbol{x}$ in equations 7.41 and 7.42 becomes, in the quasi-Newton approach

$$\Delta\boldsymbol{x} = -\boldsymbol{J}^{-1} \cdot \boldsymbol{g} \tag{7.43}$$

There are several quasi-Newton methods which all use equation 7.43 but which differ in the ways in which $\boldsymbol{J}$ or rather its inverse $\boldsymbol{J}^{-1}$ are approximated. $\boldsymbol{J}$ is selected to be a positive definite matrix for the initial iteration and $\boldsymbol{J}^{-1}$ is modified after each subsequent iteration in such a way that $\boldsymbol{J}$ remains positive definite. This ensures convergence to a minimum. The simplest positive definite matrix with which to start the process is the identity matrix $\boldsymbol{I}$. If $\boldsymbol{J}$ is set equal to

www.EngineeringBooksPDF.com

I in 7.43 that relationship becomes

$$\Delta x = -I^{-1} \cdot g = -g$$

This represents an initial iteration consisting of a step of length unity in the steepest gradient direction towards a minimum. After this initial iteration the matrix J^{-1} is modified according to the information derived from the first iteration. Recalling the Fletcher and Reeves conjugate gradient method of section 7.3.2.3, which is a first-order method based on conjugacy, that too started with a steepest gradient search and in subsequent searches the directions were modified so as to guarantee convergence. The quasi-Newton methods are similar in operation and in fact the updating of J^{-1} is done using the concept of conjugacy. Consequently quasi-Newton methods are similar in form to the Newton–Raphson second-order method but may also be thought of as conjugate gradient methods.

One of the most efficient and popular quasi-Newton methods is the *Davidon–Fletcher–Powell method (DFP method).* As with the Fletcher and Reeves method the DFP method is too complicated to be explained fully in this book. It requires a fairly high level of familiarity with matrix algebra, the properties of quadratics and conjugacy. The interested reader should consult the more specialised books listed in the bibliography for further details and proofs. The DFP algorithm is merely stated below. For notational simplicity the inverse J^{-1} of the positive definite matrix J is referred to as the matrix K. Thus $K \equiv J^{-1}$. In the following algorithm the subscript i attached to all vectors and matrices refers to an iteration number $i = 1, 2, 3, \ldots$, etc.

Step 1: Set iteration number $i = 1$. Start with an initial point x_i and an $N \times N$ positive definite symmetric matrix K_i. For the first iteration $K_1 \equiv I$, the identity matrix.

Step 2: Calculate the elements of $g_i(x_i)$, that is, first derivatives, and calculate

$$\Delta x_i = -K_i g_i$$

Step 3: Carry out a line minimisation over λ in the direction Δx_i and calculate

$$x_{i+1} = x_i + \lambda_i^* \Delta x_i$$

Step 4: Check whether x_{i+1} is the solution point by calculating g_{i+1} and comparing against zeros. If x_{i+1} is optimal, terminate the process. If not, continue with step 5.

Step 5: Update the matrix K_i to K_{i+1} where

$$K_{i+1} = K_i + M_i + P_i$$

in which

$$M_i = \lambda_i^* \frac{\Delta x_i \, \Delta x_i^T}{\Delta x_i^T \, Q_i} \qquad P_i = -\frac{(K_i \, Q_i)\,(K_i \, Q_i)^T}{Q_i^T \, K_i \, Q_i} \qquad Q_i = g_{i+1} - g_i$$

www.EngineeringBooksPDF.com

Step 6: Set iteration number $i = i + 1$ and go to step 2.

The Davidon–Fletcher–Powell method has proved to be very robust and efficient in operation. Note that although it is similar in form in steps 2 and 3 to the modified Newton–Raphson method, a second-order method, the DFP method does not calculate or use second derivatives. To be precise, therefore, it is a first-order method which is as good as if not better than conjugacy-based first-order methods such as that of Fletcher and Reeves. One aspect in which the Fletcher–Reeves method is superior to the DFP method and other quasi-Newton methods is that of computer storage requirements. The Fletcher–Reeves method only requires that two N-element column vectors, $\boldsymbol{p}_i$ and $\boldsymbol{g}_i$ in equations 7.35, are stored between iterations i and $i + 1$ whereas in the DFP method the $N \times N$ matrix $\boldsymbol{K}_i$ must be stored between these iterations together with other information such as $\boldsymbol{g}_i$, $\Delta \boldsymbol{x}_i$, λ_i^*, etc. On large problems the Fletcher and Reeves method poses fewer computer storage difficulties.

7.5 APPROPRIATE METHODS FOR ENGINEERING PROBLEMS

This chapter has provided a brief introduction to unconstrained optimisation methods. Within the categories of zeroth, first and second-order methods, only the basic concepts have been examined and those methods which find most use in solving engineering design problems described. Many books have been devoted to more comprehensive surveys of unconstrained optimisation methods and the interested reader should consult some of these for more details. As this chapter has shown, the basic ideas used in unconstrained optimisation are simple but as increased numerical efficiency is sought so the methods need to use more advanced mathematics beyond the level set in most undergraduate and post-graduate civil engineering courses. The development of new algorithms for non-linear programming is now an intensely mathematical speciality and the most recent methods lie well beyond the scope of this book.

For engineering purposes, however, the robustness of an algorithm is an aspect at least as important as its efficiency. While it may be reasonable for a research specialist to try out several possible algorithms on a problem until a successful one is found, this practice is not acceptable in an engineering design context. For engineering use an optimisation algorithm should be robust enough to produce a local minimum solution for any appropriate problem. An algorithm which fails to produce a solution to a seemingly appropriate problem as a result of numerical inaccuracies arising from its own sophistication cannot be recommended for practical engineering use. The methods described in this chapter may not be the most modern or the most efficient but they have all been widely used and have proved their robustness. Eventually some of the more recently developed methods will replace those described here when their robustness has been proved but this process is necessarily a gradual one.

The choice of which particular method is the most appropriate to solve a given engineering problem is not always easy. Nevertheless some general guide-

www.EngineeringBooksPDF.com

lines may be given. First of all the nature of the function to be minimised should be examined. An analytical solution is usually preferable to a numerical solution if it can be obtained so most engineers will usually attempt a classical differential solution before anything else. It requires only a few minutes' work to determine whether the classical approach is likely to yield results. The determining factors here are whether $f(\boldsymbol{x})$ is continuous and differentiable and whether the set of equations $\boldsymbol{g}(\boldsymbol{x}) = 0$ appear likely to be soluble. Assuming that the classical differential method for some reason does not work then a numerical solution must be sought. Attention is then redirected towards determining the characteristics of the objective function.

The first question which should be asked is whether or not the function $f(\boldsymbol{x})$ is a continuous one. The nature of the practical engineering problem represented by $f(\boldsymbol{x})$ is a help here. For example, if $f(\boldsymbol{x})$ contains several cost coefficients which have different values for different ranges of variables then $f(\boldsymbol{x})$ will be discontinuous. If $f(\boldsymbol{x})$ is not continuous the simplex method of section 7.2.2.2 is an appropriate method to use as it is least affected by discontinuities. Although sequential line minimisation methods can be used, the possible discontinuities suggest zeroth-order methods with Fibonacci for line searches and the most efficient of these methods, Powell's method, is affected by discontinuities which disrupt the sequence of conjugate directions. If the discontinuous nature of $f(\boldsymbol{x})$ arises only because of the presence of integer or discrete variables the best way of proceeding is to assume that the variables are all continuously valued and solve the problem as a continuous one, rounding-off the continuous optimum to the nearest integer or discrete solution. As was shown with linear programming in figure 3.9 this process is not rigorous but the difficulties set by integer or discrete-valued variables in NLP are far greater even than in LP.

If the function $f(\boldsymbol{x})$ is a continuous one the next questions that should be asked are: 'Is any derivative information available?' and 'How easily can it be obtained?'. If analytical first derivatives are not available they may be estimated by finite difference approximations as in equation 7.17 or 7.18 if function evaluations can be made very quickly and accurately. A better choice of method, however, in the absence of analytical derivatives is to use Powell's zeroth-order method with quadratic or cubic curve-fitting for the line searches. If analytical first derivatives are available the choice of methods is essentially between a conjugate gradient method such as Fletcher–Reeves or a quasi-Newton method such as Davidon–Fletcher–Powell. The Fletcher–Reeves method is recommended if there are any apprehensions about available computer storage or if first derivatives have been obtained by finite difference approximations, but otherwise the DFP method is recommended as being generally more efficient. With both these methods quadratic or cubic curve-fitting should be used for line minimisations; a personal preference of the author is for the quadratic.

A final possibility which should be borne in mind if analytical first and second derivatives are available and the function is a continuous one in five variables or less is that of using a modified Newton method in preference to

www.EngineeringBooksPDF.com

Fletcher-Reeves or Davidon-Fletcher-Powell. The value $N = 5$ is about the number of variables above which the numerical disadvantages of the Hessian matrix become significant.

SUMMARY

This chapter has examined methods for solving unconstrained optimisation problems. The classical differential approach for such problems is not generally applicable and numerical solution methods must be used in its place on most problems. Many different numerical optimisation methods were examined in three categories according to what derivative information was available for use in a numerical search for a minimum. In general it was shown that zeroth-order methods using only function values are less efficient than first-order methods which use first derivatives in addition to function values. The anticipated greater efficiency of second-order methods over first-order methods did not materialise except for very small problems because of the extra time needed to calculate and manipulate all the information.

Aspects of the chapter which stand out as being particularly important are line minimisation and the concept of conjugacy. Most unconstrained optimisation methods solve multi-variable problems by a sequence of searches along a line. Conjugacy is important because it provides a means of choosing good directions in which to carry out the searches. The speed and accuracy with which a line minimum can be located along a search direction is also an important factor in the efficiency of any search method. Guidelines were presented to help in making the choice of a solution method appropriate to a given problem.

Although unconstrained optimisation methods have an importance in solving many civil engineering problems it is still a fact that most engineering design problems are constrained ones. The next chapter is devoted to methods for solving non-linear constrained optimisation problems. Within this much larger class of problems it turns out that many solution methods incorporate aspects of unconstrained optimisation. Therefore, the ideas and methods of this chapter form an excellent background for the next chapter.

BIBLIOGRAPHY

Avriel, M., Rijckaert, M. J., and Wilde, D. J., (eds), *Optimization and Design* (Prentice-Hall, Englewood Cliffs, N.J., 1973); see particularly R. W. H. Sargent, 'Minimization without constraints', pp. 37-75

Dixon, L. C. W., *Non-linear Optimization* (English Universities Press, London, 1972) chapters 1 to 5

Dixon, L. C. W. (ed.), *Optimization in Action* (Academic Press, London, 1976)

Fletcher, R., and Powell, M. J. D., A rapidly convergent descent method for minimization, *Comput. J.*, **6**, No. 2 (1963) 163-8

www.EngineeringBooksPDF.com

Fletcher, R., and Reeves, C. M., Function minimization by conjugate gradients, *Comput. J.*, 7, No. 2 (1964) 149-54

Murray, W., (ed.), *Numerical Methods for Unconstrained Optimization* (Academic Press, London, 1972)

Nelder, J. A., and Mead, R., A simplex method for function minimization, *Comput. J.* 7, No. 4 (1965) 308-13

Powell, M. J. D., An efficient method for finding the minimum of a function of several variables without calculating derivatives, *Comput. J.*, 7, No. 2 (1964) 155-62

Spendley, W., Hext, G. R., and Himsworth, F. R., Sequential application of simplex design in optimization and evolutionary operation, *Technometrics*, 4, (1962) 441-59

Wilde, D. J., *Optimum Seeking Methods* (Prentice-Hall, Englewood Cliffs, N.J., 1964)

EXERCISES

7.1 The function $f = x_1^2 + 2x_2^2 + 3x_1^4 + 4x_2^4 + 5x_1^2x_2^2$ has, by inspection, a minimum at (0, 0). Find the value of f after two steepest gradient searches for the minimum of f, starting from point (1, 1). Use a single quadratic fit and $r = -1$ for each line minimisation.

7.2 Carry out two cycles of a Newton-Raphson minimisation of the function f in question 1 starting from the same point (1, 1).

7.3 A hole of 1000 m^3 volume is to be dug in flat ground. It is to be circular in plan with vertical sides. It is to be lined on the sides and base with an impervious lining so that it can be filled with liquid but the top is not to be covered.

The lining costs £10/m^2 and the excavation costs are £1/m^2 for the first 2 m depth, £2/m^2 for the next 2 m depth and £3/m^2 thereafter.

Find the dimensions of the hole so that total costs are minimised.

7.4 In the storage tank example 6.2.3 of chapter 6 the values of the unit costs are C_1 = £30/m^2, C_2 = £50/m^2, C_3 = £90/m^2 and the volume of the tank must be $V' = 10^3$ m^3. Using equation 6.12 as a cost function for the tank bracket the optimum (least cost) diameter D of the tank by a Fibonacci method starting at D = 10 m and using an initial step-length of 0.1 m.

7.5 Locate the optimum diameter of the tank in question 4 within the bracketed interval by a Fibonacci search to an accuracy of 0.1 m. What will be the likely cost of the tank?

www.EngineeringBooksPDF.com

7.6 How many trials are needed to bracket the minimum of the function

$$f = x^6 - 3x^4 - 10x^3 - x^2 - 858x + 2500$$

using gradient values only? Start at $x = 0$ with an initial step-length of 0.01, doubling each subsequent step-length.

Using gradient values only locate the minimum of f within an interval of length 0.01. How many more trials does this requires?

7.7 The salinity level, f, of the output stream from a water desalination plant depends upon four operating parameters, x_1 to x_4. Several test runs have been carried out to attempt to find values for x_1 to x_4 which achieve a minimum salinity f^*. The results for six test runs are shown in the table.

Calculate numerical first derivatives from this data and determine suitable values for x_1 to x_4 for the next trial run.

Run no.	x_1	x_2	x_3	x_4	f
1	1.00	15.00	3.50	120.0	230.0
2	1.00	15.10	3.50	120.0	247.0
3	1.00	15.00	3.40	120.0	220.0
4	1.00	15.00	3.50	121.0	205.0
5	1.05	15.00	3.50	120.0	228.0
6	1.59	12.49	2.02	120.4	280.0

7.8 Show that the Newton-Raphson method predicts a minimum at the point (2/3, 4/3) when applied to the function

$$f = x_1^4 + 2x_2^4 - 3x_1x_2^3 + 4x_1^3x_2 - 2x_1 - 4x_2$$

at a starting point (1, 1).

Use the predicted point to define a search direction from the starting point as in the modified Newton-Raphson method and find the line minimum of f, that is, substitute $x_1 = 1 - \lambda/3, x_2 = 1 + \lambda/3$ into f and minimise f over λ).

www.EngineeringBooksPDF.com

8 NON-LINEAR CONSTRAINED OPTIMISATION METHODS

Constrained non-linear optimisation problems are more difficult to solve than unconstrained problems. The previous chapter has described methods that search an unconstrained function to find its lowest value. These search methods are somewhat pedestrian because in general nothing is known about the function being searched. Many trials are needed to build up even a rough picture of how the function behaves. If a set of non-linear constraints or boundaries is added to the function dividing it into feasible and infeasible regions it is no longer sufficient to know merely how the function behaves. The positions, orientations and configurations of the boundaries of the feasible region must also be explored numerically before a constrained minimum can be found. Thus in general, even more trials will be needed to solve a constrained problem than an unconstrained problem. If a non-linear problem has many variables and many constraints the number of trials needed by a numerical direct search process to locate a minimum can be very large and uneconomical. For this reason constrained problems are not usually solved by direct numerical search methods unless they are fairly small problems. Instead various devices, approximations and transformations are used to make the constrained problem easier to solve.

This chapter begins by examining some simple ways of eliminating constraints from a problem and removing variables so as to make the problem smaller and easier to solve. Next it examines so-called Lagrangian methods for equality and inequality-constrained problems and shows how these methods essentially convert a constrained optimisation problem into an unconstrained one. Lagrangian methods are, however, of theoretical interest rather than practical value; they do not in general make very good numerical solution methods. Nevertheless, the idea of converting a constrained problem to an unconstrained one is attractive since unconstrained problems can be solved fairly efficiently. The group of methods known as penalty function methods exploits this idea and is examined in detail. Penalty function methods solve a constrained problem by solving a converging sequence of unconstrained problems.

This chapter then examines methods which do not attempt to eliminate the

www.EngineeringBooksPDF.com

constraints. One popular method in this category is that of approximating all the non-linear functions in the objective and constraints by linear functions. This method, known as sequential linear programming, is examined in detail and some of its disadvantages are described. Other direct numerical search methods are briefly described although their efficiencies on all but the smallest constrained problems are generally very low. Finally the chapter describes the method known as geometric programming which is of great value in engineering design and is totally different in concept from all other non-linear programming methods.

8.1 SIMPLE SOLUTION DEVICES

8.1.1 Elimination and Substitution

The more constraints which are appended to a problem the more difficult it usually becomes to solve. A first step in solving any constrained problem should always be to examine it carefully to see if there are any ways of removing constraints to simplify the problem. An example of this has already been demonstrated in chapter 6, example 6.2.3, the design of a liquid storage tank. As initially formulated the problem consisted of minimising a non-linear cost function, 6.9, of two variables, D and L, subject to a single equality constraint, 6.10. This is a non-linear constrained problem. It was simplified, however, by using the single equality constraint to express one variable, L, as a function of the other, D, and substituting the resulting expression for L into the cost function. This effectively removed the constraint and eliminated a variable leaving a final, much simpler problem of minimising an unconstrained function of only one variable, problem 6.12.

Whenever equality constraints are present in an NLP problem they should be used to eliminate variables if this is possible. Sometimes it is not possible to rearrange an equality to give one variable as an explicit function of the others; in this case the equality constraint cannot be removed. More usually it is possible and this course of action should be taken as it almost always simplifies the subsequent solution process.

Often some or all of the constraints appended to an NLP problem consist of limits or bounds upon the values of the problem variables. Constraints such as these can be removed by substituting appropriate new variables which do not require bounds in place of the original variables. Some useful substitutions are given below in which x_i is an original problem variable with bounds and x_j is the new problem variable which replaces it and is unbounded.

(1) If $x_i \geqslant 0$ is a constraint then variable x_i should be replaced by x_j^2 wherever it occurs. The constraint $x_j^2 \geqslant 0$ can be discarded as it is automatically true for all x_j.

(2) If $a \geqslant x_i \geqslant 0$ is present as a constraint then variable x_i should be replaced by $a \cos^2 x_j$ wherever it occurs and the constraint discarded.

www.EngineeringBooksPDF.com

(3) If $a \geqslant x_i \geqslant -a$ then x_i should be replaced by $a \sin x_j$ and the constraint removed.

(4) If $a \geqslant x_i \geqslant b$ then x_i should be replaced by $[(a + b)/2] + [(a - b)/2] \sin x_j$ and the constraint removed.

Other substitutions can be devised to allow these and other variable bounds to be eliminated. Of course at each substitution the problem becomes algebraically more complicated but should be easier to solve as fewer constraints will be present. Some substitutions can add new local optima where none previously existed.

8.1.2 Solution by Constraint Deletions

In relatively small problems of engineering design (a few variables and constraints), knowledge of the engineering problem which the NLP problem represents can be used to remove constraints. For example, consider the simply supported beam design, example 6.2.1 of chapter 6. Many possible constraints are listed there describing the possible limiting effects of bending stresses, shear stresses, deflection, cross-sectional proportions, permissible sizes, etc. For a particular set of design loads, spans, code limits, etc., it is likely that many of the constraints listed will not be active at the optimum. Only perhaps two or three will govern the design. Experience in design can often be used to discard those constraints which the designer thinks will not be active at the solution leaving only two or three which may be active. This use of engineering knowledge to reduce the size of a problem by discarding constraints can be most effective. It should be noted strongly, however, that any constraint discarded as likely to be slack at the solution should always be checked for slackness once the solution has been obtained.

This policy of discarding constraints to enable a solution to be found and then checking them against the solution can also be used in a trial and error type of solution method for small, analytically amenable problems. Suppose an NLP problem has a non-linear objective function of several variables and a single inequality constraint. Suppose that it is not known whether the constraint will be active or slack at the solution of the problem. The problem may be solved by making an assumption and later verifying the assumption. Assume that the constraint is slack at the optimum. The constraint can then be removed and an unconstrained solution found for the problem. The solution point is then substituted into the removed constraint. If the constraint is not violated the initial assumption of its slackness was correct and the unconstrained solution which has been found must be the solution of the complete problem. If the constraint is violated, however, by the unconstrained solution point the initial assumption of slackness must be incorrect. The constraint must be active at the solution of the problem. Since it must be active it must be satisfied as an equality. As an equality constraint it may be possible to remove it by using it to eliminate a variable from

www.EngineeringBooksPDF.com

the problem leaving again an unconstrained optimisation problem to be solved. The solution of this problem must represent the solution of the complete problem.

This approach may be used when two or more inequality constraints are present although care must be taken. An initial assumption must be made about the slackness or activeness of each constraint and for several constraints there will be many possible combinations of valid or invalid initial assumptions. It may require several trial solutions with different constraints before one compatible with the initial assumptions is found. An example of this is given below in section 8.1.3.

8.1.3 Example

Find the minimum of

$$f = (x_1 + 2x_2 + 3x_3 - 4)^2 + (2x_1 + x_3)^2 + 4x_2{}^2$$

subject to

$$x_1 + x_2 + x_3 \leqslant 1.5$$
$$-x_1 - 2x_2 + 3x_3 \leqslant 2$$

Firstly, assume both constraints to be slack at the solution and find an unconstrained minimum of f. This can be done analytically by setting each of the three first derivatives of f to zero and solving the three resulting linear equations in three unknowns. The unconstrained minimum of f is found to be at the point $(-0.8, 0, 1.6)$ at which point $f = 0$. Substitution of this solution into the two constraints gives

$$x_1 + x_2 + x_3 = -0.8 + 0 + 1.6 = 0.8 < 1.5$$

This constraint is, therefore, slack as was initially assumed. The second constraint gives

$$-x_1 - 2x_2 + 3x_3 = 0.8 - 0 + 4.8 = 5.6 \nleqslant 2$$

and is violated by the unconstrained solution. The second constraint does not satisfy the assumption made about it and must be active at the solution of the original problem. Now assume the first constraint to be slack and the second to be active at the optimum. The first constraint may be ignored again and the second constraint expressed as an equality giving a new problem

$$\text{Minimise } f = (x_1 + 2x_2 + 3x_3 - 4)^2 + (2x_1 + x_3)^2 + 4x_2{}^2$$
$$\text{subject to } -x_1 - 2x_2 + 3x_3 = 2$$

The equality constraint can be eliminated by substitution for x_1, that is

$$x_1 = -2 - 2x_2 + 3x_3$$

and the problem reduces to

$$\text{Minimise } f = (6x_3 - 6)^2 + (-4x_2 + 7x_3 - 4)^2 + 4x_2{}^2$$

www.EngineeringBooksPDF.com

This may be solved analytically to give a solution $f = 1.415$ at $x_2 = 0.472$, $x_3 = 0.908$. The corresponding value of $x_1 = -0.218$. Checking this point in both original constraints gives

$$x_1 + x_2 + x_3 = 1.162 < 1.5$$

The slackness of the first constraint is thus confirmed. The second constraint gives

$$-x_1 - 2x_2 + 3x_3 = 2$$

which is active, as expected. The complete problem has, therefore, been solved and the solution is $f^* = 1.415$ at the point $(-0.218, 0.472, 0.908)$.

Section 8.1 has examined several useful ways in which constrained NLP problems may be simplified and solved. For very many problems, however, when all these simplifications have been employed, the problems still remain large in size and with several constraints. Most problems require more than the above cosmetic devices for their solution. The next section 8.2 examines constrained optimisation in a more rigorous mathematical fashion and establishes the bases of more generally applicable solution methods.

8.2 LAGRANGE MULTIPLIER METHODS

In chapter 7, section 7.1, a classical analytical method for solving unconstrained optimisation problems was described. In practice it turned out that the method was not particularly useful and other methods based upon numerical rather than analytical approaches were needed to solve most practical problems. A similar situation exists with constrained optimisation. An analytical solution method exists which is not particularly good for practical problem solving. Nevertheless the analytical method does provide a sound basis for understanding how and why the numerical methods work so well and so it is worthwhile studying the classical approach as a preliminary to the numerical methods

8.2.1 Equality-constrained Problems

Consider the problem

$$\begin{aligned} &\text{Minimise } f(x_1, \ldots, x_N) \\ &\text{subject to } g_j(x_1, \ldots, x_N) = 0 \quad j = 1, \ldots, J \end{aligned} \tag{8.1}$$

The objective function f and each of the J constraint functions $g_j, j = 1, \ldots, J$ are general functions of N variables $x_i, i = 1, \ldots, N$. All the J constraints (usually $J < N$) are written such that the right-hand side is zero and are equality constraints. Note that in this chapter g is used to denote a constraint function and is not the same as g in chapter 7 where it represented the first partial derivative of f.

The classical method known as the Lagrange multiplier method solves problem 8.1 indirectly. Firstly a *Lagrangian function* $L(\boldsymbol{x}, \boldsymbol{\lambda})$ is constructed. The Lagrangian

www.EngineeringBooksPDF.com

function $L(\boldsymbol{x},\boldsymbol{\lambda})$ is made up of the objective function $f(x_1, \ldots, x_N)$ plus each of the J constraint functions $g_j(x_1, \ldots, x_N), j = 1, \ldots, J$, multiplied by a new variable $\lambda_j, j = 1, \ldots, J$, known as a *Lagrange multiplier*. Thus

$$L(\boldsymbol{x},\boldsymbol{\lambda}) = f(x_1, \ldots, x_N) + \sum_{j=1}^{J} \lambda_j g_j(x_1, \ldots, x_N) \tag{8.2}$$

The Lagrangian function $L(\boldsymbol{x},\boldsymbol{\lambda})$ is then a function of $N + J$ variables x_i, $i = 1, \ldots, N$ and $\lambda_j, j = 1, \ldots, J$. It is possible to show that a solution of the constrained minimisation problem 8.1 is also a stationary point of the Lagrangian function 8.2. Problem 8.1 can, therefore, be solved indirectly by finding a stationary point of the Lagrangian function 8.2. This is equivalent to replacing a constrained minimisation problem by an unconstrained problem which can be easily solved.

In order to show the equivalence between problem 8.1 and a stationary point of 8.2 consider a very simple problem with two variables, x_1 and x_2, and a single equality constraint, that is

$$\left.\begin{array}{l} \text{Minimise } f(x_1, x_2) \\ \text{subject to } g(x_1, x_2) = 0 \end{array}\right\} \tag{8.3}$$

Suppose that the point (x_1^*, x_2^*) is a solution of problem 8.3. Consider a point $(x^*_1 + \Delta x_1, x_2^* + \Delta x_2)$ infinitesimally close to (x_1^*, x_2^*) and expand $f(x_1^* + \Delta x_1, x_2^* + \Delta x_2)$ as a Taylor series truncated after the linear terms. Thus

$$f(x_1^* + \Delta x_1, x_2^* + \Delta x_2) = f(x_1^*, x_2^*) + \left(\frac{\partial f}{\partial x_1}\right)_* \Delta x_1 + \left(\frac{\partial f}{\partial x_2}\right)_* \Delta x_2 \tag{8.4}$$

The * subscript to the partial derivatives in 8.4 denotes values at (x_1^*, x_2^*). Since Δx_1 and Δx_2 are infinitesimals and since f is minimised by (x_1^*, x_2^*) the values of $f(x_1^*, x_2^*)$ and $f(x_1^* + \Delta x_1,\ x_2^* + \Delta x_2)$ should be the same. Thus equation 8.4 yields

$$\left(\frac{\partial f}{\partial x_1}\right)_* \Delta x_1 + \left(\frac{\partial f}{\partial x_2}\right)_* \Delta x_2 = 0 \tag{8.5}$$

Now consider the constraint. At the point (x_1^*, x_2^*) the constraint must be satisfied and it must still be satisfied at the point $(x_1^* + \Delta x_1, x_2^* + \Delta x_2)$. Thus

$$\left.\begin{array}{l} g(x_1^*, x_2^*) = 0 \\ \text{and} \\ g(x_1^* + \Delta x_1, x_2^* + \Delta x_2) = 0 \end{array}\right\} \tag{8.6}$$

Expanding $g(x_1^* + \Delta x_1, x_2^* + \Delta x_2)$ as a Taylor series in a similar fashion to that above gives

$$g(x_1^* + \Delta x_1, x_2^* + \Delta x_2) = g(x_1^*, x_2^*) + \left(\frac{\partial g}{\partial x_1}\right)_* \Delta x_1 + \left(\frac{\partial g}{\partial x_2}\right)_* \Delta x_2 \tag{8.7}$$

www.EngineeringBooksPDF.com

Substituting 8.6 into 8.7 yields

$$\left(\frac{\partial g}{\partial x_1}\right)_* \Delta x_1 + \left(\frac{\partial g}{\partial x_2}\right)_* \Delta x_2 = 0 \tag{8.8}$$

Equations 8.5 and 8.8 may now be used to eliminate the infinitesimals. Δx_2 can be found from 8.8 as

$$\Delta x_2 = -\frac{(\partial g/\partial x_1)_*}{(\partial g/\partial x_2)_*} \times \Delta x_1 \tag{8.9}$$

Substituting 8.9 into 8.5 gives

$$\left(\frac{\partial f}{\partial x_1}\right)_* \Delta x_1 - \frac{(\partial g/\partial x_1)_*}{(\partial g/\partial x_2)_*} \times \left(\frac{\partial f}{\partial x_2}\right)_* \Delta x_1 = 0$$

that is

$$\left[\left(\frac{\partial f}{\partial x_1}\right) - \frac{(\partial g/\partial x_1)}{(\partial g/\partial x_2)} \times \left(\frac{\partial f}{\partial x_2}\right)\right]_* \Delta x_1 = 0 \tag{8.10}$$

Since Δx_1 is an infinitesimal, though is not necessarily equal to zero, the only way 8.10 may be satisfied is if the terms within the $[\ ]_*$ are equal to zero, that is

$$\left[\frac{\partial f}{\partial x_1} - \frac{(\partial g/\partial x_1)}{(\partial g/\partial x_2)} \frac{\partial f}{\partial x_2}\right]_* = 0 \tag{8.11}$$

Equation 8.11 gives a condition which must be satisfied at the solution point of the problem, (x_1^*, x_2^*). Condition 8.11 may be rewritten as

$$\left[\frac{\partial f}{\partial x_1} - \frac{\partial f/\partial x_2}{\partial g/\partial x_2} \times \frac{\partial g}{\partial x_1}\right]_* = 0 \tag{8.12}$$

Now define a quantity λ, called the Lagrange multiplier, as

$$\lambda = -\left(\frac{\partial f/\partial x_2}{\partial g/\partial x_2}\right)_* \tag{8.13}$$

and condition 8.12 can be written as

$$\left(\frac{\partial f}{\partial x_1} + \lambda \frac{\partial g}{\partial x_1}\right)_* = 0 \tag{8.14}$$

Relationship 8.13 can also be rewritten as

$$\left(\frac{\partial f}{\partial x_2} + \lambda \frac{\partial g}{\partial x_2}\right)_* = 0 \tag{8.15}$$

Equations 8.14 and 8.15 represent two conditions which must hold if (x_1^*, x_2^*) is

www.EngineeringBooksPDF.com

to be a solution of problem 8.3. A third condition which must hold is that (x_1^*, x_2^*) must satisfy the constraint, that is

$$g(x_1^*, x_2^*) = 0 \tag{8.16}$$

Collecting together the three conditions 8.14, 8.15 and 8.16 gives conditions 8.17, the necessary conditions for (x_1^*, x_2^*) to be a solution of problem 8.3, that is

$$\left.\begin{aligned} \left(\frac{\partial f}{\partial x_1} + \lambda \frac{\partial g}{\partial x_1}\right)_* &= 0 \\ \left(\frac{\partial f}{\partial x_2} + \lambda \frac{\partial g}{\partial x_2}\right)_* &= 0 \\ g(x_1^*, x_2^*) &= 0 \end{aligned}\right\} \tag{8.17}$$

Now consider the Lagrangian function $L(x_1, x_2, \lambda)$ for problem 8.3. L is formed by adding to the objective function f the constraint g multiplied by a Lagrange multiplier λ. Thus

$$L(x_1, x_2, \lambda) = f(x_1, x_2) + \lambda g(x_1, x_2)$$

Necessary conditions for a stationary point of L are that

$$\frac{\partial L}{\partial x_1} = 0; \frac{\partial L}{\partial x_2} = 0; \frac{\partial L}{\partial \lambda} = 0$$

These three conditions yield 8.18

$$\left.\begin{aligned} \frac{\partial f}{\partial x_1} + \lambda \frac{\partial g}{\partial x_1} &= 0 \\ \frac{\partial f}{\partial x_2} + \lambda \frac{\partial g}{\partial x_2} &= 0 \\ g(x_1, x_2) &= 0 \end{aligned}\right\} \tag{8.18}$$

The conditions 8.18 for a stationary point of the Lagrangian function $L(x_1, x_2, \lambda)$ are exactly the same as the necessary conditions 8.17 for (x_1^*, x_2^*) to be a constrained minimum of problem 8.3.

A very simple problem 8.3 was chosen to demonstrate the equivalence of constrained minimisation and Lagrangian stationarity because the algebra can be easily followed. For larger problems in N variables and with J equality constraints the equivalence still holds although the algebra needed to prove it is much more complicated. Thus conditions for a solution of problem 8.1 and for a stationary point of the Lagrangian of problem 8.1, which is stated as 8.2, are identical and are as follows.

$$\left.\begin{aligned} \frac{\partial f}{\partial x_k} + \sum_{j=1}^{J} \lambda_j \frac{\partial g_j}{\partial x_k} &= 0 \quad k = 1, \ldots, N \\ g_j(x_1, \ldots, x_N) &= 0 \quad j = 1, \ldots, J \end{aligned}\right\} \tag{8.19}$$

www.EngineeringBooksPDF.com

The first of these conditions giving N equations comes from the requirement that $\partial L(\boldsymbol{x},\boldsymbol{\lambda})/\partial x_k = 0$ for all $k = 1, \ldots, N$. The second condition giving J equations comes from the requirement that $\partial L/\partial \lambda_j = 0$ for all $j = 1, \ldots, J$.

8.2.1.1 Example

$$\begin{aligned} &\text{Minimise } f = \tfrac{1}{2}(x_1^2 + x_2^2 + x_3^2) \\ &\text{subject to } x_1 + 2x_2 + 3x_3 = 1 \\ &\qquad\qquad 3x_1 + 2x_2 + x_3 = 2 \end{aligned}$$

The Langrangian function for this problem is

$$\begin{aligned} L(x_1, x_2, x_3, \lambda_1, \lambda_2) = \tfrac{1}{2}(x_1^2 + x_2^2 + x_3^2) + \lambda_1(x_1 + 2x_2 + 3x_3 - 1) \\ + \lambda_2(3x_1 + 2x_2 + x_3 - 2) \end{aligned}$$

Necessary conditions for a stationary point of L are that

$$\frac{\partial L}{\partial x_1} = \frac{\partial L}{\partial x_2} = \frac{\partial L}{\partial x_3} = \frac{\partial L}{\partial \lambda_1} = \frac{\partial L}{\partial \lambda_2} = 0$$

leading to the simultaneous equations

$$\begin{aligned} x_1 + \lambda_1 + 3\lambda_2 &= 0 && \text{(a)} \\ x_2 + 2\lambda_1 + 2\lambda_2 &= 0 && \text{(b)} \\ x_3 + 3\lambda_1 + \lambda_2 &= 0 && \text{(c)} \\ x_1 + 2x_2 + 3x_3 - 1 &= 0 && \text{(d)} \\ 3x_1 + 2x_2 + x_3 - 2 &= 0 && \text{(e)} \end{aligned}$$

From (a), (b) and (c) x_1, x_2 and x_3 are found as functions of λ_1 and λ_2, that is

$$\begin{aligned} x_1 &= -\lambda_1 - 3\lambda_2 \\ x_2 &= -2\lambda_1 - 2\lambda_2 \\ x_3 &= -3\lambda_1 - \lambda_2 \end{aligned}$$

Substituting into (d) and (e) gives

$$\begin{aligned} -14\lambda_1 - 10\lambda_2 - 1 &= 0 && \text{(f)} \\ -10\lambda_1 - 14\lambda_2 - 2 &= 0 && \text{(g)} \end{aligned}$$

Solving (f) and (g) gives $\lambda_1^* = 1/16$; $\lambda_2^* = -3/16$. Substituting these values gives optimal values of x_1, x_2, x_3 as

$$x_1^* = \tfrac{1}{2}; x_2^* = \tfrac{1}{4}; x_3^* = 0$$

and the value of f is then $f^* = 1/2(1/4 + 1/16) = 5/32$. The solution would have been obtained had the constraints been eliminated by substitution into the objective function as suggested in section 8.1.1

www.EngineeringBooksPDF.com

8.2.2 Inequality-constrained Problems

Suppose the constraints in problem 8.1 are inequality constraints instead of equality constraints. The general form of the problem is then

$$\left.\begin{array}{l}\text{Minimise } f(x_1, \ldots, x_N) \\ \text{subject to } g_j(x_1, \ldots, x_N) \leqslant 0 \quad j = 1, \ldots, J\end{array}\right\} \tag{8.20}$$

Any inequality constraint can be written in the form $g(x) \leqslant 0$ with suitable algebraic manipulations so problem 8.20 is quite general. How can the Lagrange multiplier method be used with inequality constraints?

The answer to this question lies in the use of slack variables. In chapter 3 on linear programming, linear inequality constraints were converted to equalities by adding or subtracting a slack variable in each constraint. The same thing can be done here. An inequality constraint of the form

$$g(x_1, \ldots, x_N) \leqslant 0$$

is converted to an equality constraint of the form

$$g(x_1, \ldots, x_N) + x_{N+1}^2 = 0$$

by adding the square of an extra slack variable x_{N+1}. The slack variable x_{N+1} is squared so that the value of the slack term is always positive whatever the value of x_{N+1}. There is no general requirement for non-negativity of variables in non-linear programming. Adding slack variables to problem 8.20 converts it to problem 8.21.

$$\left.\begin{array}{l}\text{Minimise } f(x_1, \ldots, x_N) \\ \text{subject to } g_j(x_1, \ldots, x_N) + x_{N+j}^2 = 0 \quad j = 1, \ldots, N\end{array}\right\} \tag{8.21}$$

Problem 8.21 now has $N + J$ variables. It may now be solved by the Lagrange multiplier method for equality constraints. The Lagrangian function $L(\boldsymbol{x}, \lambda)$ is

$$L(\boldsymbol{x}, \lambda) = f(x_1, \ldots, x_N) + \sum_{j=1}^{J} \lambda_j [g_j(x_1, \ldots, x_N) + x_{N+j}^2] \tag{8.22}$$

and has $N + 2J$ variables (N problem variables, J slack variables and J Lagrange multipliers). The solution of problem 8.21 is found by finding the stationary point of $L(\boldsymbol{x}, \lambda)$ given by 8.22. These stationarity conditions are

$$\left.\begin{array}{ll}\dfrac{\partial L}{\partial x_i} = 0 & i = 1, \ldots, N \\ \dfrac{\partial L}{\partial x_i} = 0 & i = N+1, \ldots, N+J \\ \dfrac{\partial L}{\partial \lambda_j} = 0 & j = 1, \ldots, J\end{array}\right\} \tag{8.23}$$

www.EngineeringBooksPDF.com

Writing down these conditions for the Lagrangian function 8.22 gives

$$\left.\begin{aligned} \frac{\partial f}{\partial x_i} + \sum_{j=1}^{J} \lambda_j \frac{\partial g_i}{\partial x_i} &= 0 \qquad i = 1, \ldots, N \\ 2\lambda_j x_{N+j} &= 0 \qquad j = 1, \ldots, J \\ g_j(x_1, \ldots, x_N) + x_{N+j}^2 &= 0 \qquad j = 1, \ldots, J \end{aligned}\right\} \tag{8.24}$$

The middle one of these three sets of stationarity conditions 8.24 is interesting. It represents stationarity of the Lagrangian function with respect to the slack variables and for each constraint $j = 1, \ldots, J$ there will be a condition of the form

$$2\lambda_j x_{N+j} = 0$$

This condition implies that if the constraint j is not active at the optimum its slack variable x_{N+j} will have some non-zero value and so the Lagrange multiplier for that constraint must be zero. Conversely, if constraint j has a non-zero Lagrange multiplier, λ_j, then that constraint's slack variable, x_{N+j} must be zero in order to satisfy this condition and a zero value of x_{N+j} implies that constraint j must be active (solved as $g_j = 0$) at the optimum. Thus values of the Lagrange multipliers resulting from 8.24 indicate which constraints are active (non-zero multipliers) and which are slack (zero values for the multipliers). The three conditions 8.24 are often referred to as the *Kuhn-Tucker conditions* for constrained optimality.

8.2.2.1 Example

The example of section 8.1.3 previously solved by constraint deletions will be solved again by the Lagrange multiplier method. The problem is

$$\begin{aligned} &\text{Minimise } f = (x_1 + 2x_2 + 3x_3 - 4)^2 + (2x_1 + x_3)^2 + 4x_2^2 \\ &\text{subject to } x_1 + x_2 + x_3 \leqslant 1.5 \\ &\qquad\qquad -x_1 - 2x_2 + 3x_3 \leqslant 2 \end{aligned}$$

The constraints with slack variables added are

$$\begin{aligned} x_1 + x_2 + x_3 + x_4^2 - 1.5 &= 0 \\ -x_1 - 2x_2 + 3x_3 + x_5^2 - 2 &= 0 \end{aligned}$$

and the Lagrangian function $L(x_1, x_2, x_3, x_4, x_5, \lambda_1, \lambda_2)$ is

$$\begin{aligned} L(x,\lambda) = {} & (x_1 + 2x_2 + 3x_3 - 4)^2 + (2x_1 + x_3)^2 + 4x_2^2 \\ & + \lambda_1(x_1 + x_2 + x_3 + x_4^2 - 1.5) + \lambda_2(-x_1 - 2x_2 + 3x_3 \\ & + x_5^2 - 2) \end{aligned}$$

www.EngineeringBooksPDF.com

For this Lagrangian the stationarity conditions 8.25 are

$$\left.\begin{aligned}
2(x_1 + 2x_2 + 3x_3 - 4) + 4(2x_1 + x_3) + \lambda_1 - \lambda_2 &= 0 \quad \text{(a)}\\
4(x_1 + 2x_2 + 3x_3 - 4) + 8x_2 + \lambda_1 - 2\lambda_2 &= 0 \quad \text{(b)}\\
6(x_1 + 2x_2 + 3x_3 - 4) + 2(2x_1 + x_3) + \lambda_1 + 3\lambda_2 &= 0 \quad \text{(c)}\\
2\lambda_1 x_4 &= 0 \quad \text{(d)}\\
2\lambda_2 x_5 &= 0 \quad \text{(e)}\\
x_1 + x_2 + x_3 + x_4^2 - 1.5 &= 0 \quad \text{(f)}\\
-x_1 - 2x_2 + 3x_3 + x_5^2 - 2 &= 0 \quad \text{(g)}
\end{aligned}\right\} \qquad (8.25)$$

These seven equations may be solved to yield values for the seven unknowns $x_1, \ldots, x_5, \lambda_1$ and λ_2. Solving first conditions (a), (b) and (c) yields

$$\left.\begin{aligned}
x_1 &= -0.8 - 0.13\ \lambda_1 + 0.74\lambda_2 \quad \text{(h)}\\
x_2 &= \quad - 0.075\lambda_1 + 0.6\ \lambda_2 \quad \text{(m)}\\
x_3 &= \ 1.6 + 0.06\ \lambda_1 - 0.88\lambda_2 \quad \text{(n)}
\end{aligned}\right\} \qquad (8.25)$$

Substituting (h), (m) and (n) into (f) and (g) gives

$$\left.\begin{aligned}
-0.145\lambda_1 + 0.46\lambda_2 + x_4^2 &= 0.7 \quad \text{(p)}\\
\text{and}\\
0.46\ \lambda_1 - 4.58\lambda_2 + x_5^2 &= -3.6 \quad \text{(q)}
\end{aligned}\right\} \qquad (8.25)$$

These conditions are solved with the aid of (d) and (e). There are several possibilities to be investigated.

(1) Try $\lambda_1 = \lambda_2 = 0$ (no constraints active) therefore $x_4 \neq 0, x_5 \neq 0$. Condition (p) gives $x_4^2 = 0.7$ but (q) gives $x_5^2 = -3.6$ which cannot be true. Thus this possibility is not valid.

(2) Try $\lambda_1 \neq 0, \lambda_2 = 0$ (constraint g_1 active) therefore $x_4 = 0, x_5 \neq 0$. Conditions (p) and (q) become

$$\left.\begin{aligned}
-0.145\lambda_1 &= 0.7\\
0.46\ \lambda_1 + x_5^2 &= -3.6
\end{aligned}\right\}$$

Solving these gives $\lambda_1 = -4.828$ and $x_5^2 = -1.379$. This again cannot be valid since x_5^2 cannot be negative.

(3) Try $\lambda_1 = 0, \lambda_2 \neq 0$ (constraint g_2 active) therefore $x_4 \neq 0, x_5 = 0$. Conditions (p) and (q) become

$$\left.\begin{aligned}
0.46\lambda_2 + x_4^2 &= 0.7\\
-4.58\lambda_2 &= -3.6
\end{aligned}\right\}$$

which can be solved to give $\lambda_2 = 0.786$ and $x_4^2 = 0.664$. Substituting into (h), (m) and (n) gives

$$x_1^* = -0.218; x_2^* = 0.472; x_3^* = 0.908$$

and the corresponding value of $f^* = 1.415$. This is the solution obtained previously in section 8.1.3. There is a final possibility to be examined.

www.EngineeringBooksPDF.com

(4) Try $\lambda_1 \neq 0, \lambda_2 \neq 0$ (both constraints active) therefore $x_4 = x_5 = 0$. Conditions (p) and (q) become

$$\left.\begin{aligned} -0.145\lambda_1 + 0.46\lambda_2 &= 0.7 \\ 0.46\ \lambda_1 - 4.58\lambda_2 &= -3.6 \end{aligned}\right\}$$

which can be solved to give $\lambda_1 = -3.425, \lambda_2 = 0.442$. Equations (h), (m) and (n) yield

$$x_1 = -0.028; x_2 = 0.522; x_3 = 1.006$$

and the corresponding $f = 1.995$. This value of f is higher than the value previously achieved of $f^* = 1.415$

Thus the solution of the problem is

$$x_1^* = -0.218; x_2^* = 0.472; x_3^* = 0.908; f^* = 1.415$$

8.2.3 Comments on the Lagrange Multiplier Method

The worked examples of the Lagrange multiplier method, particularly the last one for inequality constraints, demonstrate that an analytical method does exist for constrained non-linear optimisation problems but is a very cumbersome method requiring considerable algebraic care if correct solutions are to be obtained. Rather like the analytical methods for unconstrained optimisation, the Lagrange multiplier method cannot be thought of as a general solution method. There are very many more problems it cannot solve than those it can solve. It requires that all the functions f and g are algebraic, continuous and differentiable. Even if these conditions are satisfied it requires that the functions are easily amenable to solution. It establishes sets of simultaneous equations which must be solved. If these equations turn out to be non-linear in the variables they may not be easily soluble. There is a direct parallel between Lagrange multiplier methods and the analytical solution of unconstrained problems in that both methods are interesting but not of general practical value.

In order to solve more general constrained optimisation problems something better than Lagrange multipliers is needed. The remainder of this chapter is devoted to a study of methods which use numerical search procedures to solve constrained optimisation problems. The next section examines penalty function methods which, though numerical search is used, owe a lot in their solution philosophy to Lagrangian methods.

8.3 PENALTY FUNCTION METHODS

Lagrange multiplier methods solve constrained optimisation problems by adding the problem constraints to the objective function to form a Lagrangian function and then finding an unconstrained stationary point of the Lagrangian. Thus a constrained problem is solved by converting it to an unconstrained one. Penalty

www.EngineeringBooksPDF.com

function methods work in a very similar way; the constraints are added to the objective function by means of penalty coefficients and an unconstrained minimum of this new objective function is sought. This process is repeated several times each time using different values of the penalty coefficients selected in such a way that the sequence of unconstrained optima thus found converges to the solution of the original constrained optimisation problem.

In order to demonstrate how penalty function methods work consider firstly a problem with only equality constraints.

8.3.1 Penalty Functions for Equality-constrained Problems

Problem 8.1 is a typical equality-constrained problem and is restated below as problem 8.26.

$$\left.\begin{array}{l}\text{Minimise } f(x_1, \ldots, x_N) \\ \text{subject to } g_j(x_1, \ldots, x_N) = 0 \quad j = 1, \ldots, J\end{array}\right\} \tag{8.26}$$

This problem is converted to an unconstrained problem of the form 8.27

$$\underset{\boldsymbol{x}}{\text{Minimise}}\, P(\boldsymbol{x}, R) = f(x_1, \ldots, x_N) + R \sum_{j=1}^{J} [g_j(x_1, \ldots, x_N)]^2 \tag{8.27}$$

The function $P(\boldsymbol{x}, R)$ in 8.27 is known as the *penalty function* and it consists of two components; the first is the objective function $f(x_1, \ldots, x_N)$ of problem 8.26 and the second is an additional term which is known as the *penalty term* because it increases the value of, or penalises, the penalty function $P(\boldsymbol{x}, R)$. The penalty term consists of the square of each constraint function all summed together and multiplied by a constant, R. Note that the minimisation is to be carried out over variables $\boldsymbol{x}$ only. R is a prescribed constant, not a variable.

Examining problems 8.26 and 8.27 more carefully, suppose that the penalty coefficient R has the value zero. Then whatever the values of $g_j(x_1, \ldots, x_N)$ are for all j, the value of the penalty term will be zero and $P(\boldsymbol{x}, R)$ in problem 8.27 will be equal to $f(x_1, \ldots, x_N)$. For $R = 0$, therefore, problem 8.27 is equivalent to an unconstrained minimisation of $f(x_1, \ldots, x_N)$. The solution point of this problem $(x_1^u, \ldots, x_N^u)$ will minimise $f(x_1, \ldots, x_N)$ in problem 8.26 but will probably violate some of the J equality constraints. Now consider the two problems 8.26 and 8.27 again, this time with a small positive value of R. The previous solution for $R = 0$, $(x_1^u, \ldots, x_N^u)$, probably violated some of the constraints g_j of problem 8.26. Thus for some constraints, say $j = 1, \ldots, j'; j' \leqslant J$

$$g_j(x_1^u, \ldots, x_N^u) \neq 0$$

The value of the penalty term will, therefore, be positive, that is

$$R \sum_{j=1}^{J} [g_j(x_1^u, \ldots, x_N^u)]^2 > 0$$

www.EngineeringBooksPDF.com

The value of this penalty term will increase as R increases until it becomes much larger than the value of $f(x_1^u, \ldots, x_N^u)$. When this happens, at some large value of R, if problem 8.27 is solved again the minimising point can be expected to move from $(x_1^u, \ldots, x_N^u)$ to some new point at which the value of the dominant penalty term is greatly reduced in value. In fact if the new point is one which satisfies all the equality constraints of problem 8.26 then all the g_j^2 values in the penalty term will be zero and the value of the penalty term itself will be zero. Thus the new solution point of problem 8.27 with R large must minimise $f(x_1, \ldots, x_N)$ and at the same time make each of the constraint functions $g_j(x_1, \ldots, x_N)$ equal to zero. This point must, therefore, be $(x_1^*, \ldots, x_N^*)$, the solution point of problem 8.26.

Essentially, problem 8.27 has been constructed so that when it is solved for a very large value of the coefficient R its solution point must also solve the constrained problem 8.26. The effect of increasing the value of R from zero towards some large value is to move from a point that is an unconstrained minimum of $f(x_1, \ldots, x_N)$ but which violates the constraints, towards a point which satisfies all the constraints as well as minimising f. Increasing R tends to tighten up the constraints. This can be seen if a numerical example is examined.

8.3.1.1 Example

The example chosen is that of section 8.2.1.1 already solved by Lagrange multipliers

$$\begin{aligned} &\text{Minimise } f = \tfrac{1}{2}(x_1^2 + x_2^2 + x_3^2) \\ &\text{subject to } g_1 \equiv x_1 + 2x_2 + 3x_3 - 1 = 0 \\ &\qquad\qquad g_2 \equiv 3x_1 + 2x_2 + x_3 - 2 = 0 \end{aligned}$$

The penalty function problem equivalent to this is

$$\begin{aligned} \underset{x_1, x_2, x_3}{\text{Minimise}}\, P(x, R) = {} & \tfrac{1}{2}(x_1^2 + x_2^2 + x_3^2) \\ & + R\,\{(x_1 + 2x_2 + 3x_3 - 1)^2 + (3x_1 + 2x_2 + \\ & + x_3 - 2)^2\} \end{aligned}$$

Table 8.1 The effects of different values of the coefficient R

R	x_1	x_2	x_3	g_1	g_2	f	P
0	0	0	0	-1	-2	0	0
10^{-2}	0.09959	0.00547	0.00004	-0.88935	-1.69025	0.00497	0.04145
1	0.46712	0.24476	0.03663	0.06653	-0.07249	0.13973	0.14941
10^2	0.49964	0.24995	0.00026	0.00032	-0.00092	0.15606	0.15615
10^4	0.49999	0.24999	0.00000	0.00000	-0.00001	0.15625	0.15625
∞	0.5	0.25	0	0	0	0.15625	0.15625

www.EngineeringBooksPDF.com

Table 8.1 shows the solution of this problem for different values of R. Starting with $R = 0$, values of the variables x_1, x_2, x_3 which minimise P violate both constraints g_1 and g_2. As R increases, constraint violations become smaller and smaller until when $R = 10^4$ the values of P and f are identical to five decimal places. This represents a fully converged solution of the problem in a numerical sense. The final row of table 8.1 gives the problem solution found in section 8.2.1.1 which corresponds theoretically to a value of $R = \infty$.

8.3.2 Penalty Functions for Inequality-constrained Problems

When inequality constraints are present they too may be included in a penalty term which is added to the objective function. Problem 8.20 is an inequality-constrained problem which is restated here

$$\begin{aligned} &\text{Minimise} \quad f(x_1, \ldots, x_N) \\ &\text{subject to } g_j(x_1, \ldots, x_N) \leqslant 0 \quad j = 1, \ldots, J \end{aligned} \tag{8.28}$$

For this problem an appropriate form of the penalty function is

$$\underset{\boldsymbol{x}}{\text{Minimise}}\ P(\boldsymbol{x}, R) = f(x_1, \ldots, x_N) - R \sum_{j=1}^{J} [g_j(x_1, \ldots, x_N)]^{-1} \tag{8.29}$$

In problem 8.29 the penalty term consists of the sum of the reciprocals of the constraint functions $g_j, j = 1, \ldots, J$. Since feasible values of $x_1, \ldots, x_N$ cause the g_j to have negative values the penalty term will be negative and, for a positive penalty coefficient R, it must therefore be subtracted from $f(\boldsymbol{x})$ in order to make P larger than f (for minimisation problems). Consider problem 8.29 with some large positive value of R. Consider a feasible point $(x_1, \ldots, x_N)$. If this feasible point lies anywhere near the inequality constraint boundaries it must cause at least one of the g_j to have a very small negative value. The reciprocal of that g_j will then be very large and negative and when this is multiplied by a large positive R and subtracted from f it will give P a very large positive value. Thus the penalty term becomes very large near the constraint boundaries. This is just the opposite of the effect seen with equality constraints where the penalty term becomes zero when constraints are tight. With inequality constraints the penalty term keeps the minimising point of P away from the constraint boundaries. For large R, a numerical solution of problem 8.29 must find a solution point $(x_1^1, \ldots, x_N^1)$ which is not near the constraint boundaries. Suppose this problem has been solved giving $(x_1^1, \ldots, x_N^1)$ as the point which minimises P. If the value of the penalty coefficient R is then reduced, say by a factor of ten, and problem 8.29 is solved again it is possible for values of the $g_j, j = 1, \ldots, J$, to reduce in value without increasing the over-all penalising value of the penalty term. The new solution point $(x_1^2, \ldots, x_N^2)$ of this second problem can, therefore, approach closer to any of the constraint boundaries if this leads to a reduced value of P. The solution of problem 8.29 is, therefore, carried out several

www.EngineeringBooksPDF.com

times, each time with a reduced value of the coefficient R. Eventually as R tends towards zero those constraints in problem 8.28 which are tight (= 0) at the optimum of problem 8.28 are allowed to approach $g_j = 0$ from the negative side. Those constraints which are slack (< 0) at the optimum of problem 8.28 can remain slack with large negative values of g_j because their reciprocals will be relatively small and will be negligible when multiplied by a small R.

A simple single-variable problem demonstrates graphically how the penalty function solution operates for inequality constraints. Figure 8.1 shows the problem

$$\left.\begin{array}{l}\text{Minimise } f(x_1) \\ \text{subject to } g_1 \equiv x_1 - 3 \leqslant 0\end{array}\right\} \tag{8.30}$$

It can be seen that the function $f(x_1)$ decreases as x_1 increases until it meets the constraint boundary $x_1 = 3$. The solution of problem 8.30 is, therefore, at $x_1^* = 3$ and constraint g_1 is tight at the solution. The penalty function problem of problem 8.30 is 8.31

$$\underset{x_1}{\text{Minimise}}\ P(x_1, R) = f(x_1) - R(x_1 - 3)^{-1} \tag{8.31}$$

Starting with a large value of R the first penalty function P_1 is shown on figure 8.1. Note that when x_1 is small, P_1 lies close to $f(x_1)$ and that P_1 diverges from $f(x_1)$ as x_1 increases. As x_1 approaches the value $x_1 = 3$, P_1 becomes very large. The minimum value of P_1 is shown encircled and is nowhere near the solution of problem 8.30. The value of R is then reduced giving the function P_2.

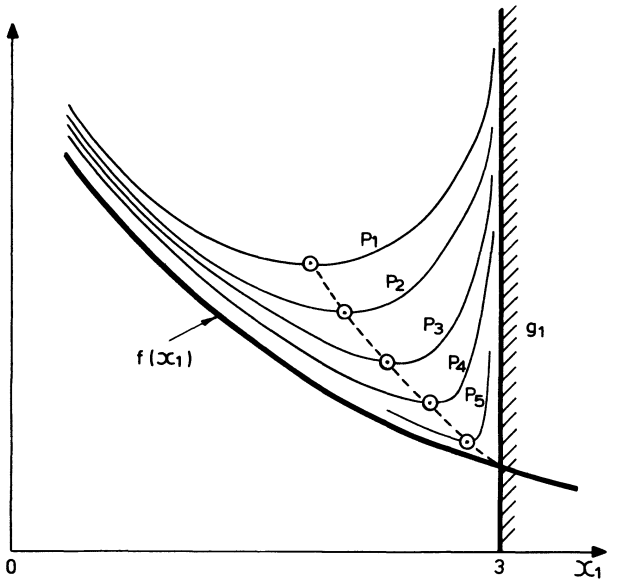


Figure 8.1 An interior penalty function

www.EngineeringBooksPDF.com

Minimisation of P_2 locates a new optimum which is slightly closer to the solution of problem 8.30. Successive reductions in the value of R give new penalty functions P_3, P_4, P_5, etc., and their successive minima approach the solution of problem 8.30 closer and closer with each reduction in R.

The essence of this penalty function method is that the solution of problem 8.30 is achieved by a sequence of solutions of problem 8.31 for a decreasing set of values of R. Problem 8.31 is, of course, an unconstrained optimisation problem and consequently this solution method is sometimes called the SUMT method (*sequential unconstrained minimisation technique*). The unconstrained optimisation methods described in the previous chapter may, therefore, be used to solve constrained problems via the penalty function approach. Some aspects of the numerical solution of penalty function problems can cause difficulties, however, and are worth noting.

Firstly, figure 8.1 shows that a feasible point (one which does not violate the constraint g_1) is needed to start the unconstrained minimisation. The penalty function P does not generally exist and, therefore, cannot be evaluated in the infeasible (cross-hatched) region. For this reason the particular form of the penalty function P defined in 8.29 and 8.31 is an *interior penalty function*. P is valid only in the interior of the feasible region. A feasible point is, therefore, needed from which to start the minimisation of the first penalty function P_1. It may not always be easy to specify a feasible starting point particularly if there are many constraints in the original problem. There are rigorous methods for finding feasible points for any constraint set provided such points exist but they are rather time-consuming to use and are beyond the scope of this book to explain in detail. Fortunately, for most engineering design problems it is not difficult to specify a feasible initial design. Structural elements, for example, may be specified to be very large and oversized in an initial design so as to satisfy any stress or displacement constraints. The need for a feasible starting point, however, can sometimes be a source of difficulty. Fortunately only one feasible starting point is needed for the whole sequence of minimisations. The minimising point for the first penalty function must automatically be feasible and can therefore be used as a starting value for the second penalty function minimisation, and so on.

Secondly, it is necessary to specify an initial value for the coefficient R of the first penalty function P_1 and also some reduction scheme for changing the value of R for the succeeding minimisations. The choice of R is arbitrary though much research has been devoted to finding ways of choosing R. Specialist texts should be consulted. An injudicious choice of R or the reduction factor can result in very slow convergence to the solution of the constrained problem and many minimisations may be required. Referring to figure 8.1 it can be seen that the sequence of unconstrained minima appear to lie on a smooth curve passing through the solution of the problem. Some penalty function methods have successfully used extrapolation techniques to generate the equation of this curve from the sequence of minima and then to extrapolate it to predict the solution of the problem. Space precludes a detailed study of how this can be done.

www.EngineeringBooksPDF.com

Often a zeroth-order unconstrained optimisation method is used for the sequence of minimisations. Powell's method is popular for this. The penalty function P is discontinuous because it does not generally exist outside the feasible region, and although first derivatives may be analytically available, they become infinitely large at the constraint boundaries so first-order methods can experience difficulties there. Whatever method is used should incorporate checks for constraint violations. When a constraint violation is detected, P and its derivatives should be allocated very high values thus directing the search back into the feasible region.

The general efficiency and speed of convergence of penalty function solutions is usually improved if the constraints are *normalised*. In fact normalisation usually improves the efficiency of any constrained optimisation method. Normalisation consists of altering all the constraints so that the constant term is unity. For example, consider two possible constraints in a structural design problem

$$\left.\begin{aligned} & g_1(\boldsymbol{x}) \equiv \delta(\boldsymbol{x}) - \delta_{\max} \leqslant 0 \\ \text{and} \\ & g_2(\boldsymbol{x}) \equiv \sigma(\boldsymbol{x}) - \sigma_{\max} \leqslant 0 \end{aligned}\right\} \tag{8.32}$$

Constraint g_1 requires that the displacement $\delta(\boldsymbol{x})$ of some point in the structure must be less than $\delta_{\max}$ which might typically be 10 mm. Constraint g_2 requires the stress $\sigma(\boldsymbol{x})$ at some point not to exceed $\sigma_{\max}$ which might typically be $200\,\mathrm{N/mm^2}$. Consider a set of variables $\boldsymbol{x}$ at which point the actual displacement $\delta(\boldsymbol{x})$ and stress $\sigma(\boldsymbol{x})$ are equal to half their limit values. Thus $\delta(\boldsymbol{x}) = 5$ and $\sigma(\boldsymbol{x}) = 100$. The values of the two constraints are then $g_1 = -5$ and $g_2 = -100$. Their reciprocals are -0.2 and -0.01 respectively and in a penalty function such as that of problem 8.29 it is clear that the term g_1^{-1} will have twenty times the penalising effect of the term g_2^{-1}. This is unrepresentative of the fact that displacement and stress are both equal to half their limit values and should, therefore, have approximately equal penalising effects.

Normalisation of the constraints achieves a much more uniform balance among the constraints. The constraints are rewritten so that the constant term in each is unity, that is

$$\left.\begin{aligned} & g \equiv \frac{\delta(\boldsymbol{x})}{\delta_{\max}} - 1 \leqslant 0 \\ \text{and} \\ & g_2 \equiv \frac{\sigma(\boldsymbol{x})}{\sigma_{\max}} - 1 \leqslant 0 \end{aligned}\right\} \tag{8.33}$$

Using these constraints, $g_1^{-1} = g_2^{-1} = -2.0$ when $\delta(\boldsymbol{x})$ and $\sigma(\boldsymbol{x})$ are half $\delta_{\max}$ and $\sigma_{\max}$, respectively. Normalisation of constraints so that they all have similar effects is almost always beneficial in constrained optimisation problems and is particularly effective in penalty function methods.

www.EngineeringBooksPDF.com

The reciprocal penalty function in problem 8.29 is perhaps the most popular interior penalty function for inequality constraints but is not the only one possible. Another possible form for problem 8.29 is

$$\underset{\boldsymbol{x}}{\text{Minimise}}\ P(\boldsymbol{x}, R) = f(x_1 \ldots, x_N) - R \sum_{j=1}^{J} \log[-g_j(x_1, \ldots, x_N)] \quad (8.34)$$

This logarithmic penalty function is very similar to the reciprocal one in that it too creates a barrier near the constraint boundaries. In addition to interior penalty function methods which operate within the feasible region and converge feasibly to a solution, there are *exterior penalty function* methods which operate very similarly to the interior ones but in the infeasible region, converging infeasibly to a solution which is feasible. A typical exterior penalty function for problem 8.28 is

$$\underset{\boldsymbol{x}}{\text{Minimise}}\ P(\boldsymbol{x}, R) = f(x_1, \ldots, x_N) + R \sum_{j=1}^{J} [f_j'(g_j)]^2 \quad (8.35)$$

in which the function $f_j'(g_j)$ takes the value zero if constraint j is satisfied (that is, if $g_j \leqslant 0$) and the value g_j if constraint j is violated (that is, if $g_j > 0$). Thus the penalty term will be positive if any constraints are violated and zero only when all constraints are satisfied. Problem 8.35 is solved several times for decreasing values of the penalty coefficient R. One advantage of exterior penalty functions is that they do not require a feasible starting point to be known or found, an infeasible point will suffice. A balancing disadvantage, however, is that the individual minima found in the sequence of minimisations are generally infeasible. In engineering design problems an interior penalty function is often preferred because the individual sequential minima are all feasible. An interior method can always be terminated before complete convergence of the sequence has been obtained and the previous minimum will represent a feasible design. This enables an engineer to monitor the sequence and terminate the solution when he is satisfied with a design. In the exterior approach a feasible design is not achieved until the entire sequence is fully converged.

Penalty function methods have been widely used in an engineering design context because they are reliable and robust. Their continuing popularity is well-deserved. Experience with them has tended to show that they work best on relatively small problems with few variables and constraints. For large problems many sequential minimisations, each requiring considerable computing time, are often necessary for convergence. Poor performance on large non-linearly constrained problems, however, is also an attribute of most other methods and does not particularly detract from penalty function methods. This section has only introduced the ideas which underlie penalty functions. Many different penalty functions have been suggested and research into improved methods continues. The bibliography at the end of the chapter lists several references for further study.

www.EngineeringBooksPDF.com

8.4 LINEARISATION METHODS

As was mentioned above, constrained non-linear optimisation problems with large numbers of variables and constraints can be very hard and tedious to solve. Most NLP methods perform poorly on large constrained problems. On the other hand very large linear programming problems can be solved simply and rapidly by the simplex method as described in chapter 3. Consequently it is very tempting to try to adapt the simplex LP method to solve non-linear problems. One idea is to replace the non-linear problem by a sequence of linear problems which can be solved by the simplex method and which converge to the solution of the non-linear problem. This approach is sometimes called *sequential linear programming*.

8.4.1 Sequential Linear Programming (SLP)

Consider the general inequality-constrained non-linear problem 8.21, restated below as problem 8.36.

$$\left.\begin{array}{l}\text{Minimise } f(x_1, \ldots, x_N) \\ \text{subject to } g_j(x_1, \ldots, x_N) \leqslant 0 \quad j = 1, \ldots, J\end{array}\right\} \tag{8.36}$$

Suppose a feasible point $\boldsymbol{x}' \equiv (x_1', \ldots, x_N')$ is known for this problem. All the non-linear functions f and $g_j, j = 1, \ldots, J$ in 8.36 may be linearised about the point $\boldsymbol{x}'$ by means of a Taylor series expansion truncated after the linear terms. Thus linearising problem 8.36 gives problem 8.37

$$\begin{array}{l}\text{Minimise } f(\boldsymbol{x}') + \sum_{i=1}^{N} \left(\frac{\partial f}{\partial x_i}\right)_{\boldsymbol{x}'} (x_i - x_i') \\ \text{subject to } g_j(\boldsymbol{x}') + \sum_{i=1}^{N} \left(\frac{\partial g_j}{\partial x_i}\right)_{\boldsymbol{x}'} (x_i - x_i') \leqslant 0 \quad j = 1, \ldots, J\end{array} \tag{8.37}$$

Rearranging problem 8.37 gives the problem

$$\begin{array}{l}\text{Minimise } \left\{ f(\boldsymbol{x}') - \sum_{i=1}^{N} x_i' \left(\frac{\partial f}{\partial x_i}\right)_{\boldsymbol{x}'} \right\} + \sum_{i=1}^{N} \left(\frac{\partial f}{\partial x_i}\right)_{\boldsymbol{x}'} x_i \\ \text{subject to } \left\{ g_j(\boldsymbol{x}') - \sum_{i=1}^{N} x_i' \left(\frac{\partial g_j}{\partial x_i}\right)_{\boldsymbol{x}'} \right\} + \sum_{i=1}^{N} \left(\frac{\partial g_j}{\partial x_i}\right)_{\boldsymbol{x}'} x_i \leqslant 0 \quad j = 1, \ldots, J\end{array} \tag{8.38}$$

The term within { } in the objective function is a constant which may be omitted for optimisation purposes. In each of the constraints the { } term is also a positive or negative constant which may be transferred to the right-hand side of the inequality. The right-hand side may be made positive if it is negative by multi-

www.EngineeringBooksPDF.com

plying through by -1 and changing the direction of the inequality. Problem 8.38 can then be written in the form

$$\left.\begin{aligned} &\text{Minimise } \sum_{i=1}^{N} a_i x_i \\ &\text{subject to } \sum_{i=1}^{N} b_{ji} x_i \geqslant c_j \quad j = 1, \ldots, J_1 \\ &\qquad\qquad \sum_{i=1}^{N} b_{ji} x_i \leqslant c_j \quad j = J_1 + 1, \ldots, J \end{aligned}\right\} \tag{8.39}$$

In problem 8.39 the $\geqslant$ constraints have been grouped together, as have the $\leqslant$ constraints. Problem 8.39 can be compared with the general form for linear programming problems, 2.26 and 3.1. It is identical apart from the absence of constraints requiring all variables x_i to be non-negative, $i = 1, \ldots, N$. These may be added if all variables in the original problem 8.36 must be non-negative. If 8.36 contains variables which are unrestricted in sign, problem 8.39 must be modified to permit this as described in section 3.5 of chapter 3. The end product is a standard LP problem which can be solved by the simplex method to yield an optimum point, $(x_1^*, \ldots, x_N^*)$.

This optimising point for the first LP problem must then be substituted into the non-linear problem 8.36 to check that the solution of the linearised problem 8.39 does not violate any of the original non-linear constraints $g_j, j = 1, \ldots, J$. Assuming that no constraint violations occur, the point $(x_1^*, \ldots, x_N^*)$ replaces $(x_1', \ldots, x_N')$ as a starting point for a completely new linear approximation 8.39, to problem 8.36. This new LP problem is solved yielding a new solution point which is used to set up a new linear approximation. Eventually the non-linear problem 8.36 will be solved by this converging sequence of LP problems.

Attractive though this method appears to be there are some serious difficulties and disadvantages. Firstly, in order to set up each LP problem all the coefficients a_i, b_{ji} and c_j in problem 8.39 must be calculated. This requires that values are calculated at $\boldsymbol{x}'$ for all functions f and g_j of problem 8.36 and for all first derivatives of f and all g_j. When analytical derivatives are not available, numerical derivatives must be calculated using, for example, equation 7.17. This can be a time-consuming and approximate process when the problem has many variables and constraints; furthermore it is a process which must be repeated at each iteration of the sequential solution process. Secondly, the need to keep solutions feasible presents several difficulties. Each approximating LP problem must be established at a feasible point of the original problem. Thus for the very first of the sequence of linear approximating problems a feasible point $(x_1', \ldots, x_N')$ must be found for the original non-linear constraint set. Methods have been devised for doing this but they can be complicated and time-consuming. More seriously, however, even

www.EngineeringBooksPDF.com

if such a point is found and the problem linearised and solved by the simplex method the solution $(x_1^*, \ldots, x_N^*)$ of this first linear problem may not be a feasible point for the original constraints. Figure 8.2 shows how this can occur.

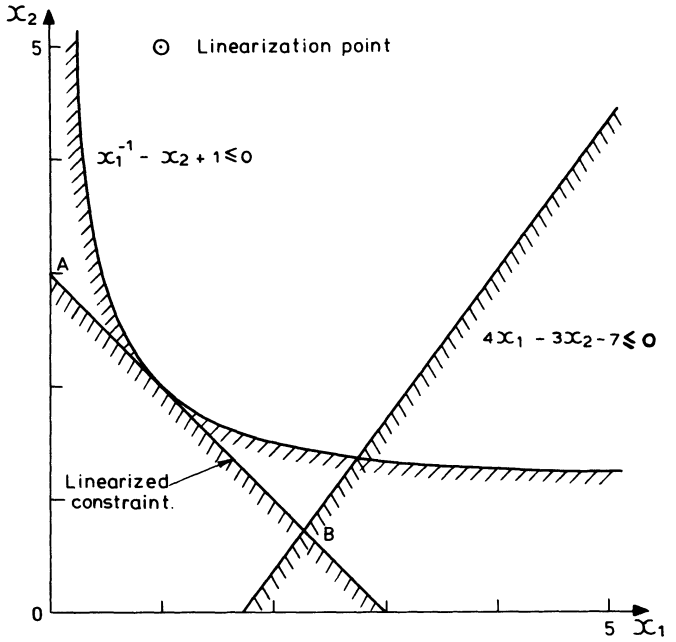


Figure 8.2 Infeasibility of linearised constraints

Figure 8.2 shows two constraints

$$x_1^{-1} - x_2 + 1 \leqslant 0$$

and

$$4x_1 - 3x_2 - 7 \leqslant 0$$
$$x_1, x_2 \geqslant 0$$

The first of these is non-linear and is the curved boundary. The second is linear. The point (1, 5) is feasible for both constraints and may be used as a starting point at which to linearise the constraints. If this is done the constraints become

$$-x_1 - x_2 + 3 \leqslant 0$$

and

$$4x_1 - 3x_2 - 7 \leqslant 0$$

Linearisation of the non-linear constraint gives the tangential constraint boundary shown in figure 8.2. The second constraint naturally remains linear. If

www.EngineeringBooksPDF.com

a linear or linearised objective function is known and the LP problem is solved by the simplex method the solution must lie at a vertex of the linear boundaries. Two vertices exist, A and B, which are feasible for the linearised constraints and one of these must be the solution (x_1^*, x_2^*) of the first approximating LP problem. Yet both these candidate LP solution points violate the original non-linear constraint. Neither of them could be used to set up a second approximating LP problem because they are infeasible. The danger of trying to linearise at an infeasible point can also be demonstrated by this example. Suppose vertex A was the solution of the first LP problem. A is the point (0, 3) and is infeasible for the non-linear constraint. If this constraint is linearised at (0, 3) using the Taylor series expansion as in problem 8.37 values of the constraint function and its first derivatives at (0, 3) will be needed. The function value, however, turns out to be infinite and the first derivative with respect to x_1 is also infinite and so no usable linear approximation exists to the non-linear constraint at the infeasible point (0, 3). If the solution point of an approximating LP problem turns out to be infeasible with respect to the original constraints it cannot be used to set up the next linearised problem; feasibility must somehow be restored before a new linearisation is made. This raises the question of how best to move from an infeasible point to a feasible one without excessively increasing the value of an objective function which has just been carefully minimised. Methods have been suggested for restoring feasibility but they are not rigorous and can be time-consuming. The possibility of infeasible points and the use of *ad hoc* methods to restore feasibility destroys any hope of proving that in general a sequence of approximating LPs will converge to the solution of a non-linear problem. The assumption of convergence is, therefore, not founded upon fact but on intuition. Unfortunately, it is very easy to find examples in which the sequence of LPs diverges instead of converging.

Perhaps the most serious disadvantage of sequential linear programming is that the solution philosophy embodied in linear programming is just inappropriate for non-linear problems. In chapter 6 it was shown that the solution of a non-linear problem may be unconstrained (even when possible constraints are present), may lie on constraint boundaries or may lie at a constraint vertex. Figure 6.3 shows this. Linear programming problems, however, always have solutions at constraint vertices. Thus the process of linearisation of a non-linear problem sets up a problem which has a vertex solution. The sequence of linear programming problems, if it converges at all, must converge upon a vertex of the non-linear constraints. An unconstrained optimum or an optimum on a constraint boundary cannot be found by the method described. This is a very serious defect of the method and many attempts have been made to remedy it. The most widely encountered remedial measure consists of adding to the constraint set of the linearised problem an extra set of move limit constraints. These constraints provide an upper and a lower bound upon the value of each variable in the problem, thereby limiting the amount by which any variable may be altered. They form a sort of box or hypercube around the approximating point

www.EngineeringBooksPDF.com

and the LP problem is solved within the box. By providing these artificial constraints it is possible for the sequence of linear programs to venture away from the original constraint vertices and optima which lie on constraint boundaries or within the unconstrained region of the objective function can in theory be found. The remedy is not without its own snags however. The size of the move limits must be specified and these is usually little available information on which to base these sizes. If the limits are made too small and tight very many approximating LPs may be needed to locate a possible optimum. If they are too large they are effectively absent and do not serve the intended purpose of assisting convergence.

Despite the deficiencies of the SLP method it has been used successfully to solve a variety of engineering design problems. It has been used particularly for structural design problems in conjunction with the finite element method of analysis. Finite element analyses are carried out on an initial feasible design to establish the necessary data for an approximating LP problem which is solved to yield a new design. This analysis/redesign cycle is continued until a satisfactory optimum design is found. Naturally, since the finite element method and the simplex method are both heavy users of computer storage, this design approach requires very large computational facilities and hence is costly. Unfortunately, it often fails to produce results for the reasons given above and cannot generally be recommended except as a last resort. Most success with the method has been reported on problems in which the original objective function is a linear one or is almost linear. Experience with SLP has tended to show that the linearisation of non-linear constraint functions does not necessarily invite disaster and can often be advantageous; linearisation of a non-linear objective function is rarely successful.

8.4.2 Sequential Quadratic Programming (SQP)

Since difficulties arise when a non-linear objective function is linearised, it is logical to try to retain much of the computational simplicity of linear programming and to make a non-linear approximation of the objective function instead of a linear one. A problem in which all the constraints are linear and the objective function is a quadratic, that is, has the form

$$f(x_1, \ldots, x_N) = \sum_{i=1}^{N} a_i x_i + \sum_{i=1}^{N} \sum_{j=1}^{N} b_{ij} x_i x_j \tag{8.40}$$

is said to be a problem of *quadratic programming* (QP). In 8.40 the as and bs are constants. Sequential quadratic programming is similar in approach to SLP in that a general non-linear constrained problem is solved by a sequence of approximating problems each of which is a QP problem. The non-linear constraints are linearised about some feasible point as in 8.37 and the non-linear objective function is approximated by a quadratic function 8.40 at the same point. A

www.EngineeringBooksPDF.com

Taylor series expansion truncated after the second-order terms can be used for this. If $\boldsymbol{x}'(\equiv x_1', \ldots, x_N')$ is the approximation point, $f(x_1, \ldots, x_N)$ is approximated as

$$f(x_1, \ldots, x_N) = f(x_1', \ldots, x_N') + \sum_{i=1}^{N} (x_1 - x_1') \left(\frac{\partial f}{\partial x_i}\right)' + \sum_{i=1}^{N} \sum_{j=1}^{N} (x_i - x_i')(x_j - x_j') \left(\frac{\partial^2 f}{\partial x_i \partial x_j}\right)' \tag{8.41}$$

The constant as and bs in 8.40 are found by comparison with 8.41. To determine all these constants all the second derivatives of $f(\boldsymbol{x})$ must be evaluated at $\boldsymbol{x}'$. If N is large this can be a considerable task and means that each QP problem in the sequence is more costly to set up than the corresponding LP problem.

The big advantage which SQP has over SLP is that the quadratic objective function allows solutions to be found anywhere within the feasible region or on the constraint boundaries. This overcomes the main objection to SLP, that it is a constraint vertex-seeking method. SQP imposes no restrictions on where the optimum may lie.

The only difference between LP and QP problems lies in the nature of the objective functions and, because of this, QP problems can be solved by a simplex LP method suitably modified to accept a quadratic objective function. The tabular solution approach described in chapter 3 may be used with each table set up as described there, that is, slack variables are introduced, the constraint equations are used to divide the variables into independent and dependent groups and the constraints then form the body of the table. In LP the row corresponding to the objective function contains the linear objective function coefficients. In QP the elements of this row are the N first derivatives of the quadratic function. Each of these derivatives is, of course, a linear function of the independent variables. Differentiating 8.40 gives the kth element of the objective function row as:

$$\frac{\partial f}{\partial x_k}(x_1, \ldots, x_N) = a_k + \sum_{i=1}^{N} 2b_{ik}x_i \quad k = 1, \ldots, N \tag{8.42}$$

At the approximating point, $\boldsymbol{x}'$, each of these elements can be evaluated by substituting $\boldsymbol{x} = \boldsymbol{x}'$ in 8.42. The table now looks exactly the same as a simplex table and the solution proceeds along the general lines of the simplex method but with some additions. In the simplex LP method an independent variable is chosen and is altered in value so that f reduces. The amount by which the chosen variable is altered is governed by two criteria; the variable itself must not become negative nor must it alter by an amount such that any dependent variable becomes negative. In quadratic programming a third criterion is added to these; the chosen variable may not be altered by an amount such that its objective function coefficient equation, 8.42, changes sign.

www.EngineeringBooksPDF.com

Since 8.42 is linear, the critical value of the chosen variable at which 8.42 becomes zero is simple to evaluate. When an appropriate change in value of the chosen independent variable has been found all the N objective function coefficients 8.42 are re-evaluated at the new point. If any dependent variable has been driven to zero a pivoting operation removes it from the dependent variables and replaces it with a non-zero independent variable exactly as in the simplex LP method. If no dependent variable has reached zero another independent variable is chosen to be altered so that f will decrease and the process is repeated. Thus the solution of a QP problem is very similar to, though slightly more complicated than, the simplex LP method.

The ability of the method to find a solution point anywhere in the feasible region means that the termination criteria are slightly more complicated than in the simplex LP method. One criterion, that all objective function coefficients are positive with all independent variables zero, is exactly as in the simplex LP method and corresponds to a vertex solution. Another termination criterion is that some of the objective function coefficients are zero whilst the rest satisfy the first criterion. This represents a solution point on the constraint boundaries but not at a vertex. A third criterion for termination is that all the objective function coefficients are zero. This corresponds to an unconstrained optimum.

SQP is much less popular than SLP despite the fact that it is a much more robust and reliable method. Its disadvantages are that more computational effort is needed to set up quadratic rather than linear programming problems. Also each QP problem is more time-consuming in solution than the corresponding LP problem. Very large problems, however, can be solved by sequential quadratic programming and in almost all respects it is recommended in preference to sequential linear programming.

SLP and SQP are not the only methods which attempt to solve constrained non-linear optimisation problems by a sequence of much simpler approximating problems. Geometric programming can also be used in a similar fashion (see section 8.6).

8.5 DIRECT NUMERICAL SEARCH METHODS

In chapter 7 when examining unconstrained optimisation problems, several direct numerical search methods were described which choose a search direction, search along it for a line minimum, choose a new search direction, search again, etc., until a solution is found. It is logical to try to do the same sort of thing with a constrained non-linear problem making appropriate allowance for the presence of constraint boundaries. Much work has been done on developing direct search methods for constrained problems. It is not proposed, however, to examine these methods here for the reason that they have proved to be much inferior to the other methods described in this chapter. If direct numerical search is to be used it is generally more efficient to use it on an unconstrained problem resulting from incorporating constraints by means of penalty functions, than to use it on

www.EngineeringBooksPDF.com

the original constrained problem. Sequential approximation methods using quadratic or geometric approximations are also much preferable to direct numerical search on constrained problems.

The difficulties which make constrained numerical search uncompetitive arise as a result of the constraints. Figure 8.3 shows some of these difficulties. Non-linear objective function contours are shown and a curved non-linear constraint boundary.

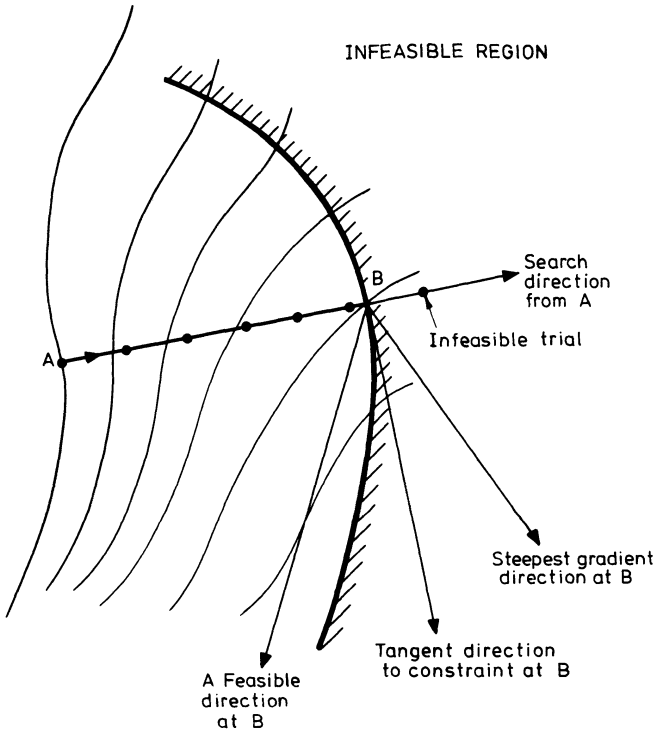


Figure 8.3 Feasible and infeasible search directions

Suppose that from some feasible point A a search direction has been chosen. In figure 8.3 a steepest gradient direction normal to the contour at A is shown. The numerical search along this direction is more complicated than in an unconstrained problem because not only must the objective function be calculated at each trial point but also all the constraints must be evaluated to check that none of them has been violated. Thus each trial requires much more calculation work than in unconstrained problems. In figure 8.3 whatever line minimisation method is used a trial will eventually be made at an infeasible point. The constraint shown will be violated. The minimum point along the direction shown in figure

www.EngineeringBooksPDF.com

8.3 lies at the intersection of the line with the constraint boundary and this point, B, must be found accurately because it is a line minimum and might even be the solution point for the whole problem. Thus a considerable amount of back-tracking and interpolation is needed to find point B.

As shown in figure 8.3 point B is not the solution of the problem but is just a constrained line minimum. Consequently a new search direction from B must be found for the next line search. If the steepest gradient direction of the objective function is calculated as was done at A it is found that this direction leads straight into the infeasible region and it cannot, therefore, be used. What other directions are there if the steepest gradient direction is unusable? One possibility is to try to follow the constraint boundary tangent at B in a direction in which the objective function decreases. Figure 8.3, however, shows that because of the curved non-linear nature of the constraint, the tangent direction at B also leads straight into the infeasible region and is also, therefore, unusable. Needless to say a considerable number of trials will have been used in establishing these directions and proving them to be unusable.

Several methods have been described which allow further progress to be made from a point such as B in figure 8.3. It is clear that feasible search directions exist along which the objective function can be reduced in value. The *feasible directions* method of Zoutendijk enables these directions to be found and searched along. There is, however, no way of finding a best feasible direction at B and so nothing can be guaranteed about the convergence or efficiency of the method on general non-linear problems. Experience with direct numerical search methods for non-linear constrained problems has not been very encouraging. Efficiencies are generally low, particularly on problems in which the constraints play a major part. No further study will be made of these methods here. Although research still continues into new methods for constrained non-linear optimisation, very little of this is concerned with direct search methods.

8.6 GEOMETRIC PROGRAMMING

Geometric Programming (GP) is a very different approach to solving non-linear optimisation problems from any of the methods so far examined. It has many peculiarities, not least of which is that it is equally applicable to unconstrained and constrained problems. It is also of particular use in solving engineering design problems. Indeed the origins of the method are in engineering optimisation. In the early 1960s the method was developed from some striking insights and observations of Clarence Zener on the nature of the solutions of engineering optimisation problems. Many researchers have filled in the mathematical background of GP bringing it to its present state of development. Of all the methods used in solving engineering design problems GP is certainly the most difficult to understand in a mathematical sense but the effort required to understand it is amply repaid by a very useful optimisation method.

www.EngineeringBooksPDF.com

8.6.1 Unconstrained Geometric Programming

Geometric programming is examined here first in relation to unconstrained problems because the necessary theory is simpler; later, constrained problems are examined. Geometric programming is concerned with *posynomial functions.* A posynomial function is simply a polynomial function in which the coefficients of all terms are positive (posynomial ≡ *pos*itive pol*ynomial*). The functions

$$f = 4x_1 + \frac{2.8x_1^3}{x_2} + \frac{x_2x_3}{x_1^{1.5}} + \frac{7x_2^2}{x_1^{12}x_3^3}$$

and

$$f = x_1^2 + 2x_1x_2 + x_2^2$$

are both posynomials but the functions

$$f = 4x_1 + \frac{2.8x_1^3}{x_2} - \frac{x_2x_3}{x_1^{1.5}} - \frac{7x_2^2}{x_3^3}$$

and

$$f = x_1^2 - 2x_1x_2 + x_2^2$$

are not posynomials because each of them has at least one term with a negative coefficient. Note that negative powers of the variables are acceptable, negative coefficients of terms are not. The algebraic definition of a posynomial function, f, is

$$f = \sum_{t=1}^{T} C_t \prod_{i=1}^{N} x_i^{a_{ti}}$$

with

$$\begin{array}{lll} C_t > 0 & & t = 1, \ldots, T \\ x_i > 0 & & i = 1, \ldots, N \\ a_{ti} & \text{unrestricted} & t = 1, \ldots, T;\ i = 1, \ldots, N \end{array} \qquad (8.43)$$

Thus the posynomial function f is the sum of T terms, each term having a strictly positive coefficient C_t. In each term the N variables x_i, $i = 1, \ldots, N$ are each raised to some power a_{ti} and all variables are producted together ($\Pi \equiv$ product). The two posynomial functions quoted above, when expressed in the form 8.43 are

$$f = 4x_1^1x_2^0x_3^0 + 2.8x_1^3x_2^{-1}x_3^0 + 1x_1^{-1.5}x_2^1x_3^1 + 7x_1^{-12}x_2^2x_3^{-3}$$

and

$$f = 1x_1^2x_2^0 + 2x_1^1x_2^1 + 1x_1^0x_2^2$$

The first of these fs has four terms ($T = 4$) and three variables ($N = 3$), the second

www.EngineeringBooksPDF.com

has $T = 3$ and $N = 2$. Note that, because the powers a_{ti} may be fractional, the variables $x_i, i = 1, \ldots, N$ must be strictly positive as indicated by $x_i > 0$ in 8.43. The quantity x^a has no meaning if x is negative and a is non-integer. The conditions upon C_t, x_i and a_{ti} in 8.43 are always implied though not always stated in what follows.

In the early 1960s Zener was concerned with engineering optimum design problems which required the minimisation of a total cost function which was made up of the sum of the costs of several individual components of the design; a very common type of function in engineering. He noted that each component cost could be expressed as a unit cost coefficient, C_t, multiplied by all the design variables in that component raised to some known powers, a_{ti}. Thus the total cost was represented as a posynomial function. Let us assume that the first of the two posynomial functions is typical of the cost functions Zener had to minimise, that is

$$\text{Minimise } f = 4x_1 + 2.8x_1^3x_2^{-1} + 1x_1^{-1.5}x_2x_3 + 7x_1^{-12}x_2^2x_3^{-3} \quad (8.44)$$

The total cost f is made up of the sum of four individual component costs. The function f is continuous and algebraic and it is tempting to try to solve problem 8.44 by the classical differential method. A stationary point of f is found by equating the first derivatives of f to zero and solving the resulting simultaneous equations. These equations are

$$\left.\begin{aligned} \frac{\partial f}{\partial x_1} &\equiv 4 + 8.4x_1^2x_2^{-1} - 1.5x_1^{-2.5}x_2x_3 - 84x_1^{-13}x_2^2x_3^{-3} = 0 \\ \frac{\partial f}{\partial x_2} &\equiv -2.8x_1^3x_2^{-2} + x_1^{-1.5}x_3 + 14x_1^{-12}x_2x_3^{-3} = 0 \\ \frac{\partial f}{\partial x_3} &\equiv x_1^{-1.5}x_2 - 21x_1^{-12}x_2^2x_3^{-4} = 0 \end{aligned}\right\} \quad (8.45)$$

They are highly non-linear and appear to be extremely difficult to solve directly. Each equation, however, represents a stationary point of f with respect to a particular variable $x_i, i = 1, 2, 3$. If each equation is multiplied through by its own x_i the system of equations 8.45 becomes

$$\left.\begin{aligned} x_1\frac{\partial f}{\partial x_1} &= 0 \\ x_2\frac{\partial f}{\partial x_2} &= 0 \\ x_3\frac{\partial f}{\partial x_3} &= 0 \end{aligned}\right\}$$

which for this numerical problem yields

$$\left.\begin{aligned} (4x_1) + 3(2.8x_1^3x_2^{-1}) - 1.5(x_1^{-1.5}x_2x_3) - 12(7x_1^{-12}x_2^2x_3^{-3}) &= 0 \\ -(2.8x_1^3x_2^{-1}) + (x_1^{-1.5}x_2x_3) + 2(7x_1^{-12}x_2^2x_3^{-3}) &= 0 \\ (x_1^{-1.5}x_2x_3) - 3(7x_1^{-12}x_2^2x_3^{-3}) &= 0 \end{aligned}\right\} \quad (8.46)$$

www.EngineeringBooksPDF.com

The system of equations 8.46 has a remarkable structure in that each of the terms within parentheses is also a term of the posynomial function f. Thus if the four terms in f are written as P_1, P_2, P_3, P_4 then problem 8.44 becomes

$$\text{Minimise } f = P_1 + P_2 + P_3 + P_4 \tag{8.47}$$

and the necessary conditions 8.46 for a minimum of f become

$$\left.\begin{aligned} P_1^* + 3P_2^* - 1.5P_3^* - 12P_4^* &= 0 \\ -P_2^* + P_3^* + 2P_4^* &= 0 \\ P_3^* - 3P_4^* &= 0 \end{aligned}\right\} \tag{8.48}$$

where

$$\begin{aligned} P_1 &= 4x_1 & P_2 &= 2.8x_1^{3}x_2^{-1} \\ P_3 &= x_1^{-1.5}x_2x_3 & P_4 &= 7x_1^{-12}x_2^{2}x_3^{-3} \end{aligned} \tag{8.49}$$

The terms $P_1, \ldots, P_4$ represent the individual component costs of the four-component design and the necessary conditions 8.48 represent relationships between and among those component costs which must hold in an optimum design. For example the last of the equations 8.48 states that in the optimum design

$$P_3^* - 3P_4^* = 0$$

that is

$$P_3^* = 3P_4^*$$

Thus the optimum design will be one in which the third component, P_3 costs three times as much as the fourth component. Conversely, any design in which the third component costs more than or less than three times the fourth component cannot be an optimum (least total cost) design. The original problem 8.44 and the necessary conditions for its solution 8.45 have now been cast into a different form of trying to find values, not for the design variables x_i, $i = 1, 2, 3$, but for the component costs P_1, P_2, P_3, P_4 such that their sum is a minimum. This is problem 8.47. The conditions 8.48 represent relationships among these component costs at optimum.

A further simplification can be made if the function f in 8.47 is further transformed. In an optimum design the sum of the element costs $P_1^* + P_2^* + P_3^* + P_4^*$ must equal the minimum total cost f^*. Thus at the solution of problem 8.47

$$P_1^* + P_2^* + P_3^* + P_4^* = f^*$$

Hence

$$\frac{P_1^*}{f^*} + \frac{P_2^*}{f^*} + \frac{P_3^*}{f^*} + \frac{P_4^*}{f^*} = 1 \tag{8.50}$$

Of course f^* is not yet known but it may be introduced as an unknown constant. Each of the terms $P_1^*/f^*, \ldots, P_4^*/f^*$ in 8.50 represents the *proportion of the total cost* represented by the four components at optimum and obviously the

www.EngineeringBooksPDF.com

sum of these cost proportions must sum to one. Introducing new variables w_1, w_2, w_3, w_4 as the four optimal proportions, such that

$$w_1 = \frac{P_1^*}{f^*}; \qquad w_2 = \frac{P_2^*}{f^*}$$
$$w_3 = \frac{P_3^*}{f^*}; \qquad w_4 = \frac{P_4^*}{f^*} \tag{8.51}$$

transforms 8.50 into 8.52

$$w_1 + w_2 + w_3 + w_4 = 1 \tag{8.52}$$

From 8.51 the ith cost proportion, w_i is given by

$$w_i = \frac{P_i^*}{f^*}$$

therefore

$$P_i^* = w_i f^* \tag{8.53}$$

If 8.53 is substituted into 8.48 for $P_1^*, P_2^*, P_3^*, P_4^*$, and f^* is deleted through, equations 8.48 become

$$\left.\begin{aligned} w_1 + 3w_2 - 1.5w_3 - 12w_4 &= 0 \\ -w_2 + w_3 + 2w_4 &= 0 \\ w_3 - 3w_4 &= 0 \end{aligned}\right\} \tag{8.54}$$

Equations 8.52 and 8.54 represent relationships among the cost proportions in the optimum design. They are four equalities in the four unknown optimal cost proportions which can, therefore, be found. Solution of 8.52 and 8.54 yields

$$w_1 = 1/7; w_2 = 10/21; w_3 = 2/7; w_4 = 2/21$$

Thus the optimal cost proportions have been found. This is a most peculiar result. The distribution of total cost among the four components of the optimum design is known but the actual costs are not yet known, nor is the actual optimum design yet known in terms of values of the design variables x_1, x_2, x_3.

f^* is the next quantity to be found and the following algebra shows how this is done using a rather odd subterfuge. f^* can be written as $(f^*)^1$ and equation 8.52 allows the power of unity to be replaced by $w_1 + w_2 + w_3 + w_4$. Thus

$$\begin{aligned} f^* = (f^*)^1 &= (f^*)^{w_1 + w_2 + w_3 + w_4} \\ &= (f^*)^{w_1} (f^*)^{w_2} (f^*)^{w_3} (f^*)^{w_4} \end{aligned}$$

From 8.53 f^* may be replaced by P_i^*/w_i, $i = 1, \ldots, N$
therefore

$$f^* = \left(\frac{P_1^*}{w_1}\right)^{w_1} \left(\frac{P_2^*}{w_2}\right)^{w_2} \left(\frac{P_3^*}{w_3}\right)^{w_3} \left(\frac{P_4^*}{w_4}\right)^{w_4} \tag{8.55}$$

www.EngineeringBooksPDF.com

Substituting the known values of $w_1, \ldots, w_4$ and substituting for $P_1^*, \ldots, P_4^*$ from 8.49 gives

$$f^* = \left(\frac{4x_1}{1/7}\right)^{1/7}\left(\frac{2.8x_1^3x_2^{-1}}{10/21}\right)^{10/21}\left(\frac{x_1^{-1.5}x_2x_3}{2/7}\right)^{2/7}\left(\frac{7x_1^{-12}x_2^2x_3^{-3}}{2/21}\right)^{2/21}$$

$$= (28)^{1/7}(5.88)^{10/21}(3.5)^{2/7}(73.5)^{2/21}$$
$$x_1^{[(1/7)+(10/7)-(3/7)-(8/7)]}x_2^{[(-10/21)+(2/7)+(4/21)]}x_3^{[(2/7)-(6/21)]}$$

$$= (28)^{1/7}(5.88)^{10/21}(3.5)^{2/7}(73.5)^{2/21}x_1^0x_2^0x_3^0$$

therefore

$$f^* = (28)^{1/7}(5.88)^{10/21}(3.5)^{2/7}(73.5)^{2/21}$$

$$= 8.0593$$

Note that in this calculation the variables x_1, x_2 and x_3 have disappeared because the powers of each variable sum to zero. Thus f^*, the least total cost of the design is now known in addition to the distribution of that total cost among the four individual components.

The actual optimal component costs $P_1^*, \ldots, P_4^*$ can now be found from 8.53 since for component i, $P_i^* = w_i f^*$. Knowing these component costs the optimal values of the design variables can be found as follows. Consider the first component.

$$P_1^* = w_1 f^* = \frac{8.0593}{7} = 1.1513$$

but from 8.49

$$P_1^* = 4x_1^*$$

thus

$$4x_1^* = 1.1513$$

therefore

$$x_1^* = 0.2878$$

Knowing the value of x_1^*, the value of x_2^* can be found from P_2^*.

$$P_2^* = w_2 f^* = \frac{10 \times 8.0593}{21} = 3.8378 = 2.8x_1^{*3}x_2^{*-1}$$

therefore

$$x_2^* = \frac{2.8 \times 0.2878^3}{3.8378}$$

$$= 0.0174$$

www.EngineeringBooksPDF.com

Knowing x_1^* and x_2^*, x_3^* can be found from either P_3^* or P_4^*. From P_3^*

$$P_3^* = w_3 f^* = \frac{2 \times 8.0593}{7} = 2.3027 = x_1^{*-1.5} x_2^* x_3^*$$

therefore

$$x_3^* = \frac{2.3027 \times 0.2878^{1.5}}{0.0174}$$

$$= 20.4440$$

The solution of the original problem 8.44 is therefore

$$f^* = 8.0593; x_1^* = 0.2879; x_2^* = 0.0174; x_3^* = 20.4440$$

and has been found in a very devious and indirect manner. Nevertheless, although it is indirect, it is a method which can in principle be used to minimise all posynomial functions. This example has been slightly contrived so as to demonstrate the principles of the solution method. Some further points need to be examined to generalise the solution method to all posynomials and the method needs to be expressed in a more concise mathematical form. This will now be done.

8.6.1.1 Positive Degree of Difficulty Problems

The posynomial function f in problem 8.44 was composed of four terms and had three variables. The key to solving the problem was not to look for optimal values of these variables but instead to look for an optimal distribution of cost, $w_1, \ldots, w_4$, among the terms of f. It is too specific to refer to these quantities w as cost proportions since the function f need not necessarily represent a cost. A more general name for them is *weights*. The weights w_1 to w_4 represent the proportion of f contributed by each of the terms of f and clearly they must always sum to one. Equations 8.52 and 8.54 are crucial to the solution method because they allow these weights to be found. Equation 8.52 represents the fact that the weights must sum to unity. Equations 8.54 come from algebraic manipulation of the necessary conditions for a minimum of f, 8.45. There are three equations in 8.54 because there are three variables x_1, x_2, x_3 in problem 8.44 and f must be stationary with respect to each of them. Thus 8.52 and 8.54 form four linear equations in the four unknown weights which can, therefore, be found uniquely. The squareness of the system of equations 8.52 and 8.54 is unfortunately not general and will only be obtained when the number of terms, T in f is equal to 1 plus the number of variables, N. In problem 8.44 $T = 4$ and $N = 3$, thus $T = N + 1$ and the weights $w_1, \ldots, w_4$ could, therefore, be found uniquely by solving 8.52 and 8.54.

What happens when $T \neq N + 1$? The case in which $T < N + 1$ turns out not to be of practical interest because posynomial functions with $T < N + 1$ do not generally have a finite unconstrained minimum. The problem of finding such a

www.EngineeringBooksPDF.com

minimum, therefore, does not exist. The case in which $T > N + 1$ is of interest, however, because a minimum can exist and this case is the one most usually found.

The quantity $(T - N - 1)$ is important in geometric programming and is referred to as the *degree of difficulty* of a problem. The previous example in which $T = N + 1$ was a *zero degree of difficulty* problem and was particularly easy to solve. The case to be examined now in which $T > N + 1$ is classed as a *positive degree of difficulty* problem because $(T - N - 1)$ is a positive integer. The case may be studied either by adding extra terms T to f in 8.44 or by deleting variables. The latter will be chosen for demonstration purposes and variable x_3 will be deleted from problem 8.44, (x_3 will be held constant at the value $x_3 = 1$). Problem 8 44 then becomes a single degree of difficulty problem

$$\text{Minimise } f = 4x_1 + 2.8x_1^{3}x_2^{-1} + 1x_1^{-1.5}x_2 + 7x_1^{-12}x_2^{2} \tag{8.56}$$

If exactly the same solution method is used as for problem 8.44 the system of equations similar to 8.52 and 8.54 for problem 8.56 are

$$\left.\begin{aligned} w_1 + w_2 + w_3 + w_4 &= 1 \\ w_1 + 3w_2 - 1.5w_3 - 12w_4 &= 0 \\ - w_2 + w_3 + 2w_4 &= 0 \end{aligned}\right\} \tag{8.57}$$

Equations 8.57 are now no longer a square set of equations because $T = 4, N = 2$ and $T > N + 1$. The missing equation in 8.57 is the one corresponding to the stationarity of f with respect to variable x_3 which is no longer a variable. The weights $w_1, \ldots, w_4$ can no longer be found uniquely.

Equations 8.57, however, may be solved as far as possible to yield

$$\left.\begin{aligned} w_1 &= -3 + 33w_4 \\ w_2 &= 2 - 16w_4 \\ w_3 &= 2 - 18w_4 \\ w_4 &= w_4 \end{aligned}\right\} \tag{8.58}$$

All the weights are now known as functions of only one unknown weight w_4. Furthermore, bounds can be found upon the optimal value of w_4 because the definition of the weights implies that no weight may be negative. Examining the first equation in 8.58 it is seen that w_4 must be greater than 1/11, otherwise w_1 would be negative. From the third equation of 8.58 w_4 must be less than 1/9, otherwise w_3 would be negative. Hence

$$0.0909 \leqslant w_4 \leqslant 0.1111$$

These represent fairly tight bounds upon the optimal value of w_4 but they do not allow the optimal value of w_4 to be found. If the solution of problem 8.56 is continued using the weight functions 8.58 instead of the unique optimal weight values as before, equation 8.55 yields f in the form

$$f = \left(\frac{P_1^*}{-3 + 33w_4}\right)^{-3+33w_4} \left(\frac{P_2^*}{2 - 16w_4}\right)^{2-16w_4} \left(\frac{P_3^*}{2 - 18w_4}\right)^{2-18w_4} \left(\frac{P_4^*}{w_4}\right)^{w_4} \tag{8.59}$$

www.EngineeringBooksPDF.com

This time f cannot be found uniquely as it is a function of w_4. f^* is the least value of f and so in order to find f^* and the optimal value of w_4, f must in some way be optimised over the variable w_4. It turns out that f must be maximised over variable w_4 and the reason for this maximisation will now be examined.

Problem 8.56 has a function f composed of four terms, $P_1, \ldots, P_4$, that is

$$f = P_1 + P_2 + P_3 + P_4$$

This may be written in the form

$$f = w_1\left(\frac{P_1}{w_1}\right) + w_2\left(\frac{P_2}{w_2}\right) + w_3\left(\frac{P_3}{w_3}\right) + w_4\left(\frac{P_4}{w_4}\right) \tag{8.60}$$

There is a mathematical relationship called the *arithmetic-geometric mean inequality*, also called *Cauchy's inequality*, that is concerned with the relationship between the arithmetic mean and the geometric mean of numbers. Geometric programming takes its name from this arithmetic-geometric mean inequality. Cauchy's inequality states that for any positive numbers $V_1, V_2, \ldots, V_T$ and positive weights $w_1, w_2, \ldots, w_T$ such that $w_1 + w_2 + \ldots + w_T = 1$ the following relationship can be proved.

$$w_1 V_1 + w_2 V_2 + w_3 V_3 + \ldots + w_T V_T \geqslant V_1^{w_1} V_2^{w_2} V_3^{w_3} \ldots V_T^{w_T} \tag{8.61}$$

with inequality only when $V_1 = V_2 = V_3 = \ldots = V_T$. This will not be proved here but is rigorously provable.

If the function f in the form 8.60 is identified as having the same form as the left-hand side of 8.61 with, generally, $V_i \equiv P_i/w_i$, $i = 1, \ldots, 4$, the following may be written

$$w_1\left(\frac{P_1}{w_1}\right) + w_2\left(\frac{P_2}{w_2}\right) + w_3\left(\frac{P_3}{w_3}\right) + w_4\left(\frac{P_4}{w_4}\right) \geqslant \left(\frac{P_1}{w_1}\right)^{w_1} \left(\frac{P_2}{w_2}\right)^{w_2} \left(\frac{P_3}{w_3}\right)^{w_3} \left(\frac{P_4}{w_4}\right)^{w_4}$$

therefore

$$P_1 + P_2 + P_3 + P_4 \geqslant \left(\frac{P_1}{w_1}\right)^{w_1} \left(\frac{P_2}{w_2}\right)^{w_2} \left(\frac{P_3}{w_3}\right)^{w_3} \left(\frac{P_4}{w_4}\right)^{w_4} \tag{8.62}$$

At the optimum, equation 8.53 defines each of the ratios, $P_1^*/w_1, \ldots P_4^*/w_4$ to be equal to f^*, thus all these ratios are equal and 8.62 may, therefore, be written as an equality at the optimum.

$$P_1^* + P_2^* + P_3^* + P_4^* = \left(\frac{P_1^*}{w_1}\right)^{w_1} \left(\frac{P_2^*}{w_2}\right)^{w_2} \left(\frac{P_3^*}{w_3}\right)^{w_3} \left(\frac{P_4^*}{w_4}\right)^{w_4} \tag{8.63}$$

The left-hand sides of 8.62 and 8.63 are, respectively, $f(\boldsymbol{x})$ and $f(\boldsymbol{x}^*)$ and it is known that generally $f(\boldsymbol{x}) \geqslant f(\boldsymbol{x}^*)$ because $f(\boldsymbol{x}^*)$ is defined as the minimum of

www.EngineeringBooksPDF.com

$f(x)$. If the right-hand sides of 8.62 and 8.63 are written as $f(w)$ and $f(w^*)$ respectively, the two relationships 8.62 and 8.63 may be combined together in 8.64.

$$f(x) \geqslant f(x^*) = f(w^*) \geqslant f(w) \tag{8.64}$$

From 8.64 it is clear that the value f^* may be found either by *minimising* f over variables x or by *maximising* f over variables w. It may be thought that it is incorrect to write $f(w), f(w^*)$ for the right-hand sides of 8.62 and 8.63 because each of the functions $P_1, \ldots, P_4$ is a function of x. Is it not more correct to write $f(w, x)$ and $f(w^*, x^*)$? In fact it is not. Returning to the first example, equation 8.55 represents the right-hand side of 8.63. When this function f^* is evaluated it is found that the powers of all variables in x turn out to be zero, thus x variables are absent from 8.55 and from the right-hand sides of 8.62 and 8.63. It is therefore correct to write these as $f(w)$ and $f(w^*)$.

Returning to the numerical example in which $T > N + 1$ this examination of Cauchy's inequality has now shown that the value of f^* in equation 8.59 will be found by maximising f over variable w_4, the single unknown weight. In substituting P_1^* to P_4^* into 8.59 only the coefficients of those terms need be substituted; the variables x_1, x_2, x_3 may be omitted because at the optimum they will all have powers of zero. Thus f^* is given by

$$f^* = \max_{w_4} \left\{ \left(\frac{4}{-3 + 33w_4}\right)^{-3+33w_4} \left(\frac{2.8}{2 - 16w_4}\right)^{2-16w_4} \left(\frac{1}{2 - 18w_4}\right)^{2-18w_4} \left(\frac{7}{w_4}\right)^{w_4} \right\} \tag{8.65}$$

This maximisation over variable w_4 of the function with { }in 8.65 may be done by any line search method and the known bounds on the optimal value of w_4, that is, $0.0909 \leqslant w_4 \leqslant 0.1111$ are useful. The optimal value of w_4 is found to be $w_4 = 0.10513$ and the corresponding optimal values of w_1, w_2, w_3 from 8.58 are $w_1 = 0.46929$, $w_2 = 0.31792$, $w_3 = 0.10766$. The optimal value f^* is $f^* = 10.7898$.

Knowing the optimal weights $w_1, \ldots, w_4$ and f^*, optimal values of variables x_1 and x_2 may be found from the relationships 8.53, that is, $P_i^* = w_i f^*$, $i = 1, \ldots, 4$. For problem 8.56 these are

$$\left.\begin{aligned} 4x_1 &= 5.06355 \\ 2.8x_1^{3}x_2^{-1} &= 3.43029 \\ 1x_1^{-1.5}x_2 &= 1.16163 \\ 7x_1^{-12}x_2^{2} &= 1.13433 \end{aligned}\right\} \tag{8.66}$$

For this problem the solution for x_1 and x_2 is simple. Sometimes it is not so easy

www.EngineeringBooksPDF.com

since these equations are non-linear. If logarithms are taken of both sides, however, a much simpler form is found

$$\left.\begin{aligned} \ln(x_1) &= 0.23577 \\ 3\ln(x_1) - \ln(x_2) &= 0.20303 \\ -1.5\ln(x_1) + \ln(x_2) &= 0.14982 \\ -12\ln(x_1) + 2\ln(x_2) &= -1.81987 \end{aligned}\right\} \tag{8.67}$$

Equations 8.67 are linear in the logarithms of x_1 and x_2 and can be very easily solved to yield $x_1^* = 1.26588, x_2^* = 1.65579$. Thus problem 8.56 has been solved.

8.6.1.2 A General Mathematical Formulation

The previous sections have examined geometric programming by means of two examples which were used to explain how problems are solved and to explore the mathematical background to the method. The logic used is somewhat tortuous, nevertheless it is rigorous. Now it is necessary to draw the several strands together into a concise general mathematical framework which can be used to minimise any posynomial function.

Geometric programming solves the following unconstrained problem

$$\left.\begin{aligned} &\underset{x_i,\, i=1,\ldots,N}{\text{Minimise}} \; f = \sum_{t=1}^{T} C_t \prod_{i=1}^{N} x_i^{a_{ti}} \\ &\text{in which} \\ &\qquad C_t > 0 \qquad t = 1, \ldots, T \\ &\qquad x_i > 0 \qquad i = 1, \ldots, N \\ &a_{ti} \text{ is unrestricted} \qquad t = 1, \ldots, T;\ i = 1, \ldots, N \end{aligned}\right\} \tag{8.68}$$

Problem 8.68 is not solved directly in the space of the variables x_i, $i = 1, \ldots, N$, by minimisation. Instead, a form of the arithmetic-geometric mean inequality, 8.61, is used to transform problem 8.68 into a completely different problem in which a function of variables w_t, $t = 1, \ldots, T$ is maximised subject to linear equality constraints. Problem 8.68 is called the geometric programming *primal problem*. The function f is the *primal objective function* and the variables x_i, $i = 1, \ldots, N$ are the *primal variables*. The problem to which the primal problem 8.68 is transformed is called the geometric programming *dual problem*. The dual problem has *dual variables* w_t, $t = 1, \ldots, T$, a *dual objective function*, $D(w_1, \ldots, w_T)$ and a set of *dual constraints* which are linear equalities. The mathematical

www.EngineeringBooksPDF.com

form of the geometric programming dual problem corresponding to the primal problem 8.68 is

$$\left.\begin{aligned} &\underset{w_t,\, t=1,\ldots,T}{\text{Maximise}}\; D = \prod_{t=1}^{T} \left(\frac{C_t}{w_t}\right)^{w_t} \\ &\text{subject to the constraints} \\ &\sum_{t=1}^{T} w_t = 1 \\ &\sum_{t=1}^{T} a_{ti} w_t = 0 \qquad i = 1, \ldots, N \\ &\text{with} \\ &w_t \geqslant 0 \qquad t = 1, \ldots, T \end{aligned}\right\} \tag{8.69}$$

The relationships between the primal and dual objective functions are that

$$f(x_1, \ldots, x_N) \geqslant f(x_1^*, \ldots, x_N^*) = D(w_1^*, \ldots, w_T^*) \geqslant D(w_1, \ldots, w_T) \tag{8.70}$$

and the relationships between the primal and dual variables at the optimum are that

$$\prod_{i=1}^{N} (x_i^*)^{a_{ti}} = \frac{w_t^* D^*}{C_t} \qquad t = 1, \ldots, T \tag{8.71}$$

Relationships 8.71 may be expressed in the simpler log-linear form

$$\sum_{i=1}^{N} a_{ti} \ln(x_i^*) = \ln\left(\frac{w_t^* D^*}{C_t}\right) \quad t = 1, \ldots, T \tag{8.72}$$

Geometric programming problems of the primal form 8.68 are not solved directly. Instead the primal problem is used to set up a dual problem 8.69 which is solved to yield D^* and $w_1^*, \ldots, w_T^*$. Having found these, relationships 8.70 show that $f^* = D^*$ and the optimal primal variables $x_1^*, \ldots, x_N^*$ are found from 8.71 or 8.72. It is logical to ask what are the advantages in converting a non-linear unconstrained problem such as 8.68 to a dual problem 8.69 which has a non-linear objective function and linear equality constraints? Superficially this transformation appears to make the problem harder to solve. The advantages lie in the relative numbers of primal variables, N, and primal terms, T. When $T = N + 1$, the zero degree of difficulty case, the dual equality constraints have a unique solution for the optimal dual variables. The dual problem then consists only of solving a set of T linear equations in T dual variables. This is far simpler than minimising a primal non-linear function over N variables. In the case of non-zero

www.EngineeringBooksPDF.com

degree of difficulty problems where $T > N + 1$ the dual problem can also often be the easier to solve. This time the dual constraints do not have a unique solution but they can be used to eliminate $N + 1$ of the dual variables from the problem expressing them as linear functions of the remaining $T - N - 1$ dual variables. The dual problem then becomes an unconstrained maximisation of the dual objective function D over the remaining $T - N - 1$ dual variables. Provided that the degree of difficulty, $T - N - 1$ is less than the number of primal variables, N, the dual problem should still be the easier to solve numerically. Thus the advantage of the GP dual problem is that it can often considerably reduce the dimensionality of a problem resulting in an optimisation over fewer variables than are present in the primal problem.

8.6.1.3 Example

A skip must be designed to carry concrete from a batching plant to the site of a large pour. The skip must have the form of a box, rectangular in plan, of length ℓ, breadth b, and with sides of depth d. The top of the skip is open. The sides of the skip cost £10/m^2 and the strengthened base costs £30/m^2. The skip is to be used to transport a total of 1500 m^3 of concrete in several journeys, each round trip costing £10. Design the skip so that the total costs of the large pour are minimised.

The volume of the skip is $\ell\, b\, d$ and, since 1500 m^3 of concrete have to be transported, the number of trips which must be made is clearly $1500/\ell bd$. The cost of these trips will therefore be £$(15\,000/\ell bd)$. The cost of the skip itself will be

$$£10(2bd + 2\ell d) + 30(\ell b)$$

Thus the total cost function to be minimised will be

$$C = 20bd + 20\ell d + 30\ell b + \frac{1500}{\ell bd}$$

Writing $b \equiv x_1, d \equiv x_2, \ell \equiv x_3$ gives a problem of the form 8.68, that is

$$\underset{x_1, x_2, x_3}{\text{Minimise}}\ f = 20x_1x_2 + 20x_2x_3 + 30x_1x_3 + 15\,000x_1^{-1}x_2^{-1}x_3^{-1}$$

From this problem the dual constraints of 8.69 can be written immediately as

$$\left.\begin{array}{rl} & w_1 + w_2 + w_3 + w_4 = 1 \\ \text{for } x_1, & w_1 \qquad + w_3 - w_4 = 0 \\ \text{for } x_2, & w_1 + w_2 \qquad - w_4 = 0 \\ \text{for } x_3, & \qquad w_2 + w_3 - w_4 = 0 \end{array}\right\}$$

These four equations in the four unknown weights can be easily solved to give

$$w_1 = w_2 = w_3 = \frac{1}{5}; w_4 = \frac{2}{5}$$

www.EngineeringBooksPDF.com

No maximisation of D in 8.69 was necessary to obtain these weights which must, therefore, be optimal weights. Thus, from the objective function of 8.69

$$D^* = \left(\frac{20}{1/5}\right)^{1/5}\left(\frac{20}{1/5}\right)^{1/5}\left(\frac{30}{1/5}\right)^{1/5}\left(\frac{15\,000}{2/5}\right)^{2/5} = £1161$$

This value of D^* is, by 8.70, equal to the minimum cost f^*. Optimal values of the primal variables x_1^*, x_2^* and x_3^* are found by solving equations 8.72, that is

$$\left.\begin{aligned}
\ln(x_1^*) + \ln(x_2^*) &= \ln\left(\frac{1161}{100}\right)\\
\ln(x_2^*) + \ln(x_3^*) &= \ln\left(\frac{1161}{100}\right)\\
\ln(x_1^*) + \ln(x_3^*) &= \ln\left(\frac{1161}{150}\right)\\
-\ln(x_1^*) - \ln(x_2^*) - \ln(x_3^*) &= \ln\left(\frac{1161}{37\,500}\right)
\end{aligned}\right\}$$

Any three of these log-linear equations yield $x_1^* = 2.782; x_2^* = 4.173; x_3 = 2.782$. Thus the problem has been solved. The skip must be 2.782 m square in plan with sides 4.173 m deep. The total cost of the skip and its operation will be £1161 of which three-fifths is spent on the skip itself and two-fifths on the trips made by the skip. Actually these values of x imply that 46.44 trips are made with the skip. Rounding this to the nearest integer number of trips, 46, and reformulating and resolving the problem yields a negligible increase in total cost which remains at £1161 to four significant figures accuracy. The skip's optimal dimensions, however, alter slightly to 2.791 m square with 4.186 m deep sides.

8.6.2 Constrained Geometric Programming

The derivation in section 8.6.1 of the primal/dual problems of unconstrained geometric programming can be extended to include posynomial constraints. Unfortunately the derivation for constrained problems is too long for this book. It can be done by further exploiting Cauchy's inequality or by a Lagrange multiplier approach. The interested reader should consult the specialist texts of Duffin, Peterson and Zener, or Beightler and Phillips for details. In this book the primal/dual relationships will merely be stated, discussed and applied to some examples.

8.6.2.1 A Constrained GP Formulation

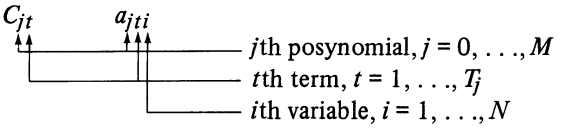
Geometric programming is concerned with posynomial functions. Unconstrained GP consists of minimising a posynomial function. Constrained GP consists of

www.EngineeringBooksPDF.com

minimising a posynomial function subject to several constraints, all of which consist of posynomial functions bounded above by unity, that is

$$\left.\begin{aligned} &\underset{x_i,\, i=1,\ldots,N}{\text{Minimise}} \quad f = \sum_{t=1}^{T_o} C_{ot} \prod_{i=1}^{N} x_i^{a_{oti}} \\ &\text{subject to} \sum_{t=1}^{T_j} C_{jt} \prod_{i=1}^{N} x_i^{a_{jti}} \leqslant 1 \qquad j = 1, \ldots, M \\ &\text{in which} \\ &\qquad C_{jt} > 0 \qquad\quad j = 0, \ldots, M;\ t = 1, \ldots, T_j \\ &\qquad x_i > 0 \qquad\quad i = 1, \ldots, N \\ &a_{jti} \text{ is unrestricted} \qquad j = 0, \ldots, M;\ t = 1, \ldots, T_j;\ i = 1, \ldots, N \end{aligned}\right\} \quad (8.73)$$

Comparing the constrained problem 8.73 with the unconstrained one 8.68, several differences are noted. The objective functions f have exactly the same posynomial form but there is an extra level of subscripts in the constrained problem. This is necessary because in the constrained problem there are posynomial functions in the constraints as well as in the objective. The extra subscript denotes which posynomial function is under consideration. It is usual to consider the constrained objective function as the zeroth posynomial function ($j = 0$). There will be T_o terms in the constrained objective function (T in the unconstrained f). The positive coefficients are C_{ot}, $t = 1, \ldots, T_o$ (C_t in the unconstrained f) and the powers are a_{oti} instead of a_{ti} in the unconstrained case. The constraints in 8.73 have the form of posynomial functions which must not be greater than unity. There are M constraint inequalities and they have an extra subscript j, $j = 1, \ldots, M$ to distinguish between constraint posynomials. Thus the kth inequality constraint will have a posynomial left-hand side with a total of T_k terms. The term coefficients will be C_{kt}, $t = 1, \ldots, T_k$, and the variable powers will be a_{kti}. The diagram below clarifies the subscript notation.



It is important to note that 8.73 is the only permissible form for constrained GP. The problem *must* be expressible as one of *minimising* a posynomial and the constraints *must* be posynomials *bounded above by unity*. If a general problem can be written in this precise form it can be solved by GP; if it does not fit this form then the duality relationships shortly to be stated do not apply and cannot be used. Thus problems of maximisation or involving lower bound constraints cannot be solved directly by GP unless they can be algebraically manipulated into the form 8.73. Note, for example, that the usual way of converting a maxi-

www.EngineeringBooksPDF.com

misation problem to one of minimisation – multiplying through by -1 – cannot be used as this would give negative coefficients which are expressly prohibited.

Problem 8.73 is the constrained GP primal problem. The GP dual problem corresponding to it is

$$\left.\begin{aligned}
&\underset{w_{jt}, j=0,\ldots,M;\ t=1,\ldots,T_j}{\text{Maximise}} \quad D = \prod_{j=0}^{M} \prod_{t=1}^{T_j} \left(\frac{C_{jt} w_{jo}}{w_{jt}}\right)^{w_{jt}} \\
&\text{subject to the constraints} \quad \sum_{t=1}^{T_o} w_{ot} = 1 \\
&\sum_{j=0}^{M} \sum_{t=1}^{T_j} a_{jti} w_{jt} = 0 \qquad i = 1, \ldots, N \\
&\text{with } w_{jo} \triangleq \sum_{t=1}^{T_j} w_{jt} \qquad j = 1, \ldots, M \\
&w_{oo} \triangleq 1 \\
&\text{and } w_{jt} \geqslant 0 \qquad j = 0, \ldots, M;\ t = 1, \ldots, T_j
\end{aligned}\right\} \tag{8.74}$$

Problem 8.74 is the constrained equivalent of the unconstrained GP dual problem 8.69. The forms are very similar although 8.74 is slightly more complicated. Examining 8.74 more closely, the dual variables are the w_{jt}s instead of w_ts. Thus there is a dual variable corresponding to *each term* of the primal problem including those terms in the constraints. The dual objective function, D, is again to be maximised and is simply a product over *all terms* (including the constraints) of the primal problem, of $(C_{jt} w_{jo}/w_{jt})^{w_{jt}}$. This is similar to the product of $(C_t/w_t)^{w_t}$ in problem 8.69 but with the extra level of subscripts. The element w_{jo} is extra but this is defined for $j = 0$ to be equal to 1 ($\triangleq$ means 'defined to be equal to') and for $j = 1, \ldots, M$ to be equal to the sum of the dual variables for the jth primal constraint. Thus w_{jo} is not an extra variable but is a known linear function of the dual variables. The constraints in the dual problem 8.74 are very similar to those in problem 8.69. The first equality requires the sum of the objective function weights to be unity. The other linear equalities, one for each primal variable, are derived from the stationarity conditions and are similar to those of problem 8.69 but with the summation over all posynomial terms in the objective function and constraints.

The relationships between primal and dual objective functions is similar to 8.70, that is

$$f(x_1, \ldots, x_N) \geqslant f(x_1^*, \ldots, x_N^*) = D(w_{jt}^*) \geqslant D(w_{jt}) \tag{8.75}$$

The relationships between primal and dual variables at the optimum are similar

www.EngineeringBooksPDF.com

to 8.71 but with modifications to include the posynomial constraint functions.

$$\prod_{i=1}^{N} (x_i^*)^{a_{jti}} = \left\{ \begin{array}{ll} \dfrac{w_{ot}^* D^*}{C_{ot}} & j = 0, t = 1, \ldots, T_o \\ \\ \dfrac{w_{jt}^*}{w_{jo}^* C_{jt}} & j = 1, \ldots, M; t = 1, \ldots, T_j \end{array} \right\} \quad (8.76)$$

Expressed in log-linear form 8.76 becomes

$$\sum_{i=1}^{N} a_{jti} \ln(x_i^*) = \left\{ \begin{array}{ll} \ln\left(\dfrac{w_{ot}^* D^*}{C_{ot}}\right) & j = 0; t = 1, \ldots, T_o \\ \\ \ln\left(\dfrac{w_{jt}^*}{w_{jo}^* C_{jt}}\right) & j = 1, \ldots, M; t = 1, \ldots, T_j \end{array} \right\} \quad (8.77)$$

Relationships 8.73 to 8.77 represent a complete statement of the primal/dual constrained GP problem. They correspond in all respects to relationships 8.68 to 8.72 for unconstrained problems. Some examples show how problems with constraints are solved.

8.6.2.2 Example

$$\begin{array}{ll} \text{Minimise} & f = 2x_1^{3}x_4 + 3x_2^{-3}x_4 \\ \text{subject to} & 6x_1^{-0.25}x_2x_3^{-0.75}x_4^{-1} \leqslant 1 \\ & 4x_1^{-0.5}x_3^{2}x_4 + 5x_1^{-0.5}x_3x_4 \leqslant 1 \end{array}$$

This problem has five posynomial terms and four primal variables and is therefore a zero degree of difficulty problem. It is of the primal form 8.73. The dual constraints of 8.74 can immediately be written as

$$\left.\begin{array}{lrrrrr} & w_{01} + w_{02} & & & & = 1 \\ \text{for } x_1, & 3w_{01} & - 0.25w_{11} & - 0.5w_{21} & - 0.5w_{22} & = 0 \\ \text{for } x_2, & - 3w_{02} + & w_{11} & & & = 0 \\ \text{for } x_3, & & - 0.75w_{11} + & 2w_{21} + & w_{22} & = 0 \\ \text{for } x_4, & w_{01} + w_{02} - & w_{11} + & w_{21} + & w_{22} & = 0 \end{array}\right\}$$

Solving these five linear equations in the five unknown dual variables gives optimal values as

$$w_{01}^* = \frac{1}{3}; w_{02}^* = \frac{2}{3}; w_{11}^* = 2; w_{21}^* = \frac{1}{2}; w_{22}^* = \frac{1}{2}$$

From the definitions of 8.74

$$\begin{aligned} w_{00}^* &= 1 \\ w_{10}^* &= w_{11}^* = 2 \\ w_{20}^* &= w_{21}^* + w_{22}^* = \tfrac{1}{2} + \tfrac{1}{2} = 1 \end{aligned}$$

www.EngineeringBooksPDF.com

No maximisation of D was needed to find these optimal weights. Thus the optimal dual objective function is

$$D^* = \left(\frac{2 \times 1}{1/3}\right)^{1/3} \left(\frac{3 \times 1}{2/3}\right)^{2/3} \left(\frac{6 \times 2}{2}\right)^{2} \left(\frac{4 \times 1}{1/2}\right)^{1/2} \left(\frac{5 \times 1}{1/2}\right)^{1/2}$$

$$= (6)^{1/3}(4.5)^{2/3}(6)^{2}(8)^{1/2}(10)^{1/2}$$

$$D^* = 1594.8$$

From 8.75 this value of D^* is equal to f^*. Optimal values for the primal variables are found by solving equations 8.77, that is

$$\left.\begin{aligned}
3\ln(x_1^*) \qquad\qquad + \ln(x_4^*) &= \ln\left(\frac{\frac{1}{3} \times 1594.8}{2}\right) \\
-3\ln(x_2^*) \qquad + \ln(x_4^*) &= \ln\left(\frac{\frac{2}{3} \times 1594.8}{3}\right) \\
-0.5\ln(x_1^*) + 2\ln(x_3^*) + \ln(x_4^*) &= \ln\left(\frac{\frac{1}{2}}{1 \times 4}\right) \\
-0.5\ln(x_1^*) + \ln(x_3^*) + \ln(x_4^*) &= \ln\left(\frac{\frac{1}{2}}{1 \times 5}\right) \\
-0.25\ln(x_1^*) + \ln(x_2^*) - 0.75\ln(x_3^*) - \ln(x_4^*) &= \ln\left(\frac{1}{2 \times 6}\right)
\end{aligned}\right\}$$

These log-linear equalities yield

$$x_1^* = 10.1422; x_2^* = 0.0896; x_3^* = 1.2500; x_4^* = 0.2548$$

and the problem is solved.

8.6.2.3 Example

$$\begin{aligned}
&\text{Minimise } f = x_1x_2^2x_3^{-1} + x_1^{-1}x_2^{-3}x_4 + x_1x_3 \\
&\text{subject to } \quad 2x_1^{-1}x_3x_4^{-2} + 3x_3x_4 \leqslant 1 \\
&\qquad\qquad\qquad\qquad 5x_1x_2 \leqslant 1
\end{aligned}$$

Six posynomial terms and only four primal variables make this a single degree of difficulty problem. The dual constraints 8.74 are:

$$\left.\begin{aligned}
w_{01} + w_{02} + w_{03} &= 1 \\
w_{01} - w_{02} + w_{03} - w_{11} + w_{21} &= 0 \\
2w_{01} - 3w_{02} + w_{21} &= 0 \\
-w_{01} + w_{03} + w_{11} + w_{12} &= 0 \\
w_{02} - 2w_{11} + w_{12} &= 0
\end{aligned}\right\}$$

www.EngineeringBooksPDF.com

Solving these equalities to express all dual variables in terms of one dual variable, say w_{01}, gives

$$\left.\begin{aligned} w_{01} &= w_{01} \\ w_{02} &= 8w_{01} - 4 \\ w_{03} &= -9w_{01} + 5 \end{aligned}\right\} \qquad w_{00} = 1$$

$$\left.\begin{aligned} w_{11} &= 6w_{01} - 3 \\ w_{12} &= 4w_{01} - 2 \end{aligned}\right\} \qquad w_{10} = 10w_{01} - 5$$

$$w_{21} = 22w_{01} - 12 \qquad w_{20} = 22w_{01} - 12$$

Inspection of these dual variables shows that for all ws to be non-negative, w_{01}^* must lie within the range

$$0.5455 \leqslant w_{01}^* \leqslant 0.5556$$

Substituting the dual variables into the dual objective function D gives

$$D = \left(\frac{1}{w_{01}}\right)^{w_{01}} \left(\frac{1}{8w_{01} - 4}\right)^{8w_{01}-4} \left(\frac{1}{-9w_{01} + 5}\right)^{-9w_{01}+5}$$

$$\times \left(\frac{2(10w_{01} - 5)}{6w_{01} - 3}\right)^{6w_{01}-3} \left(\frac{3(10w_{01} - 5)}{4w_{01} - 2}\right)^{4w_{01}-2} (5)^{22w_{01}-12}$$

To find D^* and w_{01}^* this function D must be maximised over the single variable w_{01}. This may be done by any of the line search methods of chapter 7. w_{01}^* has already been bracketed to lie somewhere between 0.5455 and 0.5556 and since D is known as an algebraic function of w_{01} derivative information can be used in the line maximisation. Consequently a line maximisation using gradient values will be used. In fact instead of maximising D directly it is simpler to maximise the natural logarithm of D. The optimal w_{01}^* will be the same. Taking logs

$$\ln(D) = w_{01} \ln\left(\frac{1}{w_{01}}\right) + (8w_{01} - 4) \ln\left(\frac{1}{8w_{01} - 4}\right) + (-9w_{01} + 5)$$

$$\ln\left(\frac{1}{-9w_{01} + 5}\right) + (6w_{01} - 3) \ln\left(\frac{2(10w_{01} - 5)}{6w_{01} - 3}\right)$$

$$+ (4w_{01} - 2) \ln\left(\frac{3(10w_{01} - 5)}{4w_{01} - 2}\right) + (22w_{01} - 12) \ln(5)$$

Differentiating with respect to w_{01} gives, after some simplification

$$\frac{d\ln(D)}{dw_{01}} = \ln\left\{\left(\frac{1}{w_{01}}\right)\left(\frac{1}{8w_{01} - 4}\right)^{8}\left(\frac{1}{-9w_{01} + 5}\right)^{-9}\left(\frac{2(10w_{01} - 5)}{6w_{01} - 3}\right)^{6}\right.$$

$$\left.\times \left(\frac{3(10w_{01} - 5)}{4w_{01} - 2}\right)^{4} (5)^{22}\right\} = \ln\left\{\frac{(-9w_{01} + 5)^9 (10w_{01} - 5)^{10}\, 2^6 \times 3^4 \times 5^{22}}{w_{01}(8w_{01} - 4)^8 (6w_{01} - 3)^6 (4w_{01} - 2)^4}\right\}$$

www.EngineeringBooksPDF.com

Inspection of this derivative reveals that its value is large and positive at the lower limit, $w_{01} = 12/22$. As w_{01} increases all elements within the logarithm increase except for the term $(-9w_{01} + 5)^9$ which decreases. The derivative can only become zero when this term becomes very small, that is, close to the upper limit of $w_{01} = 5/9$. Examining values of the derivative near this upper limit quickly yields $w_{01}^* \approx 0.55546$. The optimal dual variables are then

$$\left.\begin{aligned} w_{01}^* &= 0.55546 \\ w_{02}^* &= 0.44368 \\ w_{03}^* &= 0.00086 \end{aligned}\right\} \quad w_{00} = 1$$

$$\left.\begin{aligned} w_{11}^* &= 0.33276 \\ w_{12}^* &= 0.22184 \end{aligned}\right\} \quad w_{10}^* = 0.55460$$

$$w_{21}^* = 0.22012 \qquad w_{20}^* = 0.22012$$

Thus D^* is found as

$$D^* = \left(\frac{1}{0.55546}\right)^{0.55546} \left(\frac{1}{0.44368}\right)^{0.44368} \left(\frac{1}{0.00086}\right)^{0.00086}$$

$$\times \left(\frac{2 \times 0.55460}{0.33276}\right)^{0.33276} \left(\frac{3 \times 0.55460}{0.22184}\right)^{0.22184} (5)^{0.22012}$$

$$D^* = 6.6532$$

This value of D^* is equal to f^* by 8.75. Optimal values of the primal variables are found by solving equations 8.77. Six such equations are available but only four are required and they yield

$$x_1^* = 0.12423; x_2^* = 1.60998; x_3^* = 0.08713; x_4^* = 1.53028$$

Thus the problem has been solved.

8.6.3 Comments on Posynomial GP

As was mentioned towards the end of section 8.6.1.2 the greatest advantage of the dual GP problem is that it can reduce the dimensionality of the primal problem to be solved. When the primal problem has non-linear posynomial constraints the dual problem has the added advantage of a much simpler form. Geometric programming transforms a non-linearly constrained primal problem into a dual form which can be solved as an unconstrained non-linear problem in a reduced number of variables. When the dual problem has zero degrees of difficulty its solution is almost trivially simple. For one degree of difficulty problems, hand solution is still very easy using simple line search methods. For problems with more than one degree of difficulty more effort is required and hand calculations must yield to computer solutions. Any of the methods described in chapter 7 for solving multi-variable unconstrained problems may be used with simple checks incorporated to ensure that all dual variables remain non-negative. The dual prob-

www.EngineeringBooksPDF.com

lem has analytical derivatives so first-order or second-order (quasi-Newton) unconstrained methods can be used. The author has used Fletcher and Reeves' conjugate gradient method to solve GP problems of up to one hundred degrees of difficulty.

In a computational sense, GP has one advantage over other constrained optimisation methods in that it has a standard primal form, problem 8.73. If a problem fits the form 8.73 it should be possible to obtain a computed solution to the problem merely by inputting data defining the numbers of variables and constraints and all coefficients and exponents. Given this information, the computer can set up the appropriate dual problem, perform the maximisation and transform back to give a solution of the primal problem automatically. Computer programs that do this have been written but are not yet widely available. Given such programs, GP problems would be as easy to solve for the computer user as are LP problems at this time. The standard problem format 8.73 makes this feasible and adds to the potential of GP.

The posynomial problem 8.73 may be shown to be a convex problem. This means that the solution of a posynomial geometric programming problem will be a *global* optimum. This is rather a double-edged sword. It means that a globally optimal solution can be guaranteed for any problem that fits the form 8.73 and this is a considerable bonus. It also means, however, that any problem known to possess local optima cannot be solved by geometric programming.

When solving constrained GP problems it is sometimes found that the dual variables corresponding to the terms of some constraints are zero when D is maximised. If for some constraint j all the dual variables corresponding to that constraint, w_{jt}, $t = 1, \ldots, T_j$ are zero then by definition $w_{jo} = 0$. These dual variables have no effect upon D and this merely means that constraint j is slack at the optimum. In transforming back to find values for the optimal primal variables by using 8.76 or 8.77 the equations corresponding to terms in slack constraints should be omitted.

In this section 8.6 only the basic GP method has been described with sufficient mathematical background as was necessary to derive the results. Several proofs were omitted. The interested reader should consult the specialist texts listed in the bibliography for more details of this rather complicated, though very useful, optimisation technique.

8.6.4 Generalised (Non-posynomial) GP

The standard form 8.73 of posynomial geometric programming is, as stated above, a computational asset. It is also a practical disadvantage because very often problems arise which do not quite fit the standard form. Clearly, maximisation problems do not fit nor do lower-bound constraints. If these are algebraically manipulated and converted to minimisation or upper-bound constraints it is usually found that the manipulation has caused some term coefficients to become negative and this does not fit the standard form either. If negative coef-

www.EngineeringBooksPDF.com

ficients could be accommodated within the GP method then these difficulties would disappear. The method would then be able to handle maximisation or minimisation and upper or lower-bound constraints and would be much more generally applicable.

Unfortunately the primal/dual relationships of GP depend on the convexity of the problem and that convexity depends on the coefficients all being positive. This is a fundamental impasse that cannot be perfectly resolved. It is possible, however, to circumvent the impasse. This is done by making posynomial approximations to all non-posynomial functions in the problem and solving the non-posynomial problem as a converging sequence of posynomial GP problems. In this book it is not possible to describe the several ways of approximating general NLP problems by posynomial GP sequences. The book by Beightler and Phillips should be consulted. Considerable success, however, has been achieved in solving general problems by sequential geometric programming. At the core of this approach is the solution of posynomial GP problems as described in this chapter. When sequential GP is used on a general problem all the convexity and globality properties disappear and the sequence can only be guaranteed to locate a local optimum. This is no worse a guarantee, however, than any offered by other constrained optimisation methods.

SUMMARY

This chapter has examined several approaches to the solution of constrained non-linear optimisation problems. Simple methods such as the selection of active constraints can sometimes be effective on small problems and Lagrange multiplier methods for both equality and inequality-constrained problems have the advantage of mathematical rigour. They also provide the theoretical background to constrained optimisation but do not form usable numerical solution methods except for simple problems. Penalty function methods owe much in concept to Lagrangian methods and they are much used in solving design optimisation problems. Interior and exterior penalty functions were considered.

Linearisation methods, particularly sequential LP, were then examined. It was shown that although SLP superficially has much to commend it, it is beset by fundamental difficulties when applied to general non-linear problems. A theoretically much better method, though one little used, is sequential quadratic programming. Approximation methods such as SLP and SQP have been most used in solving very large non-linear constrained problems with many variables and constraints. Direct numerical search methods were not studied in detail because they are generally inferior to penalty function methods. The difficulties of adequately searching a constrained space were examined. Finally, geometric programming was studied. GP was derived in some detail for unconstrained posynomial problems and the primal/dual nature of the method was explained. The extension of GP to constrained problems was studied by means of numerical

www.EngineeringBooksPDF.com

examples and, finally, consideration was briefly given to generalised non-posynomial geometric programming.

This chapter and chapter 7 have examined the solution of non-linear-optimisation problems in a mathematical and numerical fashion. Chapter 6 established that almost all engineering design problems are non-linear. The next chapter redirects attention towards civil engineering problems and explains how the methods described in chapters 7 and 8 may be used to solve civil engineering design and planning problems.

BIBLIOGRAPHY

Avriel, M., Rijckaert, M. J., and Wilde, D. J., (eds) *Optimization and Design* (Prentice-Hall, Englewood Cliffs, N. J. 1973)

Beightler, C. S., and Phillips, D. T., *Applied Geometric Programming* (Wiley, New York, 1976)

Duffin, R. J., Peterson, E. L., and Zener, C., *Geometric Programming – Theory and Application* (Wiley, New York, 1967)

Fiacco, A. V., and McCormick, G. P., *Non-linear Programming. Sequential Unconstrained Minimization Techniques* (Wiley, New York, 1968)

Fox, R. L., *Optimization Methods for Engineering Design* (Addison-Wesley, Reading, Mass., 1971)

Rao, S. S., *Optimization theory and applications* (Wiley Eastern, New Delhi, 1978)

Wilde, D. J., and Beightler, C. S., *Foundations of Optimization* (Prentice-Hall, Englewood Cliffs, N. J., 1967)

Wilde, D. J., *Globally Optimal Design* (Wiley Interscience, New York, 1978)

Zener, C., *Engineering Design by Geometric Programming* (Wiley Interscience, New York, 1971)

EXERCISES

(Note: A variety of methods may be used to solve these problems and it is instructive if as many different ways as possible are used to obtain and to check the solutions.)

8.1 Minimise $f = x_1^2 + 2x_2^2 - 3x_1x_2 - 4x_1$
subject to $5x_1 + x_2 = 6$

8.2 Minimise $f = x_1^2 + x_2^2 + x_3^2$
subject to $x_1 - x_2 = 0$
$x_1 + x_2 + x_3 = 1$

www.EngineeringBooksPDF.com

8.3 Minimise $f = x_1^2 + x_2^2 - 2x_1 - 10x_2 + 26$

$$-x_1^2 + x_2^2 + 4x_1 \leqslant 4$$
$$x_1 + x_2 \geqslant 3$$

8.4 Maximise $f = 10x_1 + 5x_2 + 2x_1x_2 - x_1^2 - 2x_2^2$

subject to
$$x_1 + 2x_2 \leqslant 20$$
$$3x_1 - 2x_2 \leqslant 10$$
$$x_1 \leqslant 5$$

8.5 The solution of the problem

Maximise $f = x_1 + \alpha x_2$

subject to
$$x_1^2 + x_2^2 \leqslant 13$$
$$-x_1 + x_2 \leqslant 2$$

is located at the point $x_1^* = 2, x_2^* = 3$. What is the value of the positive coefficient α and the maximum value of f?

8.6 Minimise the following functions by geometric programming

(a) $f = \dfrac{10}{x_1x_2} + \dfrac{5}{x_2} + 2x_1^2x_2^3$

(b) $f = \dfrac{100}{x_1x_2x_3} + 10x_1x_2 + 20x_1x_3 + 30x_2x_3$

(c) $f = x_1^2 + 2x_2 + \dfrac{3}{x_1x_2} + 4x_1x_2$

8.7 Minimise $f = \dfrac{10}{x_1^2} + 2x_2^2x_3$

subject to $\dfrac{2x_1^2}{x_2^2} + \dfrac{3x_2}{x_3} \leqslant 1$

$$x_1, x_2, x_3 > 0$$

8.8 Minimise $f = \dfrac{1}{x_1x_2^2x_3^3}$

subject to
$$x_1^3 + x_2^2 + x_3 \leqslant 1$$
$$x_1, x_2, x_3 > 0$$

8.9 Figure 8.4 shows the general arrangement of an overhead crane which must be designed to lift a maximum of 100 kN. The winding gear located at C has an automatic power cut-out which cuts off power to the winding gear when the vertical displacement of C reaches 10 mm. Find the cross-sectional areas of all

www.EngineeringBooksPDF.com

the truss members so that a minimum volume of material is used in the truss and the cut-out operates when the design lifting capacity is reached. Ignore buckling effects in compression members. The elastic modulus, $E = 2.1 \times 10^5$ N/mm^2. Hint: Use Lagrange multiplier method to minimise truss volume subject to a displacement constraint.

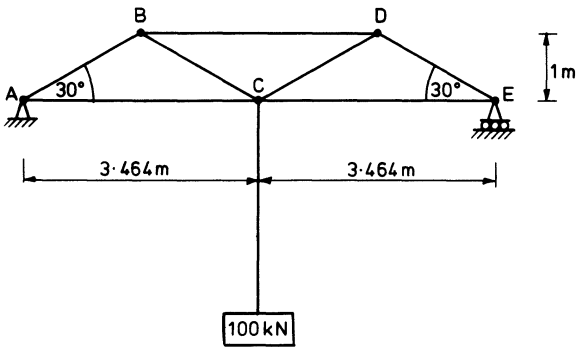


Figure 8.4

8.10 An oil storage depot for a refinery is to be planned consisting of a number of tanks similar to that shown in figure 6.2. Sufficient tanks to store 10^6 m^3 of oil must be provided. Unit costs are as follows. The circular cylinders (tank sides) cost £10/m^2 of surface area. The hemispheres (tank ends) cost £40/m^2 of surface area. The tank supports, foundations and ancillary works cost £10^{-5}/m^6 (that is, cost is proportional to the square of the volume of the tank). How many tanks should be provided, at what total cost and of what dimensions if the total cost is to be as small as possible?

8.11 A trapezoidal canal is required to have a cross-sectional area of 50 m^2. Find values for the canal dimensions b, d and ϕ as shown in figure 8.5 so that the discharge will be as large as possible. Discharge is maximum when mean velocity is maximum. Mean velocity increases with hydraulic radius, A/p, where A is the cross-sectional area of flow and p is the wetted perimeter.

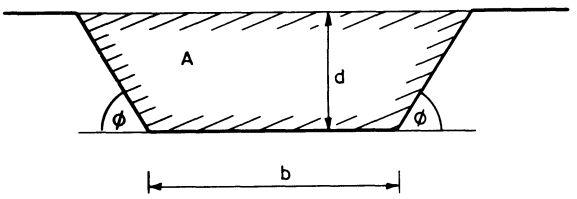


Figure 8.5

www.EngineeringBooksPDF.com

8.12 A rectangular cross-section bar 5 m long has breadth, b and depth, d and is pin-ended. For part of its life it must carry only a uniformly distributed load of 10 kN/m as a beam and its maximum deflexion must not exceed 15 mm. For the rest of its life it must carry only an axial compressive load of 500 kN and must have a factor of safety of 2 against Euler buckling failure. Find the dimensions b and d of the bar such that the total volume of material in the bar is as small as possible. The elastic modulus of the bar material is 2.1×10^5 N/mm^2.

www.EngineeringBooksPDF.com

9 NON-LINEAR OPTIMISATION IN CIVIL ENGINEERING

Chapter 6 showed that most civil engineering design problems are non-linear ones and consequently that optimal design requires non-linear optimisation methods. Chapters 7 and 8 have examined non-linear optimisation from a numerical viewpoint and have developed several solution methods. This chapter swings attention back towards civil engineering and shows how practical civil engineering problems may be modelled mathematically so as to use some of the solution techniques of the previous chapters.

An important aspect of the mathematical modelling of practical problems is that the nature of the solution required determines the form of the mathematical model. A model for a preliminary feasibility study can be very different from a macro or a micro-design model. A feasibility model only requires a skeleton design sufficient to allow the nature and performance of the design to become apparent and to permit broad cost estimates to be made. In macro-design the feasibility study skeleton must have more detailed flesh added and design models at this stage should be capable of producing sufficiently accurate results for a general arrangement drawing to be made for the design. In the micro-design stage a mathematical model should be capable of adding detail to the general arrangement drawing. For any project there is, therefore, a hierarchy of mathematical models, all different and each one tailored to produce a specific type of end result.

Often very simple mathematical models can be used, particularly in the early stages of design. Simplicity should perhaps be the guiding principle of all mathematical modelling of practical projects. Start with a very simple model, making sensible assumptions and approximations, and evaluate the results of the model very critically. As the shortcomings of the model become apparent the model can be expanded in those areas so that reality is better approximated. Thus the model can grow in the directions indicated by the results until the results become sensible enough and accurate enough for confidence to be placed in them. As has been noted earlier in this book there are always some subjective factors in

www.EngineeringBooksPDF.com

any design that cannot be mathematically modelled. No model can ever be complete for this reason. Each model, and its results, is merely another stage in the process of decision-making which culminates in the completed civil engineering project.

In chapters 7 and 8 the solution methods which were derived were illustrated by simple numerical examples capable of solution by hand. For larger numerical problems with more variables and constraints a computer may be needed to obtain solutions. Most of the methods of this book may be programmed for rapid computer solution. This chapter is more concerned with formulating mathematical problems to model practical civil engineering projects than with solving the resulting problems. It is assumed in this chapter, therefore, that once a practical problem has been formulated in one of the standard forms studied earlier, the solution of the problem can be achieved either by hand or, more usually, by computer.

EXAMPLE 9.1 – A PUMPED PIPELINE

A new oil refinery is proposed at a site that is 20 km from a port at which tankers will unload. The oil is to be conveyed from port to refinery by means of a pipeline. The terrain is mostly flat and the oil must be pumped along the pipeline. Explore this pumped pipeline project using mathematical optimisation models.

This example demonstrates the different levels of mathematical modelling that can be used to investigate different stages of the design of the project. The pumped pipeline forms only a small but an essential part of the refinery project and during the planning phase of the refinery a very rough schematic design for the pipeline will be needed so that broad estimates of cost can be assessed. A very simple model can be used for this purpose.

Since the terrain is flat it can be assumed that the pumps must provide all the pressure head needed to drive the oil down the pipeline. The pressure head is dissipated along the length of the pipeline by friction between liquid and pipe. For a pipe of length L and internal diameter D the frictional head loss h is approximately given by

$$h = \frac{\alpha L Q^2}{D^5} \tag{9.1}$$

where Q is the volume of liquid flowing in the pipe per second and α is an ascertainable constant depending upon pipe roughness, liquid viscosity, etc. It can be assumed that some desired volumetric flow rate Q is known from the nature and requirements of the port facility and the refinery. Thus the variables in 9.1 are the pipe size D and the head loss h. The pump must provide a total

www.EngineeringBooksPDF.com

pressure head H which must be at least as large as h to maintain flow. The power output required from the pump is proportional to the weight of liquid pumped per second multiplied by the pressure head produced. Thus the pump power output, P, can be expressed as

$$P = \beta QH \tag{9.2}$$

where β is again an ascertainable constant. Making a rough approximation that the cost of a pipe is proportional to its diameter and that the cost of a pumping station is proportional to the power it produces, the cost of a single pump and a length L of pipeline is

$$C = C_1 LD + C_2 \beta QH \tag{9.3}$$

where C_1 and C_2 are unit cost coefficients for the pipeline and for the pump, respectively. Thus a very simple mathematical model for the pumped pipeline is

$$\begin{aligned} &\text{Minimise } C = C_1 LD + C_2 \beta QH \\ &\text{subject to } H \geqslant \frac{\alpha LQ^2}{D^5} \end{aligned} \tag{9.4}$$

Problem 9.4 may be reorganised into geometric programming form as

$$\begin{aligned} &\text{Minimise } C = (C_1 L)\ D + (C_2 \beta Q)\ H \\ &\text{subject to } (\alpha LQ^2)\ H^{-1}\ D^{-5} \leqslant 1 \end{aligned} \tag{9.5}$$

in which the terms in parentheses are constants.

Solving problem 9.5 using the GP dual, the dual constraints are

$$\begin{aligned} & w_{01} + w_{02} = 1 \\ \text{for } D,\ & w_{01} - 5w_{11} = 0 \\ \text{for } H,\ & w_{02} - w_{11} = 0 \end{aligned} \tag{9.6}$$

These three equations can be solved for the three dual variables w to give

$$w_{01} = 5/6;\ w_{02} = 1/6;\ w_{11} = 1/6$$

Thus, since $w_{01} = 5w_{02}$, the optimal design will be one in which the pipeline costs five times as much as the pumps. The total cost will be

$$\begin{aligned} C^* &= \left(\frac{C_1 L}{5/6}\right)^{5/6} \left(\frac{C_2 \beta Q}{1/6}\right)^{1/6} (\alpha LQ^2)^{1/6} \\ &= 6LQ^{1/2} \left(\frac{C_1^5 C_2 \alpha \beta}{5^5}\right)^{1/6} \end{aligned} \tag{9.7}$$

www.EngineeringBooksPDF.com

The values of the optimal primal variables are then found to be

$$D^* = Q^{1/2}\left(5\,\frac{C_2}{C_1}\,\alpha\beta\right)^{1/6} \tag{9.8}$$

$$H^* = \frac{L}{Q^{1/2}}\left(\frac{C_1}{5\beta C_2}\right)^{5/6}\alpha^{1/6} \tag{9.9}$$

Examining these results it is seen that the pipe diameter D^* does not depend upon the length of the pipeline but the pressure head required H^* is proportional to L. This means that if a single pumping station is built at the head of a 20 km long pipeline, the internal pressure in the pipe close to the pump will be very high and the pressure at the refinery end will be low. For a constant diameter pipe the large pressures close to the pump will require a thick-walled pipe but at the refinery end a thick-walled pipe will not be necessary. The model, however, is not able to reflect the cost effects of such pipe design considerations.

Clearly the great difference in pressures at the ends of the pipeline present difficulties which must be overcome. One way of doing this is not to use a single pump for the whole length of pipe but to use several equally spaced pumps along the length of pipe. If n pumps are used, spaced a distance L/n apart, the total cost (C^* from 9.7) remains the same since $n \times L/n = L$ as before. The pipe diameter (D^* from 9.8) remains the same, but the maximum pressure head required (H^* from 9.9) is reduced from H^* to H^*/n. Thus the required number of pumps n may be found by equating H^*/n to the maximum permissible pressure in a pipe of diameter D^*. Figure 9.1 shows the effect on maximum pressures and head losses of replacing a single pump and pipeline by five equally spaced pumps.

Only an extremely simple model, problem 9.4, has been used to study the pipeline project yet it has allowed the balance between pumping costs and pipe costs to be explored and has given a feel for the interrelationships which exist in the design. Problem 9.4 however, has now outlived its usefulness for further study as it is too simple to permit a more detailed design to be made. Its shortcomings are apparent; for example, the model assumes that n pumps of power P/n will cost the same as a single pump of power P, an assumption which is intuitively untrue. The model must now be expanded and modified to represent reality more accurately.

Let n be the number required of equally spaced pumps each producing a pressure head H. A useful rule of thumb for the initial cost of the pumps is the six-tenths rule which reflects that the initial cost of any power plant is roughly proportional to the power output raised to the six-tenths power. Thus the initial costs of the pumps can be represented by $\tilde{C}_2 n(\beta QH)^{0.6}$ with $\tilde{C}_2$ a cost coefficient. Lifetime operating costs of the pumps can be included as they are likely to be significant. Assuming operating costs to be proportional to power output gives a cost term $\tilde{C}_3 n\beta QH$ in which $\tilde{C}_3$ should be representative of future operating costs

www.EngineeringBooksPDF.com

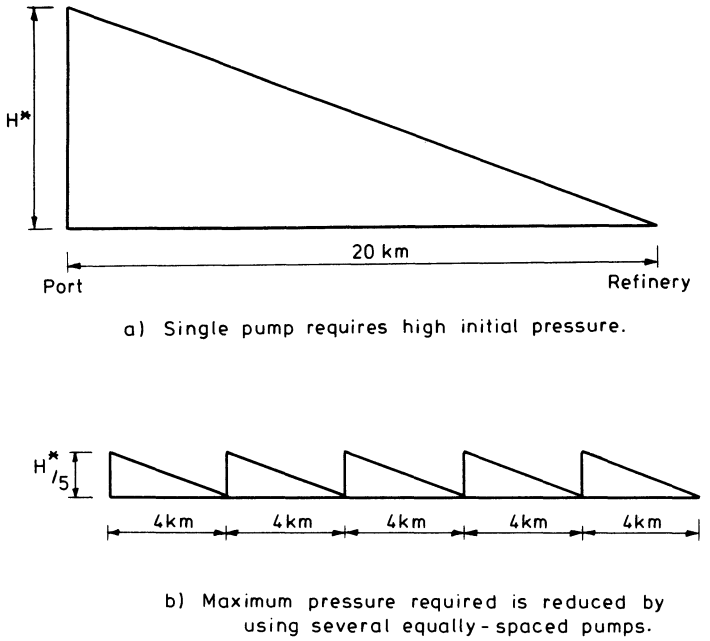


Figure 9.1 Pressure distributions for single-pump and five-pump pipelines

over the lifetime of the pumps. The pipe cost functions can also be improved by modelling them as a non-linear function of the diameter, D. Pipe costs increase more rapidly than linearly as the diameter increases and the quadratic cost term $\tilde{C}_1 L(D + \gamma D^2)$ reflects this non-linearity. Coefficients $\tilde{C}_1$ and γ can be found by regression analysis of pipe cost data.

In addition to the constraint that the head, H, produced by the pumps must exceed the pipe head losses, an extra constraint must be added to limit the maximum pressure in a pipe to $\tilde{H}$, the permissible pipe pressure corresponding to the pipe cost data. If these improvements and modifications are collected together into a new mathematical model the following geometric programming problem is obtained

$$\left.\begin{aligned} &\text{Minimise } C = (\tilde{C}_1 L)\, D + (\tilde{C}_1 L\gamma)\, D^2 + (\tilde{C}_2 \beta^{0.6} Q^{0.6})\, nH^{0.6} \\ &\qquad\qquad + (\tilde{C}_3 \beta Q)\, nH \\ &\text{subject to } (\alpha L Q^2)\, D^{-5}\, n^{-1}\, H^{-1} \leqslant 1 \\ &\qquad (\tilde{H}^{-1})\, H \leqslant 1 \end{aligned}\right\} \quad (9.10)$$

In problem 9.10 the terms within parentheses are constants. The variables are D, n and H; the problem has six terms and three variables and is, therefore, a two-degree-of-difficulty problem. It will not be solved here. Solving this prob-

www.EngineeringBooksPDF.com

lem would yield a more reliable cost estimate and design. It would not necessarily yield an integer value for n, however, so some rounding of n must be done and some further simple modifications made to determine a sensible design for the pipeline.

Feasibility studies can often be carried out using simple models such as those above. So long as the model is representative of all the major effects within the problem, the simpler it is the easier it will be to set up and solve. Feasibility studies are concerned only with major effects, with a very broad perspective of the project and with the balance among cost elements. In this example the cost balance arose because as the pipe size decreases the pipe costs decrease but more head must be provided to overcome viscosity effects so pumping costs increase. This balance can be determined by very simple models.

The feasibility model, problem 9.10 and its successors, should determine a rough design for the pipeline. The number of pumping stations and the pump power output from each should be known, as should an approximate pipe diameter. A design such as this, however, must be improved further to accommodate the practical realities that pipe sizes are discrete quantities and that pump sizes are also discrete. The macro-design stage must convert the continuous impractical feasibility study into a discrete practical engineering design. For this pipeline example an obvious choice of model for the macro-design stage is that of a serial system (see chapter 5). Dynamic programming is ideally suited to handle the discrete nature of pipe sizes and pump power outputs. Figure 9.2 shows the pumped pipeline design expressed as a serial system. Each stage in the system consists of a pump and the length of pipe between the pump and the next pump (or the refinery for the last stage). There are, therefore, n sequential stages. The state variable is the pressure head $\bar{H}$ in the pipe at the end farthest from the pump. The decision variables at each stage are the pump power output (and hence

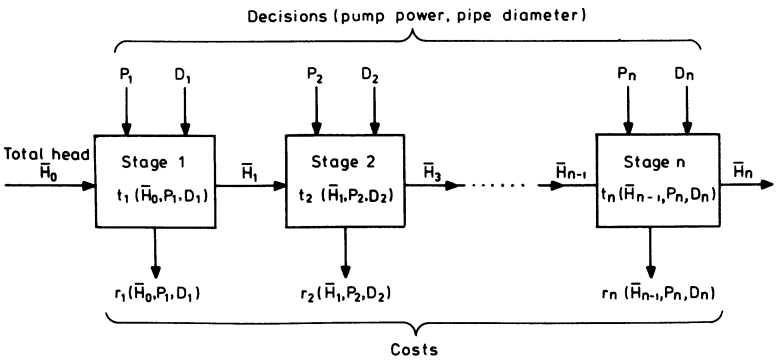


Figure 9.2 The pumped pipeline as a serial problem

www.EngineeringBooksPDF.com

the pressure head added) and the pipe diameter D (which determines the pipe head losses). Naturally only discrete values for P and D need be used.

This representation as a serial system allows other practical aspects of the problem to be incorporated easily. For example, the terrain may not permit pumping stations to be spaced out at exactly equal distances. Some pipe lengths may have to cross rivers or roads or be diverted around obstructions and will, therefore, be longer and more expensive than others. The serial framework and DP can easily accommodate these factors. The dynamic programming solution to the macro-design problem of figure 9.2 should produce a fully practical general arrangement of the major elements of the pipeline. The pipe sizes will correspond to commercially available pipes and the pumps also will correspond to readily available specifications. Furthermore, the over-all design will be the cheapest of all possible alternatives.

The micro-design stage of this pumped pipeline is perhaps less interesting from a systems viewpoint although much detailed design remains to be done: the pipe must be laid or supported, structures must be designed to house the pumps, river or road crossings must be designed. The micro-design stage must produce complete details for the whole project and there may well be some parts of this micro-design in which non-linear optimisation and the general systematic approach may be of use. Rather than attempting to tie this whole chapter to a single example, it is more useful to examine other applications of non-linear optimisation separately.

9.2 MICRO-DESIGN OF ENGINEERING ELEMENTS

The basic building blocks of many civil engineering projects are the beams, slabs, columns, walls, panels, etc., which are found in almost all construction projects. Much of the micro-design stage of any project consists of the detailed design of many of these basic elements. Since these elements are common to all projects their micro-design can be repetitive and time-consuming. For this reason small computers are increasingly being used for the design of common components to cut the design time for these elements. Non-linear optimisation provides a very convenient context in which to use computers to assist in the design of common components.

Figure 9.3 shows cross-sections of three typical structural beam elements: one in steel, one a reinforced concrete T-beam, one a precast concrete I-beam. At the micro-design stage of such elements the beam lengths are usually known as are the applied loads and bending moments. The designer's task is to provide a cross-sectional design which will safely equilibrate the known loads. He therefore has to select values for each of the variables x in the particular cross-section he is designing. Non-linear optimisation can be used computationally to provide the designer very rapidly with a set of cost-efficient values for these variables corresponding to a safe design: a good design suggestion which the designer may

www.EngineeringBooksPDF.com

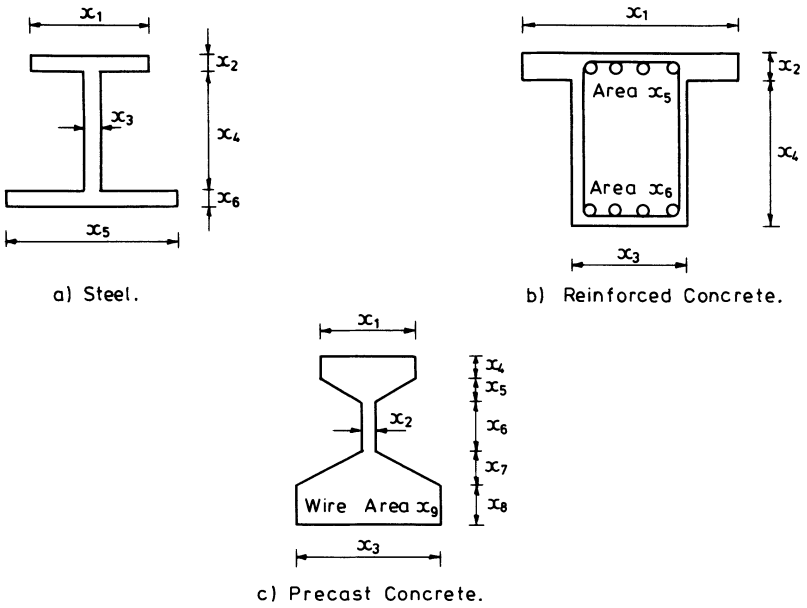


Figure 9.3 Typical beam element cross-sections

modify according to his subjective appraisal of its appropriateness.

As an example of how the optimisation problems can be constructed, consider the design of a steel beam such as that shown in figure 9.3a. An unequal flange section is chosen to demonstrate how non-standard sections can be easily designed. An equal flange beam would lead to a much simpler design model. Suppose the length of the beam L is known and that the six variables for which values must be found are as shown on figure 9.3a. Assume also that the maximum live bending moment M_L which the beam must carry is known as is the maximum live shearing force S.

Since the beam is made of steel, the cost of the steel plates will dominate the cost of the beam and will be approximately proportional to the volume of steel used. Fabricational costs such as welding will be significant but will be almost the same whatever cross-sectional dimensions are chosen. Their variation is not sufficient to warrant their inclusion in an objective function. A suitable objective function to be minimised is, therefore, the cross-sectional area of the section

$$\text{Minimise } f = x_1x_2 + x_3x_4 + x_5x_6 \tag{9.11}$$

The maximum bending moment to be carried is composed of the known live bending moment M_L plus the dead weight bending moment M_D. A conservative design will always result if these two are added and M_D is calculated for a simply supported beam at its centre. The maximum bending moment is then

www.EngineeringBooksPDF.com

$$M_{\max} = M_L + \frac{\rho L^2}{8}(x_1x_2 + x_3x_4 + x_5x_6)$$

in which ρ is the density of the steel of the beam.

The bending stresses in the extreme fibres of each flange must not exceed some known permissible values as specified by codes of practice. Since this is an unequal flange section these maximum bending stresses will be different and must be calculated carefully. Firstly the position of the horizontal neutral axis of the section must be found. Assume that y is the depth to the neutral axis measured from the top surface of the top flange. Then it is found that

$$y = \frac{\left[\frac{x_1x_2^2}{2} + x_3x_4\left(x_2 + \frac{x_4}{2}\right) + x_5x_6\left(x_2 + x_3 + \frac{x_6}{2}\right)\right]}{(x_1x_2 + x_3x_4 + x_5x_6)} \tag{9.12}$$

Knowing y, the moment of inertia of the cross-section can be expressed as

$$I = \frac{x_1x_2^3 + x_3x_4^3 + x_5x_6^3}{12} + x_1x_2\left(y - \frac{x_2}{2}\right)^2 + x_3x_4\left(x_2 + \frac{x_4}{2} - y\right)^2 + x_5x_6\left(x_2 + x_4 + \frac{x_6}{2} - y\right)^2 \tag{9.13}$$

The maximum stress in the extreme fibres of the top and bottom flanges can then be written as

$$\frac{M_{\max}\, y}{I} \text{ and } \frac{M_{\max}}{I}(x_2 + x_4 + x_6 - y)$$

Assuming for descriptive purposes that the top flange is in compression and the bottom flange is in tension, two stress constraints can be written:

$$\frac{M_{\max}\, y}{I} \leqslant \sigma_c \tag{9.14}$$

$$\frac{M_{\max}}{I}(x_2 + x_4 + x_6 - y) \leqslant \sigma_t \tag{9.15}$$

where σ_c and σ_t are maximum permissible stress values determined from codes of practice. Often, in codes of practice, permissible stress values are not given as constants but as variables which depend on section properties. Such is the case with σ_c which is usually found in tabular form for different values of the beam slenderness ratio, L/r, (length/radius of gyration) and D/T (over-all depth/compression flange thickness). The reason for giving σ_c in this form is so that lateral buckling of the compression flange may be prevented by specifying σ_c so that buckling cannot occur. For any known beam σ_c is found by interpolation of the tabulated values. This method of finding σ_c does not lend itself easily to computer methods or to non-linear optimisation. It would be much preferable if σ_c

www.EngineeringBooksPDF.com

were specified as a non-linear function of L/r and D/T because a functional relationship would be much more easy to incorporate in a computer program however complicated algebraically it might be. Using the tabulated values, however, an approximating function for σ_c may be fitted to the data by means of a polynomial in L/r and D/T. It is then a simple matter to express r, D and T as functions of the design variables x_1 to x_6. σ_c is then found as a function of x_1 to x_6 and forms the right-hand side of constraint 9.14. This is not done here as the algebra is complicated. A similar procedure can be adopted for other permissible code stresses that are not absolute constants.

Considering next the shearing stresses, codes of practice often prescribe a maximum average shearing stress, τ_{av}, and a peak shearing stress, τ_{max}, neither of which may be exceeded in the beam. The average shearing stress including dead weight shearing leads to a constraint

$$S/(x_1x_2 + x_3x_4 + x_5x_6) \leqslant \tau_{av} - \frac{\rho L}{2} \tag{9.16}$$

Peak shearing stress occurs at the neutral axis of the cross-section and leads to the constraint

$$\left[S + \frac{\rho L}{2}(x_1x_2 + x_3x_4 + x_5x_6)\right]\left[x_1x_2(y - \frac{x_2}{2}) + \frac{x_3(y - x_2)^2}{2}\right]$$
$$/Ix_3 \leqslant \tau_{max} \tag{9.17}$$

Constraints 9.14 to 9.17 represent the main structural actions of the beam but a variety of other constraints may be necessary. There will be a minimum permissible thickness for the steel plates, providing lower bounds on variables x_2, x_3 and x_6. The designer may wish to specify upper bounds on the over-all depth of the section and the widths of the flanges so that the beam cross-section fits the available space and matches up with other members. In addition, codes of practice suggest upper and lower bounds for the depth/width ratio of the cross-section so that instability effects are not encountered. These form further constraints. One feature which has not yet been considered in this design problem is that of providing vertical stiffeners for the web to prevent web-buckling. The need for such stiffeners, their number and spacing depends upon the proportions of the cross-section and web and flange thicknesses. It is possible to add their cost into the objective function by means of new variables describing their number, thickness and size and to form algebraic constraints which enable the variables to be assigned optimal values.

It is quite possible, in fact, to formulate a model for the complete, automatic, least cost design of a steel beam or for the other beam types shown in figure 9.3. Many such programs have been written and used successfully. It requires considerable computer programming effort, however, to produce a beam design program attempting to incorporate facilities to handle all possible load com-

www.EngineeringBooksPDF.com

binations and all possible constraints which a designer might wish to include, as well as generally being an all-embracing program. The resulting programs tend to be large, needing large computer facilities, and unwieldy and complicated to use. It seems more logical to use the computer as an aid to the designer rather than as a replacement for him. A design aid program for steel beam design would incorporate the objective function 9.11, bending and shearing stress constraints 9.14 to 9.17 and a menu of size and proportion constraints from which the designer can choose those he wants to include. The resulting program can be small and fast to use and would produce an economical initial design. It would leave to the designer the ultimate choice of whether the design looks right. Also, the designer would have to complete the design, rounding sizes where necessary, adding web stiffeners if they are needed and generally checking the engineering practicality of the beam.

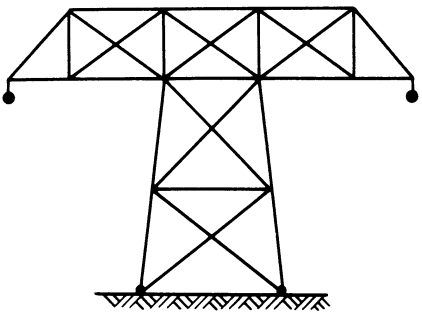
The steel beam example of figure 9.3a has been used to demonstrate the use of non-linear optimisation models in the design of engineering elements. The fact that the same simple elements occur so frequently in designs means that a suite of small programs, one for each of the beams in figure 9.3 and for other common elements such as columns and slabs, can be of considerable value in a design office. Time spent in repetitive design is reduced by the use of computers while the skills of the designers are retained and enhanced. Furthermore, the use of optimisation methods ensures that the resulting designs are cost efficient.

Experience has shown that very many of the simple common elements of civil and mechanical engineering can be optimally designed by appropriate small computer programs. Element micro-design problems are usually quite small but are highly non-linear. The steel beam design problem shows this high non-linearity. Usually the number of variables is small, less than ten, but there may often be a large number of constraints which may be active or slack as the loading and design necessitates. For these small, highly non-linear, multiply constrained element design problems several solution algorithms have been used successfully – geometric programming, penalty functions, direct constrained search methods are possible candidates.

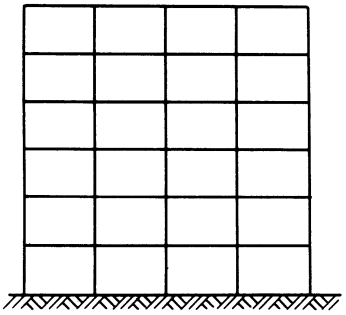
9.3 DESIGN OF MULTI-ELEMENT STRUCTURAL SYSTEMS

Two typical design problems which are frequently encountered in the macro-design stage are shown in figure 9.4. Figure 9.4a shows a large truss composed of members carrying predominantly axial forces. Secondary moments and shears are present in the members if rigid joints are used but for macro-design these secondary effects are ignored. Figure 9.4b shows a rigid-jointed structural frame composed of many beams and columns. This section examines briefly how macro-designs for structures such as these may be obtained using non-linear optimisation methods.

www.EngineeringBooksPDF.com



a) Structural truss.



b) Structural frame.

Figure 9.4 Multi-element structural systems

9.3.1 Optimum Truss Design

Steel trusses are a common feature of everyday life. Electricity pylons, overhead support structures for railway electric power cables and crane structures are examples. Each of these structures may be built in large numbers to the same design and, therefore, it is worthwhile making that initial design on a cost optimal basis. Each time a structure is built to that design a cost saving is made and the cumulative savings over all the identical structures can be large. Figure 9.4a shows a two-dimensional pylon structure for demonstration purposes. Real pylons are of course three-dimensional and this merely increases the number of members but does not affect the principles discussed here. It is assumed here that the general arrangement of the truss, its geometry, numbers and lengths of members are known. The macro-design problem considered here is that of finding a safe size for each member such that the total cost of the pylon is as small as possible. If the numbers and layout of members are not known the problem of design is far more complicated and is still a topic for further research. Since members carry predominantly axial forces, the designer's task is simply to find

www.EngineeringBooksPDF.com

a suitable cross-sectional area for each member. A typical truss, however, may have several hundred members so this simple problem is perhaps not so simple after all.

A structure such as the pylon of figure 9.4a must be capable of equilibrating several different and independent externally applied loads. Firstly, there is a symmetrical load case, a downwards force applied by the cables at their attachment points. Secondly, it is necessary to consider an unsymmetrical load case corresponding to the cables on one side of the pylon being absent perhaps carried away in a storm. The breaking of the cables should leave the pylon intact. Thirdly, a horizontal wind load should be considered in addition to both normal and unsymmetrical loading. Thus during its life the pylon may have to withstand several different extreme load cases and it must be designed to equilibrate them all. Pylons, and indeed most trusses, are statically indeterminate structures thus even if all the applied load cases are precisely known the axial forces in all the members cannot be uniquely found. They depend on the sizes of the members and these sizes are the problem variables.

For a steel truss composed of N members, the cost of the truss will be dominated by the cost of the steel in the members and, since steel cost is proportional to weight and volume, an appropriate objective function for macro-design purposes is

$$\text{Minimise } f = \sum_{i=1}^{N} L_i A_i \tag{9.18}$$

in which L_i, A_i are respectively the (known) length and (unknown) cross-sectional area of member i, $i = 1, \ldots, N$. This objective function does not include the cost of the joints at member ends which is a significant cost item but not one that has any large influence on member sizes. A major constraint upon the size of each member is that the axial stress caused by any of the independent external load cases must not exceed some maximum permissible value. If F_{ji} is the axial force in member i caused by applied load case $j, j = 1, \ldots, J$, the maximum stress constraint has the form

$$\frac{\max_{j=1,\ldots,J} \langle F_{ji} \rangle}{A_i} \leqslant \bar{\sigma} \qquad i = 1, \ldots, N \tag{9.19}$$

where $\bar{\sigma}$ is the known permissible stress. Note that $\bar{\sigma}$ may be different in tension and compression. It is assumed that if the maximum force is known, the appropriate $\bar{\sigma}$ will be used corresponding to the sign of the force. There will be N constraints like 9.19, one for each member. Since the truss is statically *indeterminate* the F_{ji}s in constraints 9.19 are not constants but are unknown functions of the member sizes A_i, $i = 1, \ldots, N$. In the case of a statically *determinate* truss the F_{ji}s will be constants determined by analysis for each of the load cases. For such a determinate truss, minimising 9.18 subject to constraints 9.19 leads to an optimum design in which each member reaches its maximum permissible

www.EngineeringBooksPDF.com

stress under at least one load case. This is known as a fully stressed design. Unfortunately, most practical trusses are not statically determinate and the solution of the problem of minimising 9.18 subject to 9.19 is not a trivial task. Constraints 9.19 do not form a convex set because the F_{ji}s depend upon the member areas and it has been proved that a fully stressed statically indeterminate truss is not necessarily optimal.

To add further difficulty to this truss design problem there are frequently other constraints present in addition to member stress constraints. It is usually necessary to limit the displacements of certain joints of these light flexible trusses. Typically in the pylon the attachment points for the cables may be required not to deflect by more than some prescribed amount δ. For joint k of the truss loaded by load case j the deflection constraint can be written as

$$\sum_{i=1}^{N} \frac{F_{ji}\, U_{ki}\, L_i}{A_i E_i} \leqslant \delta_k \tag{9.20}$$

In constraint 9.20 U_{ki} is the axial force in member i of the truss caused by a unit load applied to the truss at joint k in the direction in which the displacement is to be limited to a maximum permissible value of δ_k. E is the elastic modulus of the steel of the truss. Note that constraint 9.20 is a summation over all members of the truss and so may have many hundreds of terms in a large truss structure. A constraint of the form of 9.20 will be required for each joint at which the displacement is to be restricted and for each applied load case $j = 1, \ldots, J$. There may, therefore, be many constraints like 9.20.

In specifying a statically indeterminate layout for the truss it is assumed that no members vanish in the optimum design. In order to ensure that this does not happen and that no member is assigned an impracticably small area it is usual to prescribe a minimum size for all members. This leads to a set of N constraints 9.21.

$$A_i \geqslant A_{\min} \qquad i = 1, \ldots, N \tag{9.21}$$

In addition to constraints 9.19 to 9.21 it may be necessary to ensure that slender compression members do not fail by buckling. This can be done by modifying the right-hand sides of constraints 9.19 for compression members to

$$\min \langle \bar{\sigma}, \frac{\pi}{L} \sqrt{(\alpha E k \max_j <F_{ji}>)} \rangle$$

instead of $\bar{\sigma}$. α is the required safety factor against Euler buckling and k is the ratio I/A^2 that will be constant for all members having geometrically similar cross-sections differing only in scale. Another design consideration of slender steel trusses that can cause severe difficulties is the dynamic response of the truss. It is sometimes necessary to place constraints on the natural frequencies of the truss. Dynamic response constraints are too complex to be considered here.

The reason for the above detailed formulation of the truss optimum design problem is to demonstrate a fundamental difference in the natures of simple

www.EngineeringBooksPDF.com

element design problems and multi-component system design problems. The truss design problem consists of minimising 9.18 subject to constraints 9.19 to 9.21. Whereas element design problems have only a few variables, this system design problem may have many hundreds. The functions involved in element design were highly non-linear without any regular pattern to them; in this system design problem there is considerable pattern or structure in the mathematical model. The objective function is a linear summation over all variables. The displacement constraints are also summations of the reciprocals of all variables. Stress and minimum size constraints are also reciprocal functions, one variable per constraint. This problem is totally different from that of element design.

One solution method for the truss design problem is as follows. A complete set of member sizes is guessed and the truss is fully analysed to determine all the F_{ji}s and U_{ki}s. These 'constants' are entered into the design problem and the transformation of variables $z_i = 1/A_i$ is made which yields a problem of the form

$$\begin{aligned} &\text{minimise } f = \sum_{i=1}^{N} L_i/z_i \\ &\text{subject to } \sum_{i=1}^{N} a_{ji}\, z_i \leqslant b_j \qquad j = 1, \ldots, M \\ &\qquad\qquad 0 \leqslant z_i \leqslant c_i \qquad i = 1, \ldots, N \end{aligned} \tag{9.22}$$

In problem 9.22 all the 'constants' a_{ji}, b_j, c_i are known once member forces have been determined. Problem 9.22 has only linear constraints in the variables z but the objective function is non-linear. Several optimisation methods can be used to solve this problem. An obvious candidate is sequential quadratic programming. In fact for problems like this with linear constraints there are many possible methods similar to quadratic programming that handle the constraints in a simplex LP-type fashion while optimising a non-linear objective function. Problem 9.22 can therefore be solved quite easily to yield optimal values of the variables z and hence also of the variables A. Since all the member areas, however, will have been altered by the optimisation process from the initially guessed values, all the member forces and hence the 'constants' in problem 9.22 will also have changed. A new analysis is performed with the latest set of area variables to determine new member forces and new values for the 'constants'. Having set up a new problem 9.22 it is re-optimised to yield a further new set of member areas. This process of analysis/optimisation/analysis/optimisation is repeated until successive sets of member areas are sensibly the same. A final analysis then completes the process and confirms the validity of the optimum design.

The truss optimum design problem can therefore be solved by an iterative process of analysis and redesign. Assuming that member forces remain constant during optimisation, correcting them afterwards is not entirely rigorous and for this reason no convergence proofs for the iterative process can be given. Nevertheless convergence is usually quite rapid. Since the over-all problem is of a non-

www.EngineeringBooksPDF.com

convex nature the resulting optimum design can only be guaranteed to be locally optimal.

The pattern or structure of the optimum truss design problem has marked it out as a candidate for much research. There is a large amount of literature on solving optimum truss design problems using special purpose methods that exploit the regular structure of the problem.

9.3.2 Optimum Frame Design

Figure 9.4b shows a rigid-jointed frame, composed of beams and columns, that is typical of the steel or concrete frames found in many multi-storey buildings. How can frames such as figure 9.4b be designed for least cost? The design problem is more complicated than that of truss design because the frame stiffness is achieved by using rigid connections between members and consequently members are subjected to axial forces, bending moments and shearing forces whereas in the truss the externally applied loads cause only axial forces. Trusses in steel are relatively light and flexible compared with concrete frames. Concrete is relatively inflexible; large displacements cause cracking and a loss of structural integrity. Also since the frame usually has the wall panels infilled in a building structure, displacements of large magnitude could damage the infill panels and must be prevented. Consequently very rigid designs are normally used for such frames. Nevertheless, the optimum design problem for frames such as figure 9.4b is similar in nature to that for trusses; the number of members can be large and, assuming a characteristic size variable for each member, the number of variables in the problem is frequently large. The main performance requirements for the frame are that external loadings, of which there may be several independent cases, should cause stresses and displacements within permissible limits. These lead to constraints on each separate element of the frame and also to constraints that involve summations over all frame elements. Consequently frame design and truss design are classed together in terms of macro-design; in detail the problems are very different but in over-all nature they are similar displaying regular patterns or structure.

In chapter 2, example 2.3 described the rigid-plastic design of a steel frame and showed that the design problem could be cast into linear programming form – a highly specific sort of problem structure. In that example for each member the characteristic variable chosen was the fully plastic moment M_P which is a function of the member cross-sectional geometry. Frames such as figure 9.4b are frequently designed so that ultimate collapse just occurs under factored-up working loads. Consequently many frame optimum design programs use this design approach and employ linear programming or sequential LP as solution algorithms. Clearly in a frame such as that of figure 9.4b there will be very many possible arrangements of plastic hinges and failure modes. Much research has been done on developing ways of selecting as constraints for the problem only a few most likely candidates from the large number of possible collapse modes. Also

www.EngineeringBooksPDF.com

rigid-plastic design makes many assumptions about the behaviour of the structure under load that are gross approximations of reality. For example, it assumes that the frame material has an infinitely large elastic modulus thus no displacement occurs until the yield stress of the material is reached. Then it assumes that the stress remains constant for all material strains. This stress-strain characteristic is only a very rough approximation for steel and is far worse for concrete. For practical design purposes many design methods have replaced the rigid-plastic assumption by stress-strain characteristics that more accurately represent steel or concrete. This effectively complicates the design problem and introduces non-linearities instead of linearities. Approximating these non-linear problems by a sequence of LP problems has proved useful as a solution method because the degree of non-linearity is not high.

As with optimum truss design a lot of research has been done on optimum frame design and many practical design computer programs exist. The interested reader should consult the literature for further information. Both truss design and frame design are essentially macro-design problems resulting in optimal values for single characteristic variables for all the members. They are sizing programs and can be thought of as design aids. They do not produce final designs but only design suggestions which for the model used are optimal. Further micro-design of each element is necessary within the limits of characteristic variables set by the macro-design. The models described in this section are, therefore, only intermediate stages in the entire design process.

9.4 OTHER NON-LINEAR PROBLEMS

This chapter has concentrated on design problems because they are almost always non-linear. This is not to suggest that non-linear optimisation is confined only to the design phase. In fact non-linear problems can also be found in planning, construction and operation. It has earlier been noted in connection with linear programming that LP problems arising from practical applications appear to be linear only when viewed with a very broad perspective. As details are added or when the problem is examined more closely the linearity often dissolves into discreteness or non-linearity. A major source of non-linear optimisation problems lies in applications which started as linear programming but which developed non-linearities as improvements in modelling were made. In section 9.3.2 the LP problem for rigid-plastic design became non-linear as more realistic material properties replaced idealised properties.

It is often the objective function of an LP problem that develops non-linearities. Very often a linear objective function can develop into a quadratic function converting the whole problem to one of quadratic programming. Figure 9.5 shows how this can happen.

Suppose there is an LP problem of the transportation type. Some commodity is being conveyed from several sources to several destinations. Variables are the amounts carried between source i and destination j. Problem constraints represent

www.EngineeringBooksPDF.com

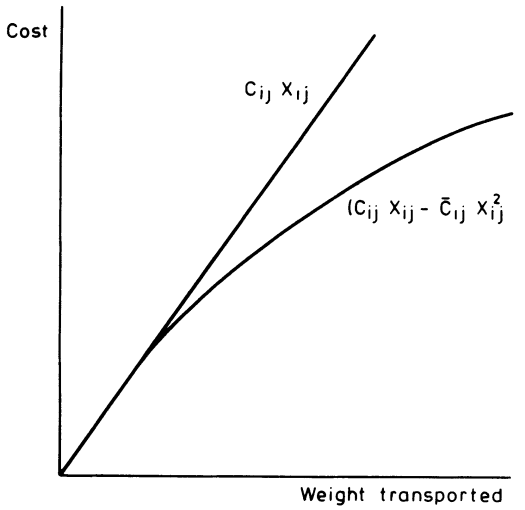


Figure 9.5 Economies of scale lead to non-linear costs

the supply availability of the commodity at the sources and the demand requirements at the destinations and are linear constraint functions. The usual objective function is the sum of the transportation costs on all source–destination pairs. This objective function which is to be minimised assumes that the cost of transporting X_{ij} tonnes of the commodity between i and j is directly proportional to the weight transported and is $C_{ij}\ X_{ij}$. In figure 9.5 this linear cost is shown by the straight line on a graph of cost against weight. In reality transportation costs are often amenable to economies of scale. Really large quantities of the commodity can be carried at a lower unit cost than small quantities because of the possibilities of vehicles carrying full loads instead of partially full loads and of using larger capacity more cost efficient vehicles. A truer representation of transportation costs is given by the curved line on figure 9.5. As the weight transported increases, the effective unit cost decreases. This discount effect or economy of scale effectively changes the linear objective function into a non-linear one. A quadratic function $(C_{ij}\ X_{ij} - \bar{C}_{ij}\ X_{ij}^2)$ can be used to represent the non-linear costs and the total cost to be minimised is then the sum of all these quadratic functions. For this optimisation problem quadratic programming will provide the solution.

This example is typical of many problems encountered in which economies of scale convert a linear programming problem into a non-linear programming problem. At the beginning of this chapter a general principle of modelling was suggested that models should start simply and should be modified and improved both in the light of the results and according to how well the model corresponded with reality. Linear programming is one model form that is simple to construct and can yield excellent results for many problems. In view of the principle above

www.EngineeringBooksPDF.com

it is only to be expected that as a linear model is required to approximate reality more closely it will develop non-linearities that require different solution techniques.

SUMMARY

This chapter has examined ways in which non-linear optimisation methods can be used to solve practical civil engineering problems. Design problems almost always turn out to be non-linear and it has been shown that feasibility studies, macro-design and micro-design of any project all require different types of mathematical models. Non-linear optimising models can be used to explore the feasibility of a project using a broad perspective and simple models. Successive models can incorporate more detail as it is seen to be necessary and can be used to explore the macro and micro-design stages. There is no such thing as a single optimising model for a project. Design is achieved by using many different models each of which is discarded in favour of a better model when its results are no longer seen to represent reality as well as is desired.

Models for the macro-design of common multi-component structures were seen to be large with possibly many hundreds of variables and constraints. This large problem size is often offset by the regular pattern or structure of the functions involved. This structure can be exploited to simplify the solution methods required. In contrast, models for the micro-design of common engineering elements are usually small and compact. They usually have only a few variables but the functions are highly non-linear and have very little pattern or structure to them. Nevertheless, their solution can usually be found by a variety of non-linear optimisation methods. Uses of non-linear optimisation methods are not confined to the design phase. They can be found throughout planning, construction and operation, very often as natural developments of linear programming problems.

BIBLIOGRAPHY

The technical journals are undoubtedly the best source for further reading on non-linear optimisation applications to problems of relevance to civil engineering. Since the range of applications is very wide, many papers have been published in this area. To narrow down the field it is best to have a specific engineering problem in mind and to consult recent issues of a technical journal covering that engineering topic. The divisional publications of the Americal Society of Civil Engineers or the abstracting journals such as Engineering Abstracts often provide good starting points. So great is the volume of literature that a general survey of applications in civil engineering, or even a short selective bibliography, would be unrepresentative.

www.EngineeringBooksPDF.com

10 PROBABILISTIC DECISION-MAKING

All the previous chapters of this book have been concerned with decision-making methods in which the consequences of a decision are predictable with certainty. The decision to dig a rectangular hole 2 m x 1 m x 1 m deep on a flat site has the predictable consequence that 2 m^3 of excavated material will be produced. The unique consequence of the decision is unavoidable and plans can be made for disposing of this material which is certain to be produced. Not all decisions are like this. In some cases the consequences of a decision are governed by chance and a whole spectrum of possible outcomes may result from a single decision. The decision to sink a new borehole for oil does not have a predictably certain consequence. Oil may or may not be found and, if it is located, may or may not be present in sufficient volume to be exploitable. Since the consequences of the decision to sink the borehole are not unique but are governed by chance, subsequent planning is made more difficult. Special methods for planning under uncertainty are needed and this chapter describes several.

Problems and methods for planning under uncertainty are wide and varied. This chapter does not attempt a general coverage. Instead, several practical examples of a civil engineering nature are examined and appropriate probabilistic solution methods are developed. The chapter begins with a simple introduction to the difference between deterministic and probabilistic quantities and then explores the nature of risk in probabilistic decision making. It is assumed that the reader has some mathematical background in probability and topics in this area necessary to the understanding of this chapter are briefly reviewed.

A major topic in this chapter is the use of expected values in decision-making. Several methods are examined, through examples, culminating in decision trees and utility values. The use of expected values in maintenance and replacement problems is also described.

The second major topic of this chapter is reliability. Methods for calculating the reliability of single components and multi-component systems are introduced and emphasis is placed on design methods for components and systems to achieve some desired reliability. This section concludes with a qualitative study

www.EngineeringBooksPDF.com

of different reliability-based optimisation formulations pointing out the disadvantages of each. Finally, the connection between system failure and its possible consequences of injury or death is examined in an optimisation context.

This chapter is necessarily selective in its choice of topics. Subjects such as simulation, queuing theory and inventory theory are certainly important ones with many civil engineering applications. They have been omitted with regret in favour of those chosen only because they are perhaps more easily accessible in other texts.

10.1 DETERMINISTIC AND PROBABILISTIC QUANTITIES

It is useful to begin by examining probabilistic quantities since in reality few quantities are deterministic. Imagine an anemometer permanently fixed to the top of a building to measure wind velocities. Suppose it is arranged that at noon every day the wind speed will be recorded. Over a period of months or years a large number of readings will be taken and they will show a large range of different values for the single measured quantity of noon wind speed. This quantity is termed a *probabilistic* one because it has a range of probable values and it is not possible to predict with certainty what tomorrow's noon wind speed will be. It may be possible to examine the records of previous noon wind speeds and perform statistical analyses of the recorded data. This might allow predictions to be made of the form: 'Tomorrow there is a 60 per cent chance that the noon wind speed will be somewhere in the range 5 m/s to 10 m/s' but they would not allow a single value to be predicted with certainty.

In contrast, if the anemometer is also able to record its own above ground height at the same time each day then, because the instrument is permanently attached to the building, the recorded height data over a long period would be constant. This height quantity is *deterministic* because it has a single, predictable value. It is possible to predict with certainty that tomorrow the height above ground of the instrument will be the same as it was today. This prediction ignores extraneous occurrences such as theft of the instrument or collapse of the building which do not concern us in this context. Provided that both instrument and building remain intact the recorded quantity, height above ground, will not change and so will be a deterministic quantity. Or will it? Is this height quantity really deterministic? In fact, the building will expand or contract as inside and outside temperatures change from day to day and from season to season and consequently on some days the recorded height will be fractionally larger than on others. Thus even this quantity which appears to be entirely predictable and deterministic is really probabilistic. The range of variations in height, however, will be small and very small in comparison with the mean height so for most practical purposes it is convenient to assume the height to be deterministic instead of probabilistic.

In reality very few quantities are truly deterministic but it is common practice to assume them to be so if the range of variation of the quantity is small enough

www.EngineeringBooksPDF.com

to have an entirely negligible effect on any calculations which must be made using that quantity. Everyone knows that the human animal has ten toes. Actually this is a deterministic approximation to a probabilistic quantity; some people, as a result of various misfortunes, do not have ten toes. For most purposes the assumption of ten toes per person is entirely reasonable. There are instances, however, when this assumption may not be reasonable. Clearly the orthopaedic shoe industry cannot use this assumption. It may be of concern to that industry to be able to estimate how many people in the UK have only eight toes, or no toes at all, in order to regulate the manufacture of special footwear. In this case the probabilistic nature of the quantity is of more importance than the conventional deterministic assumption. Pursuing this example a little further the normally assumed value of ten toes per person is a *characteristic value* used to replace the probabilistic nature of the quantity for most calculations. Ten is also the *peak value* for the distribution of the quantity – most people in a random sample have ten toes. Here the peak and characteristic values are the same. This is not generally true for all quantities as will be seen later. The *mean value* of number of toes per person is likely to be of the order of 9.99, that is, in a random sample of 100 people someone will probably have lost a toe. In this example the mean value is not of much interest and it would be misleading to use it instead of the characteristic value. For a probabilistic quantity it can sometimes be difficult to choose a single value to represent that quantity deterministically for calculation purposes.

In civil engineering characteristic values are used very frequently often without any appreciation of the fact that the quantities they represent are probabilistic in nature. It is much easier to use characteristic values than to view quantities in a probabilistic way. Most of the time this is perfectly reasonable. A designer may choose to use 10 mm diameter reinforcing bars in a concrete beam. Actually the diameter of such bars may vary slightly; bar diameter is actually a probabilistic quantity but the manufacture of reinforcing bars is so controlled that the range of variation is very small. In this instance 10 mm is the characteristic or nominal diameter and it probably corresponds to the large-sample mean diameter of bars which are nominally 10 mm diameter. Design calculations can be made using this characteristic value with negligible errors.

In the same design, however, the engineer might specify concrete with a characteristic ultimate strength of 20 kN/m^2. In doing this the designer should be aware that concrete can vary very widely in quality. If a number of supposedly identical concrete cylinders are cast from the same concrete mix and cured in the same environment before testing, they will give a fairly wide range of ultimate strength results. This variability is important. Some test specimens may be considerably below the mean strength. In this example it is not possible to use the mean strength as a characteristic value for design purposes since then much of the concrete would be below mean strength and failure might occur. What then should be used as a characteristic value? One often used value is the 95 percentile, that is, the concrete must be such that of all samples tested the ultimate strength must be at least as large as the characteristic value in 95 per cent of the samples.

www.EngineeringBooksPDF.com

Naturally, if the 95 percentile is chosen for design purposes, the actual concrete must be sampled and tested as it is placed to verify that this 95 percentile value is achieved. In this example the use of a deterministic equivalent for a probabilistic quantity must be accompanied by checks and safeguards to ensure that the variation in the probabilistic quantity does not invalidate calculations made using the deterministic value. It may seem that this is a rather elaborate mechanism to enable a deterministic value to be used. Would it not be better and simpler to accept the probabilistic nature of concrete and to perform design calculations in a probabilistic rather than a deterministic fashion? The answer to this question is that it might be better but it would certainly not be simpler. Probabilistic design is anything but simple and existing design methods using probabilistic quantities are underdeveloped and laborious. Therefore, engineers still tend to go to extreme lengths to try to find some characteristic deterministic value to represent a quantity which is really probabilistic in nature.

Throughout this book all quantities have been assumed to be deterministic. The only exception has been in chapter 4 where PERT used probabilistic activity durations for critical path construction planning. It would be possible to go systematically through this book replacing deterministic approximations to quantities by probabilistic distributions and to re-examine linear programming, dynamic programming, geometric programming and all the other topics in a probabilistic fashion. Many methods have been developed for solving linear and non-linear programming problems that have probabilistic variables or data. In civil engineering, however, these methods are very little used because for the most part the existence of usable deterministic characteristic values and methods for the sensitivity analysis of the results renders them unnecessary. Consequently probabilistic linear and non-linear programming will not be covered in this book. Instead this chapter introduces several types of decision-making problems that cannot be solved in a deterministic fashion – problems in civil engineering that must be solved probabilistically. It is perhaps helpful to classify decision-making problems into three types: problems that are deterministic in nature and can be solved deterministically without any need for checks or post-solution analysis, problems on the deterministic/probabilistic borderline that can be solved deterministically but require some sort of sampling or checking or post-solution analysis to verify the validity of the solution, and problems that are fundamentally probabilistic in nature and can only be solved probabilistically. This chapter is concerned only with the last class of problems at the extreme probabilistic end of the spectrum. What types of problem are these?

10.2 PROBABILISTIC DECISION-MAKING PROBLEMS

A feature common to all probabilistic problems is the element of *risk*. The outcome of any policy decision in a deterministic model is completely predictable and, therefore, has no risk associated with it. For example, investing £100 in a bank deposit account bearing 10 per cent interest guarantees a return of £110

www.EngineeringBooksPDF.com

after one year. The outcome or return is completely predictable. In a probabilistic model the outcome of any policy decision is not predictable and may have a range of values. Investing £100 in shares on the stock market does not guarantee anything after one year. The return depends on the economic performance of the companies whose shares were bought. If the investor's judgement of the market is good he may obtain a return much greater than £110 but there is also a risk that he may obtain a low return or even lose all the investment. If the investment is made in the shares of traditionally 'safe' companies the risk may be small and the returns may only be marginally better than those offered by a bank deposit. If, on the other hand, the £100 is invested in the shares of a new oil-prospecting company the return may be very large if that company is successful in its prospecting and locates a new oil field. There is also a very large risk, however, that no new oil will be located and the investor might lose all his £100 investment.

The risk element is, therefore, always present in probabilistic decision-making. Characteristically, profits tend to be small when the risk is low. High profits are usually accompanied by high risk. In the civil engineering industry an example of the effects of risk is afforded by the contract bidding or tendering process. A contractor wishes to bid for the construction work on a new project. He has two risk factors to contend with – the risk that his bid will not secure the contract and the risk that, having secured it, he may not be able to complete the work for the bid price. If he puts in a very low bid for the work he obviously has a high chance of being awarded the contract but because his price is low his profit margin may be small and he may also have a high chance of making a loss on the work. Conversely, if he puts in a very high bid so as to minimise any likelihood of making a loss on the work his chance of actually winning the contract will be reduced. Ideally the contractor would like to have some logical means of selecting a bid price which will give him a good chance of both obtaining the contract and making a profit from it. Clearly in this problem nothing is certain; characteristic deterministic values have no meaning or value here. If a decision-making model is to be used it must be one that is based on the probabilistic nature of the quantities involved. Furthermore, assuming that such models and methods for solving this problem exist and assuming that they permit the calculation of a 'best' bid price that gives the contractor a good chance of winning the contract and making profit from it, there are still no guarantees that if he makes the suggested bid he will win the contract and profit from it. The nature of chance and risk is such that uncertainty is always present.

Another class of problems in which deterministic models and methods play no part are those concerned with reliability. It was mentioned earlier in the context of concrete ultimate strengths that a characteristic (95 percentile) value is often used for design purposes to enable simple deterministic design methods to be used. The use of a characteristic value implies that out of all the concrete to be used, up to 5 per cent may be understrength. Since it is not possible to predict exactly where the understrength concrete will be, there exists the possibility that

www.EngineeringBooksPDF.com

it might be distributed within the structure in such a way that the structure might fail or collapse under working loads. Of course the 95 percentile value is such that this failure possibility is very low. For most structures it is so low as to be negligible as a factor in design calculations. For some structures, however, the risk of failure cannot be ignored – a concrete pressure vessel for a nuclear reactor is a good example. The consequences of the failure of such a vessel are so potentially enormous that the risk of failure must be carefully calculated and designed for. Methods for doing this are probabilistically based. Once again it is worth noting that because probabilistic quantities are involved, the probability of failure of *all* artefacts always exists. The chance element does not permit a zero probability of failure. A structure designed to last for a thousand years using the most sophisticated probabilistic methods may collapse after one day because of the chance element associated with all probabilistic quantities.

Within this chapter it is only possible to introduce some ideas about probabilistic decision-making methods. It is assumed that the reader is already familiar with some aspects of probability. The next section provides a brief refresher in this area and defines some of the terms used in the rest of the chapter.

10.3 RANDOM VARIABLES AND THEIR PROPERTIES

A *random variable* describes the possible outcomes of a chance process and it may have discrete or continuous values. The numbers of people attending a football match might be described by a discrete-valued random variable; an anemometer would record wind speed as a continuous random variable. The complete set of all possible values of a random variable is called the *sample space*. A discrete random variable usually has a finite sample space whereas a continuous random variable usually has an infinite sample space. These two types of random variable are considered separately because, although they share much in common, their mathematical treatments are very different.

10.3.1 Discrete Random Variables

Consider a die with six faces numbered from 1 to 6. The face-up value can be described by a discrete random variable with a sample space of six, that is, when the die is rolled one of six values will show face-up when the die stops rolling. Associated with each of the six possible values is a measure of the likelihood of occurrence of that value. Rolling the die and noting the face-up value constitutes an *event*. If the die is a fair (unbiased) one then the probabilities of each possible value emerging face-up after a single event are the same. No one value is more likely to show up than any other. Furthermore if the die is rolled very many times and the face-up value is recorded it is to be expected that each of the values 1 to 6 will be recorded approximately the same number of times. If we choose one particular value, say 5, and let n_5 be the number of times 5 shows

www.EngineeringBooksPDF.com

up out of a total of N rolls of the die, we can define the *probability of the event five* as

$$p(5) = \lim_{N \to \infty} \left(\frac{n_5}{N} \right) \tag{10.1}$$

The term $p(5)$ is read as 'the probability that the event has the value 5'. For a fair die it is to be expected that $p(5) = 1/6$ since at each roll any one of the six values is likely to show up. Another way of interpreting this probability is as a *frequency*. The event 5 should occur with a frequency of once every six rolls, averaged over a large number of rolls. Generalising the definition 10.1 to a general event with a value E gives

$$p(E) = \lim_{N \to \infty} \left(\frac{n_E}{N} \right) \tag{10.2}$$

that is, the probability of the event having the value E is equal to the frequency with which E occurs over an infinite number of events.

Clearly for the fair six-valued die $p(1) = p(2) = p(3) = p(4) = p(5) = p(6) = 1/6$. From the definition 10.2 it is obvious that for a discrete-valued quantity $p(E)$ must lie between zero and unity and furthermore that the sum of the probabilities of all events in the sample space must be unity. Thus for a total sample space of T we can define

$$0 \leqslant p(t) \leqslant 1; \sum_{t}^{T} p(t) = 1 \tag{10.3}$$

An event with a probability of zero is an impossible event. If its probability is one it is a certain event. In frequency terms $p(t) = 0$ implies that event t never occurs, $p(t) = 1$ implies that t always occurs. If the six-face die had the value 4 on each face then $p(4) = 1$ but $p(5) = 0$.

The die example provided a useful means of introducing a formal definition of the probability of an event (10.2). This definition gives a means of calculating an event probability from the results of previous experiments. However, to obtain the result that $p(5) = 1/6$ for the die it was not actually necessary to perform all the previous rolls; the result could be deduced by examining the circumstances of the experiment. Very often, probabilities are estimated in this way, rather than by actually conducting all the experiments. Indeed, in many circumstances it is just not possible to perform any experiments. For example, consider the problem of forecasting the result of a football game between two teams A and B. Three outcomes are possible: team A wins, team A loses, the result is a tie. Each of these three outcomes has a probability and these probabilities must satisfy conditions 10.3. They cannot be estimated from equation 10.2, however, unless teams A and B have recently played a very large number of games with each other, all with identical teams in the same location under the same conditions. Equation 10.2 is, therefore, useful as a definition, but is not of great

www.EngineeringBooksPDF.com

value in estimating probabilities. For the football game no rigorous, analytical calculations of probabilities are possible. The probabilities must be deduced rather than calculated and furthermore, for the football game, the deduced probabilities will be subjective rather than objective in nature. The person estimating probabilities of the three outcomes must weight carefully all the information he has on the teams, their strengths and weaknesses, their recent performances against other teams, and as a result of this mental deliberation he will assign probabilities. Suppose he decides that in his opinion a tie is most likely, a win for team A less likely than a tie, and a win for team B less likely than a win for team A. Then the probabilities he might assign might be: team A wins – 0.3, team A loses – 0.2, a tie – 0.5. These probabilities owe nothing in their calculation to the definition 10.2 but they can still be interpreted in the spirit of definition by *imagining* a series of identical games between A and B. Out of every ten games in the series, five would be tied, three would be won by team A and two by team B. In this chapter many of the numerical values assigned to probabilities are subjective estimates rather than objective values derived from infinitely repeatable experiments, yet the definition 10.2 underlies even the subjective estimates.

An important aspect of probability is that concerned with the probabilities of occurrence of combinations of events. This is an essential feature of reliability as will be shown later. For the present two definitions will be presented. The first concerns mutually exclusive events. Events are called *mutually exclusive* if one and only one of them can occur. In the die example all the events are mutually exclusive if only one die is rolled. The occurrence of either of two mutually exclusive events E_1 and E_2 is defined as follows

$$p(E_1 \text{ or } E_2) = p(E_1) + p(E_2) \tag{10.4}$$

For the die, the probability of obtaining either a 4 or a 5 is, by 10.4, simply $p(4) + p(5) = 1/6 + 1/6 = 1/3$. Definition 10.4 extends to more than two alternatives in an additive fashion, that is

$$p(E_1 \text{ or } E_2 \text{ or } \ldots \text{ or } E_k) = p(E_1) + p(E_2) + \ldots + p(E_k) \tag{10.5}$$

Another valuable definition is that of *independent events*. Two events are independent (or statistically independent) if the occurrence of one in no way affects the occurrence of the other. The probability of occurrence of two independent events E_1 and E_2 is defined as their *joint probability*

$$p(E_1E_2) = p(E_1) \times p(E_2) \tag{10.6}$$

which is extendable to more than two independent events by relationship 10.7

$$p(E_1E_2 \ldots E_k) = p(E_1) \times p(E_2) \times \ldots \times p(E_k) \tag{10.7}$$

Using the die to provide an example, consider the probability of obtaining a

www.EngineeringBooksPDF.com

sequence of values $4 \rightarrow 5 \rightarrow 6$ in three consecutive rolls of a single die. By definition 10.7, this probability is

$$p(456) = p(4) \times p(5) \times p(6) = (1/6) \times (1/6) \times (1/6) = (1/216)$$

Each roll of the die is independent of all subsequent or previous rolls so the events 4, 5 and 6 are independent. Firstly it is necessary to roll a 4 and the probability of this is 1/6. Then, in the next roll a 5 must be rolled and the probability of this happening is also 1/6 so the joint probability of the sequence 4, 5 is the product of these probabilities, that is, 1/36. Only once in every 36 rolls can we expect to get the sequence 4, 5 and when it is obtained there is only a one in six chance that it will be immediately followed by rolling a 6. Thus $p(456) = 1/216$.

10.3.2 Continuous Random Variables

In the preceding section the random variable was considered to take only discrete values. When the random variable is continuous it may have infinitely many values. Definition 10.2 for the probability of an event no longer holds or rather it gives a probability of zero for the achievement of any specific value. For continuous variables we can only give meaning to the probability of achieving a value which lies within a specified range of values in the infinite sample space. Figure 10.1 shows the difference between discrete and continuous probability functions.

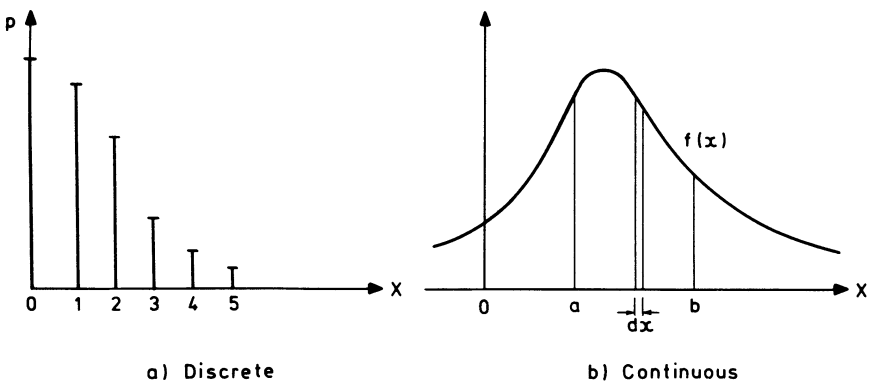


Figure 10.1 Discrete and continuous probability functions

Let X be a continuous random variable and x the range of values it can have (often $-\infty \leqslant x \leqslant +\infty$). Define $f(x)$ to be the *probability density* of variable X. $f(x)$ can be thought of as a 'probability per unit length of x' rather than as a

www.EngineeringBooksPDF.com

probability. Then the probability that the variable X will have a value between a and b (see figure 10.1b) is defined to be

$$p(a \leqslant X \leqslant b) = \int_a^b f(x)\mathrm{d}x \tag{10.8}$$

If variable X represents, for example, the total cost of a project, relationship 10.8 defines the probability that the cost will be between £a and £b. Note that the probability increases as the gap between a and b increases and that as a approaches b the probability approaches zero. This is rather like finding a needle in a haystack. That the needle is in the haystack is certain but, by searching a very small volume of the haystack only, one is very unlikely to find the needle. Corresponding to relationship 10.3 for discrete random variables are the following properties for continuous random variables

$$f(x) \geqslant 0; \quad \int_{-\infty}^{\infty} f(x)\mathrm{d}x = 1 \tag{10.9}$$

10.3.3 Distribution Functions for Random Variables

The two preceding sections have defined probability functions for discrete and continuous random variables. Often it is necessary to determine not the probability of an event or the probability that the variable lies within known limits but the probability that the variable has a value less than or equal to X, or that the event has a value less than or equal to one prescribed value. The *distribution function* defines these probabilities. Figure 10.2 shows distribution functions corresponding to the probability functions of figure 10.1. For a continuous

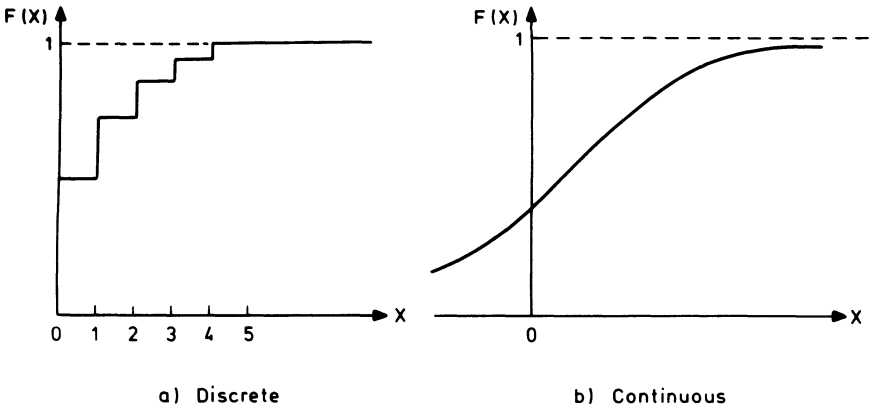


Figure 10.2 Discrete and continuous distribution functions

www.EngineeringBooksPDF.com

random variable the distribution function $F(x)$ is defined as the probability that x has a value less than or equal to x, that is

$$F(x) = p(X \leqslant x) = \int_{-\infty}^{x} f(x)\mathrm{d}x \tag{10.10}$$

Note that 10.10 is obtained from 10.8 simply by inserting $a = -\infty$ and $b = x$ as the limits. Figure 10.2b shows $F(x)$.

For a discrete random variable the distribution function is shown in figure 10.2a and the probability that an event will have a value less than or equal to k is

$$F(k) = p(E \leqslant k) = \sum_{t}^{k} p(t) \tag{10.11}$$

Thus for the die $F(5)$ is the probability that the number rolled will be less than or equal to 5 and is

$$F(5) = p(1) + p(2) + p(3) + p(4) + p(5) = 5/6$$

10.3.4 Properties of Random Variables

The probability and distribution functions of a random variable contain all the information needed to perform calculations. It is often much easier, however, to use certain gross properties of the functions rather than the functions themselves. One of the most useful properties in this context is the *mean value* of the function which is often called the *expected value* of the variable.

For a discrete variable that can have a total of T discrete values, the mean or expected value is defined as follows. Suppose that out of a total of N events, n_1 of the outcomes have the value x_1, n_2 have the value x_2, etc., up to n_T having the value x_T. Then the mean value of the variable, $\bar{X}$, equal to the expected value of X which can be written as $E(X)$, is

$$\bar{X} = E(X) = \frac{\sum_{t=1}^{T} x_t n_t}{N} = \sum_{t=1}^{T} x_t \left(\frac{n_t}{N}\right) \tag{10.12}$$

From 10.2, however, the ratio n_t/N is defined as $p(t)$ for a large number of events N. Thus

$$\bar{X} = E(X) = \mu_x = \sum_{t}^{T} x_t p(t) \tag{10.13}$$

www.EngineeringBooksPDF.com

In 10.13 μ_x is an alternative symbol often used to represent a mean value. Using 10.13 on the six-face die gives the mean or expected value as

$$\mu_x = E(X) = 1\left(\frac{1}{6}\right) + 2\left(\frac{1}{6}\right) + 3\left(\frac{1}{6}\right) + 4\left(\frac{1}{6}\right) + 5\left(\frac{1}{6}\right) + 6\left(\frac{1}{6}\right)$$

$$= 21/6 = 3.5$$

For a continuous random variable the mean is defined as

$$\bar{X} = E(X) = \mu_x = \int_{-\infty}^{\infty} xf(x)\mathrm{d}x \tag{10.14}$$

The mean of a function locates the position of the function average on the x-axis. Another important property of the probability function is concerned with describing whether the function is closely packed around the mean as in a very spiky function or whether it resembles a flatter rolling hill-shaped function. A measure of how close the function is to the mean is given by the *standard deviation* of the function. The standard deviation σ_x is defined via the *variance*, Var(x) of the function as

$$\sigma_x = \sqrt{[\mathrm{Var}(x)]} \tag{10.15}$$

where, for a discrete random variable

$$\begin{aligned} \mathrm{Var}(x) &= E([x - \mu_x]^2) \\ &= E(x^2 - 2\mu_x x + \mu_x{}^2) \\ &= E(x^2) - 2\mu_x E(x) + E(\mu_x{}^2) \\ \mathrm{Var}(x) &= E(x^2) - \mu_x{}^2 \end{aligned} \tag{10.16}$$

Thus for the die

$$\mathrm{Var}(x) = \frac{1}{6}(1^2 + 2^2 + 3^2 + 4^2 + 5^2 + 6^2) - 3.5^2$$

$$= 2.9167$$

and the standard deviation is

$$\sigma_x = \sqrt{[\mathrm{Var}(x)]} = 1.7078$$

For a continuous random variable the standard deviation is defined by 10.15 in which the variance is expressed continuously as

$$\mathrm{Var}(X) = \int_{-\infty}^{\infty} (x - \mu_x)^2 f(x)\mathrm{d}x \tag{10.17}$$

As mentioned above the standard deviation measures the spikiness or flatness of the probability function about its mean. The spikier the function, the lower the values of the standard deviation and variance.

www.EngineeringBooksPDF.com

10.3.5 Common Probability Functions

To elucidate aspects of a discrete random variable the example of rolling a fair die has been used several times. In describing the die as a fair one it was implied that the probabilities of outcomes resulting in each of the six values were all the same. This is known as a *discrete uniform distribution*. Intuition and perhaps experience of rolling die suggested that this form of probability function was a very reasonable assumption to make. Had the die not been a fair one then the uniform distribution would not have been appropriate. It would have been necessary either to determine probabilities experimentally using 10.2 or to use some other appropriate mathematical description of the probabilities.

The discrete uniform distribution is just one of very many probability functions each of which is an appropriate descriptor of different types of random variable encountered in the real world. For continuous variables, in order to calculate probabilities and distributions using 10.8 or 10.10, it is necessary to know the mathematical form of the function $f(x)$ that describes the variable. The integrals cannot be evaluated until $f(x)$ is specified. Fortunately, many possible functional forms for $f(x)$ have been studied and, although the integrals necessary for them are often difficult to perform analytically, in many cases they can be evaluated using tabulated values obtainable from statistical tables. It is usual, therefore, to examine carefully the random variable of interest and see whether it can be accurately approximated by any of the available mathematical probability functions. If it can, then it is usually much easier to perform any necessary calculations using the mathematical function rather than the actual variable.

10.3.5.1 Normal or Gaussian Functions

The *normal* or *Gaussian* function is possibly the best-known of all probability functions. Its symmetrical bell-shape is shown in figure 10.3a. The equation of the normal curve is

$$f(x) = \frac{1}{\sigma_x\sqrt{(2\pi)}}\, e^{-[(x-\mu_x)^2/2\sigma_x{}^2]} \quad \text{for } -\infty < x < \infty \qquad (10.18)$$

In 10.18 σ_x and μ_x are the standard deviation and the mean defined earlier. The normal function fits the observed behaviour of very many practical engineering quantities such as steel strength, wind loads, etc., and it is often used to represent variables which are observed to be almost normal in their behaviour.

One reason for the popularity and usefulness of the normal function is that calculations using it can easily be done by using tabulated values. The normal function can be transformed mathematically to the *standard normal* or *unit normal* function which has a mean $\mu = 0$ and a standard deviation $\sigma = 1$. Figure 10.3b shows the standard normal function.

www.EngineeringBooksPDF.com

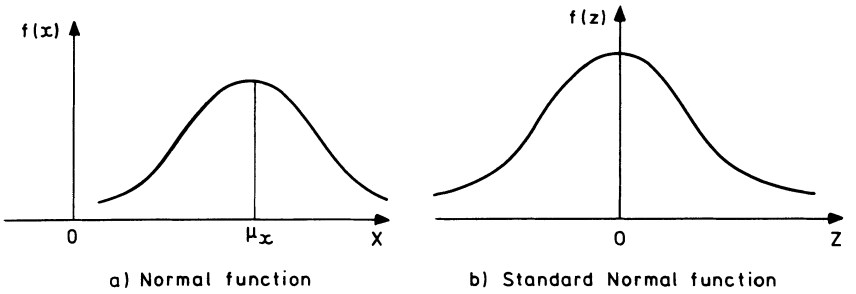


Figure 10.3 Normal and standard normal functions

As an example of how calculations are performed, suppose that random variable X is approximated by the normal function, 10.18, with a mean $\mu_x = 10.0$ and standard deviation $\sigma_x = 2.0$ and that it is necessary to calculate the probability that X lies between values $a = 8.0$ and $b = 9.0$. From 10.8

$$p(a \leqslant x \leqslant b) = \int_a^b f(x)\mathrm{d}x = \frac{1}{\sigma_x\sqrt{(2\pi)}} \int_a^b \mathrm{e}^{-[(x-\mu_x)^2/2\sigma_x^2]}\,\mathrm{d}x$$

Substituting $z = (x - \mu_x)/\sigma_x$; $\mathrm{d}z/\mathrm{d}x = 1/\sigma_x$ gives

$$p(a \leqslant x \leqslant b) = \frac{1}{\sqrt{(2\pi)}} \int_{(a-\mu_x)/\sigma_x}^{(b-\mu_x)/\sigma_x} \mathrm{e}^{-z^2/2}\,\mathrm{d}z \qquad (10.19)$$

The right-hand side of 10.19 is exactly the same as would have been obtained by evaluating $p([a - \mu_x]/\sigma_x \leqslant z \leqslant [b - \mu_x]/\sigma_x)$ for a standard normal variable z with $\sigma_z = 1$ and $\mu_z = 0$. Table 10.1 lists properties of the standard normal function. For each value of z a value is listed for $f(z)$ and $F(z)$. As a result of the symmetry of z about the zero mean $f(z) = f(-z)$ and $F(-z) = 1 - F(z)$. Then for the numerical values stated above, 10.19 yields

$$\begin{aligned} p(8.0 \leqslant x \leqslant 9.0) &= \frac{1}{\sqrt{(2\pi)}} \int_{-1}^{-1/2} \mathrm{e}^{-z^2/2}\,\mathrm{d}z \\ &= \frac{1}{\sqrt{(2\pi)}} \left\{ \int_{-\infty}^{-1/2} \mathrm{e}^{-z^2/2}\mathrm{d}z - \int_{-\infty}^{-1} \mathrm{e}^{-z^2/2}\,\mathrm{d}z \right\} \\ &= F\left(-\frac{1}{2}\right) - F(-1) \\ &= F(1) - F\left(\frac{1}{2}\right) \\ &= 0.841345 - 0.691463 \end{aligned}$$

therefore

$$p(8.0 \leqslant x \leqslant 9.0) = 0.149882$$

www.EngineeringBooksPDF.com

Table 10.1 The standard normal function

z	$f(z)$	$F(z)$	z	$f(z)$	$F(z)$
0.0	0.398942	0.500000	1.8	0.078950	0.964070
0.1	0.396952	0.539828	1.9	0.065616	0.971284
0.2	0.391043	0.579260	2.0	0.053991	0.977250
0.3	0.381388	0.617912	2.1	0.043984	0.982136
0.4	0.368270	0.655422	2.2	0.035475	0.986097
0.5	0.352065	0.691463	2.3	0.028327	0.989276
0.6	0.333225	0.725747	2.4	0.022395	0.991802
0.7	0.312254	0.758036	2.5	0.017528	0.993790
0.8	0.289692	0.788145	2.6	0.013583	0.995339
0.9	0.266085	0.815940	2.7	0.010421	0.996533
1.0	0.241971	0.841345	2.8	0.007915	0.997445
1.1	0.217852	0.864334	2.9	0.005952	0.998134
1.2	0.194186	0.884930	3.0	0.004432	0.998650
1.3	0.171369	0.903199	3.5	0.000873	0.999767
1.4	0.149727	0.919243	4.0	0.000134	0.999968
1.5	0.129518	0.933193	4.5	0.000016	0.999996
1.6	0.110921	0.945201	5.0	0.000002	0.999999
1.7	0.094049	0.955435			

Thus simple transformations permit a normal function to be transformed to a standard normal function for which tabulated data makes calculations easy.

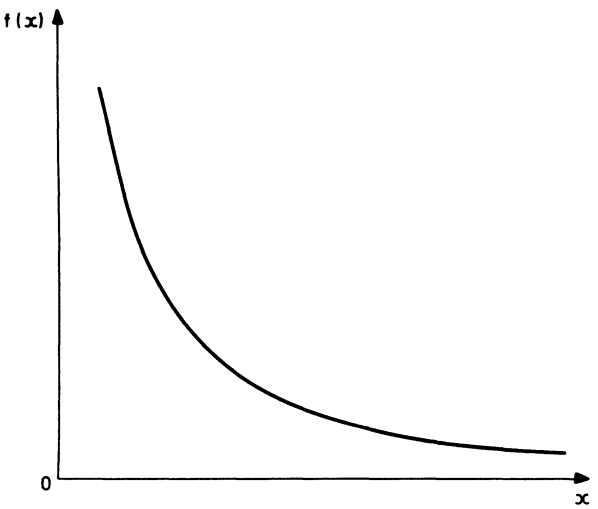


Figure 10.4 The exponential function, $f(x) = \lambda e^{-\lambda x}$

www.EngineeringBooksPDF.com

10.3.5.2 Exponential Function

The exponential function is defined by

$$f(x) = \lambda\, e^{-\lambda x} \qquad \text{for } x > 0 \qquad (10.20)$$

where λ is a positive constant. Figure 10.4 shows the typical shape of the exponential function which, like the normal function, can be used to represent many engineering quantities. In general it represents situations in which probabilities of occurrence of small values of x are high but of high value are low. An example is that of river pollution (see exercise **10.2**).

There are many other probability functions which space does not allow to be described here. It is now necessary to move on from this review of definitions in the probability area and to examine how some of these ideas can be used in solving probabilistic decision-making problems.

10.4 THE USE OF EXPECTED VALUES FOR DECISION-MAKING

Definition 10.13 of the expected value or mean value of a discrete random variable can be viewed as a weighted mean.

$$E(x) = \sum_{t}^{T} x_t p(t) \qquad (10.13)$$

Each of the possible outcome values x_t in the sample space of T values is weighted by multiplying it by the probability of occurrence of that value. The expected value is then the sum of all the probability-weighted values over the sample space. A very useful decision-making procedure can be developed by calculating the expected values of different policies and selecting the policy with the optimal expected value. This procedure is known as *Bayes' principle* which states that the decision-maker should select that action or policy which corresponds to the minimum expected loss. The following example shows how this works.

Example 10.4.1 – Contract Tendering

A contractor wishes to tender for a construction project. The preparation of a bid will cost him an estimated £10 000. A continuous range of possible bids is open to him but he has selected five discrete bids for further study – a high one, a low one and three others between these extremes. On the high bid of £1.5m he estimates he would make £750 000 profit but the probability of such a high bid winning the contract is, he estimates, only 0.02. The low bid of £0.8m has a much higher chance of winning the contract but would yield only £80 000 profit. Details of all the five bids are given in table 10.2. What bid should the contractor make?

www.EngineeringBooksPDF.com

Table 10.2 The contractor's possible bids

Bid (£m)	1.5	1.2	1.0	0.9	0.8
Cost (£1000)	10	10	10	10	10
Profit (£1000)	750	450	255	165	80
Probability of success	0.02	0.08	0.15	0.25	0.5

Each of the five bids has two possible outcomes: a successful bid will result in a profit to the contractor, an unsuccessful bid will result in a loss. For each bid the probability of success is known and so the probability of not winning the contract will be unity minus the probability of success. Note that the estimates of probability of success are subjectively rather than analytically based. For each bid the expected value of loss is calculated. For the £1.5m bid the gross profit is £750 000 from which the £10 000 bid cost must be deducted leaving a net loss of -£740 000 (that is, a net profit of £740 000). The probability of making this net loss is 0.02. If the contract is not won, the net loss will be £10 000 and the probability of making this loss is (1 - 0.02) = 0.98. Thus the expected value of the loss to the contractor of the £1.5m bid is given by 10.13 as

$$E(\text{£}1.5\text{m}) = \text{£}[(-740\,000 \times 0.02) + (10\,000 \times 0.98)]$$
$$= -\text{£}5000$$

Thus the expected loss on this bid is -£5000, that is, an expected profit. Note that this value could have been obtained from the expected value of the gross loss (-£750 000 x 0.02) plus the bid cost of £10 000 since the bid cost is incurred whether or not the bid is successful.

Performing a similar calculation for the £1.2m bid gives an expected loss of

$$E(1.2\text{m}) = \text{£}[(-450\,000 \times 0.08) + 10\,000] = -\text{£}26\,000$$

For the £1.0m bid

$$E(\text{£}1.0\text{m}) = \text{£}[(-255\,000 \times 0.15) + 10\,000] = -\text{£}28\,250$$

and for the other bids

$$E(\text{£}0.9\text{m}) = \text{£}[(-165\,000 \times 0.25) + 10\,000] = -\text{£}31\,250$$
$$E(\text{£}0.8\text{m}) = \text{£}[(-\ 80\,000 \times 0.5\ ) + 10\,000] = -\text{£}30\,000$$

Bayes' principle requires that the action which results in the minimum expected loss should be taken. For the five possible bids the minimum expected loss is -£31 250 for the £0.9m bid. This suggests that on the basis of the data in table 10.2 the contractor should enter a bid of £0.9m as his 'best bet'. The actual expected value of £31 250 profit has no monetary significance and does not mean that he can expect to make this profit. The chance element still exists; the bid may be successful in which case the contractor should make a gross profit of £165 000 or the bid may fail in which case he will lose £10 000. The value of

www.EngineeringBooksPDF.com

-£31 250 loss is simply the lowest on an artificial value scale used to rank the possible bids. If any of the figures in table 10.2 are altered then the resulting calculations would change. The contractor should, therefore, examine all the data carefully and ensure that it is as representative and accurate as it can be.

Example 10.4.2

This example is an extension of the previous contract tendering example. Suppose that the site for the construction work is in a geological area in which sub-surface water can severely hinder construction operations. If the contractor is lucky the particular site chosen might be dry right down to below foundation level. If he is less lucky he might encounter water during the excavation and consequently might have to spend money on pumping and dewatering equipment thus reducing his profit. If he is very unlucky he might encounter severe flooding of the excavations which would require very expensive remedial drainage of the site and protection for the foundations. This would severely reduce his profit.

The contractor now estimates that the profits shown before in table 10.2 represent the profits he would make only if the site is dry. If the site requires dewatering, his profits as tabulated will be reduced by £50 000 and, if flooding is bad enough to require drainage and foundation protection, the tabulated profits would be reduced by £150 000. Suppose that he estimates probabilities for the three site conditions as follows

site dry,	probability 0.1
site moderately wet,	probability 0.6
site very wet,	probability 0.3

What should be his bid now?

In this extended example there are now four discrete outcomes possible for each bid

(1) contract bid unsuccessful
(2) contract bid successful, site dry
(3) contract bid successful, site moderately wet
(4) contract bid successful, site very wet.

Assuming that the probabilities of winning the contract remain the same as detailed in table 10.2, the weights or probabilities which should be associated with each of these four outcomes are as follows. Consider the £1.5m bid. Outcome (1), an unsuccessful bid, has a 0.98 probability as before and would incur a loss of £10 000. Outcome (2), a successful bid and a dry site has a probability of $0.02 \times 0.1 = 0.002$. The reason for this is that the success of the bid and the dryness of the site are independent of each other and so by virtue of 10.6 the joint probability of them both occurring simultaneously is equal to the product of their independent probabilities. The net loss from outcome (2) would be

www.EngineeringBooksPDF.com

-£740 000. Similarly the probability of outcome (3) is 0.02 x 0.6 = 0.012 and the net loss would be -£690 000. For outcome (4) the probability is 0.02 x 0.3 = 0.006 and the net loss would be -£590 000. Combining these together to give an expected value of loss for the £1.5m bid yields

$$E(£1.5\text{m}) = £[(10\,000 \times 0.98) - (740\,000 \times 0.002) - (690\,000 \times 0.012) - (590\,000 \times 0.006)]$$
$$= -£3500$$

For the £1.2m bid a similar calculation is performed giving

$$E(£1.2\text{m}) = £[(10\,000 \times 0.92) - (440\,000 \times 0.08 \times 0.1) - (390\,000 \times 0.08 \times 0.6) - (290\,000 \times 0.08 \times 0.3)]$$
$$= -£20\,000$$

For the other three bids the following expected losses are obtained

$$E(£1.0\text{m}) = -£17\,000; E(£0.9\text{m}) = -£12\,500; E(£0.8\text{m}) = £7500$$

Comparing these five expected losses it is seen that the minimum loss is afforded by the £1.2m bid which shows an expected net profit of £20 000. Bayes' principle suggests that £1.2m should be the contractor's bid. The low bid of £0.8m shows an expected loss of £7500 and is clearly too low to offer the contractor any good chance of making profits. Also the high bid of £1.5m shows a low expected profit because of the low probability of winning the contract. All results, however, are based only on the problem data and it would be prudent to check out all that data and probability estimates as deeply as possible. To re-iterate a previous point, although £20 000 is the expected value of the net profit from a £1.2m bid, this money will not actually be made. There are four possible outcomes: a £10 000 loss and three possible profits of £440 000, £390 000 or £290 000. Which of these is realised still depends on chance.

Some comments on examples 10.4.1 and 10.4.2 are appropriate. First of all the three probabilities which refer to the site condition are mutually exclusive as defined in section 10.3.1. Only one of them will occur. Also the three site conditions are *collectively exhaustive* in that they describe all possible states of the site which must be either dry, moderately wet or very wet. Since the three events are collectively exhaustive their probabilities *must* sum to unity and of course they do since 0.1 + 0.6 + 0.3 = 1.

The probabilities for each bid of winning or losing the contract are also mutually exclusive and collectively exhaustive. Thus it was possible to write the probability of losing the contract as unity minus the probability of winning the contract. Considering the probabilities of each bid succeeding, it is clear that these probabilities are mutually exclusive in that only one of the five bids can be made but they are not collectively exhaustive. There remains the possibility that some other contractor may win the contract with a bid that does not correspond to one of the five alternatives examined here. The important feature of these probabilities is that they must accurately represent the *relative* chances of

www.EngineeringBooksPDF.com

success of each of the bids, that is, the chance of success with the £0.8m bid is twice that with the £0.9m bid and twenty-five times that of the £1.5m bid. Also the relative chances of each bid succeeding or failing must be correct.

10.4.3 Decision Trees

Example 10.4.2 is more complicated than example 10.4.1 and it is comparatively harder in example 10.4.2 to keep track of the joint probabilities. As the complexity of the problem increases still further, this simple expected value method becomes more difficult to handle. Decision trees provide a means of tracking all the alternatives in a larger problem. An example shows how decision trees are constructed and used.

Example 10.4.3.1

This example is a further extension of the contract tendering example 10.4.2. A résumé of the data for that problem is given in table 10.3.

Suppose that the contractor is still worried about the possibility of encountering wet site conditions. One option open to him is to commission a private site geological survey before he makes his bid. Such a survey would cost him £5000 and would forecast the state of the site as dry (D), moderately wet (MW) or very wet (VW). With this forecast the contractor would feel safer about making a bid. Of course the survey forecast will not be 100 per cent accurate. Probabilities that

Table 10.3 Contractor's bidding data

Bid (£m)	1.5	1.2	1.0	0.9	0.8
Cost (£1000)	10	10	10	10	10
Probability of success	0.02	0.08	0.15	0.25	0.5
Profit, D, $p = 0.1$	750	450	255	165	80
Profit, MW, $p = 0.6$	700	400	205	115	30
Profit, VW, $p = 0.3$	600	300	105	15	-70

Table 10.4 Probabilities of correspondence between forecast and actual site conditions

		Actual site condition		
		D	MW	VW
Forecast site condition	D	0.8	0.15	0.05
	MW	0.15	0.7	0.15
	VW	0.05	0.15	0.8

www.EngineeringBooksPDF.com

the actual site conditions correspond to the forecast conditions are given in table 10.4. Should the contractor have a private survey done or not? What should his bid be?

A decision tree is a graphical means of displaying the many alternative decisions and chance outcomes that exist in this problem. Conventionally, a node in the tree at which a conscious decision must be made is represented by a square node and nodes at which chance alternatives exist are represented by circular nodes. Decision trees are constructed by logically ordering the chance and decision nodes and assigning a branch to each chance outcome or decision alternative.

In this problem the first node in the tree is a decision node and represents the contractor's very first step in the logical decision-making process. The first node represents the decision whether or not to commission the private site survey. One branch emanating from the first node represents the decision to use the survey, the other branch represents the decision not to use the survey. If the survey is not used then everything corresponds to example 10.4.2 which has already been solved. This solution will be repeated here in decision tree form for explanation purposes before examining the other branch of the tree.

Having decided not to use the survey, the contractor must next decide upon his bid. This decision is represented by a square decision node at the end of the 'no survey' branch and five branches lead from this decision node corresponding to the five possible bids the contractor can make. The next event in the logical sequence is that the bid made is either successful or unsuccessful in winning the contract. This is a chance outcome and so each of the five bid branches ends in a circular chance node. Each of these chance nodes has two branches emerging from it corresponding to the two possible outcomes, success (S) or failure (F) of that bid to win the contract. Each of these two branches has the probability of that outcome written beside the branch. If the outcome is failure to win the contract (F) then the branch terminates and the value of the loss incurred as a result of that outcome is written beside the end of the terminating branch. In this example the loss will be £10 000 for submitting a failing bid. If the outcome is success (S) and the contract is won, the next logical event is the chance occurrence of site conditions. Thus each success branch has a terminal chance node and three further branches emanating from that node. These three branches correspond to the three possible site conditions D, MW and VW and beside each branch is written the probability of occurrence of that side condition as stated in table 10.3. Each of these branches then terminates and beside each of the branch ends is written the loss which will be achieved by following that particular branch down from the first node to termination. Thus the loss associated with the branches 'no survey' → '£1.5m bid' → 'S' → 'D' is, from table 10.3, £(-750 000 + 10 000) = -£740 000. Each terminating branch is appropriately marked with the loss incurred by that policy and chance path. Figure 10.5 shows the decision tree constructed thus far.

The other main branch of the decision tree corresponding to the use of the

www.EngineeringBooksPDF.com

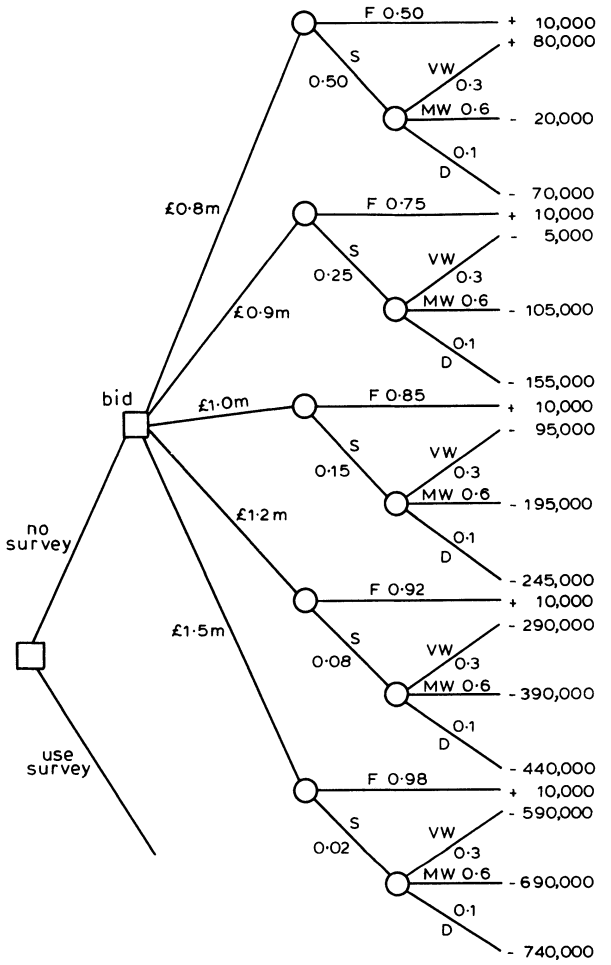


Figure 10.5 Partial decision tree of contract tendering problem

survey remains to be constructed but first the 'no survey' branch will be evaluated to demonstrate how this is done. Firstly the node or nodes nearest the ends of the decision tree are sought. If the node is a decision node the value of the minimum expected loss on any branch out of that node is chosen and is associated with that node. If the node is a chance node the expected value of the loss along all branches emerging from that node is calculated and is associated with the node. In this example the five nodes corresponding to the site condition are all equally close to the end of the tree and are closer than the five success/failure nodes. These five site condition nodes are chosen, therefore, and because they are chance nodes the expected loss value must be calculated over all branches

www.EngineeringBooksPDF.com

leading from each node. This expected loss is given simply by the loss value at the end of each emergent branch multiplied by the probability value on that branch and summed over all emerging branches.

Thus for the top site condition branches on figure 10.5 the expected loss is

$$+80\,000 \times 0.3 - 20\,000 \times 0.6 - 70\,000 \times 0.1 = +5000$$

For the next set of site condition branches the expected loss is

$$-5000 \times 0.3 - 105\,000 \times 0.6 - 155\,000 \times 0.1 = -80\,000$$

For the remaining three sets of site condition branches the expected losses are, reading from top to bottom of figure 10.5

$$-95\,000 \times 0.3 - 195\,000 \times 0.6 - 245\,000 \times 0.1 = -170\,000$$
$$-290\,000 \times 0.3 - 390\,000 \times 0.6 - 440\,000 \times 0.1 = -365\,000$$
$$-590\,000 \times 0.3 - 690\,000 \times 0.6 - 740\,000 \times 0.1 = -665\,000$$

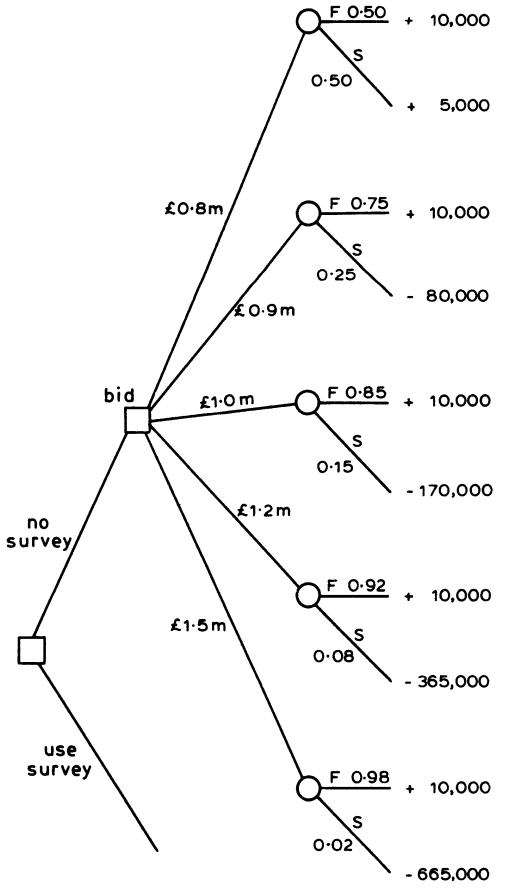


Figure 10.6 Partial evaluation of the decision tree of figure 10.5

www.EngineeringBooksPDF.com

What has been done here is that the chance outcomes of the site conditions have been replaced by the expected loss from those outcomes. The branches have been eliminated. Figure 10.6 shows the decision tree of figure 10.5 after these calculations have been performed. This process is often called *rolling back the decision tree* for obvious reasons.

The next stage consists of further rolling back using the same principle as has just been used. The nearest nodes to the end of the tree are now the contract success/failure nodes and all five are equally close to the end. They are all chance nodes so it is necessary to calculate for each node an expected loss value over all branches emerging from each node. For the topmost node of figure 10.6 this expected loss is simply the value at the end of each branch multiplied by the branch probability summed over all branches, that is

$$+10\,000 \times 0.5 + 5000 \times 0.5 = +7500$$

and for the other nodes in descending order on figure 10.6

$$+10\,000 \times 0.75 \quad -80\,000 \times 0.25 = -12\,500$$
$$+10\,000 \times 0.85 \; -170\,000 \times 0.15 = -17\,000$$
$$+10\,000 \times 0.92 \; -365\,000 \times 0.08 = -20\,000$$
$$+10\,000 \times 0.98 \; -665\,000 \times 0.02 = -3500$$

Thus all the terminal branches on figure 10.6 may be deleted and replaced by these expected loss values at the end of each of the five bid branches.

Continuing the roll-back process, the nearest node to the end of the tree is again sought. This time it is the 'bid' node, a decision node. Since it is a decision node, the roll-back rule requires that the minimum of the expected loss values at the ends of all branches emerging from the decision node should be chosen and associated with that node. In this example the minimum of the expected loss values just calculated is -20 000 and this is the value chosen. Note that all the expected loss values above are the same as those calculated in example 10.4.2 and that the decision tree has forced the same choice of bid as was made in that example, that is, the £1.2m bid. At present it is not possible to roll the tree back any further as the 'use survey' branch has no values associated with it yet. This branch must now be constructed and evaluated.

Figure 10.7 shows the 'use survey' branch of the decision tree. In fact this branch is much larger than the 'no survey' branch, having sixty terminals instead of the twenty on figure 10.5. For this reason figure 10.7 shows only a part of the branch. The diagram starts with the decision to use the private site survey. The next event is the chance node representing the chance outcomes of the survey, that is, the survey will conclude that the site is either dry (D), moderately wet (MW) or very wet (VW). Branches are shown for each of these chance outcomes. Having received the site survey the contractor must now tender his bid so each of these three branches ends in a 'bid' decision node. For space reasons on the diagram only the bid node on the MW branch is developed further but similar branch structures emanate from each of the other two bid nodes. Each of the

www.EngineeringBooksPDF.com

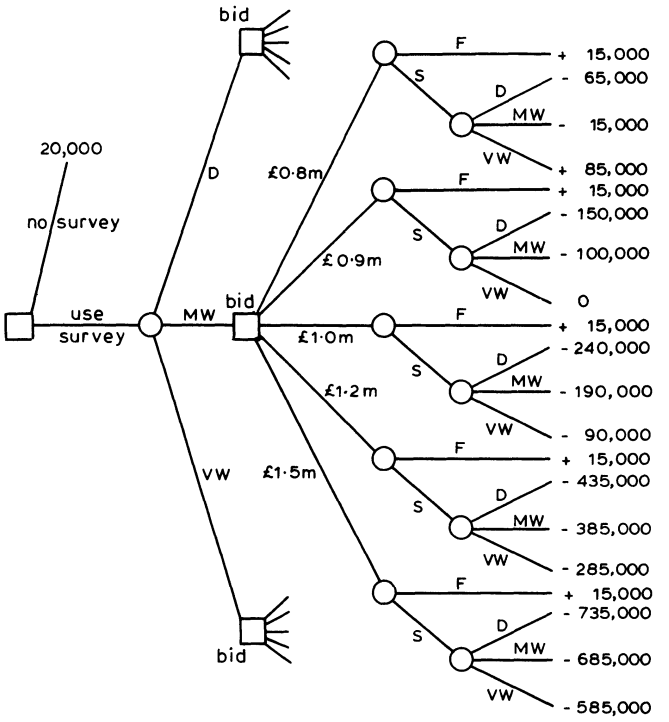


Figure 10.7 Schematic diagram of survey branch

five possible bids has its own branch terminating in a chance node representing the chance outcome of the failure (F) or success (S) of that bid in securing the contract. Each of the F branches terminates in a loss value of £15 000 representing the cost of the survey plus the tendering cost. Each of the S branches ends in a chance node representing the chance outcome of the site condition as D, MW or VW. These are necessary because, although the site survey forecast a moderately wet site, the forecast is not 100 per cent certain. There is still a chance that the forecast was wrong. Thus there are three branches from each site condition chance node each terminating in a loss value corresponding to that path through the tree.

Figure 10.7 has no probabilities stated beside the chance outcomes. The reason for this is not just lack of space on the diagram. These probabilities need further thought and consideration before they are calculated.

10.4.3.1.1 Prior and Posterior Probabilities

In assigning the probabilities of 0.1, 0.6 and 0.3 to site conditions D, MW and VW along the 'no survey' branch, the contractor was simply using the best esti-

www.EngineeringBooksPDF.com

mates available to him. These estimates were specified as data in table 10.3. All the probabilities in that table are known as *prior probabilities* because they are available from the start and before any decisions are made.

In deciding to use the survey, however, the contractor has implied that he wants to pay for more accurate estimates of site condition probabilities. These more accurate values must be calculated and added to figure 10.7. Why and how have the site condition probabilities changed?

The reason they have changed is that the contractor now knows more about the site as a result of the private survey than he would have known had he not used the survey. By using the survey he will have a prediction of the site condition *before* he bids instead of the site condition being a mere chance outcome. For example, if the site survey forecasts a dry site he can arrange to make his bid on the assumption of a dry site with, from table 10.4 an 80 per cent chance that the site will be dry. There is only a 5 per cent chance that the site will be very wet after a dry forecast whereas if he had access to no forecast the contractor would have to assume a 30 per cent chance of a very wet site. When a prior probability is modified as a result of extra information becoming available, the resulting changed probabilities are called *posterior probabilities.*

Posterior probabilities can be calculated from the known values of the prior probabilities and the confidence probabilities of the additional information. The derivation of the mathematical relationships among the probabilities is not presented here but can be stated as follows: the posterior probability of outcome X_k (of a total of K possible outcomes) where the forecast experimental outcome is X_j is equal to the prior probability of X_k multiplied by the confidence probability of X_k occurring given a forecast X_j, divided by the sum over K of these products of prior and confidence probabilities.

Table 10.5 Calculation of posterior probabilities

		D	MW	VW		
Prior probabilities		0.1	0.6	0.3		
Forecast	D	0.8	0.15	0.05		Confidence probabilities
	MW	0.15	0.7	0.15		
	VW	0.05	0.15	0.8	Σ	
Forecast	D	0.08	0.09	0.015	0.185	Prior x confidence probabilities
	MW	0.015	0.42	0.045	0.48	
	VW	0.005	0.09	0.24	0.335	
Forecast	D	0.4324	0.4865	0.0811		Posterior probabilities (= products/Σ)
	MW	0.0313	0.8750	0.0937		
	VW	0.0149	0.2687	0.7164		

www.EngineeringBooksPDF.com

The site survey example shows how posterior probabilities are calculated from this definition. Table 10.5 contains the calculations.

Three columns correspond to the three possible site conditions D, MW or VW. The prior probabilities taken from table 10.3 are followed by the confidence probabilities of table 10.4 relating a forecast to an actual site condition. Below these are entered the columnwise products of the prior and confidence probabilities. Each of these three rows is summed to give a Σ value. The final block of the table gives the posterior probabilities and consists of each value in the block above divided by the Σ value appropriate to its row. The Σ values are themselves important because they represent the probabilities of the site condition survey resulting in a dry forecast ($p = 0.185$), a moderately wet forecast ($p = 0.48$) and a very wet forecast ($p = 0.335$).

These newly calculated probabilities can now be entered on the 'use survey' branch of the decision tree. On figure 10.7 only the branch corresponding to a survey resulting in an MW site condition is shown. The value $p = 0.48$ should be written beside this main branch with 0.185 and 0.335 beside the main branches D and VW, respectively. Exploring further down the MW branch there are five nodes at which actual site conditions are represented by branches D, MW and VW. Against each of these branches should be written the posterior probabilities from table 10.5 of the site condition being D, MW or VW for a forecast of MW, that is, for D, $p = 0.0313$, for MW, $p = 0.875$ and for VW, $p = 0.0937$.

The decision tree is now complete and the roll-back process can begin. Rolling back first of all to the actual site condition chance nodes requires the calculation of the expected loss value for the three chance outcomes at each node. For the topmost branching on figure 10.7, corresponding to the £0.8m bid, the expected loss is

$$-(65\,000 \times 0.0313) - (15\,000 \times 0.875) + (85\,000 \times 0.0937)$$
$$= -£7195$$

and for the other four site condition nodes in figure 10.7 in ascending order of bid

$$-(150\,000 \times 0.0313) - (100\,000 \times 0.875) + 0$$
$$= -£92\,195$$

$$-(240\,000 \times 0.0313) - (190\,000 \times 0.875) - (90\,000 \times 0.0937)$$
$$= -£182\,195$$

$$-(435\,000 \times 0.0313) - (385\,000 \times 0.875) - (285\,000 \times 0.0937)$$
$$= -£377\,195$$

$$-(735\,000 \times 0.0313) - (685\,000 \times 0.875) - (585\,000 \times 0.0937)$$
$$= -£677\,195$$

The next set of nodes on figure 10.7 in the roll-back process are the contract success/failure chance nodes. The probabilities associated with outcomes F or S

www.EngineeringBooksPDF.com

are exactly as detailed in table 10.3 for each bid with the probability of F being unity minus the probability of S. Calculating expected loss values over these two outcomes for each bid gives values in ascending order of bid sum as follows.

For the £0.8m bid the expected loss is

$$(15\,000 \times 0.5) - (7195 \times 0.5) = +£3902.5$$

For the £0.9m to the £1.5m bids the expected losses are respectively -£11 799, -£14 579, -£16 376 and +£1156. The next node in the roll-back is a decision node corresponding to the bid. Here Bayes' principle requires that the smallest of the possible expected loss values should be selected. This is clearly the value of -£16 376 just calculated for the £1.2m bid. The next node is a chance node corresponding to the possible site condition forecasts D, MW or VW. There are probabilities of 0.185, 0.48 and 0.335 associated respectively with these three outcomes, as already noted, and an expected loss value of -£16 376 associated with the MW branch. Figure 10.7 was not able to depict the D and VW branches in full but their arrangements are both similar to that of the MW branch shown. A roll-back process similar to that just described for the MW branch can be carried out on the D and VW branches. The resulting expected loss values are -£18 081 for the D branch corresponding to a £1.2m bid and -£11 328 for the VW branch also corresponding to a £1.2m bid. Verification of these values is left to the reader. Since the survey forecast is a chance node, the expected value of these three loss outcomes must be calculated. This gives an expected loss along the complete 'use survey' branch of

$$-(18\,081 \times 0.185) - (16\,376 \times 0.48) - (11\,328 \times 0.335)$$
$$= -£15\,000$$

Comparing this expected loss of -£15 000 using the site survey with the previously calculated expected loss of -£20 000 achieved without the benefit of the private survey, it is seen that the contractor should *not* use the survey. Using it merely reduces the expected profit by an amount equal to the £5000 cost of the survey. The reason for this lies in the fact that calculations show that the contractor should make the same bid of £1.2m for all possible outcomes of the site survey forecast. Furthermore this is precisely the same bid he would make as a result of following the 'no survey' branch. The conclusion to be drawn from this example is that the contractor should not use the private site survey and should make a bid of £1.2m for the contract.

Decision trees like that described in this example form a valuable analytical tool for the decision maker. They share with critical path methods and all the network methods of chapter 4 the virtue of imposing logic upon a seemingly amorphous problem. In order to construct the decision tree for a problem it is necessary to think very carefully about the problem, to sort out chance outcomes from decision outcomes and to create an ordered sequence of chance and decision nodes and branches. The act of constructing the decision tree can itself

www.EngineeringBooksPDF.com

be of value to the decision-maker by virtue of the logical ordering of the thought processes needed to create the tree from an amorphous problem.

The roll-back process using a least expected loss criterion also provides a logical basis for decision-making. In this example the decision tree analysis showed quite clearly that the contractor would not benefit by spending money on a private site survey. In a practical situation this would be a very positive and useful result. It should perhaps be pointed out that this result is specific only to this example. It does not imply that site surveys or the seeking of additional information is always unnecessary. In this case the additional information was not of sufficient benefit and added nothing to those site condition probabilities given in table 10.3 which presumably were also determined from site investigations.

10.4.4 Utility Functions

In example 10.4.3.1 the expected value of monetary loss was used to determine a set of decisions for the contractor. There are many examples for which financial loss is not an appropriate decision-making criterion. Indeed the contracting industry can furnish examples. Suppose for instance that the contractor in the above example has had a bad year, the number of new contracts coming up is very few and the contractor really needs a new project to keep his plant and work-force busy. In this set of circumstances the contractor may desire to put in a much lower bid than that indicated by the decision tree analysis using the principle of minimising expected losses. In that example the minimum expected loss was actually an expected profit. Perhaps the contractor would be happy to accept less expected profit if this would increase his chance of actually winning the contract. If this is the case the contractor is using the wrong criterion in the previous example. Can he use some alternative criterion which reflects his personal objectives and preferences in a better way than does the expected loss criterion?

There is such a criterion, known as a *utility function*, which is often used in decision-making instead of expected loss. Essentially the utility function approach converts all monetary outcomes on the decision tree to utility values on a scale of preferences determined by the decision-maker. Referring to figure 10.5 as an example, this shows the 'no survey' branch of the contract tendering problem, example 10.4.3.1. The range of possible outcomes shown lies between a loss of £80 000 and a profit of £740 000. Consider the contractor's possible replies to the following question: 'Given the present circumstances of your business, what would you consider to be an acceptable loss and what would you consider to be a reasonable profit?'. If his circumstances were good with plenty of work and a healthy thriving business, the contractor might reply that £10 000 was an acceptable loss and that £740 000 would be a good profit. He has to accept a possible loss of £10 000 or presumably he would not be bidding but he does not need the work so badly as to risk a loss of £80 000. He is seeking profit and the

www.EngineeringBooksPDF.com

reasonable profit value of £740 000 reflects his sense of optimism. On the other hand, if the contractor's circumstances are bad, his answers might be £80 000 as a loss value and £100 000 as a profit value. His reasonable profit value and the fact that he is prepared to risk a loss of £80 000 reflect his pressing need for work in order to stay in business.

For the optimistic contractor the value or *utility* of the contract lies in the range +£10 000 to -£740 000 and for the other set of circumstances the utility lies in the range +£80 000 to -£100 000. The utility function criterion requires that the utility range be determined as above and replaced by a proportional scale ranging from zero to unity. Thus for the contractor in poor business circumstances, +£80 000 loss is given a utility value of 0.0 and -£100 000 loss is given a utility value of 1.0. All the possible monetary outcomes on figure 10.5 are then ranked according to this scale and are given an appropriate utility value. Thus a loss of +£10 000 would be given a utility value of

$$(80\,000 - 10\,000)/(100\,000 + 80\,000) = 0.3889$$

Referring to figure 10.5 the outcomes for the £0.8m bid then have utility values as follows

+£10 000 loss has a UV of 0.3889
+£80 000 loss has a UV of 0.0
-£20 000 loss has a UV of 0.5556
-£70 000 loss has a UV of 0.8333

All the outcomes of the other bids are given UVs on the same scale. If an outcome has a monetary value beyond the ends of the range it is given a UV value at the appropriate end of the range, that is, 0.0 or 1.0. The effect of this is to compress or distort the monetary outcomes to conform to a utility value scale which corresponds to the preferences of the contractor. In the case of the optimistic contractor there is very little distortion; his utility scale covers almost the whole range of monetary outcomes. For the contractor in poor business circumstances the utility scale is very distorted with all the very high profit outcomes being given the same utility value of 1.0.

Having assigned utility values to all outcomes the roll-back process is carried out exactly as before but calculating expected values of utility at each chance node instead of expected loss and at each decision node selecting the decision which yields *maximum utility*. Thus the objective of decision analysis using utility values is to maximise the value of the possible decision policies with the value scale determined by the decision-maker's own preferences and circumstances. The roll-back process using utility values is left to the reader. The maximum utility of the 'no survey' branch of figure 10.5 is 0.5021 and corresponds to a bid of £0.9m. The result for the complete decision tree using utility functions is that the site survey should not be used and the contractor should bid £0.9m.

In this example the utility scale was used in place of monetary outcomes. One

www.EngineeringBooksPDF.com

useful feature of utility functions is that they can be used to replace any scale of outcomes not just those representing money. The decision-maker merely has to specify the desirability of each possible outcome on a utility value scale. Thus they can be used to rank several simultaneous criteria. The reader should consult more specialised literature for a fuller treatment of these very useful utility functions.

10.5 MAINTENANCE AND REPLACEMENT PROBLEMS

A large class of problems for which expected values can be used for decision-making is that concerned with policies for the maintenance, repair and replacement of plant and equipment. All equipment is likely to fail. If it fails during service its repair costs and the costs associated with the temporary unavailability of the equipment can be large. A policy of repairing or replacing the equipment only when it fails is called a *corrective policy*. An alternative to a corrective policy is a *preventive policy* in which the equipment is repaired or replaced when it fails and also at regular intervals during its life. It is usually less costly to repair a piece of equipment during off-shift hours than during working hours. Depending on the piece of equipment one of the many possible maintenance policies should be better in terms of expected cost than the others. Since equipment failures are random events, the best policy can only be determined probabilistically. An example demonstrates this.

Example 10.5.1 – Maintenance of a Cutting Head

The cutting head of a tunnelling machine is very vulnerable to wear and disintegration (failure). When the head fails it must be replaced. The probability of failure of the cutting head can be expressed as a cumulative distribution function $F(T)$, of the time in use, T, of the cutting head. A typical distribution function for the observed failures of cutting heads is tabulated in table 10.6.

Table 10.6 Distribution function of time to failure for cutting heads

T (hours)	12	24	36	48	60
$F(T)$	0.05	0.3	0.7	0.9	1.0

The cost of replacing a cutting head during working hours, including all associated costs of the delay, is £1200. The cost of replacing a cutting head during non-working hours is only £500. What is an appropriate maintenance policy for the cutting head?

Many possible maintenance policies exist but only two will be examined here. The first is a purely corrective policy in which nothing is done until the head

www.EngineeringBooksPDF.com

fails and then it is replaced. The second is a preventive policy in which failures are replaced when they occur and the head is also replaced at regular time intervals during off-shift hours in the hope of reducing on-shift failures. For this second policy it is necessary to determine the best time interval between off-shift replacements.

Table 10.6 gives the *distribution* function of time to failure. From this it can be seen that no head lasts for more than 60 hours. The *density* function $f(T)$ can be deduced from table 10.6 using the relationship

$$f(T_1 - T_2) = F(T_2) - F(T_1)$$

that is, the probability of the head failing in the interval $T_1 - T_2$ is equal to the probability of failure in the interval $0 - T_2, (F(T_2))$ minus the probability of failure in the interval $0 - T_1, (F(T_1))$. Table 10.7 lists these derived probability densities. In contrast to the contract tendering example in which probabilities were subjective estimates, the probability functions in this example are objectively based, derived from operational data on the failures of cutting heads.

Table 10.7 Probability densities of cutting head failures

Interval (hours)	0-12	12-24	24-36	36-48	48-60
f(interval)	0.05	0.25	0.4	0.2	0.1

Consider first the corrective policy of replacing cutting heads as and when they fail in service. To determine the cost of such a policy the expected life of a cutting head must be determined. The means of the intervals in table 10.7 are located at 6, 18, 30, 42 and 54 hours and the probability densities are associated with each of these mean times. The expected life of a cutting head is then

$$(6 \times 0.05) + (18 \times 0.25) + (30 \times 0.4) + (42 \times 0.2) + (54 \times 0.1)$$
$$= 30.6 \text{ hours}$$

The replacement cost is £1200 and this sum can be expected to be incurred every 30.6 hours thus this corrective policy will cost £1200/30.6 = £39.22 per hour.

Now consider a preventive maintenance policy in which the head is replaced as and when it fails in service and also is replaced every x hours. Several possible values for x will be examined. Firstly, consider x = 12 hours. The probability that the head will fail in the first 12 hours is 0.05 so the expected cost of replacing the head during working hours will be £1200 x 0.05 = £60.00. The cost of replacing the existing head every 12 hours in off-shift time is £500 per replacement so the total cost of all replacements will be £60 + £500 = £560 over a 12 hour period. This works out to be £46.66 per hour.

Consider now another replacement interval of x = 24 hours. The probability that the head will fail in the first 12 of these 24 hours is 0.05. The failure probability for the second 12 of the 24 hours is 0.25. A head which fails in the first

www.EngineeringBooksPDF.com

12 hours, however, would be replaced and the replacement head might itself fail in the second 13 hour interval (but the first 12 hour period of its life). The probability of this is 0.05 x 0.05 = 0.0025, that is, the joint probability of consecutive heads failing, each within 12 hours. Thus the total probability of having to replace failed cutting heads over a 24 hour period is the sum of the individual failure outcomes, 0.05 + 0.25 + 0.0025 = 0.3025. The expected cost of these replacements is £1200 x 0.3025 = £363.00. The cost of routine head replacement after a 24 hour period is again £500 giving a total repair cost of £863.00 over 24 hours, that is, £35.96 per hour. This is less than both the corrective policy and the 12 hour preventive policy. The 36 hour preventive policy should now be examined to see whether it is even cheaper than the 24 hour preventive policy.

Firstly, the cost of in-service head replacements must be calculated. There are many more possible failure outcomes than for the 24 hour policy and the possible failure events are listed in table 10.8 along with the probability of occurrence of each pattern of failures. Note that table 10.8 contains all possible combinations of failures.

Table 10.8 Possible head failures over a 36 hour period

Interval (Hours)			
0-12	12-24	24-36	Probability
F	–	–	0.05
–	F	–	0.25
–	–	F	0.4
F	F	–	0.0025
F	–	F	0.0125
–	F	F	0.0125
F	F	F	0.000625
			$\Sigma = 0.728125$

In table 10.8 the symbol F denotes a head failure within a time interval and the symbol – denotes no failure. Thus row 1 denotes a head failure during the first 12 hours and the replacement head working satisfactorily thereafter. The probability of this outcome is 0.05. Row 5 shows a head failure in the first 12 hours with the replacement head failing during the second 12 hours of its life (the third 12 hour period of the 36 hour maintenance policy). The probability of this happening is $f(0\text{-}12) \times f(12\text{-}24) = 0.05 \times 0.25 = 0.0125$. Summing up all possible failure probabilities gives a 0.728125 chance that in-service replacements will be needed. The expected cost of replacements will be £1200 x 0.728125 = £873.75 and the cost of routine head replacement after 36 hours is £500 giving a total cost of £1373.75. This works out at £38.16 per hour. This is,

www.EngineeringBooksPDF.com

therefore, more expensive than the 24 hour preventive maintenance policy which should, therefore, be selected. There are many other possible maintenance policies that could also be evaluated. For example, it may be possible to reduce maintenance costs even further by replacing during the scheduled period only a head that has not been replaced since the last scheduled maintenance, or by replacing heads a fixed length of time after the last failure. Expected costs make it possible to examine all these policies and to select the best.

There are many other classes of problems in which the use of expected values can provide a means of making decisions in the presence of chance and uncertainty. Expected values provide a framework for ranking the mean effects of random outcomes and of selecting the most appropriate policy.

10.6 RELIABILITY

A major class of problems within the probabilistic domain is that concerned with the reliability of engineered systems. Reliability, REL, is related to probability of failure PF by the general relationship

$$REL = 1 - PF \tag{10.21}$$

This relationship applies to individual components within a system and to whole systems. As was mentioned in section 10.1 and as equation 10.21 implies, reliability only exists in the probabilistic domain; when everything is certain and deterministic failure is not a chance outcome, it may be prevented with certainty by appropriate design. As section 10.1 attempted to show, however, very little is truly deterministic: some quantities almost manage to be deterministic; with many the range of variation in value may be very small but absolute certainty is a very rare quality. Consequently, since almost everything is probabilistic, the concept of reliability must apply in some degree very extensively.

This section introduces some basic features of reliability problems. Two questions arise as a consequence of equation 10.21; 'How can the probability of failure of a single component be determined?', and 'How can the probability of failure of a complex multi-component system be found?'. These two questions, vital to the calculation of reliability, are considered below. A third question is perhaps of more immediate concern to decision-makers and asks how decisions can be made about the planning, design and operation of engineering works in which reliability is a major factor.The question of whether it is possible to design things which have a specified reliability is also studied and is shown to be beset by many difficulties.

10.6.1 Reliability of a Single Component

The reliability of a single component is related by equation 10.21 to the probability of failure of the component and to study reliability it is first necessary to define what is meant by failure. Extreme cases of component failure are obvious:

www.EngineeringBooksPDF.com

a pipe may burst under pressure, a structural member may break when loaded or an electrical component may burn out under power. In these examples failure is sudden, obvious and catastrophic. The component moves rapidly from a working state to a non-working state. A definition of failure need not necessarily include this ultimate non-functioning or broken state. Indeed, in much engineering design, components must be considered to have failed long before they actually cease to function. For example, concrete structures are sometimes deemed to have failed when surface cracks open up under load to a specified width. Some steel structures are deemed to have failed if the elastic limit is ever exceeded. In both these examples absolute failure of the structure in the sense of collapse or disintegration may still be a long way off but technically failure has occurred when some predefined limit is exceeded. Failure can, therefore, correspond to *any* state of the component as defined by the design engineer. A failed component is one that does not meet, however marginally, the specified criteria of performance.

To examine the probability of failure of a component the most general formulation consists of supposing that the component has associated with it a quantity which represents the reactive or resistance capability of the component and that this resistance quantity, R, is a random variable. For example, suppose that the reliability of a single steel tension member is being investigated and that as a criterion of failure the elastic limit or yield stress has been selected. Thus the bar is deemed to have failed when its elastic limit is exceeded. The resistance quantity, R, can be represented by the axial force in the bar *at failure*. Since the properties of steel are not uniform or homogeneous, the elastic limit and hence R for the bar are probabilistic in nature. Carrying out a series of tensile tests on bars as similar as possible to the bar of interest would yield a range of values for R that can, therefore, be represented as a probability density. A typical density function for R is shown in figure 10.8. As expected, it shows a clustering of values around a peak but with some sample bars being significantly stronger or weaker than others. The most interesting comment that can be made about the bar is that its failure will probably occur within the range of axial forces covered by the density function of R.

Now consider the component with resistance R to be subjected to active load or force A and that A is also a random variable whose density function is shown on figure 10.8. In the steel bar example the bar may be part of a structure subjected to wind or wave loading or to any other type of varying load. A represents the magnitude of loading transmitted through the structure to the bar in question. On figure 10.8 it can be seen that most of the values of A are lower than those of R and are clustered around a peak but that occasionally very low and very high values of A are experienced.

Both the active random variable A and the reactive random variable R can be shown on the same diagram because they share a common value scale, the horizontal axis. In the example of the steel bar this axis would measure force values. In figure 10.8 it can be seen that the reaction R is generally much larger in magni-

www.EngineeringBooksPDF.com

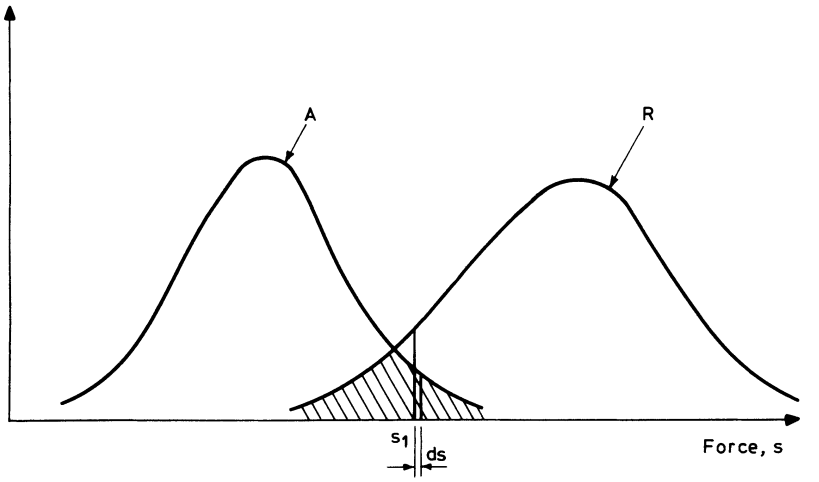


Figure 10.8 Active and reactive probability functions for a single-component reliability calculation

tude than the action A. This represents a sensible design that has a reactive capability which is greater than usually applied actions. In the shaded triangular region, however, where the tails of the two density functions intersect, the action can exceed the reactive capacity of the component. This shaded region of figure 10.8 corresponds to possible failures of the component. Failure can, therefore, be defined to occur when the action is greater than the reactive capacity of the component. Thus the probability of failure of the component is the probability that for all possible horizontal axis values A exceeds R, that is

$$\text{PF} = p(A > R) \tag{10.22}$$

Denoting the horizontal axis scale by s ($s \equiv$ force for the bar example) consider a particular value of force s_1 and another value $s_1 + \mathrm{d}s$ infinitesimally close to it. What is the probability that the active force A will have a value within this small interval s_1 to $s_1 + \mathrm{d}s$? This can be written as $p(s_1 \leqslant A \leqslant s_1 + \mathrm{d}s)$ and from 10.8 this probability is

$$p(s_1 \leqslant A \leqslant s_1 + \mathrm{d}s) = \int_{s_1}^{s_1+\mathrm{d}s} f_A(s)\mathrm{d}s \tag{10.23}$$

In equation 10.23 $f_A(s)$ is the density function of variable A in terms of the force quantity s on figure 10.8.

The component will fail *at force* s_1 whenever two conditions or events occur simultaneously. Firstly, the active force must have a value within the infinitesimally small range just above s_1 and equation 10.23 gives the probability of this event. Secondly, the reactive capacity of the component must be less than or

www.EngineeringBooksPDF.com

equal to s_1. What is the probability of this event? The probability that the reactive capacity R will be less than or equal to s_1 is written $p(R \leqslant s_1)$ and is, from equation 10.10

$$p(R \leqslant s_1) = F_R(s_1) = \int_{-\infty}^{s_1} f_R(s)\mathrm{d}s \tag{10.24}$$

$F_R(s_1)$ is the distribution function of variable R at strength s_1. Since A and R are independent random variables the probability that equations 10.23 and 10.24 will occur simultaneously is their joint probability, that is, the product of the independent probabilities (equation 10.6). Thus the probability of failure of the bar at a force value within a range infinitesimally above s_1 is

$$PF(s_1 + \mathrm{d}s) = F_R(s_1) \int_{s_1}^{s_1+\mathrm{d}s} f_A(s)\mathrm{d}s \tag{10.25}$$

Equation 10.25 gives the probability of failure within the range s_1 to $s_1 + \mathrm{d}s$. The total probability of failure of the component at all force values s_1 is given by integrating 10.25 over the whole force range $-\infty \leqslant s_1 \leqslant +\infty$. Some simplification of the integration limits within equation 10.25 yields equation 10.26 which is the total probability of failure of the component

$$\mathrm{PF} = \int_{-\infty}^{\infty} F_R(s) f_A(s)\mathrm{d}s \tag{10.26}$$

Equation 10.26 is one that in theory allows the probability of failure of a component to be calculated from the probability functions of the active and reactive variables. In practice, however, these probability functions often involve exponentials (for example, equations 10.18 and 10.20) and the integrand within equation 10.26 is usually algebraically intractable. Thus in most cases the definite integral in equation 10.26 can only be evaluated numerically using Simpson's rule or other more specialised quadratures.

From the point of view of decision-making and design the intractability of equation 10.26 is unfortunate. The designer may wish to design a component to have a specified reliability which implies a specified PF value. The designer usually has no control over the active random variable A but he can define and control the reactive variable R. In the steel bar example the cross-sectional area of the bar is under the designer's control. By specifying a small area the designer effectively shifts the entire R probability function on figure 10.8 to the left, reducing the reactive capacity. The intersecting tails area of A and R will be increased and so the probability of failure will increase. By increasing the cross-sectional area the R function is moved to the right and the probability of failure is reduced. There is, however, no easy way of choosing a cross-sectional area to give some required PF. In most cases several trial sizes must be investigated, per-

www.EngineeringBooksPDF.com

forming a numerical integration of equation 10.26 at each trial. Several of the methods described in chapter 7 can be used to perform this design using the fewest number of trials. As a final comment it should be noted that increasing the bar's cross-sectional area will decrease PF and hence will increase the component reliability. Increasing the bar's area will also increase the component cost. Thus increased component reliability is only achieved at extra cost.

10.6.2 Reliability of a Multi-component System

In many engineering projects components are connected together to form multi-component systems. For example, each pipe in a water supply network is a component and each beam or column or slab in a structure can be a component. Each component plays its assigned part in the proper functioning of the whole system. From the engineering analysis and design point of view it is sometimes necessary to be able to determine the reliability of a whole system rather than just that of individual components. If the reliability of each individual component in a multi-component system is known it is theoretically possible to calculate the over-all reliability of the whole system. These calculations are difficult, however, because the interactions of components are different within different systems. In some systems the failure of one component may cause failure of the whole system; in others a single component failure may not cause the whole system to fail.

Figure 10.9 shows some multi-component systems. In figure 10.9a the components are arranged in a chain. This is known as a *series* system, sometimes called a *weakest-link* system. Defining REL_i and PF_i to be respectively the reliability and failure probability of component $i = 1, \ldots, N$ in an N-component system, and REL_s, PF_s to be the whole system reliability and failure probability, equation 10.21 holds. Thus

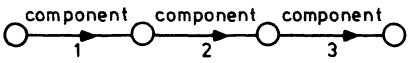
$$REL_s = 1 - PF_s; REL_i = 1 - PF_i \qquad i = 1, \ldots, N$$

For a series system composed of components whose reliabilities are independent (that is, the failure of one component has no effect on the failure probability of another) the system reliability is defined as

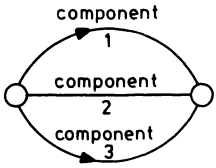
$$REL_s = \prod_{i=1}^{N} REL_i = \prod_{i=1}^{N} (1 - PF_i) \tag{10.27}$$

The derivation of equation 10.27 is as follows. A series system will fail if any one of the components fails. REL_s depends on all components not failing. The probability that component i does not fail is REL_i and so by equation 10.7 the joint probability of all N components not failing is simply the product of all the REL_i values over all N components. The name weakest link became attached to a series system because, like any chain, it will fail when any link fails. Furthermore, since REL_i cannot exceed unity for any component, 10.27 implies that the

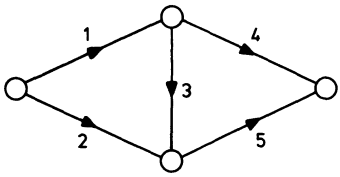
www.EngineeringBooksPDF.com



a) A series system with three components.



b) A parallel system with three components.



c) A general five component system.

Figure 10.9 Multi-component systems

system reliability cannot ever exceed the reliability of its least reliable component (or weakest link).

Figure 10.9b shows what is termed a *parallel* system, sometimes called a *fail-safe* system. The system reliability for a parallel system is defined as

$$REL_s = 1 - \prod_{i=1}^{N} PF_i = 1 - \prod_{i=1}^{N} (1 - REL_i) \qquad (10.28)$$

Clearly in figure 10.9b the failure of one component does not cause the whole system to fail. Provided the components are independent they must all fail for the system to fail. The probability that they will all fail is, by equation 10.7, the product of their individual failure probabilities and this is then the whole system failure probability from which equation 10.28 follows. The name fail-safe reflects the fact that if a component fails the whole system can still function safely.

Figures 10.9a and b demonstrate that system reliability depends both on individual component reliabilities and on the ways in which individual components are connected together. In many practical systems with many components the arrangement of components is neither series nor parallel but consists of a complex

www.EngineeringBooksPDF.com

network of which figure 10.9c is a simple example. The calculation of system reliability for general component arrangements is more difficult and specialised texts should be consulted. Only one method will be described here for the example of figure 10.9c.

Let R_i be the component reliability of component $i = 1, \ldots, 5$ on figure 10.9c. In chapter 4 the idea of paths through networks was explored. In this example there are three possible paths through the network; paths 1-4, 2-5 and 1-3-5. Taken separately each of these paths is a series system and so equation 10.27 can be used to determine the reliabilities R_{p1}, R_{p2} and R_{p3} of these paths. Thus, using equation 10.27

$$R_{p1} = R_1R_4 ; R_{p2} = R_2R_5 ; R_{p3} = R_1R_3R_5$$

The whole system will fail if all three paths are broken. The reliability R_s of the whole system depends on all three paths not failing simultaneously. The three *paths*, therefore, form a parallel reliability system. Thus 10.28 can be used to calculate R_s, that is

$$\begin{aligned} R_s &= 1 - (1 - R_{p1})(1 - R_{p2})(1 - R_{p3}) \\ &= 1 - (1 - R_1R_4)(1 - R_2R_5)(1 - R_1R_3R_5) \\ R_s &= R_1R_4 + R_2R_5 + R_1R_3R_5 - R_1^2R_3R_4R_5 - R_1R_2R_4R_5 \\ &\quad - R_1R_2R_3R_5^2 + R_1^2R_2R_3R_4R_5^2 \end{aligned}$$

The elements such as R_1^2, R_5^2 in these expressions for R_s can be replaced by R_1, R_5 respectively since they represent the joint probability with itself of a component not failing and this is simply equal to the component reliability. Thus the whole system reliability of figure 10.9c is

$$\begin{aligned} R_s = {} & R_1R_4 + R_2R_5 + R_1R_3R_5 - R_1R_3R_4R_5 - R_1R_2R_3R_5 \\ & - R_1R_2R_4R_5 + R_1R_2R_3R_4R_5 \end{aligned} \qquad (10.29)$$

A typical design problem might be to determine individual component reliabilities such that the whole system has a specified reliability. If all components are to be identical with reliability R, and the required system reliability is 0.99 then equation 10.29 becomes

$$2R^2 + R^3 - 3R^4 + R^5 = 0.99$$

which may be solved numerically to give $R = 0.942$.

Figure 10.9c represents a very simple multi-component system. As the complexity of the network increases, system reliability calculations become more difficult and the problems associated with designing the system grow in magnitude.

Example 10.6.2.1 – A Reliability Design Problem

A heavy industry company extracts cooling water for its manufacturing processes from a river and pumps it to the individual processes. Figure 10.10 shows

www.EngineeringBooksPDF.com

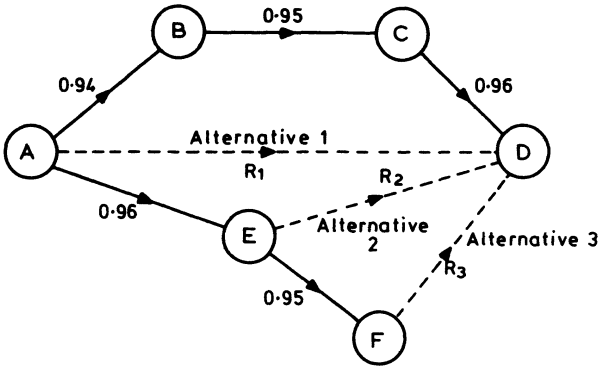


Figure 10.10 Water supply system of example 10.6.2.1

a schematic arrangement of the cooling water network. A is the river source and B, C, D, E and F are manufacturing processes requiring cooling water. Each of the arcs of the network represents a pump and pipeline and each arc can be idealised as a single component. The values beside each arc represent reliability values for the arcs.

The company is concerned about the poor reliability of the cooling water supply to process D. Interruptions to the supply to D occur too frequently and each time cause considerable loss of production in D. The company wishes to achieve a 95 per cent reliability of water supply at D and is investigating several ways of achieving this as cheaply as possible. Three alternatives are possible

(1) construct a new pumped pipeline linking A to D
(2) construct a new pumped pipeline linking E to D
(3) construct a new pumped pipeline linking F to D

Each alternative entails a cost which is proportional to the length of pipeline required and inversely proportional to the square root of the probability of failure of the pumped pipeline component, that is

$$C_i = K_i L_i / \sqrt{(\mathrm{PF}_i)} \tag{10.30}$$

For the three alternatives selected above values of $K_i L_i$ are £14 000, £12 000 and £11 800 respectively. Which alternative should the company select?

Examining alternative 1, the new arc from A to D will mean that water can reach D via two independent paths. The first path is ABCD, a three-component series system. From equation 10.27 the reliability of this path is (0.94) x (0.95) x (0.96) = 0.85728. This is the reliability of the present water supply system to process D. The reliability of the new link from A to D direct is R_1 and R_1 must be found to make the reliability of supply to D equal to 0.95. The two

www.EngineeringBooksPDF.com

possible paths to D are independent and form a parallel system so the system reliability, from equation 10.28 will be

$$R_S = 0.95 = 1 - (1 - 0.85728)(1 - R_1)$$

that is

$$0.95 = 0.85728 + 0.14272R_1$$

which yields the required reliability value R_1 for the link AD as

$$R_1 = \frac{0.95 - 0.85728}{0.14272} = 0.6497$$

Substituting this value into equation 10.30 gives the new link cost as

$$C_1 = \frac{14\,000}{\sqrt{(1 - 0.6497)}} = \frac{14\,000}{0.5919} = £23\,653$$

Now examining alternative 2 the new arc from E to D with reliability R_2 will also create two possible supply paths to D. One is ABCD as before with a path reliability of 0.85728. The second path is AED, a two-component series system for which equation 10.27 gives the reliability as (0.96) x (R_2). As before, these two paths form a parallel system and the total system reliability will, therefore, be

$$R_S = 1 - (1 - 0.85728)(1 - 0.96R_2)$$

As R_S must be equal to 0.95 this becomes

$$0.95 = 0.85728 + 0.13701R_2$$

that is

$$R_2 = 0.6767$$

The cost of this new link from E to D is then

$$C_2 = \frac{12\,000}{\sqrt{(1 - 0.6767)}} = \frac{12\,000}{0.5686} = £21\,104$$

For the third alternative the path reliabilities are 0.85728 for ABCD, and (0.96) x (0.95) x (R) = $0.912R_3$ for AEFD where R_3 is the reliability of the new link FD. These two paths form a parallel system for which the reliability is

$$R_S = 1 - (1 - 0.85728)(1 - 0.912R_3) = 0.95$$

This yields R_3 = 0.7124 and the link cost for FD is

$$C_3 = \frac{11\,800}{\sqrt{(1 - 0.7124)}} = \frac{11\,800}{0.5363} = £22\,003$$

Comparing the three alternatives, the second is the cheapest and should be chosen. The company should, therefore, design and build a pumped pipeline be-

www.EngineeringBooksPDF.com

tween E and D to have a reliability of 0.6767. The third alternative has a smaller length cost but must be designed to a greater reliability and hence will cost more.

In this example equation 10.30 is artificially contrived for simplicity but it does contain the important feature that costs increase as the reliability increases. For the impossible completely reliable component for which PF_i = zero the cost C_i would be infinite.

10.6.3 Cost versus Reliability

In the last example, 10.6.2.1, the manufacturing company decided to invest capital to improve reliability and to save production losses. The way they chose to do this was to specify some desired reliability and examine ways of achieving this reliability at minimum cost. Essentially, this is an optimisation problem of the form

$$\begin{aligned} &\text{Minimise: capital costs} \\ &\text{subject to: system reliability} \geqslant \text{desired reliability} \end{aligned} \tag{10.31}$$

This problem form reflects the dominant nature of money as the objective function in industry and business. Reliability is relegated to a lesser role as a constraint. There are, however, difficulties associated with the form of problem 10.31. Firstly, it is necessary to specify a desired reliability value and this can be difficult. Although reliability is well understood in industry it is difficult to pluck from the air some desired numerical value. Secondly, suppose some value can be deduced by examining data and a desired reliability has been set. If problem 10.31 is then solved and the resulting minimum cost is too high for the company's budget, what happens next? Presumably a reduced reliability value must be specified and problem 10.31 solved again to yield a smaller minimum cost and this process repeated until some company budget limit is not violated. This process suggests that a monetary budget limit should be a constraint with reliability as an objective function.

In fact, this argument whether money is an objective or a constraint is a very common one. In the area of reliability there is an alternative problem form to 10.31. This is

$$\begin{aligned} &\text{Maximise: system reliability} \\ &\text{subject to: capital costs} \leqslant \text{budget limit} \end{aligned} \tag{10.32}$$

Problem 10.32 is an inversion of problem 10.31. If the least cost solution of 10.31 is provided as a budget limit in 10.32, the maximum system reliability obtained by solving 10.32 will equal the desired reliability specified in 10.31. Since it is often easier to specify a budget limit rather than a desired reliability, problem 10.32 is often used in preference to 10.31 in reliability optimisation contexts.

Problem 10.32, however, is still not entirely satisfactory as a general formulation. The maximum reliability for a particular budget limit will be obtained but

www.EngineeringBooksPDF.com

further analysis will be required to determine whether this reliability is adequate for the system being designed. The difficulty with both problems 10.31 and 10.32 is that costs and reliabilities are measured on different scales and it is consequently hard to compare and contrast the results. If reliability could be measured on a cost scale these difficulties would be ameliorated. There is a way of doing this. Equation 10.21 states that the probability of failure of a system is simply related to the system reliability. When a system fails costs will be incurred. In example 10.6.2.1 these are the costs of lost production in process D. These failure costs should be calculable and so an expected loss value can be found by multiplying failure costs by $(1 - REL_S)$. These expected failure losses are then a function of the system reliability REL_S and can be added to the capital costs which are also dependent upon REL_S. This leads to the problem formulation:

$$\underset{REL_S}{\text{Minimise:}} \text{ capital costs + expected failure losses} \qquad (10.33)$$

Problem 10.33 is an unconstrained optimisation problem and variables within it are the individual component reliabilities which combine together in the system reliability REL_S.

Figure 10.11 shows the general form of problem 10.33 graphically. Cost is plotted vertically and system reliability horizontally. The graph shows that capital costs increase as system reliability increases. This means that, as may be expected, a system that does not fail costs more to build than one that does. The

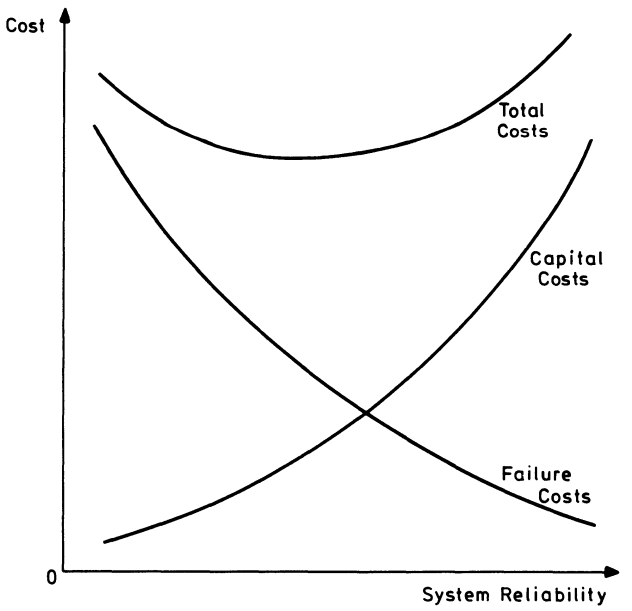


Figure 10.11 Graphs of costs against reliability

www.EngineeringBooksPDF.com

graph for failure costs decreases as reliability increases, a result which accords with intuition. The third graph represents the sum of the two previous graphs and is a convex graph with a global minimum. This is the graph of the unconstrained function in problem 10.33.

Logically, problem 10.33 is more satisfactory than problems 10.31 and 10.32 because the common cost basis makes the results easy to interpret. Even so, there are difficulties associated with it. Capital costs are incurred once only when the system is built but failure costs will be spread over the lifetime of the system. Consequently, some delicate economics calculations must be made to one or both sets of costs to ensure their equivalence so that they may be added together. Problem 10.33 therefore corresponds to the minimisation of total construction and failure/repair costs. Although such a long-term least cost policy is entirely reasonable it is not necessarily always appreciated at corporate decision-making levels where more immediate short-term benefits often appear attractive. Over thirty or forty years the total costs of buying and maintaining one Rolls-Royce car may turn out to be much smaller than the total costs of buying and maintaining several successive compact cars but relatively few people will buy a Rolls-Royce because of the very large initial capital required.

So far in this discussion of reliability-based decision-making a system failure has been assumed to incur some financial loss. These monetary losses occur through loss of production during system failure and also through the repair costs themselves for a failed system. There are, however, other possible consequences of a system failure that cannot be reduced to a simple cost. In many cases a system failure can lead to the injury or death of people who use the system or even of the general public. A cost cannot be assigned to the personal injury or death of anyone for the purposes of engineering decision-making. When injury or death is a possible outcome of a system failure then a reliability-based design procedure must always have the objective of maximising the system reliability. As has been shown, however, reliability can always be improved provided that money is available to do it. In many instances the consequences of this are that enormous amounts of money must be spent in order to achieve a high reliability level. The manned spaceflight program and the development of nuclear power stations are two good examples where system reliability needs dominate all other considerations. A very large portion of the costs of these projects can be directly attributed to reliability considerations. System failure probabilities of the order of 10^{-6} to 10^{-8} have been used as design criteria for some very critical systems. Absolute reliability, as we have seen, is an impossibility in a world which is essentially governed by chance. The unfortunate consequence of this is that, however carefully the design is carried out and however small the probability of failure is by design, the random nature of chance is such that, despite all the hardest efforts of designers, failure can always occur.

As a final comment on the topic of reliability it should be noted that this section has considered only a definition of failure in which an active agent exceeds a reactive capacity both of which are defined probabilistically. Deliberately ex-

www.EngineeringBooksPDF.com

cluded (in the second paragraph of section 10.1) are extraneous failures unrelated to these probability distributions. Thus gross errors in design, manufacture or use of a system that can be a major cause of failures do not fit into the treatment given in this section. It can be argued that, in cases such as nuclear reactor pressure vessel design and other high integrity designs, the human error factor is a more likely cause of failure than the action/reaction effect described here. This may be so but human error is far more difficult to locate and to estimate and is now the subject of much research. The fact remains that chance plays a large part in all aspects of civil engineering and from a civil engineering systems viewpoint it is necessary to appreciate the nature of chance and to make decisions that reflect this nature. Of all the areas of civil engineering systems these problems dominated by chance are perhaps the least well known and are consequently the most in need of new decision-making methods.

SUMMARY

This chapter has examined problems of a civil engineering nature in which chance plays a major part. Special methods based upon probability concepts are required for decision-making in chance-dominated problems. Only a few special areas of a vast subject were examined. The use of expected values and Bayes' principle produces a logical and much used decision-making method for a wide variety of problems. Risk is an unavoidable consequence of decision-making when the chance element is concerned. When the decision-making process is concerned with engineering design, risk becomes synonymous with probability of failure. Methods for evaluating failure probability and its complement, reliability, were described for single components and multi-component systems. Despite the analytical rigour and algebraic elegance of reliability calculation methods, there are still many conceptual and philosophical problems surrounding reliability that make its results often less than satisfying. Some of the reasons for this were qualitatively explored. The chance aspects of civil engineering systems remain underexplored.

BIBLIOGRAPHY

Ang, A. H. S., and Tang, W. H., *Probability Concepts in Engineering Planning and Design, Vol. 1. Basic principles* (Wiley, New York, 1975)

Benjamin, J. R., and Cornell, C. A., *Probability, Statistics and Decision for Civil Engineers* (McGraw-Hill, New York, 1970)

Freudenthal, A. M., Garrelts, J. M., and Shinozuka, M., The analysis of structural safety, *Proc. ASCE*, **92**, No. ST1, (1966) 267-325

Fry, J., *Probability and its Engineering Uses* (Van Nostrand, Princeton, N. J., 1965)

Hahn, G. J., and Shapiro, S. S., *Statistical Models in Engineering* (Holden-Day, San Francisco, 1967)

www.EngineeringBooksPDF.com

Lloyd, D. K., and Lipow, M., *Reliability: Management, Methods and Mathematics* (Prentice-Hall, Englewood Cliffs, N.J., 1962)
Rau, J. G., *Optimization and Probability in Systems Engineering* (Van Nostrand Reinhold, New York, 1970)
Sandler, G. H., *System Reliability Engineering* (Prentice-Hall, Englewood Cliffs, N.J., 1963)
Schlaiffer, R. O., *Analysis of Decisions under Uncertainty* (McGraw-Hill, New York, 1969)
Shooman, M. L., *Probabilistic Reliability – An Engineering Approach* (McGraw-Hill, New York, 1968)
Smith, C. O., *Introduction to Reliability in Design* (McGraw-Hill, New York, 1976)
Tillman, F. A., Hwang, C.-L., and Kuo, W., *Optimization of Systems Reliability* (Marcel Dekker, New York, 1980)

EXERCISES

10.1 Test samples of concrete designed to have a characteristic strength of 20 N/mm^2 show strengths which are Gaussian distributed with a mean of 29.2 N/mm^2 and a standard deviation of 4.8 N/mm^2. Calculate the 95 percentile strength (the strength that is equalled or exceeded by 95 per cent of the samples) of the concrete.

10.2 To control pollution in a river the mean daily concentration, X, of a specified pollutant has been measured over a long period of time yielding the following results.

X (mg/m^3)	0.0	1.0	2.0	3.0	4.0	5.0
% of days on which X is equalled or exceeded	100	67.0	44.9	30.1	20.2	13.5

(a) Show that the concentration level may be modelled by the exponential probability density function, equation 10.20, and calculate the coefficient, λ.

(b) Maximum permissible value of X is 8.0 mg/m^3. What is the probability that this level will be reached or exceeded on any one day?

(c) What is the average return period (in days) of this dangerous pollution condition if the daily pollution levels are statistically independent?

10.3 A contractor is considering alternatives of buying, long-term leasing and short-term hiring for a large mobile crane. Usage of the crane depends on his work-load: over a five-year period the contractor estimates that his work-load will be high (probability 0.3), medium (probability 0.5), low (probability 0.2).

www.EngineeringBooksPDF.com

If he buys or long-term leases the crane he thinks that this will attract work and the above probabilities change to 0.4, 0.5 and 0.1 respectively.

To buy the crane costs £280 000 but after five years its resale value will be £70 000. Maintenance and operating costs for his own crane will be £50 000/year for a high work load, £40 000/year for a medium and £30 000/year for a low work load. If he buys the crane he can hire it out on short-term hire when he is not using it himself. Each such rental will bring him a net profit of £40 000 but he estimates there is only a 0.3 chance that a renter will be available when the crane is not working. He estimates that the crane will be available once for short-time hire if his work-load is high twice if it is medium and three times if it is low.

On long-term rental for five years the rental cost will be £50 000/year with operating costs of £20 000, £17 000 and £15 000/year for high, medium and low work-loads respectively. There is no resale value nor sub-rental possibilities on long-term lease.

Short-term rental of a crane costs £40 000 per hiring plus operating costs of £10 000 per hiring. The contractor estimates that a high work-load would entail 6 hirings, a medium load 4 hirings and a low work-load 2 hirings. The probability that a crane will be available for hire when needed is 0.4.

Over a five-year period he estimates that the benefits of using a mobile crane will be £100 000/year for a high work load, £50 000/year for a medium and £20 000/year for a low work-load.

The contractor is aware that during this five-year period a very large contract will be let to start construction at the start of year six. A crane either owned or on long-term lease is an essential prerequisite to bidding for this contract. If he owns his crane and he gets the contract he must forgo his resale value but can expect to make £500 000 profit. If he is long-term leasing and gets the contract he must renew his lease at a cost of £150 000 to make this profit. It will cost him £20 000 to bid whether or not he is successful but he feels that there is a 0.5 chance of success. What course of action should the contractor take?

10.4 A roofing company has offered its work-force a 10 per cent pay rise but this offer has been rejected and a strike threatened unless a 15 per cent pay rise is granted. To pay 15 per cent instead of 10 per cent means that the unit cost of roofing must rise from £1.30/m^2 to £1.48/m^2.

A large roofing contract is about to be let by the government involving 1 000 000 m^2 of roofing. The contract will be awarded on the basis of the lowest cost/m^2. A bid cannot be made if the work-force is on strike. Management are certain that the strike will take place unless the 15 per cent rise is awarded and estimate that they will lose £100 000 as a result of the strike. Management are also certain that the strike will eventually collapse but are unsure whether the work-force will return to work with the 10 per cent rise before the contract bid is due. It is estimated that the probability of the strike ending before the due date is 0.7, after the due date, 0.3.

www.EngineeringBooksPDF.com

Possible contract bids and management estimates of the probabilities that these bid prices will secure the contract are given below. What should the company do?

Bid price £/m^2	Probability of getting contract
1.50	0.8
1.55	0.7
1.60	0.5
1.65	0.3
1.70	0.1

10.5 A contractor operates six identical bulldozers on a construction site. Probabilities of breakdown of a single bulldozer are as given below.

Months since maintenance	Breakdown probability in preceding month
1	0.1
2	0.2
3	0.3
4	0.4

The cost of repairing a bulldozer after in-service breakdown averages £300 per repair. The cost of regular overhaul and maintenance during off-shift hours averages £100 per bulldozer. Evaluate and compare corrective and preventive maintenance policies and select the best policy.

10.6 What is the cost of a preventive maintenance policy for bulldozers in exercise **10.5** if only those bulldozers that have not suffered breakdown in the previous period between overhauls are maintained during off-shift hours?

10.7 In the network shown in figure 10.12 each of the six components has a reliability of 0.9. It is proposed to add two new components BF and EC as shown by the broken lines. If these new components both have the same reliability, R, what value should R have so that the reliability of the whole system is 0.99?

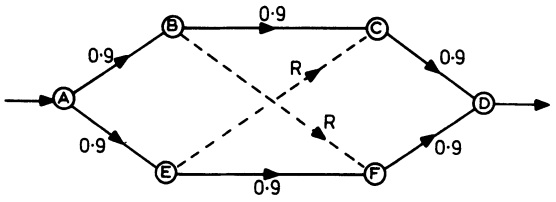


Figure 10.12

www.EngineeringBooksPDF.com

10.8 A contractor has gathered together plant for the purposes of making a large concrete pour. Concrete is produced in a mixer capable of making 500 m^3 in an 8-hour shift. Its probability of failure is 0.15 in any shift. In the event of mixer breakdown there is an older standby mixer capable of producing 350 m^3/shift but with a failure probability of 0.3/shift. This is only used when the first mixer fails.

Concrete from the mixer is carried to the site of the pour by a fleet of ten small dumpers each capable of moving 62 m^3/shift. Each dumper breaks down on average once every five shifts. Concrete from the dumpers is directed into the pour by means of a tower crane and concrete hopper. This crane can handle 600 m^3/shift and its breakdown rate is 0.05/shift. What volume of concrete can this system be expected to pour during a typical 8-hour shift?

www.EngineeringBooksPDF.com

SOLUTIONS TO EXERCISES

CHAPTER 3

3.1 $x_1^* = 4, x_2^* = 1; f^*(\max) = 18$

3.2 $x_1^* = 8, x_2^* = 10; f^*(\min) = 38$

3.3 $x_1^* = 1/3, x_2^* = 4/3; f^*(\min) = 3$

3.4 $x_1^* = 4, x_2^* = 4/3, x_3^* = -1; f^*(\max) = 22/3$

3.5 $x_1^* = 0, x_2^* = -10, x_3^* = -6, x_4^* = 0; f^*(\min) = -14$

3.6 $x_1^* = 7.5, x_2^* = 2.5, x_3^* = 0; f^*(\max) = 17.5; 1 \leqslant c \leqslant \infty; f^*(\max) = 7.5c + 2.5$

3.7 Minimise $f = 5x_1 + 5x_2$

3.8 Production policy: 35 units of type 1, 55 units of type 2, 10 units of type 3; gross profit = £325. Policy remains optimal for new wage rate; company's net profit reduced from £175 to £55 if deal is agreed

3.9 x_{ijk} = m^3 of aggregate size k transported from source j to plant i
Profit = £27 750 − C^* where C^* is found from

Minimise $C = 9x_{ACL} + 10x_{BCL} + 8.5x_{ACM} + 9.5x_{BCM} + 10.3x_{ADL} + 10.8x_{BDL} + 11x_{ADM} + 11.5x_{BDM} + 11.5x_{ADF} + 12x_{BDF} + 12.5x_{AEM} + 12.5x_{BEM} + 11.5x_{AEF} + 11x_{BEF}$

www.EngineeringBooksPDF.com

Subject to the constraints

$$\left.\begin{aligned} x_{ACL} + x_{ADL} &\leqslant 180 \\ x_{ACM} + x_{ADM} + x_{AEM} &\leqslant 210 \\ x_{ADF} + x_{AEF} &\leqslant 120 \end{aligned}\right\} \text{production at A}$$

$$\left.\begin{aligned} x_{BCL} + x_{BDL} &\leqslant 225 \\ x_{BCM} + x_{BDM} + x_{BEM} &\leqslant 262.5 \\ x_{BDF} + x_{BEF} &\leqslant 150 \end{aligned}\right\} \text{production at B}$$

$$\left.\begin{aligned} x_{ACL} + x_{ADL} + x_{BCL} + x_{BDL} &= 300 \\ x_{ACM} + x_{ADM} + x_{AEM} + x_{BCM} + x_{BDM} + x_{BEM} &= 350 \\ x_{ADF} + x_{AEF} + x_{BDF} + x_{BEF} &= 150 \end{aligned}\right\} \text{mix grading}$$

$$\left.\begin{aligned} x_{ACL} + x_{BCL} &\leqslant 200 \\ x_{ACM} + x_{BCM} &\leqslant 150 \\ x_{ADL} + x_{BDL} &\leqslant 125 \\ x_{ADM} + x_{BDM} &\leqslant 150 \\ x_{ADF} + x_{BDF} &\leqslant 100 \\ x_{AEM} + x_{BEM} &\leqslant 150 \\ x_{AEF} + x_{BEF} &\leqslant 150 \end{aligned}\right\} \text{supply of aggregate}$$

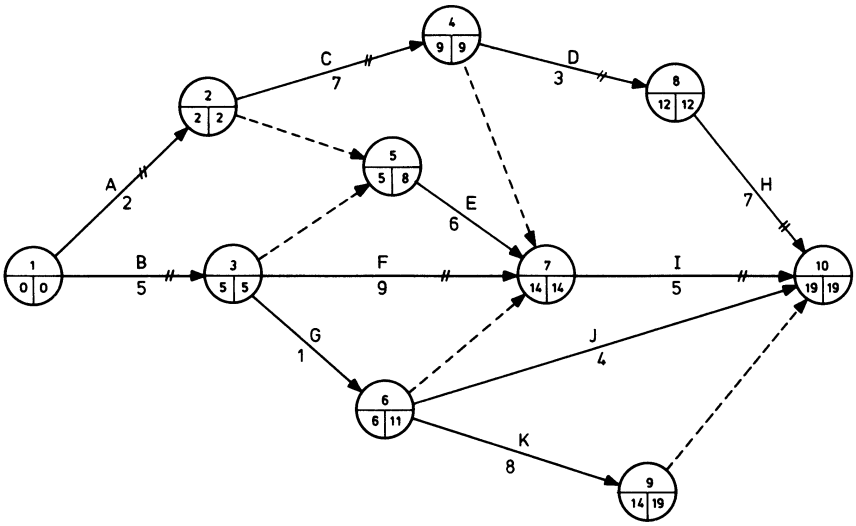
$$\text{All } x_{ijk} \geqslant 0$$

CHAPTER 4

4.1 Non-critical activity floats are

Activity	*Total float*	*Free float*	*Independent float*
E	3	3	0
G	5	0	0
J	9	9	4
K	5	0	0

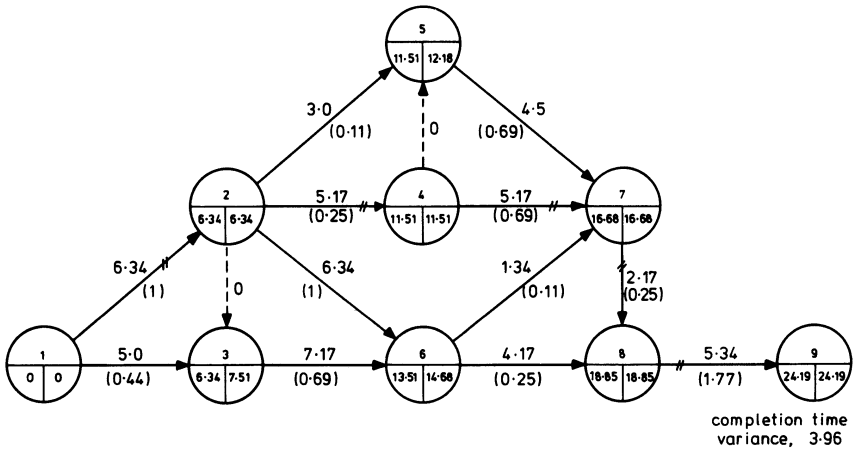
www.EngineeringBooksPDF.com



4.2 23 days; no effect on completion time; time increases to 26 days unless excavator is converted in which case completion time is 24 days

4.3 Steelwork for 5–7 at end of month 15 followed by that for 6–7 and 6–8 at end of month 16; (project duration 26 months instead of 27 months for other alternative)

4.4



www.EngineeringBooksPDF.com

4.5

Event number	*Earliest*	*Latest*
1	0	0
2	2	3
3	6	8
4	7	7
5	12	12

4.6 Longest, 15; shortest 11

4.7 10 flow units

4.8 Flows in arcs 1-3, 3-5 are 4 units; flows in arcs 1-4, 4-5 are 3 units; flows in all other arcs are zero; cost = 302

4.9 44 units

4.10 Cheapest land line system costs 82 units

CHAPTER 5

5.1 Optimal policy is to save: A , 0 weeks; B , 1 week; C , 3 weeks; D , 0 weeks; least extra cost = £39 500

5.2 Optimal allocation: year 1 , 3 buses; year 2 , 3 buses; year 3 , 0 buses; maximum profit = 110.25 units

For 5 buses allocation is: year 1 , 2 buses; year 2 , 3 buses; year 3 , 0 buses; maximum profit = 105 units

5.3 Optimal allocation: job 1 , 4 men; job 2 , 3 men; job 3 , 3 men; benefit = 19.528 units; allocation for 8 men: job 1 , 3 men; job 2 , 2 men; job 3 , 3 men; benefit = 16.083 units

5.4 $x_1^* = 5, x_2^* = 1; f^* = 21$

5.5 (a) Production is 4, 4, 4, 2, 4. Least cost = £10 100
(b) Production is 1, 4, 4, 2, 4. Least cost = £ 8100
(c) Production is 1, 2, 4, 2, 4. Least cost = £ 6900
(d) Production is 1, 2, 1, 2, 4. Least cost = £ 5300

5.6 Activity 1-2 reduces to 5 weeks duration; activity 4-5 reduces to 4 weeks; activity 7-8 reduces to 7 weeks; least total cost = £900

www.EngineeringBooksPDF.com

CHAPTER 7

7.1 $x_1 = 0.09357, x_2 = -0.04068; f = 0.01238$

7.2 $x_1 = 0.41740, x_2 = 0.36256; f = 0.71182$

7.3 Depth = 4 m; diameter = 17.841 m; cost = £6242

7.4 $11.2 \leqslant D^* \leqslant 13.3$ (metre units)

7.5 $12.2 \leqslant D^* \leqslant 12.3$ (metre units); cost $\leqslant$ £24 055

7.6 10 trials bracket $2.55 \leqslant x^* \leqslant 5.11$; solution is $2.99 \leqslant x^* \leqslant 3.00; f^* \leqslant 133$ requiring 8 more trials

7.7 Trial 6 is on steepest gradient direction; fit quadratic and place next trial at (0.73, 16.16, 4.19, 119.83); value of f estimated by quadratic is $f = 89.46$

7.8 $\lambda^* = 0.7266; x_1^* = 0.7578, x_2^* = 1.2422; f^* = -3.5879$

CHAPTER 8

8.1 $x_1^* = 1.07576, x_2^* = 0.62121; f^* = -4.37879$

8.2 $x_1^* = x_2^* = x_3^* = 1/3; f^* = 1/3$

8.3 $x_1^* = 4, x_2^* = 2; f^* = 18$

8.4 $x_1^* = 5, x_2^* = 3.75; f^* = 53.125$

8.5 $\alpha = 1.5; f^* = 6.5$

8.6 (a) $x_1^* = 1, x_2^* = 1.25743; f^* = 15.9054$
(b) $x_1^* = 2.02141, x_2^* = 1.34761, x_3^* = 0.67380; f^* = 136.204$
(c) $x_1^* = 0.88458, x_2^* = 0.78249; f^* = 9.45033$

8.7 $x_1^* = 0.59252, x_2^* = 1.08178, x_3^* = 8.11336; f^* = 47.4733$

8.8 $x_1^* = 0.42529, x_2^* = 0.48038, x_3^* = 0.69231; f^* = 30.7071$

8.9 Areas of members AB, BC, CD, DE are all 952.4 mm^2; areas of members AC and CE are 824.8 mm^2; area of member BD is 1649.6 mm^2; truss volume = 19.048×10^6 mm^3.

www.EngineeringBooksPDF.com

8.10 Use 12 tanks, diameter 25.15 m, cylinder length 150.92 m; total cost = £3.218 x 10^6

8.11 $\phi = \pi/3$; b = 6.204 m; d = 5.373 m

8.12 b = 99.7 m; d = 145.9 mm; bar volume = 0.07277 m^3

CHAPTER 10

10.1 21.28 N/mm^2

10.2 (a) $\lambda = 0.4$
(b) 0.04
(c) 25 days

10.3 Long-term lease and bid on contract; expected profit = £150 000; expected profits by buying are £125 400; by short-term hire are £34 000

10.4 Pay the 15 per cent rise and bid £1.60/m^2 for the contract; expected profit = £45 000; expected maximum profit with strike is only £22 500

10.5 Corrective: £720/month; preventive: £579/month for a two-month maintenance interval

10.6 £470.60/month for a three-month maintenance interval

10.7 $R = 0.779$

10.8 461.25 m^3 governed by the mixers

www.EngineeringBooksPDF.com

INDEX

Activities *see* Construction planning networks
Arithmetic-geometric mean inequality 273
Artificial variables 53–8

Backwards solution of DP problems 150–1, 157, 159
Bayes' principle 325
Bellman's principle 139
Binary LP 70
Bracketing in unconstrained optimisation 192–3, 214–6
Branched serial systems 157–9

Cauchy's inequality 273
Characteristic value 312
Concave function 180–3
Conjugacy 209–12, 223, 231–2
Constrained non-linear optimisation *see* Non-linear optimisation (constrained)
Construction planning networks 82–105
  activities (defined) 84, 86, 92, 95
  activity float 93–6, 99–100
  allocation of activity durations 89, 101–2
  critical path (defined) 93, 95–6, 129
  critical path method 83, 93, 125, 129
  drawing a network 83–8
  events (defined) 84, 86, 89, 92
  event numbering algorithm 88, 90
  event slack 93
  event time algorithms 89–92, 98–100, 109
  network analysis table 95
  PERT 96–7
  project control by network 104–5, 125
  resource allocation 102–4, 125
  time–cost optimisation 101–2
Construction phase of a project 4, 7–8, 17–22, 75, 78, 82–105, 129–48, 325–43
Construction phase examples
  allocation of a crane 144–8
  contract tendering 325–40
  critical path problems 82–105, 129–43
  earthworks 17–22, 29, 40–1, 67, 73–4
  plant maintenance 340–3
Continuous DP 161–3
Convexity 167, 179–183, 185
Cost modelling 13, 20–21, 24, 28, 77, 101–2, 169–70, 267, 307–8, 352–5
Critical path *see* Construction planning networks
Cubic fitting 205–6
Curve fitting in non-linear optimisation 203–6, 233

Decision trees 329–40
Degree of difficulty 271
Density function 318–9
Design phase of a project 4, 6–7, 17, 25–9, 75, 78, 82–3, 167–76, 291–307, 343–55
Design phase examples
  beam design 171–4, 297–301
  drainage design 152–9
  macro-design 6–7, 167, 291, 296–7, 301–7
  micro-design 6–7, 167, 291, 297–301
  pipe selection 174–5
  pumped pipeline 292–7
  reliability design 343–52
  rigid plastic design 17, 25–9, 70–1, 73, 75, 306
  storage tank 175–6
  truss and frame structures 301–7
DFP method 231–3
Directed networks 108–19
  circuits 88, 110–1
  maximum flow problems 112–5
  minimum cost flow problems 115–6
  path problems 108–11, 125, 129–30, 134

www.EngineeringBooksPDF.com

Directed network examples
airport terminal planning 118
sewage treatment planning 116–7
traffic planning 117
Direct search methods *see* Non-linear optimisation
Discrete random variables *see* Random variables
Discrete-valued variables 16, 38, 70–3, 78, 129, 189, 233
Distribution function 319–320
Dual simplex method 66
Duality
in LP 38, 63–6, 78
in GP 275–81
Dynamic programming 129–63
Bellman's principle of optimality 139
branched serial systems 157–9
continuous DP 161–3
decision variables 136
efficiency of DP 159–61
explicit enumeration 159
implicit enumeration 160
multiple state variables 161
optimal control 151
recurrence relationship 139, 158, 162
return function 136, 139, 158, 162
reversal of direction of solution 150–1, 157, 159
serial system representation 136
stage (defined) 135
state variable (defined) 136
traceback method 135, 157
transition function 136, 158, 162
Dynamic programming examples
critical path 129–43
drainage design 152–9
pumped pipeline 296–7
purification process 148–52, 156
tower crane allocation 144–8

Event, in construction planning *see* Construction planning networks
Event, probabilistic 315–7, 328
Expected value 96, 313, 320–1, 325–43
Explicit enumeration *see* Dynamic programming
Exponential distribution 325, 356
Exterior penalty function method 256

Fail-safe systems 348–53
Feasibility studies 5, 291–6
Feasible directions method 265
Feasible points, regions 32–7, 177
Fibonacci method 196–204, 206, 233
First-order methods *see* Non-linear optimisation (unconstrained)
Fletcher–Reeves method 223–4, 231–3
Float *see* Construction planning networks
Flow problems in networks, graphs 112–6

Gaussian distribution 322–4
Geometric programming 265–86
arithmetic–geometric mean inequality 273
Cauchy's inequality 273
constrained posynomial GP 278–84, 293–5
degree of difficulty (defined) 271
generalised non-posynomial GP 285–6
positive degree of difficulty problems 271–5, 282–4, 293–5
posynomial (defined) 266
primal–dual forms 275–6, 279–81
sequential GP 263–4
unconstrained posynomial GP 266–78
Global optima 179
Graph problems 119–25
postman problem 120–1
salesman problem 121–2
spanning trees 122–4
Graphical representation of 2-variable problems 31–5, 167, 176–8, 182–3
Graphical solution of LP problems 31–5
Grid search 191–2, 206

Hamiltonian circuit 121
Hessian matrix 188–9

Implicit enumeration *see* Dynamic programming
Independent events, probabilistic 317
Infeasible points, regions 32–7, 177
Integer LP problems 70–3
Integer valued variables 16, 38, 70–3, 78, 129, 189
Interior penalty function method 253–6

Joint probabilities 317

Kuhn–Tucker conditions 247

Lagrangian methods 237, 241–9
equality constraints 241–5, 286
inequality constraints 246–9, 286
Kuhn–Tucker conditions 247
Lagrange multipliers 241
Lagrangian function (defined) 241–2
Linearising methods for non-linear optimisation 257–63
Linear programming 15–81
artificial variables 53–8
binary LP 70
duality 38, 63–6, 78
symmetric dual problem (defined) 63–4
use in sensitivity analysis 66, 78

www.EngineeringBooksPDF.com

dual simplex method 66
feasible points, regions 31–7
flowchart of simplex method 51–2
graphical solution of LP problems 31–5
infeasible points, regions 32–7
integer LP 70–3
mixed integer LP 70
negative variables 67–9, 78
non-negativity requirement 31
phase I method 52–8, 65
phase II 53
pivoting 46–7, 54–8
revised simplex method 66
rounding of LP solutions 71–3, 78
sensitivity analysis 58–63, 77–8
constraint right-hand sides 58, 60–3, 78
objective function coefficients 58–60, 78
practical uses 62–3, 77
use of duality in sensitivity analysis 66, 78
simplex method 17, 37, 38–58
slack variables 39
transportation problem 67, 73–4, 307–8
Linear programming examples
earthworks 17–22, 73–4
network problems 97–102, 110, 112–6
precasting plant 22–5, 74
rigid plastic design of a frame 25–9, 70–1, 73, 75
water resource management 75–7
Line minimisation methods 191–206, 214–6
bracketing the minimum 192–3, 214–5
cubic fitting 205–6
Fibonacci method 196–203, 204, 206, 233
grid and random search 191–2, 206
interval reduction methods 194–6, 215–6
quadratic fitting 203–5
Local optima 167, 179

Macro-design 6–7, 167, 291, 296–7, 301–7
Mean value 312, 320–1, 325
Micro-design 6–7, 167, 291, 297–301
Minimax strategy 196
Mixed integer LP 70
Modified Newton–Raphson method 228–30, 233
Multi-component system reliability 347–52
Mutual exclusivity 317, 328

Negative variables in LP 67–9, 78
Networks 82–125
Newton–Raphson method 225–30
Non-linear optimisation (unconstrained) 186–234
classical differential methods 186–90
Hessian matrix 188
optimality check 188–9
first-order methods 213–24, 232
conjugacy 223–4
Fletcher–Reeves method 223–4, 231–3
line minimisations 214–6
numerical derivatives 213–4
steepest gradient method 217–22
second-order methods 224–33
DFP method 231–3
modified Newton–Raphson method 228–30, 233
Newton–Raphson method 225–30
quasi-Newton methods 230–2
zeroth-order methods 186, 190–213
conjugacy 209–12, 223
grid and random search 191–2, 206
line minimising methods 191–206
non-linear simplex method 212–3, 233
pattern direction searches 208–9, 223
Powell's method 209–12, 223–4, 233, 255
sequential line minimisations 206–8
Non-linear optimisation (constrained) 237–87
constraint substitutions 238–9
constraint trial deletions 239–41, 286
direct constrained search 263–5, 286
feasible directions method 265
geometric programming 265–86
Lagrangian methods 241–9
linearisation methods 257–63, 264, 286
sequential GP 263–4
sequential LP 257–61, 286, 306–7
sequential QP 261–3, 307
normalisation of constraints 255
penalty functions 237, 249–56, 286, 301
equality constraints 250–2
exterior penalty function 256
inequality constraints 252–6
interior penalty function 253–6
SUMT 254

www.EngineeringBooksPDF.com

Non-linear optimisation examples
beam design 171–4, 297–301
pipe selection 174–5
pumped pipeline 292–7
storage tank 175–6
truss and frame structures 301–7
Non-negativity in LP 31
Normal distribution 322–4
Normalisation of constraints 255
Numerical derivatives 213–4

Objective functions 13–14
Operation phase of a project 4, 8–9, 17, 22–25, 75, 78, 116, 168, 307
Operation phase examples
maintenance of plant 340–3
precasting plant 17, 22–5, 29, 74
purification process 148–52, 156
water resource management 75–7
Operations research 3
Optimal control 151
Optimisation processes
formal 10, 15
informal 10

Path problems in networks 88–96, 98–100, 108–11, 119–25, 130–42
Pattern direction 208–9, 223
Peak value 312
Penalty functions *see* Non-linear optimisation (constrained)
PERT 96–7
Phase I method *see* Linear programming
Phases of a project 4–11
Pivoting *see* Linear programming
Planning networks 82–105
Planning phase of a project 4–6, 74–5, 78, 82–3, 116, 168, 292, 307
Planning phase examples
airport terminal facilities 118
pumped pipeline 292–7
sewage treatment 116–7
tendering 325–40
traffic planning 117
Posterior probabilities 334–6
Postman problem 120–1
Posynomial (defined) 266
Powell's method 209–12, 223–4, 233, 255
Primal–dual forms
GP 275–6, 279–81
LP 63–6
Prior probabilities 334–336
Probabilistic quantity (defined) 311
Probabilistic decision-making methods
Bayes' principle 325
decision trees 329–40
expected value criteria 320–1, 325–43
posterior probabilities 334–6
prior probabilities 334–6
utility functions 338–40
see also Reliability, Random variables
Programme (defined) 15
Project control by network 102–5
Project phases 4–11

Quadratic curve fitting 203–5
Quadratic programming 261–4, 307
Quasi-Newton methods 230–2

Random variables 315–25
Bayes' principle 325
continuous random variables 318–25
density functions 318–9
discrete random variables 315–8
distribution functions 319–20
event probability 315–6
expected value 320–1, 325–43
exponential function 325
frequency interpretation 316
independent events 317
joint probabilities 317
mutual exclusivity 317, 328
normal (Gaussian) distribution 322–4
standard deviation 321
uniform distribution 322
variance 321
Recurrence relationship of DP 139, 158, 162
Reliability 343–55
cost–reliability models 352–5
multi-component systems 347–52
fail-safe 348–52
weakest link 347–8, 350–2
single component reliability 343–7
Resource allocation 74, 102–4, 125
Return function in DP 136, 139, 158, 162
Revised simplex method 66
Risk 313–14
Roll-back of decision trees 331–3, 336–7
Rounded discrete solutions 71–3, 78, 233

Saddle point 229
Salesman problem 121–2
Scientific method 11
Second-order methods *see* Non-linear optimisation (unconstrained)
Sensitivity analysis *see* Linear programming
Sequential GP 263–4
Sequential LP 257–61, 286, 306–7
Sequential QP 261–3, 307
Serial systems 129–63
Serial system (defined) 136
Serial system reliability 347–8, 350–2

www.EngineeringBooksPDF.com

Shortest paths 82, 106, 108–11, 116–9, 120–5, 134–5
Simplex method (LP) *see* Linear programming
Simplex method (non-linear) 212–13, 233
Slack variables 39, 246–7
Spanning trees 122–4
Stage (DP) 135
Standard deviation 321
State variable (DP) 136
Steepest gradient method 217–22
SUMT 254
Symmetric duality 63–4
Synthesis 1, 168
System (defined) 2
Systematic (defined) 2
Systematic decision-making method 11–14, 18, 20, 21, 23, 25, 29, 37, 168, 185
Systems engineering 3

Tendering 325–40
Traceback in DP 135, 157–9
Transition function in DP 136, 158, 162
Transportation problem 18, 67, 73–4, 307–8
Trees 122–4, 329–40

Unconstrained problems *see* Non-linear optimisation (unconstrained)
Undirected networks 119–25
  postman problem 120–1
  salesman problem 121–2
  trees 122–4
Uniform distribution 322
Unimodal function 180
Utility functions 338–40

Vertex, of constraints 34–7, 176–8

Weakest-link systems 347–8, 350–2

Zeroth-order methods *see* Non-linear optimisation (unconstrained)

www.EngineeringBooksPDF.com