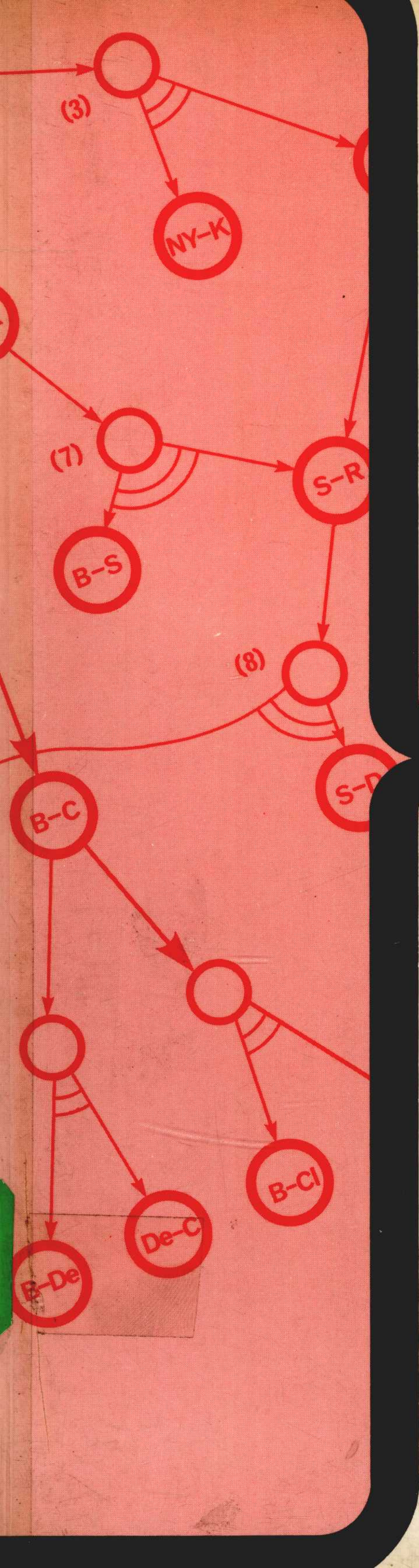


Graph Theory in Operations Research

T B Boffey

Graph Theory in Operations Research

T. B. Boffey
University of Liverpool

M

First published 1982
Reprinted (with corrections) 1984

Published by
THE MACMILLAN PRESS LTD
London and Basingstoke
Companies and representatives
throughout the world

Typeset in 10/12 Press Roman by
CAMBRIAN TYPESETTERS
Farnborough, Hants

Printed in Hong Kong

ISBN 0 333 28213 2
ISBN 0 333 28214 0 pbk

Macmillan Computer Science Series

Consulting Editor
Professor F. H. Sumner, University of Manchester

S. T. Allworth, *Introduction to Real-time Software Design*

Ian O. Angell, *A Practical Introduction to Computer Graphics*

G. M. Birtwistle, *Discrete Event Modelling on Simula*

T. B. Boffey, *Graph Theory in Operations Research*

Richard Bornat, *Understanding and Writing Compilers*

J. K. Buckle, *The ICL 2900 Series*

Derek Coleman, *A Structured Programming Approach to Data**

Andrew J. T. Colin, *Fundamentals of Computer Science*

Andrew J. T. Colin, *Programming and Problem-solving in Algol 68**

S. M. Deen, *Fundamentals of Data Base Systems**

J. B. Gosling, *Design of Arithmetic Units for Digital Computers*

David Hopkin and Barbara Moss, *Automata**

Roger Hutty, *Fortran for Students*

H. Kopetz, *Software Reliability*

A. M. Lister, *Fundamentals of Operating Systems, second edition**

G. P. McKeown and V. J. Rayward-Smith, *Mathematics for Computing*

Brian Meek, *Fortran, PL/I and the Algols*

Derrick Morris and Roland N. Ibbett, *The MU5 Computer System*

John Race, *Case Studies in Systems Analysis*

B. S. Walker, *Understanding Microprocessors*

I. R. Wilson and A. M. Addyman, *A Practical Introduction to Pascal*

* The titles marked with an asterisk were prepared during the Consulting Editorship of Professor J. S. Rohl, University of Western Australia.

Preface

Graphs and networks are used as models in many fields of study as shown by the recent collection of articles edited by Wilson and Beineke (1979). This book is concerned with the application of graphs and networks to some of the problems that arise in industry, local government, transport and other areas. The term 'operations research' (to be interpreted in a wide sense) is adopted in the title to indicate the area from which topics are selected.

Books have been published for a number of years on applied graph theory. The field has grown to the extent that books devoted to applications in particular subjects are fully justified. Indeed I found that, within the size limitation imposed, I had to be selective as to the OR topics to be covered. The final choice is a personal one and reflects my interests and knowledge; no doubt others would have come up with a different selection.

The text grew out of a course that has been given for several years to final-year undergraduates in operations research and computing. It is designed primarily for courses in OR, but it will also be of interest to those on computing and transport studies courses as well. It is also hoped that the practising operations researcher will find it of value, and effort has been devoted to providing realistic examples (though necessarily scaled down in size).

The background mathematical knowledge required has purposely been kept to a minimum in order to make the material accessible to a wider readership. The principal prerequisites are a working knowledge of set theoretic notation and matrices, and exposure to a first course on linear programming. A prior knowledge of duality concepts would be helpful but is not necessary, the required dual results being developed as they are needed (section 10.3 is an exception to this general rule).

It has been convenient to adopt a variety of abbreviations. Some are general, some are names of algorithms (usually derived from the originators' names) and some (such as SCP for the set covering problem) are names of standard problems. Lists of these abbreviations are given on p. ix. The reader should consult these lists if ever there is any doubt as to the meaning of a particular abbreviation.

During the preparation of this book I have greatly benefited by the advice and comments of colleagues, and it is a pleasure to acknowledge this indebtedness. I would specifically like to mention Donald Davison, Steve Filbin, Chris Pursglove, Graham Rand, Grahame Settle and Derek Yates. I would also like to express my gratitude to Iain Buchanan who, as referee, made many

helpful suggestions leading to the improvement of the book. Thanks are also due to Mrs Betty Jones and Mrs Pam Billingsly for typing the manuscript and to David Sherratt and Ken Chan for assistance with the GATE text editing system; they were always willing to help. Finally I would like to thank my wife Pamela for her help and support – it is to her and to my parents that the book as a whole is dedicated.

ACKNOWLEDGEMENTS

The diagrams of figure 1.2 are taken, by permission, from the *Operational Research Quarterly* from the articles indicated in the caption. Figure 1.5 is reprinted, by permission, from M. Folie and J. Tiffin, Solution of a multi-product manufacturing and distribution problem, *Management Science*, **23**, 3 (Nov, 1976), © 1976 The Institute of Management Sciences. Several exercises are (usually modified versions of) questions from University of Liverpool examination papers; these are indicated by the letters LU followed by the year in which they were set. Finally, example 6.1 is due to Frank Wharton.

Permission to use the above material is gratefully acknowledged.

Abbreviations

GENERAL

B & B	Branch-and-bound
BF	Breadth-first
CPM	Critical path method
DF	Depth-first
DK(f)	Dinic–Karzanov network
ES (EF)	Earliest start (finish) of an activity
ILP	Integer linear programming problem
LB	Lower bound
L(G)	Arc-to-vertex dual of G
LP	Linear programming problem
LS (LF)	Latest start (finish) of an activity
MILP	Mixed integer linear programming problem
MST	Minimal spanning tree
PD	Project duration
PERT	Project evaluation and review technique
SD-tree	Shortest distance tree

PROBLEMS

AP	(Linear) assignment problem
ATSP	Asymmetric travelling salesman problem
1-CD	1-centdian problem
p-CP	*p*-centre problem
1-CPP	1-centre problem in the plane
CPP	Chinese postman problem
LSCP	Location set covering problem
MSCP	Maximal set covering problem
p-MP	*p*-median problem
1-MPP	1-median problem in the plane
OTSP (s,f)	Open travelling salesman problem (from s to f)
SCP	Set covering problem
SLP	Simple location problem
TRP	Transshipment problem

TSP	Travelling salesman problem
VRP	Vehicle routing problem

ALGORITHMS

AB	To solve AP using alternating bases
AT	To find alternating trees
BF	Breadth-first search subroutine for algorithm (FF)
BKE	To solve SLP (Bilde–Krarup–Erlenkotter)
BP (FP)	Backward pass (forward pass) for CPM
CRASH	For reducing project durations
CW	Savings method for VRP (Clarke–Wright)
D	To find a shortest path (Dijkstra)
Fd (Fd′)	To find a shortest path (Ford)
FF	To find a maximal flow (Ford–Fulkerson)
Fk	To order vertices of a project network (Fulkerson)
Fl	To find all shortest paths (Floyd)
FW	For non-linear minimum cost flow problems (Frank–Wolfe)
G	To find 1-median of a tree (Goldman)
GNR	To find a p-centre (Garfinkel–Neebe–Rao)
HM	Hungarian method to solve AP
K	To find an MST (Kruskal)
L	To level resource profiles
N	To find a shortest path (Nemhauser)
OOK	Out-of-kilter algorithm for least cost flow
P	To find an MST (Prim)
PT	To form a precedence tree
RA	For resource constrained problem
SWEEP	The Gillett–Miller angular approach to VRP
Y	To find the K shortest paths (Yen)

NOTATION

ϕ	The empty set
2^X	Set of all subsets of X
$\{x \mid P\}$	Set of all elements x which satisfy condition P
$\{x\}$	Set containing just the element x
$\lceil\ \rceil$ ($\lfloor\ \rfloor$)	Smallest integer not less than (largest integer not greater than)
$\lvert S \rvert$	Number of elements in set S

Contents

1 Introduction

What is a graph? Consider first the commonly used concept defined in terms of Cartesian coordinates; figure 1.1a shows a graph of a real valued function f of a real continuous variable x where

$$y = f(x) = x^3 - 6x^2 + 9x + 16 \qquad -1\tfrac{1}{4} \leqslant x \leqslant 4\tfrac{3}{4}$$

Figure 1.1b also shows a graph, this time of

$$y = g(x) = 2x \bmod (5) \qquad 0 \leqslant x \leqslant 4$$

In both cases x can take on a (continuous) infinity of values and the graph consists of continuous lines containing an infinite number of points.

Consider again the function g, this time with x restricted to the set $\{0, 1, 2, 3, 4\}$. The graph is now given by the five points marked on figure 1.1c. Of course, since x is restricted to such a small range, all values of $g(x)$ can be listed explicitly

$$g(0) = 0, \quad g(1) = 2, \quad g(2) = 4, \quad g(3) = 1, \quad g(4) = 3$$

and the operation of g can be illustrated as in figures 1.1d and e. These provide alternative ways of showing the graph of g on $\{0, 1, 2, 3, 4\}$ and such representations are feasible when x can take on discrete values only. Such representations, which are akin to the popular concept of a network, will be those of relevance in this book.

The central concept of a graph, as understood in graph theory and throughout this book, is that it is a set of entities called *vertices*, which are interrelated via certain correspondences called *links*. A graph is often thought of as a set of points (representing vertices) with interconnecting lines (representing links) and is frequently pictured in this way. Links may be directed (with directions shown by arrows), in which case they are called *arcs*, or they may be undirected and called *edges*. Graphs are directed (undirected) if their only links are arcs (edges). Graphs can also be pictured in other ways, as shown by the examples in figure 1.2, all of which are taken from an OR journal (*Operational Research Quarterly*).

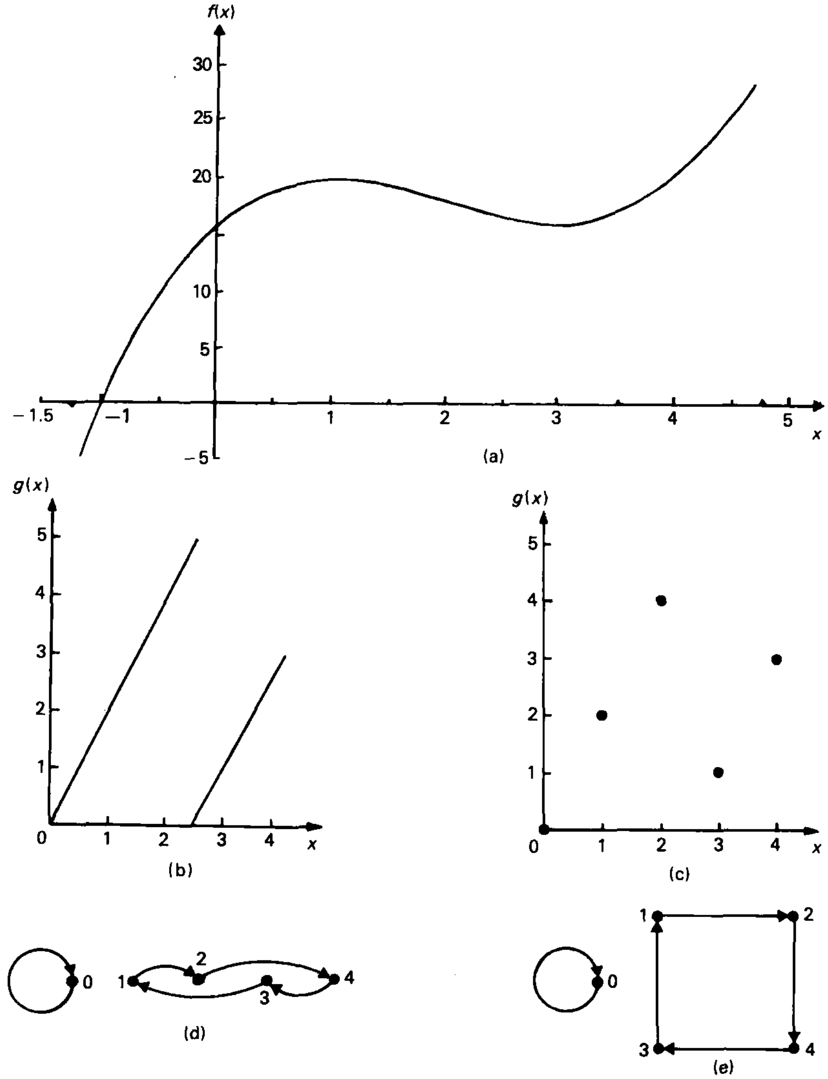


Figure 1.1 (a) Graph of $f(x) = x^3 - 6x^2 + 9x + 16$ for $-1\frac{1}{4} \leqslant x \leqslant 4\frac{3}{4}$; (b) graph of $g(x) = 2x \bmod(5)$ for $0 \leqslant x \leqslant 4$; (c), (d) and (e) give three representations of $g(x)$ restricted to $x \in \{0,1,2,3,4\}$

Example (a) is an idealised diagram of a computer with a network of connections via multiplexers to local input stations. The graph is undirected and no closed curves are formed by the edges; connected graphs with this property are called *trees* (section 2.3). Vertices are represented in three ways – by dots, open circles and a rectangle.

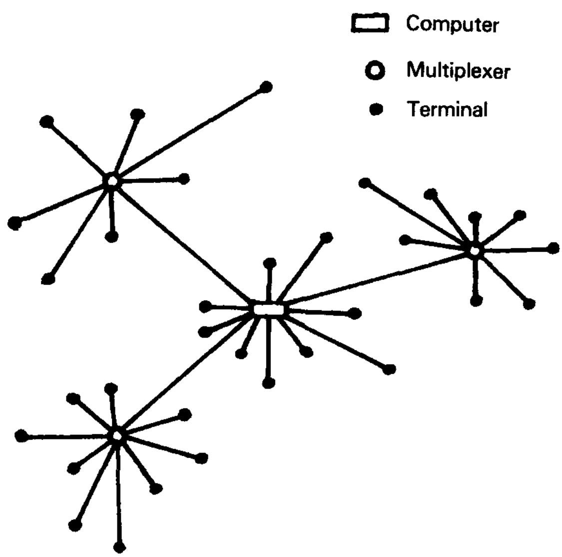
Example (b) gives a flow chart of a queueing theory model of the internal transport system of a steelworks. The graph is directed, the arrows giving the directions of the arcs. Vertices are represented by rectangular and diamond-shaped boxes and with text written inside the boxes providing *labels*, or names, for the vertices.

Example (c) is an adaptation of a decision tree put forward in relation to a decision as to whether or not to expand rev-counter production capacity by installing extra equipment or by putting employees on overtime. Vertices are represented by large dots and rectangles, and are labelled. Extra information is given alongside the vertices.

Example (d) shows the feedback loops of a systems dynamics model of the shipping industry. The graph is directed and the arcs form many closed curves, or circuits. Vertices are identified merely by having incident arcs and by their labels (the items of text). Arcs are shown in this example by curved rather than straight lines.

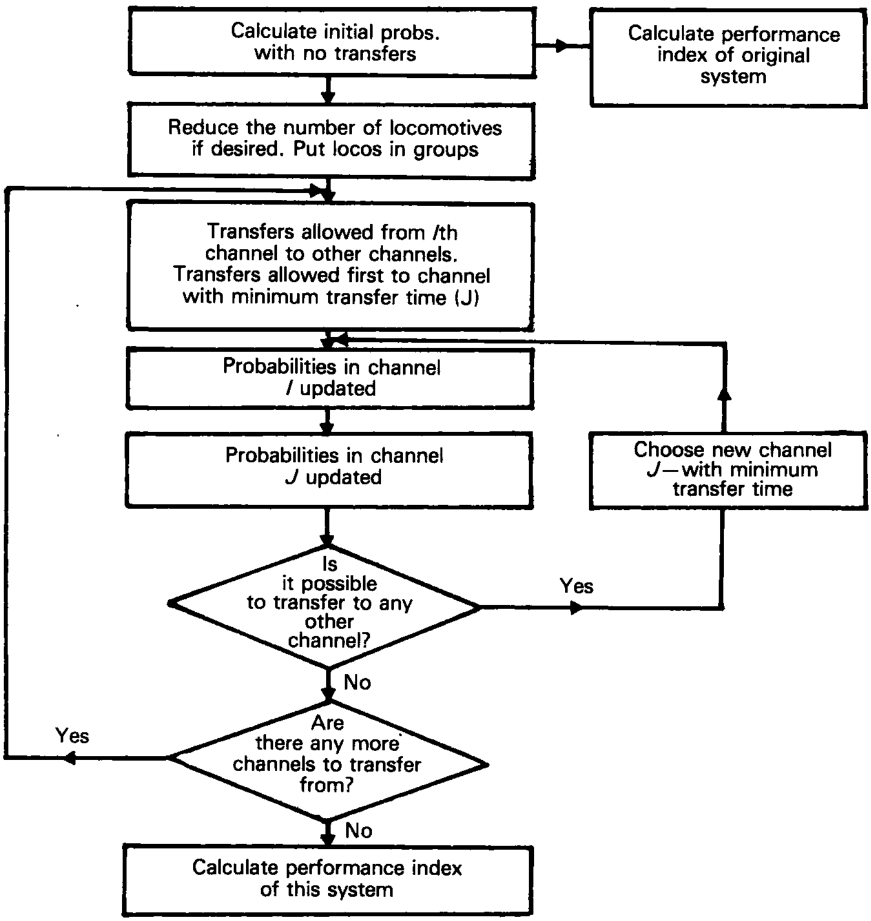
Thus it is seen that pictures of graphs can take a variety of forms. Vertices may be represented in a variety of ways and may or may not be labelled. The disposition of the vertices relative to each other is not a property of graph theory and different placings, although leading to different pictures, represent the same graph. Arcs and edges may be represented by straight or curved lines and may or may not be labelled.

Six problems will now be posed to illustrate the wide range of situations to which graph theory is applicable.



(a)

Figure 1.2 Examples of graphs from: (a) Drinkwater (1977); (b) Corkindale (1975); (c) Moore and Thomas (1973) – modified; (d) Taylor (1976)



(b)

Figure 1.2 *continued*

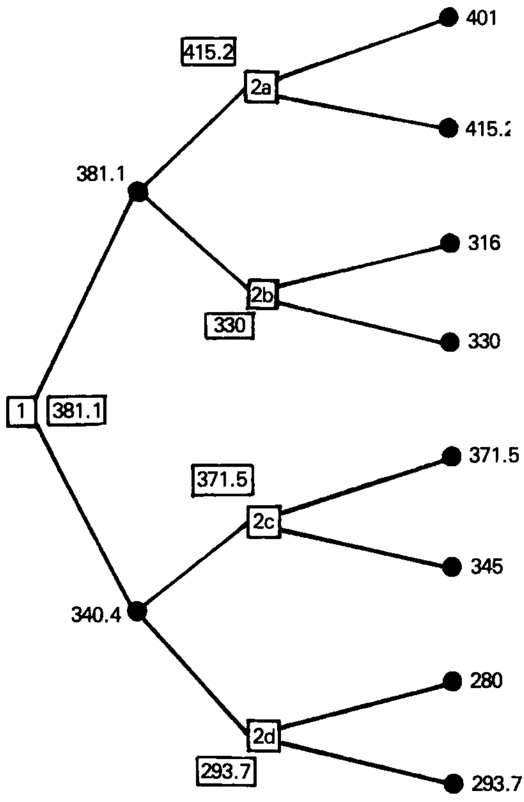


Figure 1.2 *continued*

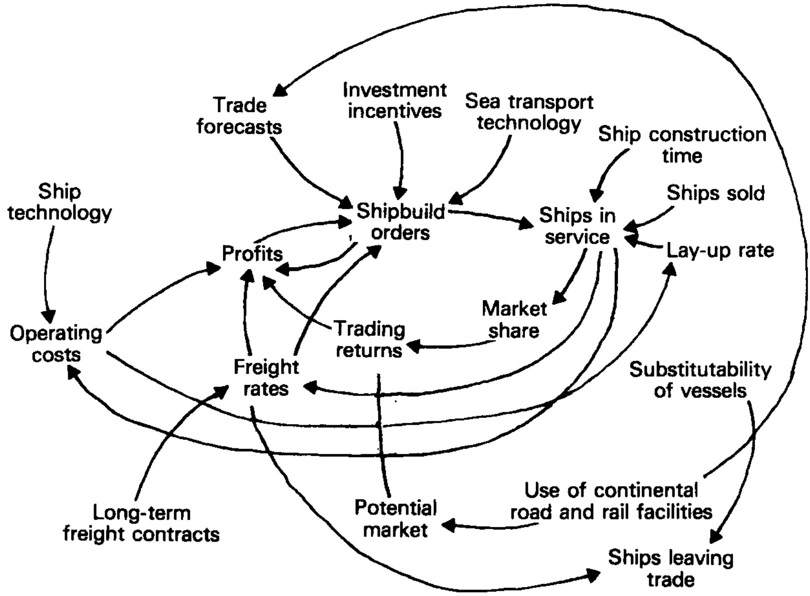


Figure 1.2 *continued*

(1) Layout Planning and Planarity (Seppänen and Moore, 1970)
An architect was asked to design a special-purpose one-storey building incorporating 5 rooms $A, B, \ldots, E$ of specified sizes etc., in such a way that each room has an outside wall and

A is next to B, C and D
B is next to A and C
C is next to B, A and D
D is next to A, C and E
E is next to D

The architect produces the plan shown in figure 1.3a. At this stage the client notices that B is relatively far away from D, and thinking this to be a disadvantage asks whether the plan can be rearranged so that B is next to D also. The architect thinks that this is not possible but is not sure. By showing that the so-called dual graph (section 6.2) of figure 1.3b is non-planar (section 2.1), the impossibility is established.

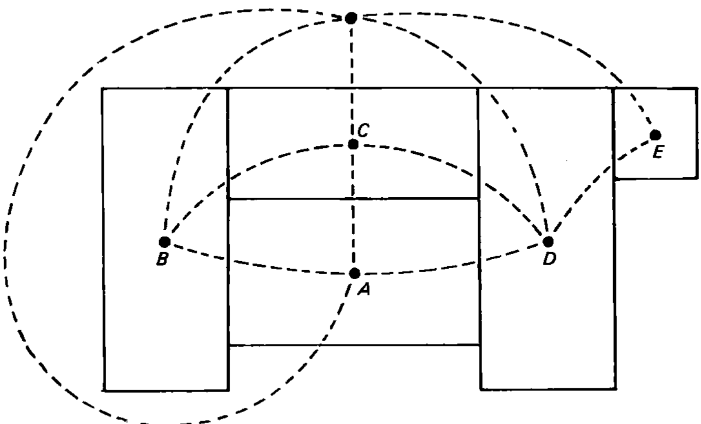
Similar problems arise when assigning work areas on a shop floor to satisfy specified adjacencies (Seppänen and Moore 1970, Francis and White 1974).

(2) Production Planning

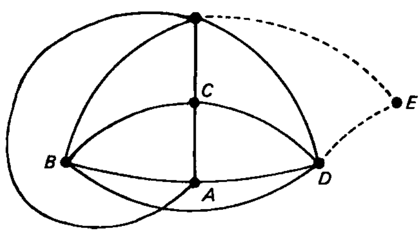
A manufacturing company is asked to supply d_i units of a bulky item at the end of each of the next 6 months, that is at times $i = 1, \ldots, 6$. The company has limited storage capacity for this item which has the effect of restricting the stock on hand, s_i, at the start of each month to at most 5 units and a stock holding cost of 1 is incurred per unit per month stocked. The initial stock is 3 units and it is decided to run the stock down to zero after the present contract has been honoured. Because of variations in the cost of labour, raw materials, etc., it is estimated that the cost p_i of producing a unit in month i is given by

Month	1	2	3	4	5	6
d_i	1	1	0	3	3	4
p_i	11	13	13	12	14	13

If it is possible to produce at most 2 units in any month what should the production policy be?



(a)



(b)

Figure 1.3