NETWORK FLOW PROGRAMMING

Paul A. Jensen
J. Wesley Barnes

NETWORK FLOW PROGRAMMING

Paul A. Jensen
J. Wesley Barnes

Operations Research Group
Department of Mechanical Engineering
University of Texas at Austin

John Wiley & Sons
New York Chichester Brisbane Toronto

Library of Congress Cataloging in Publication Data:

Jensen, Paul A 1936–
Network flow programming.

Includes index.
1. Network analysis (Planning) 2. Programming (Mathematics) I. Barnes, J. Wesley, joint author. II. Title.

T57.85.J39 658.4′032 79-26939
ISBN 0-471-04471-7

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

PREFACE

Today, both theoretical and applied interests in the use of network flow programming are experiencing an expansion unrivaled by that of any other optimization technique. Perhaps the most important reasons for this expansion are the fresh advances in network computational methods that allow analysts to solve problems of enormous size formerly unassailable by any other approach. Networks have been used in innumerable applications to represent such things as inventory systems, river systems, distribution systems, precedence ordering of events, flowcharts, and organization charts. In fact, the network representation is such a valuable visual and conceptual aid to the understanding of the relationships between events and objects that it is used in virtually every scientific, social, and economic field.

In this book, we present a synthesis of the more important techniques, both recent and traditional, that are related to network flow programming. The level of discussion is easily within the reach of first-year graduate or advanced undergraduate students. The two background requirements are a reasonable facility with a general-purpose programming language, such as Fortran IV, and an understanding of Linear Programming that is consistent with a good undergraduate text in Operations Research. The population possessing this background includes most persons holding undergraduate degrees in Computer Science, Engineering, Operations Research, Applied Mathematics, and Business Administration. However, the most important and largest segment of our intended audience is the practitioner.

With this audience in mind, we do three things. First, we emphasize concepts rather than theoretical proofs. Such proofs abound in the literature, and at the end of the book an extensive bibliography is provided which covers the theoretical underpinnings of the subject. Second, we include in the text a coherent computational package of efficient algorithms that will solve any of the various network programming problems addressed in the book. Third, we take great care to present both the relationships between Linear Programming and Network Flow Programming and the interrelationships that join the various network flow programming problems and their suggested techniques of solution.

In the process of achieving these three objectives, we have formulated a reasonably complete exposition of the total spectrum of single commodity network flow programming problems. This presentation ranges from basic topics such as network storage techniques and handling representations to such advanced topics such as stochastic and generalized networks.

The principal strengths of this book can be summarized as follows:

The description of the various network flow programming problems in a consistent and unified notation. This makes evident the strong relationships between the problems and the relationships to the underpinning linear programming theory.

The modular nature of the manipulation and optimization algorithms and programs. Algorithms are presented in small packages that are easily understood by the student. Complex operations are performed by collections of modules. Modularization also provides for easy substitution of algorithms for experiments that test alternative computational approaches.

Emphasis on the computational aspects of algorithms. The recent emergence of network flow programming is primarily due to the computational procedures used to store and manipulate networks and network components in the computer. The serious student or user of network programming cannot neglect this aspect.

The method of presentation of the algorithms. We present algorithms at three levels: a rigorous flowchart using notation related to the computer programs (the form of the flowcharts is particularly compact and easily understood), a parallel English language description, and an example.

The scope of the material covered. We cover all of the important single commodity network flow programming problems. This includes extensive discussion of generalized networks and networks with separable nonlinear cost functions. In particular, this gathering of subjects is an especially useful one not currently available in existing texts.

Although most of the material in this book derives from previously published work, some appears here for the first time. The results and algorithms given here are immediately applicable to a wide variety of "real-world" problems. The computer codes, which we designed to be especially useful for teaching, can be obtained from us for a nominal handling charge. In addition, a solutions manual containing detailed answers to all exercises in the book is available to instructors and practitioners from John Wiley on written request.

There is ample material for a two-semester course, with some augmentation—perhaps with case studies. The first semester would comprise the material of Chapters 1 through 7. The second semester would deal with the more advanced material in the remaining chapters and the augmentations selected by the instructor. A particularly able class should be able to cover Chapters 1 through 10 in a single concentrated semester. These comments are based on our more than five years of experience teaching this material.

We express our thanks to the great number of graduate students in the Mechanical Engineering Department's Operations Research graduate program at the University of Texas at Austin who gave us assistance in the development of this book. Finally, special thanks are due to Margaret Jensen, who typed the numerous revisions of the manuscript.

Paul A. Jensen
J. Wesley Barnes

CONTENTS

CHAPTER 1	**NETWORK FLOW MODELS**	1
1.1	Introduction	1
1.2	Relationships between Network Flow Programming Problems	3
1.3	Specializations of the Pure, Linear, Minimum Cost Flow Problem	4
1.4	The Network-with-Gains Model	7
1.5	A Preview of Things to Come	8
1.6	Historical Perspective	9
1.7	Exercises	10
CHAPTER 2	**MODELING APPLICATIONS OF NETWORK PROGRAMMING**	11
2.1	Introduction	11
2.2	Slack Node Parameters	11
2.3	The Pure, Linear, Minimum Cost Flow Problem—Example Applications	12
2.4	The Transportation Problem—Example Applications	23
2.5	The Assignment Problem—Example Applications	28
2.6	The Shortest Path Problem—Example Applications	34
2.7	The Maximum Flow Problem—Example Applications	39
2.8	Networks with Gains—Example Applications	42
2.9	Historical Perspective	48
	Exercises	50

CHAPTER 3	FORMALIZATION OF NETWORK MODELS	59
3.1	Network Notation	59
3.2	Two Useful Transformations	65
3.3	The Algebraic Network Model	66
3.4	The Linear Programming Model for the Pure Minimum Cost Problem	69
3.5	Terminology from Graph Theory	73
3.6	The Expanded and Marginal Networks	76
3.7	Networks with Nonlinear Arc Costs	79
3.8	Solution Algorithms for the Pure Minimum Cost Flow Problem	83
3.9	Limitations of Network Flow Programming	85
3.10	Historical Perspective	86
	Exercises	87
CHAPTER 4	NETWORK MANIPULATION ALGORITHMS	91
4.1	Computational Cost	91
4.2	Network Representations	92
4.3	Method of Algorithm Description	97
4.4	Reading and Storing the Network	99
4.5	Tree Representation	103
4.6	Equivalent Algorithms Using a Preorder Traversal List	116
4.7	Flow Manipulation Algorithms	121
4.8	Historical Perspective	123
	Exercises	123
CHAPTER 5	THE SHORTEST PATH PROBLEM	125
5.1	Introduction	125
5.2	Representation as a Minimum Cost Flow Problem	126
5.3	All Admissible Arc Costs Positive	129
5.4	Some Arc Costs Negative with No Negative Cycles	132
5.5	Negative Cycles	136
5.6	A Nonbasic Algorithm	138
5.7	A Dual Shortest Path Algorithm	139
5.8	Historical Perspective	142
	Applied Exercises	144
	Theoretical Exercises	145
CHAPTER 6	THE MAXIMUM FLOW PROBLEM	147
6.1	Problem Statement	147
6.2	Physical Interpretation of the Dual	148
6.3	Theoretical Results	150
6.4	Basic and Nonbasic Algorithms	153

6.5	Flow Augmenting Algorithms	154
6.6	Historical Perspective	164
	Applied Exercises	165
	Theoretical Exercises	166
CHAPTER 7	**PURE MINIMUM COST FLOW PROBLEMS**	167
7.1	The Maxflow Method to Obtain a Primal Feasible Solution	168
7.2	The Artificial Arc Method to Obtain a Primal Feasible Solution	170
7.3	A Nonbasic Primal Algorithm	175
7.4	A Basic Primal Algorithm	186
7.5	A Dual Node Infeasible Algorithm	194
7.6	Historical Perspective	201
	Exercises	202
CHAPTER 8	**THE OUT-OF-KILTER ALGORITHM**	203
8.1	Network Model	207
8.2	Linear Programming Model	208
8.3	Kilter States	210
8.4	Flow Change	214
8.5	Potential Change	219
8.6	An Example Application of the OKA	223
8.7	Historical Perspective	227
	Exercises	227
CHAPTER 9	**NETWORK MANIPULATION ALGORITHMS FOR THE GENERALIZED NETWORK**	231
9.1	Introduction	231
9.2	The Network with Gains Model	233
9.3	Linear Programming Model	234
9.4	The Dual of the Linear Program	241
9.5	Representation of Basis Networks	242
9.6	Flow for Generalized Networks	254
9.7	The Node Potentials	268
9.8	Historical Perspective	275
	Exercises	275
CHAPTER 10	**GENERALIZED MINIMUM COST FLOW PROBLEMS**	281
10.1	The Generalized Shortest Path Problem	281
10.2	All Arc Costs Positive—All Gains Less than or Equal to One	283

10.3	Negative Arc Costs—Arc Gains Greater than One	285
10.4	Dual Generalized Shortest Path Algorithm	293
10.5	Generalized Minimum Cost Flow Algorithms	307
10.6	The Flow Augmentation Approach	308
10.7	A Generalized Primal Minimum Cost Flow Algorithm	315
10.8	Historical Perspective	333
	Exercises	334
CHAPTER 11	**THE CONVEX MINIMUM COST FLOW PROBLEM**	339
11.1	Convex Cost Functions	339
11.2	The Character of Solutions	343
11.3	Physical Network Flow Problems	344
11.4	Cost Functions That Depend on Random Variables	351
11.5	Piecewise Linear Approximation	355
11.6	Implicit Piecewise Approximation	357
11.7	Historical Perspective	363
	Exercises	364
CHAPTER 12	**CONCAVE COSTS**	367
12.1	Applications	368
12.2	Notation	371
12.3	Exhaustive Enumeration	372
12.4	Implicit Enumeration	376
12.5	Lower Bound	378
12.6	The Implicit Enumeration Algorithm	381
12.7	Example	385
12.8	Historical Perspective	388
	Exercises	389
	REFERENCES	391
	INDEX	403

ALGORITHMS

CHAPTER 4

1 READ 101
2 ORIGS 102
3 TERMS 102
4 ORIG 103
5 TERM 103
6 TPATH 107
7 ROOT 108
8 DELTRE 111
9 ADDTRE 113
10 TRECHG 116
11 ROOTP 118
12 DELTREP 119
13 ADDTREP 120
14 TREINT 121
15 FLOCHG 122
16 MFLO 122

CHAPTER 5

1 DSHORT 129
2 PSHORT 133
3 STARTD 134
4 SMNSP 134
5 SHORT 137
6 NBSHORT 139
7 DUALSP 141

CHAPTER 6

1 MAXFLO 156
2 LABEL1 157
3 FPATH 158
4 FPATH2 160
5 LABEL2 160

CHAPTER 7

1 PHASE1 168
2 ARTIFIC 172
3 STARTM 174
4 PRIMAL1 175
5 SHORTM 178
6 DSHRTM 178
7 PSHRTM 179
8 SMNSPM 180
9 PRIMAL 187
10 SELECT 189
11 INCREMENT 199
12 SHORT2 200

CHAPTER 8

1 READOK 208
2 KILTER 213
3 ADOKA 215
4 FIND 216
5 FLOW 217
6 FPATH3 218
7 OKAMFLO 219
8 POTENT 222

CHAPTER 9

1 TRAIL 246
2 ROOTG 247
3 ADDTREG 251
4 MFLOG 266
5 FLOWG 266
6 DUAL 271
7 CYCLE 273

CHAPTER 10

1 DSHRTG 283
2 PSHRTG 286
3 SMNSPG 287
4 DULSPG 302
5 INCREMG 314
6 ARTIFG 316
7 PATHPG 323
8 PGAINS 332

CHAPTER 11

1 CONVEX 361
2 HFUN 361
3 SMNCNVX 362
4 ORGSCNV 363

CHAPTER 12

1 READCC 372
2 ENUMER 381
3 SETUP 383
4 SET 383
5 RELAX 383
6 ROUND 384
7 SEPRAT 384
8 RESET 385

CHAPTER 1
NETWORK FLOW MODELS

1.1 INTRODUCTION

As illustrated in Figure 1.1, a *network* is a collection of *nodes* and *arcs*. This representation is useful for modeling a wide range of physical and conceptual situations. Networks have been used in innumerable applications to represent such things as inventory systems, river systems, distribution systems, precedence ordering of events, flowcharts, and organization charts. The network representation is such a valuable visual and conceptual aid to the understanding of the relationships between events and objects that it is used in virtually every field of scientific, social, and economic endeavor.

Some practical situations that can be represented by a network also have the characteristic of flow; that is, water may flow in a pipe network, traffic may flow in a street network and products may flow in a distribution network. Models of such situations are called *network flow models*. In this book, we restrict our attention to models of this type and, as we will see, many problems not obviously in this class can be represented by network flow models.

Further, we consider network flow models in which the amount of flow in each arc is controllable and the objective is to choose values for the arc flows that optimize some measure of effectiveness. To illustrate, suppose Figure 1.2 defines a network flow model. Each arc in the network has flow directed as specified by the arrow head of the arc. In this model, each arc has been assigned three parameters: a *lower bound*, which is the minimal amount that can flow over the arc; a *capacity*, which is the maximum amount of flow that the arc can carry; and a *cost* for each unit of flow that passes through the arc.

Since a time period is implied in most network formulations, flows and capacities are usually stated in terms of "flow-per-unit-time." If no lower bound parameter is present on an arc, it is assumed that the lower bound is zero. In

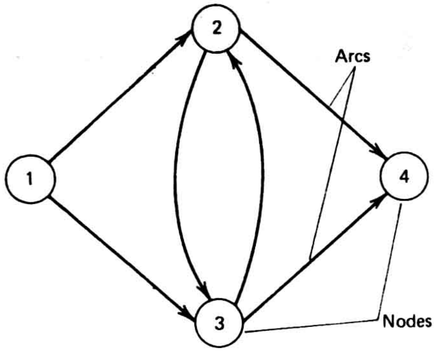

Figure 1.1
Network

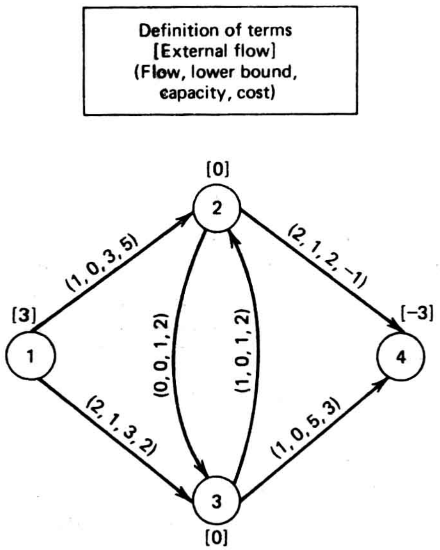

Figure 1.2
Network Flow Problem with Solution

addition lower bounds present no practical problems since a very simple transformation, discussed in Chapter 3, may be used to remove nonzero lower bounds from any network. The required quantities of flow entering or leaving the network at each node are also specified. These parameters of the nodes are called *external flows*. A positive external flow enters the network at a node and a negative external flow leaves the network. Flow is conserved at each node. Thus, the flow entering a node from the arcs of the network plus the external flow at the node must equal the flow leaving the node on the arcs of the network.

The flows on the arcs are controllable within the limits, or constraints, set by arc capacities, conservation of flow, and node external flows. Clearly, these arc flows are the decision variables of an optimization problem. The optimization problem is to choose the arc flows, within the above restrictions, to minimize the total cost of the flow. As the reader may easily verify, the flows shown in Figure 1.2 are optimal for the given parameter values.

1.2 RELATIONSHIPS BETWEEN NETWORK FLOW PROGRAMMING PROBLEMS

The problem of optimizing some objective subject to constraints is called a *mathematical programming problem*. Because all the problems considered in this text are defined by a network that carries flow, we use the term *network flow programming problem*. The problem of Figure 1.2 is a specific example of the pure, linear, minimum cost flow problem. The schematic relationships joining this basic problem to other network flow programming problems that are considered in this text are shown in Figure 1.3. The central point in this figure is the pure, linear, minimum cost flow problem. The problems listed to the left are less general in the sense that they are specializations of this basic problem. Problems listed to the right are more general in that this basic problem is in some way a specialization of each of these problems. The general linear programming problem is also shown in this figure to indicate its relationship to the network programming problems. Algorithms have been identified to solve each of the problem classes in Figure 1.3. Algorithms for the less general problems are more efficient, in computational time and memory requirements, than those for the more general problems. Algorithms defined for a more general problem can solve a less general form of that problem, while the converse is rarely true.

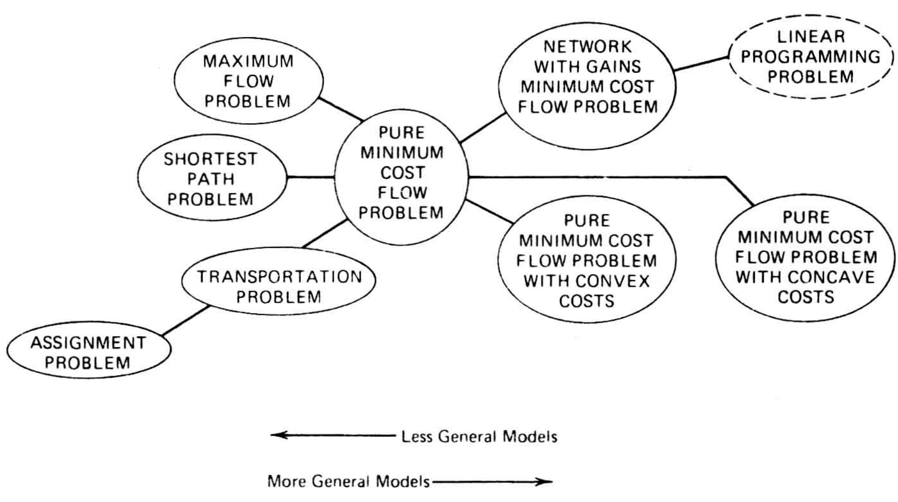

Figure 1.3
Network Flow Programming Problems Relationships

1.3 SPECIALIZATIONS OF THE PURE, LINEAR, MINIMUM COST FLOW PROBLEM

Transportation Problem

Of frequent use in practice, the *transportation problem* is a special case of the minimum cost flow problem where the network representation has a distinct form: the nodes can be partitioned into two sets N_1 and N_2 such that all arcs originate in N_1 and terminate in N_2. Three other special properties are:

1. All arcs have infinite capacities, which allows omission of the capacity parameter from the arc parameter list.
2. All nodes have nonzero fixed external flows.
3. The sum of the external flows over all nodes is zero.

An example transportation problem network model, with its associated optimal flows, is presented in Figure 1.4.

Assignment Problem

An important specialization of the transportation problem is the *assignment problem*, in which both $|N_1| = |N_2|$ and all demands and supplies are unity. Given the cost associated with pairing any object $i \epsilon N_1$ with any object $j \epsilon N_2$, the problem is to find an exhaustive one-to-one pairing of the two sets' elements that minimizes the sum of the pairing costs. This problem is illustrated by the network of Figure 1.5. Again the flows present in Figure 1.5 are optimal and

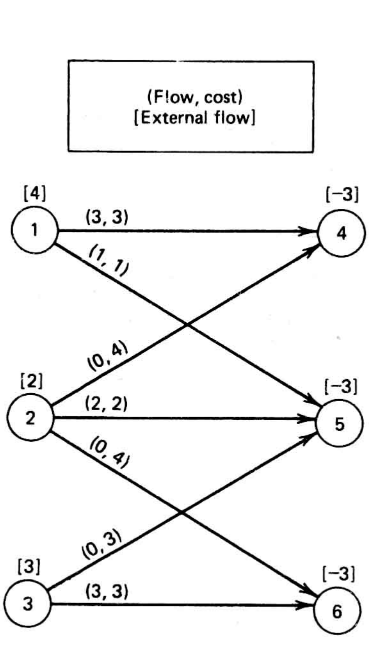

Figure 1.4
Example Transportation Problem

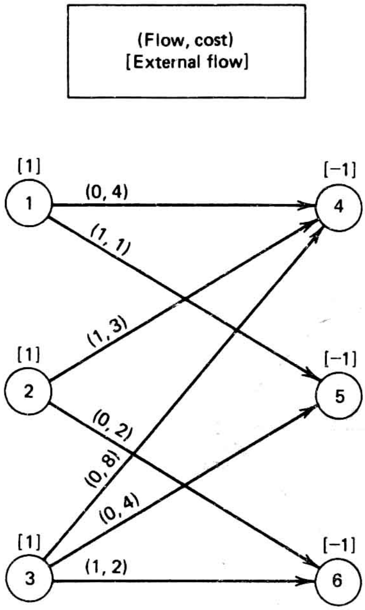

Figure 1.5
Example Assignment Problem

imply the following optimal pairings: node 1 and node 5, node 2 and node 4, and node 3 and node 6.

Shortest Path Problem

In the *shortest path problem*, two nodes are designated as the *source* and the *sink*. The arc cost is commonly given the physical interpretation of *arc length*. The *optimal path* is that sequence of arcs connecting the source to the sink such that the sum of the arc costs on the path is minimized. An example shortest path problem appears in Figure 1.6. The optimal flow pattern in Figure 1.6 implies the shortest path consists of arcs (1, 3), (3, 4), and (4, 5).

Since nodes that are neither source nor sink will always have a fixed external flow of zero, no external flow designation is given for nodes 2, 3, and 4. This convention will be followed consistently throughout the book; that is, the absence of fixed external flow parameters will be interpreted as zero external flow.

Maximum Flow Problem

For the *maximum flow problem*, arc capacity is the only relevant parameter. Once a source node and a sink node are identified, the problem is to maximize the flow passing from source to sink. An example maximum flow problem is shown in Figure 1.7, where node 1 is the source and node 6 is the sink. One of the several alternate optimal flow patterns is also given in Figure 1.7.

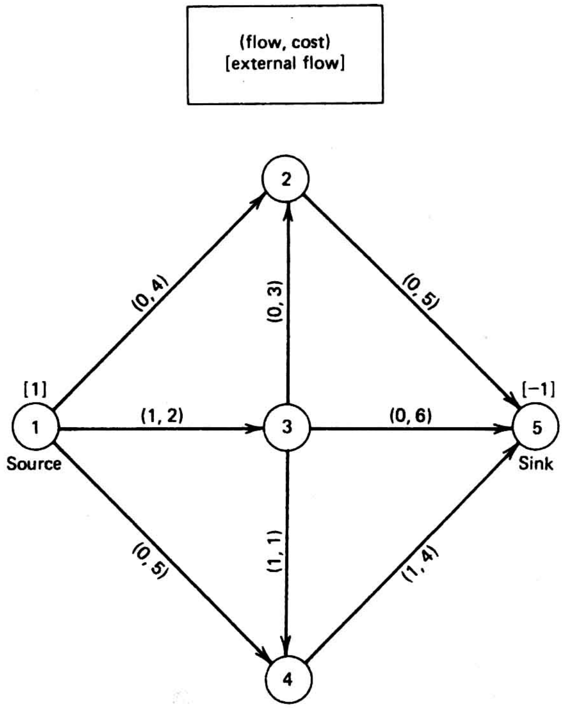

Figure 1.6
Example Shortest Path Problem

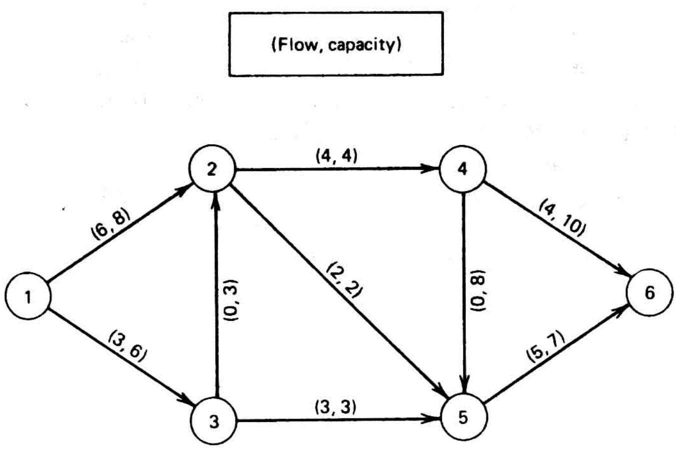

Figure 1.7
Example Maximum Flow Problem