

Mathematical Aspects
of
Scheduling and Applications

by

R. BELLMAN

A. O. ESOGBUE

and

I. NABESHIMA

Mathematical Aspects
of
Scheduling and Applications

by

R. BELLMAN

*University of Southern California
Los Angeles, USA*

A. O. ESGHBUE

*Georgia Institute of Technology
Atlanta, USA*

and

I. NABESHIMA

*University of Electro-Communications
Tokyo, Japan*



PERGAMON PRESS

OXFORD · NEW YORK · TORONTO · SYDNEY · PARIS · FRANKFURT

UK	Pergamon Press Ltd., Headington Hill Hall, Oxford OX3 0BW, England
USA	Pergamon Press Inc., Maxwell House, Fairview Park, Elmsford, New York 10523, U.S.A.
CANADA	Pergamon Press Canada Ltd., Suite 104, 150 Consumers Road, Willowdale, Ontario M2J 1P9, Canada
AUSTRALIA	Pergamon Press (Aust.) Pty. Ltd., P.O. Box 544, Potts Point, N.S.W. 2011, Australia
FRANCE	Pergamon Press SARL, 24 rue des Ecoles, 75240 Paris, Cedex 05, France
FEDERAL REPUBLIC OF GERMANY	Pergamon Press GmbH, 6242 Kronberg-Taunus, Hammerweg 6, Federal Republic of Germany

Copyright © 1982 R. Bellman, A. O. Esogbue, I. Nabeshima

All Rights Reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means: electronic, electrostatic, magnetic tape, mechanical, photocopying, recording or otherwise, without permission in writing from the publishers

First edition 1982

Library of Congress Cataloging in Publication Data

Bellman, Richard Ernest, 1920–
Mathematical aspects of scheduling and applications.
(International series in modern applied mathematics
and computer science; v. 4)

Includes bibliographies.

I. Scheduling (Management) I. Esogbue, Augustine O.
II. Nabeshima, Ichiro, 1921– III. Title. IV. Series.
TS157.5.B44 1982 658.5'3 81–15809

British Library Cataloguing in Publication Data

Bellman, R.

Mathematical aspects of scheduling and applications
(International series in modern applied
mathematics and computer science; v. 4)

I. Scheduling (Management)—Mathematics

I. Title II. Esogbue, A. O. III. Nabeshima, I.

IV. Series

658.5'3 TS157.5

ISBN 0-08-026477-8 Hardcover

ISBN 0-08-026476-X Flexicover

Printed in Great Britain by A. Wheaton & Co. Ltd., Exeter

Preface

THROUGHOUT the two major epochs in human history, in both the mechanical age and now the systems age, mankind has been beset with the arduous problem of optimal utilization of the usually limited resources in accomplishing the variegated tasks or objectives. The problem is thus not new although contemporary manifestations of this problem in forms such as the energy crisis or inflation may proffer it that coloration. Further, the problem appears recurrent and eternal. An attempt to cope with such problems demands that one develop some plan or schedule. The analysis and study of such plans particularly, with respect to their optimality and other properties under various scenarios of objectives and constraints, constitutes an exciting field known as Scheduling Theory. This theory thus deals with a concretization of continuing human experience.

The purpose of the book is, however, not a prosaic rendition of history. Rather, it is an attempt to syphon out of both the mechanical age (when man was building machines to analyze and do work) and the systems age (when the synthesis of the total problem and interrelationships are the focus of attention) certain pervasive problems that occur and are important enough to warrant the focus of our attention. We then present the kernel of these problems in mathematical terms and devise mathematical solutions to them. As always, we study their intrinsic properties and, whenever we can, generalize to other situations. Similar objectives appear in numerous works in the field. What we have done that is unique is to integrate our experiences of over two decades of research work in the field, from different vistas of experience with problems and techniques into one compendium.

The book is particularly timely. First, as we said, the problem of scheduling is a perennial one and nearly as old as history. Second, mankind is currently going through a special period in which critical shortages of important resources needed to continue civilization have been flung to the limelight. The plea to conserve is not only a plea to schedule one's consumption. It is more than that. It is also a plea for efficient utilization of available resources including optimal combinations of resources. These are issues of central interest in scheduling theory. Third, the existing books in the field are either considerably outdated or are severely limited in their scope of treatment. Fourth, the problem of optimal systems design and control bears striking interdependence with that of optimal scheduling.

We have attempted to capture the essential problems in scheduling as well

as treat them via the most effective techniques. We have introduced material in this book that are considerably current, novel, and well researched. Sometimes, these techniques and problems have not even appeared in the open literature. Other times, they may have only appeared in the form of technical reports of some research institute or in scientific journals. Further, we have exploited our familiarity with the literature of such fields as mathematics, economics, operations research, management science, computer science, engineering and medicine, to motivate our problems and methods of attack.

This book does not and is not intended to discuss everything there is to know about scheduling. For example, dynamic problems involving stochastic arrival times of jobs are not considered. Inclusion of such topics here would have led to a much lengthier book than this one or to the sacrifice of the depth of coverage of each of the topics in this book. We decided to err on the side of depth than breadth.

This book is intended as a graduate text or reference work in a course usually entitled Scheduling Theory or Control Theory in most universities' departments of mathematics, operations research, management science, computer science or engineering. It will also be useful to economists and planners. It gives an almost didactic treatment of the subject. The central techniques employed can be picked up along the way although previous familiarity with dynamic programming and integer programming is useful particularly with the advanced algorithms. The reader will find the exercises at the end of each chapter instructive and challenging. The comments and bibliography section will be useful in pursuing an in-depth study of the material covered in each chapter.

The book consists of twelve chapters. It begins with network problems—a very special subgroup of mathematical programming problems. The shortest-path problem, one of the fundamental problems in mathematics, is the problem of tracing a path of shortest length through a finite network. This problem occurs in many fields and we devote the first two chapters to it. In Chapter 1, we show how a simple dynamic programming argument yields a fundamental nonlinear equation. Thus, dynamic programming converts a combinatorial problem into an analytic one. Simple approximate techniques yield upper and lower bounds for the solution of this equation. These bounds can be easily interpreted in terms of approximate policies. We also exploit such bounds and other fathoming criteria suggested by branch and bound strategies to solve large-scale traveling-salesman problems. Applications to control theory and other parts of the calculus of variations are also discussed. In Chapter 2 we show that this problem can be used to treat many problems in artificial intelligence and that many popular mathematical games can be interpreted in these terms. The role and use of computers is stressed.

In Chapter 3 we classify scheduling problems and briefly discuss their

solution approaches. The machine scheduling problem is the problem of processing n items on m machines in an efficient manner. The problem is remarkably difficult. We discuss the many methods that can be applied, and give many examples. Among these methods are the branch and bound method, backtrack programming, dynamic programming, and various combinatorial methods.

The only analytic result known is due to Selmer Johnson for the case of two machines. If there are more than two machines, it is quite likely that no simple analytic result exists. However, for the permutation flow-shop problem with makespan objective, there have been many efficient analytic results under some restricting conditions, as described in Chapter 7.

Since these are finite problems, it might be thought they could be handled by enumeration, particularly with the speed of the digital computer. This is not the case, mathematical analysis is definitely needed, to identify and/or extend the solvable cases. Also we require an approximation method with guaranteed accuracy, or with simple efficient heuristics.

To see this, two numbers are convenient to keep in mind. First of all, $10! = 3,628,800$; secondly, a year has approximately 3×10^7 seconds.

Consequently, if each case takes one second to examine, we see that $10!$ cases takes about a month. Since $20!$ is more than $10^{10} \times 10!$, we see that $20!$ cases cannot be examined by enumeration. Even if each case requires only a microsecond, it is not feasible to examine $20!$ cases. In combinatorial problems of this type, it is not uncommon to meet numbers such as $100!$ or $1000!$. Thus mathematical analysis, whenever possible, helps us reduce the computational drain tremendously.

Chapters 4 through 6 discuss different problems and techniques of machine sequencing. Chapter 5 discusses capacity expansion problems and introduces the new technique of imbedded state space dynamic programming for reducing dimensionality so that larger problems can be solved. Chapter 6 considers an important class of network problems with nonserial phase structures and exploits dimensionality reduction techniques such as the pseudo-stage concept, branch compression and optimal order elimination methods to solve large-scale nonlinear network scheduling problems.

In Chapters 7 through 11 we consider the flow-shop scheduling problem under different objectives and constraints. We present several novel analytic results and employ various ingenious techniques of branch and bound including backtrack programming, lexicographical search method and unified multi-stage combinatorial algorithms. Applications to the increasingly important area of parallel processing are discussed. Approximate solutions for these interesting cases are also presented. In Chapter 11 we address machine scheduling problems involving sequence-dependent set-up times and present novel efficient dynamic programming formulations especially for the three-machine problem.

The book is concluded with a chapter on the job-shop-scheduling problem. Using the disjunctive graph viewpoint and the techniques of BAB, EXTBAB, dynamic programming, important analytical and computational results are derived.

What we have attempted in this work is to present a unified treatment of the many problems and techniques of solution. We want to point out how many problems exist in these domains and what opportunities there are. These are new parts of applied mathematics. What is particularly interesting about these problems is that they require a blend of analysis, algebra, topology, computer science, and a knowledge of how the problems arise.

We are grateful to a long list of friends, colleagues, and graduate students who have collaborated with us in various phases of our research efforts. In particular, we acknowledge the contributions of Burton Corwin and Thomas Morin.

RICHARD BELLMAN
Santa Monica, California

AUGUSTINE ESOGBUE
Atlanta, Georgia

ICHIRO NABESHIMA
Chofu-shi, Tokyo, Japan

Contents

1. Network Flow, Shortest Path and Control Problems	1
1.1. Introduction	1
1.2. The Routing Problem	1
1.3. Dynamic Programming Approach	2
1.4. Upper and Lower Bounds	3
1.5. Existence and Uniqueness	5
1.6. Optimal Policy	8
1.7. Approximation in Policy Space	8
1.8. Computational Feasibility	9
1.9. Storage of Algorithms	10
1.10. Alternate Approaches	10
1.11. "Traveling-salesman" Problem	10
1.12. Reducing Dimensionality Via Bounding Strategies	12
1.13. Stochastic Traveling-salesman Problem	14
1.14. Applications to Control Theory	16
1.15. Stratification	16
1.16. Routing and Control Processes	17
1.17. Computational Procedure	18
1.18. Feasibility	19
1.19. Perturbation Technique	20
1.20. Generalized Routing	21
1.21. Discussion	22
Bibliography and Comments	22
2. Applications to Artificial Intelligence and Games	26
2.1. Introduction	26
2.2. An Operational Point of View	26
2.3. Description of a Particular Problem	26
2.4. Imbedding	27
2.5. A Fundamental Equation	28
2.6. Geometric Ideas	29
2.7. Conversion of a Decision Process into an Equation	29
2.8. The Concept of a Solution	29
2.9. Determination of the Solution	30
2.10. Determination of a Number	31
2.11. Decision-making by Computers	31
2.12. Enumeration	32
2.13. Writing a Program	33
2.14. Storage	33
2.15. Dimensionality	34
2.16. Structure	34
2.17. Heuristics	35
2.18. Discussion	35
2.19. Application to Game Playing	36
2.20. Mathematical Abstractions	36
2.21. Intelligent Machines	36
2.22. The Wine-pouring Problem	36

2.23. Formulation as a Multistage Decision Process	37
2.24. Cannibals and Missionaries	39
2.25. Formulation as a Multistage Decision Process	40
2.26. Chinese Fifteen Puzzle	40
2.27. The Puzzle Again	41
2.28. Feasibility	41
2.29. Doable Positions	42
2.30. Associated Questions	42
2.31. The Original Puzzle	42
2.32. Lewis Carroll's Game of Doublets	43
2.33. Chess and Checkers	43
2.34. Solving Puzzles by Computer	43
2.35. Discussion	44
Bibliography and Comments	44
3. Scheduling Problems and Combinatorial Programming	48
3.1. Introduction	48
3.2. Classification of Scheduling Problems	48
3.3. Sequencing Problem	48
3.4. Project Scheduling Problem	50
3.5. Assembly-line Balancing Problem	51
3.6. Mutual Relations	51
3.7. Summary of Combinatorial Programming as a Solution Technique	52
3.8. State Transformation Process	52
3.9. Branch and Bound Method (BAB Method)	54
3.10. Relative Error of Approximate Value and Heuristic Algorithm for Deciding a Reliable Approximate Solution (Extended BAB Method)	56
3.11. Backtrack Programming and Lexicographical Search Method	58
Bibliography and Comments	58
4. The Nature of the Sequencing Problem	61
4.1. Introduction	61
4.2. Assumption and Objective Functions	63
4.3. Objective Function (Measure of Performance)	64
4.4. Objective Functions of the other Type	66
4.5. Classification of Sequences and Non-numerical Judgement	68
4.6. Judgement for Determination	69
4.7. Generation of Feasible Sequences	74
4.8. Potentially Optimal Sequence and Nonoptimal Sequence	74
4.9. Determination of Potentially Optimal Sequences and Nonoptimal Sequences	74
4.10. Classification and Generation of Schedules	76
4.11. Semiactive Schedule and Inadmissible Schedule	77
4.12. Start and Completion Times of Each Operation	78
4.13. Active Schedule and Nonactive Schedule	80
4.14. α -Optimal Schedule and Non- α -optimal Schedule	81
4.15. Generation of Active Schedules	82
Bibliography and Comments	85
5. Sequencing Involving Capacity Expansion	87
5.1. Introduction	87
5.2. A Simple Expansion—Sequencing Problem—the One-dimensional Version	87
5.3. Why Dynamic Programming?	89
5.4. Conventional Dynamic Program (DPI) for the One-dimensional Sequencing Problem	89

5.5. The Imbedded States Space Dynamic Program (DP2) for the One-dimensional Sequencing Problem	90
5.6. Discussion	93
5.7. Formulation	94
5.8. An Illustrative Example	94
5.9. Reducing M&E Imbedded State Space for Large-capacity Expansion Problems	96
5.10. Discussion	98
5.11. Multi-dimensional Sequencing Problem—Theory	99
5.12. A Graphical Illustration	101
5.13. Computational Experience on Real-world Problems	102
5.14. Variations and Extensions	104
5.15. Miscellaneous Exercises	104
Bibliography and Comments	106
6. Sequencing Problems with Nonserial Structures	109
6.1. Introduction	109
6.2. Serial Multistage Sequencing Processes	110
6.3. Nonserial Multistage Sequencing Processes	111
6.4. CPM-Cost Problem: the Basic Model	114
6.5. CPM-Cost Problems with Serial Structures	115
6.6. CPM-Cost Problems with Nonserial Phase Structure	117
6.7. Nonserial Networks: Project Cost Minimization Approach [PCM]	117
6.8. Project Time Minimization Approach [PTM]	120
6.9. A Dynamic Programming Model of the PTM Problem	120
6.10. Example 1. The Pseudo-stage Concept and CPM-Cost Problem	121
6.11. Example 2. A CPM-Cost Problem with Many Paths Departing from Junction	123
6.12. Example 3. A Complex Nonserial System as in Fig. 6.10	125
6.13. Discussion	127
Bibliography and Comments	128
7. Analytical Results for the Flow-shop Scheduling Problem	132
7.1. Introduction	132
7.2. Characteristics of Schedules	132
7.3. Calculation of Makespan	133
7.4. Determination of Machine Idle Time and Waiting Time of Job	135
7.5. Flow Time $T_k(i_q)$	136
7.6. Recurrence Relation for Flow Time $T_k(i_q)$	137
7.7. Expressions of Makespan	138
7.8. Calculation of Machine Idle Time	139
7.9. Flow Network Expression in Critical Path Method	141
7.10. Critical Path Length between Two Nodes and Makespan	142
7.11. Two-machine Min-Makespan Problem	146
7.12. Solution by Using Machine Idle Time	147
7.13. Solution by Dynamic Programming	149
7.14. Solution by using Flow Time $T_2(i)$	152
7.15. Working Rules	154
7.16. General Working Rule	154
7.17. Restricted Working Rule	158
Bibliography and Comments	160
8. Flow-shop Scheduling Problems: Analytical Results II and Extensions	162
8.1. Introduction	162

8.2.	Three-machine Min-Makespan Problem	162
8.3.	Limited Solution under Special Processing Times	163
8.4.	Solution by Using Machine Idle Time	163
8.5.	Formulation by Flow Times $T_2(i)$, $T_3(i)$	167
8.6.	Flow Times $T_2(i)$, $T_3(i)$ and Makespan	167
8.7.	Recurrence Relation Between $T_2(i_q)$, $T_3(i_q)$ ($q = 1, 2, \dots, n$)	167
8.8.	Dynamic Programming-type Formulation	168
8.9.	Values of $T_3(i)$, $T_3(ij)$, and $T_3(j)$, $T_3(ji)$	169
8.10.	Proof by Using Flow Times	170
8.11.	Case (a)	170
8.12.	Case (b)	171
8.13.	Limited Solution under Other Special Processing Times	173
8.14.	Sufficient Conditions for Two Adjacent Jobs to be in a Definite Order in the Optimal Sequence	175
8.15.	Sufficient Inequalities not Including $T_2(l)$ and $T_3(l)$	176
8.16.	Sufficient Inequalities that Satisfy the Transitive Property	178
8.17.	Two-machine Min-Makespan Problem where Time Lags Exist	182
8.18.	On the Generalizations to m Machine Permutation Scheduling	185
	Bibliography and Comments	197
9.	Flow-shop Scheduling Problems: General Solutions	200
9.1.	Introduction	200
9.2.	m Machine Min-Makespan Problem ($m \geq 3$, no Passing is Allowed)	200
9.3.	Concrete Procedures in BAB Algorithms	201
9.4.	BAB Algorithms for Flow Shops	204
9.5.	Standard Lower Bound	205
9.6.	Efficiency of BAB Algorithm	207
9.7.	Relations Between Machine Order, Processing Times, and Optimal Sequence	208
9.8.	Revised Lower Bound and Composite Lower Bound	210
9.9.	Backtrack Programming and Lexicographical Search Method	216
9.10.	m Machine Min-Mean Completion Time Problem	219
9.11.	A Lemma and Analytical Results	220
9.12.	A BAB Algorithm	222
9.13.	m Machine Min-Penalty Cost by Tardiness Problem	225
9.14.	Backtrack Programming and Lexicographical Search Method	226
9.15.	A BAB Algorithm and its Lower Bound	230
	Bibliography and Comments	233
10.	Flow-shop Scheduling Problems: Unified Algorithms and Approximate Solutions	235
10.1.	Introduction	235
10.2.	Unified Multistage Combinatorial Algorithms for Setup-time Imbedded Processing Times	235
10.3.	A Formulation by the State Transformation Process	236
10.4.	Simple Function $q(F, P(S_k))$ for Each Objective Function F	238
10.5.	Relation Between Min-Makespan Problem and Min-Other Objective Problems	241
10.6.	Unified Multistage Combinatorial Algorithm	242
10.7.	Application to Parallel Scheduling	243
10.8.	Numerical Examples of Min-Mean Intermediate Waiting-time Problem and Related Parallel Scheduling	243
10.9.	Unified Multistage Combinatorial Algorithm where Explicit Setup Times Exist	244

10.10. Setup Times	245
10.11. A Formulation by the State Transformation Process	245
10.12. Approximate Algorithms for Min-Makespan Problem	251
10.13. Application of a Specified Function to Construct an Approximate sequence	251
Bibliography and Comments	256
11. Flow-shop Scheduling under Sequence Dependence Conditions	258
11.1. Sequence Dependent Setup Times	258
11.2. Problem Definition	259
11.3. Optimality of Permutation Schedules	260
11.4. Dynamic Programming Formulations for ID and DI	261
11.5. Forward DP Recursion for Problem Type ID	261
11.6. Computational Aspects	263
11.7. Backward DP Recursion for Problem Type DI	263
11.8. Implications of Theorem 3	266
11.9. Computational Aspects	267
11.10. Discussion of the Dynamic Programming Approach	267
11.11. Optimal Schedules for Problem Types DII, IDI and IID	269
11.12. Dominance Conditions for Problem Types DII, IDI, and IID	270
11.13. Backward Dynamic Programming Formulations	271
11.14. Reducing Dimensionality	274
11.15. Selection of Value for c_1	275
11.16. Computational Requirements	276
11.17. Extensions	277
Bibliography and Comments	279
12. The Job-shop Scheduling Problem	281
12.1. Introduction	281
12.2. The $n \times 2$ Min-Makespan Problem	282
12.3. Graphical Solution of the $2 \times m$ Min-Makespan Problem	283
12.4. Graphical Representation of the Schedule	284
12.5. Construction of the Path	286
12.6. Approaches by Integer Linear Programming	289
12.7. Standard Formulation by ILP	290
12.8. BAB Algorithms by Following Active Schedule Generation Procedure	291
12.9. BAB Algorithm by Active Schedule Generation Procedure	291
12.10. BAB Algorithm I	292
12.11. Lower Bound in BAB Algorithm I	293
12.12. Graph $G_0 = (X, Z)$ Expressing Orderings	296
12.13. Disjunctive Graph Representation by Limited Machine Availability	297
12.14. BAB Algorithm on Disjunctive Graphs by following an Active Schedule Generation Procedure	298
12.15. BAB Algorithm by Resolving the Pairs of Disjunctive Arcs on a Disjunctive Graph	301
12.16. Representation by Graph $G_0 = (X, Z)$	301
12.17. Conflict Set	302
12.18. Disjunctive Graph	302
12.19. Complete Selection of Disjunctive Arcs and Schedules	302
12.20. Formulation of Two Types	303
12.21. BAB Algorithm by Partial Selection of Disjunctive Arcs	304
12.22. Generalization of the Sequencing Problem: The Multiproject Scheduling Problem with Limited Resources	307
12.23. Solution Based on Disjunctive Graphs	307
12.24. Problem Statement	308
12.25. Conflict Set	308

12.26. Disjunctive Graphs Based on Conflict Sets of Activities	309
12.27. Formulation by Disjunctive Graph	309
12.28. BAB and EXTBAB Algorithms by Partial Selection of Disjunctive Arcs	311
12.29. BAB Algorithm	311
12.30. Justification of Proposed BAB Algorithm and Remarks	313
12.31. Numerical Examples by BAB Algorithm and Related EXTBAB Algorithm	315
Bibliography and Comments	318

Author Index	323
---------------------	------------

Subject Index	327
----------------------	------------

CHAPTER 1

Network Flow, Shortest Path and Control Problems

1.1. Introduction

In this chapter we wish to discuss briefly an interesting class of problems with wide ramifications. On the one hand, they can be viewed as abstractions and natural generalizations of a control process. Furthermore, they provide us with an entry into the rugged terrain of scheduling processes and into the study of some other types of decision processes of combinatorial type, such as pattern recognition and numerous other problems arising in the field of artificial intelligence. The application of these ideas to artificial intelligence will be given in Chapter 2. Finally, in our continuing quest for feasible computational procedures, they furnish considerable motivation for the creation of sophisticated decomposition techniques based upon topological considerations. We will do this in Section 1.16 when we make an application of these techniques to the calculus of variations. Let us also mention that questions of this nature occur more and more frequently in connection with the execution of complex computer problems.

In abstract terms, we are interested in a discrete control process of the following type: Let p be a generic element of a finite set S and $T(p, q)$ be a family of transformations with the property that $T(p, q) \in S$ whenever $p \in S$ and $q \in D$, the decision space, again taken to be discrete. We wish to determine a sequence of decisions, q_1, q_2, \dots , which transform p , the initial state, into p_N , a specified state, in an optimal fashion.

We are really interested in questions of feasibility. However, in order to handle this imprecise concept we narrow our sights and consider optimization.

A problem of great contemporary interest, the "routing problem", is a particular case of the foregoing. We will use it as our leitmotif.

1.2. The Routing Problem

Consider a set of N points, numbered $1, 2, \dots, N$, with N the terminal point, as shown in Fig. 1.1.

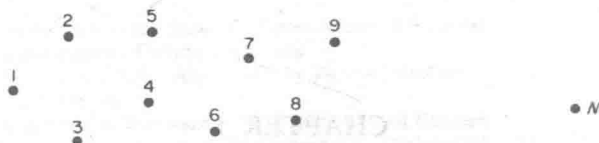


FIG. 1.1

We suppose that there exists a direct link between any two points i and j which requires a time t_{ij} to traverse, with $t_{ii} = 0$, $i = 1, 2, \dots, N$. In all that follows we take $t_{ij} \geq 0$. What path do we pursue, starting at 1, passing through some subsets of the points $2, 3, \dots, N-1$, and ending at N , which requires a minimum time?

Two possible paths are shown in Figs. 1.2 and 1.3. The first goes directly to N ; the second goes through every point always moving from a lower order number to a higher order one before reaching N .



FIG. 1.2

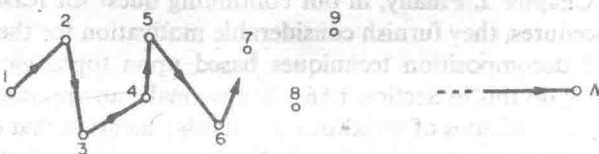


FIG. 1.3

Analytically, we are required to minimize the expression

$$T(i_1, i_2, \dots, i_k) = t_{1i_1} + t_{i_1i_2} + \dots + t_{i_k N}, \quad (1.1)$$

where (i_1, i_2, \dots, i_k) is some subset of $(2, 3, \dots, N-1)$.

Exercises

1. Does a minimizing path ever contain a loop, i.e. pass through the same point twice?
2. How many different admissible paths are there?
3. Can the problem presently be solved by direct enumeration with a digital computer for $N = 100$? Assume that we can enumerate and compare paths at the rate of one per microsecond.

1.3. Dynamic Programming Approach

To treat this problem by means of dynamic programming, we imbed the original problem within the family of problems consisting of determining the

optimal path from i to N , $i = 1, 2, \dots, N-1$. Let

$$f_i = \text{minimum time to go from } i \text{ to } N, i = 1, 2, \dots, N-1, \quad (1.2)$$

and set $f_N \equiv 0$. Then the principle of optimality yields the relation

$$f_i = \min_{j \neq i} [t_{ij} + f_j], \quad i = 1, 2, \dots, N-1, \quad (1.3)$$

with the "boundary condition" $f_N = 0$.

This functional equation is different in form from the usual dynamic programming equation, being implicit rather than explicit. Further, we note its dependence on one (the current node) variable rather than two (the stage and state) variables. The unknown function occurs on both sides of the equation which means that the solution cannot be obtained by any direct iteration. Consequently, some interesting analytic questions arise:

- Does (1.3) possess a solution?
- Does it possess a unique solution?
- Provided that it does possess a unique solution, how can the equation be used to provide the optimal path?
- How can the solution be obtained computationally?

The question of computational feasibility is, as usual, the thorniest one and one that forces us to consider new and alternate methods. Some aspects of this question will be discussed now; others will be described in the references.

Exercises

- In some cases we are interested not only in the path of shortest time, but also in the next best path and generally the k th best path. Let $f_i(2)$ be defined as time required to go from i to N following a second best path. Obtain a functional equation similar in form connecting $f_i(2)$ and f_i .
- Similarly, obtain a functional equation for $f_i(k)$, the time associated with the k th best path.
- Obtain a functional equation corresponding to (1.3) for the case where the time to go from i to j depends upon the "direction" in which i is entered, which is to say upon the point from which i is reached.
- What is the form of the equation if we cannot go from every point to every other point?

1.4. Upper and Lower Bounds

Prior to a demonstration of the existence and uniqueness of the solution of (1.3), let us show how easy it is to obtain upper and lower bounds for the solution (or solutions) of (1.3). An "experimental proof" of uniqueness is thus available in any particular case if we can show computationally that the upper and lower bounds coincide.

4 Mathematical Aspects of Scheduling and Applications

Let the sequence $\{\phi_i^{(r)}\}, r = 0, 1, 2, \dots$, be defined for each i in the following fashion:

$$\phi_i^{(0)} = \min_{j \neq i} t_{ij}, \quad i = 1, 2, \dots, N-1, \quad \phi_N^{(0)} = 0,$$

$$\phi_i^{(r+1)} = \min_{j \neq i} [t_{ij} + \phi_j^{(r)}], \quad i = 1, 2, \dots, N-1, \quad \phi_N^{(r+1)} = 0. \quad (1.4)$$

Let us recall that we are assuming that $t_{ij} \geq 0$.

It is clear that $\phi_i^{(r)} \geq 0$ and thus that

$$\phi_i^{(1)} = \min_{j \neq i} [t_{ij} + \phi_j^{(0)}] \geq \min_{j \neq i} t_{ij} = \phi_i^{(0)}. \quad (1.5)$$

Hence, inductively,

$$\phi_i^{(0)} \leq \phi_i^{(1)} \leq \dots \leq \phi_i^{(r)} \leq \phi_i^{(r+1)} \leq \dots \quad (1.6)$$

Let us now show inductively that

$$\phi_i^{(r)} \leq f_i, \quad (1.7)$$

where f_i is any nonnegative solution of (1.3). We have

$$\phi_i^{(0)} = \min_{j \neq i} t_{ij} \leq \min_{j \neq i} [t_{ij} + f_j] = f_i, \quad (1.8)$$

whence (1.7) follows from (1.4) via an induction.

Since the sequence $\{\phi_i^{(r)}\}, r = 1, 2, \dots$, is uniformly bounded and monotone increasing, we have convergence for each i . Let

$$\phi_i = \lim_{r \rightarrow \infty} \phi_i^{(r)}. \quad (1.9)$$

Then (1.7) yields

$$\phi_i \leq f_i, \quad (1.10)$$

and (1.4) shows that ϕ_i is itself a solution. Hence, ϕ_i is a lower bound for any solution.

To obtain an upper bound, let us introduce the sequence $\{\psi_i^{(r)}\}$ where

$$\psi_i^{(0)} = t_{iN}, \quad i = 1, 2, \dots, N,$$

$$\psi_i^{(r+1)} = \min_{j \neq i} [t_{ij} + \psi_j^{(r)}], \quad i = 1, 2, \dots, N. \quad (1.11)$$

Then

$$\psi_i^{(0)} = t_{iN} \geq \min_{j \neq i} [t_{ij} + f_j] = f_i, \quad (1.12)$$

and again an induction establishes

$$\psi_i^{(0)} \geq \psi_i^{(1)} \geq \dots \geq \psi_i^{(r)} \geq f_i. \quad (1.13)$$