GROUP THEORETIC AND RELATED APPROACHES TO FIXED CHARGE PROBLEMS

A THESIS

Presented to

The Faculty of the Division of Graduate

Studies and Research

by

Ronald Lee Rardin

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in the School of Industrial and Systems Engineering

Georgia Institute of Technology

December, 1973

GROUP THEORETIC AND RELATED APPROACHES

TO FIXED CHARGE PROBLEMS

Approved:

V. E. Unger, Chairman

M. S. Bazaraa

J. J. Jarvis

C. M. Shetty

W. R. Smythe, Jr.

Date approved by Chairman: 10/25/73

ACKNOWLEDGMENTS

I owe my thanks and appreciation to numerous persons for their contributions to the success of the graduate studies which culminated in this dissertation. Among the faculty, Dr. Paul Gray provided the encouragement which brought me into the graduate program at Georgia Tech and carried me through the early part of my classwork. Dr. Mokhtar Bazaraa offered friendship, guidance and inspiration as a teacher, advisor and member of my dissertation committee. The other members of my dissertation committee, Drs. Jarvis, Shetty and Smythe, also provided encouragement and useful advice. Perhaps most important of all is the contribution of my dissertation advisor, Dr. Ed Unger. His encouragement, guidance and friendship have provided an ideal environment for my dissertation research.

In addition to these faculty members, inestimable aid has also been provided by my fellow graduate students. Special credit is due Dr. Warren Langley and Dr. Jeff Kennington, who provided constant sounding boards for my ideas as the research leading to this dissertation developed.

Finally, I wish to thank my wife Blanca for her patience and understanding over the past two years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

SUMMARY

In this dissertation special problem structures related to the group theoretic formulations of fixed charge linear programming problems (FCP) and fixed charge network problems (FCNP) are investigated in the context of penalty-oriented branch-and-bound procedures. A number of penalty problems derived from the group formulation by reimposing relaxed constraints are described and compared. Problem structures of FCP and FCNP related to such penalty problems are characterized and exploited to produce efficient techniques for solving the continuous relaxations of FCP and FCNP, and for selecting, constructing and solving the penalty problems.

A number of the proposed techniques for FCNP's are implemented in a computerized algorithm and tested on a set of randomly-generated problems including both general FCNP's and the special cases of the fixed charge transportation and the warehouse location problems. Computational results indicate the procedure will efficiently solve such FCNP's when the number of variables with fixed charges is as large as 100.

CHAPTER I

INTRODUCTION

A great deal of recent research in the development of operations research methods has focused on finding computationally efficient solution procedures for linear integer programs. In spite of this broad attention, however, no generally effective algorithm for such problems has yet been presented. Thus, much of the most recent research in integer programming has turned to identification and exploitation of the simplified problem structures found in important special cases of linear integer programs. In some cases (e.g. set covering) such specialized research has produced promising results.

The purpose of the research reported in this dissertation is to pursue such an investigation and exploitation of problem structures for a class of integer programs known as *fixed charge problems*. As will be discussed in succeeding sections, this class includes many of the integer programs most often encountered in application situations, yet no computationally adequate solution procedures have been developed.

The particular structures of fixed charge problems investigated in this research are those related to the group theoretic formulation of the problem as defined in Chapters II and III. Over the past several years, Gomory, Johnson and others have presented a number of theoretical results and related computational schemes for general linear integer programs which derive from this formulation. The research reported herein seeks to apply these ideas to fixed charge problems by identifying extensions and simplifications in the approaches related to the special structures of fixed charge problems, and developing a combination of the approaches appropriate for a computationally efficient solution procedure.

1.1 Formulation of the Fixed Charge Problem

Mathematically the fixed charge problem can be formulated[1]

$$\min \sum_j \zeta_j(x_1) + c_2^T x_2$$

$$\text{s.t.} \quad A_1 x_1 + A_2 x_2 = b$$

$$u_1 \geq x_1 \geq 0$$

$$u_2 \geq x_2 \geq \ell_2$$

where

$$\zeta_j(x_1) = \begin{cases} f_j + v_j x_{1j} & \text{if } x_{1j} > 0 \\ 0 & \text{otherwise,} \end{cases}$$

f, u_1, v and x_1 are n_1-vectors, c_2, ℓ_2, u_2 and x_2 are n_2-vectors, A_1 is an m by n_1 matrix, A_2 is an m by n_2 matrix, b is an m-vector, x_{ij}, f_j and v_j are the components of x_1, f and v, and $f > 0$. The problem is

---

[1]See Section 1.4 for a full explanation of notational conventions.

thus a general linear program with the added features that all decision variables are bounded, and that a subset of the variables have a fixed charge assessed in addition to the usual variable costs of linear programming whenever they take on positive values. More general formulations could be presented where some variables were not explicitly bounded. However, practical applications usually permit introductions of some artificially high bound when none is given explicitly, and bounds on the variables with fixed charges are required for the analysis which follows. Thus upper and lower bounds on all variables will be assumed throughout this dissertation.

The importance of linear programs with fixed charges has been well known since the early 1950's. Many problems arising in government and industry have capital investment, setup or similar initial costs which must be assessed if particular alternatives are to be part of a problem solution. Examples are the construction costs of distribution or service facilities, development costs for product lines, setup costs for industrial processes, overhead and administrative costs of organizational units, purchase and installation costs for capital equipment, and personnel training costs for processes and procedures. Problems with such cost structures can often be adequately modeled by linear constraints, but the presence of fixed charges produces the more complicated objective function of the above formulation.

More generally Gray [46] has shown that any linear program with a separable concave objective function can be approximated by the fixed charge problem formulation presented above through the use of piecewise linearization. Thus a large class of nonlinear programming problems are also among those which could be at least approximately solved if an adequate solution procedure for fixed charge problems were available.

By the introduction of an n_1-vector of 0-1 integer variables w, and letting w_j and u_{1j} be the components of w and u_1, the above formulation of the fixed charge problem can be converted to the mixed-integer program

$$\begin{aligned} \text{min.}\quad & v^T x_1 + c_2^T x_2 + f^T w \\ \text{s.t.}\quad & A_1 x_1 + A_2 x_2 = b \\ & x_1 \geq 0 \\ & u_2 \geq x_2 \geq \ell_2 \end{aligned}$$

$$\left.\begin{aligned} -x_{1j} + u_{1j} w_j &\geq 0 \\ w_j &= 0 \text{ or } 1 \end{aligned}\right\} \quad j=1,2,\ldots,n_1.$$

The two formulations are clearly equivalent because the added variables and constraints assure the entire fixed charge f_j will be assessed whenever $x_{1j} > 0$, and the assumption that $f_j > 0$ guarantees no optimal solution can have $w_j > 0$ if $x_{1j} = 0$.

Though the above formulation is the standard mixed-integer statement of the fixed charge problem, it will be convenient for the analysis which follows to use the revised mixed-integer formulation

$$
(\mathrm{FCP})\quad
\begin{array}{lrcl}
\text{min.} & c_1^T x_1 + c_2^T x_2 + c_s^T s & & \\
\text{s.t.} & A_1 x_1 + A_2 x_2 & = & b \\
 & -x_1 + y - s & = & 0 \\
 & u_1 \geq y & \geq & 0 \\
 & u_1 \geq x_1 & \geq & 0 \\
 & u_1 \geq s & \geq & 0 \\
 & u_2 \geq x_2 & \geq & \ell_2 \\
 & y \equiv 0 \bmod u_1, & &
\end{array}
$$

where c_s, c_1, s and y are n_1-vectors with

$$
\left.\begin{array}{l} c_{sj} = f_j/u_{1j} \\ c_{1j} = v_j + c_{sj} \end{array}\right\} \quad j=1,\ldots,n_1.
$$

The equivalence of this formulation with the previous one follows by the substitution

$$\left.\begin{aligned} y_j &= u_{1j} w_j \\ s_j &= y_j - x_{1j} \end{aligned}\right\} \quad j=1,\ldots,n_1$$

because the addition of redundant upper bounds on s and x_1 cannot affect the solution, and each fixed charge f_j will continue to be assessed exactly when $x_{1j} > 0$. In this formulation, however, the fixed charge is assessed in pro rata amounts c_{sj}, some of which are part of the cost of x_{1j} and the remainder of which are carried by the slack variables s_j. The congruence constraint on y_j assures these two amounts will sum to the full fixed charge whenever x_{1j} is positive, and the fact that $c_{sj} > 0$ guarantees no optimal solution to FCP will incur any of f_j when $x_{1j} = 0$.

1.2 Formulation of Fixed Charge Network Problems

Where the constraints and objective function of FCP correspond to finding the least cost circulation of a single commodity through a directed network, the problem becomes part of an important subset of fixed charge problems known as *fixed charge network* or *fixed charge transshipment problems.* In particular, let (E_1, E_2) be the node-arc incidence matrix of a directed network, i.e. a matrix with columns which correspond to arcs of the network and consist entirely of zeros except for a -1 in the row corresponding to the origin node of the arc and a +1 in the row corresponding to the destination node of the arc. Then, in a formulation consistent with the definition of FCP given in the previous section, the fixed charge network problem can be stated

$$
\text{(FCNP)}\quad
\begin{array}{llllll}
\text{min.} & c_1^T x_1 + c_2^T x_2 & & & + c_s^T s & \\
\text{s.t.} & E_1 x_1 + E_2 x_2 & & & & = 0 \\
 & -x_1 & + y & - & s & = 0 \\
 & & u_1 & \geq & y & \geq 0 \\
 & & u_1 & \geq & x_1 & \geq 0 \\
 & & u_1 & \geq & s & \geq 0 \\
 & & u_2 & \geq & x_2 & \geq \ell_2 \\
 & & y & \equiv & 0 \bmod u_1 . &
\end{array}
$$

This special class of fixed charge problems has long been recognized as worthy of independent study for at least two reasons. First, any transportation or distribution problem where fixed charge cost structures are present, but where all flows can be expressed in terms of a single commodity, can be formulated as FCNP. Thus many of the important fixed charge problems mentioned in the previous section are actually fixed charge network problems.

In addition, the very special structure of the constraint matrices E_1 and E_2 permits many computational simplifications. A vast amount of research in the past two decades has focused on identifying and exploiting the special structure of such matrices.

1.2.1 The Fixed Charge Transportation Problem

When the distribution system associated with a fixed charge network problem is further simplified by having no transshipment points (i.e. every node is either a supply point or a demand point for the commodity), and (possibly zero) fixed charges are associated with each arc, the problem can be formulated in the even more specialized form of a *fixed charge transportation problem.* In its simplest form such a problem can be stated

$$\min \sum_{i=1}^{\hat{m}} \sum_{j=1}^{\hat{n}} \hat{\zeta}_{ij}(\hat{x}_{ij})$$

$$\text{s.t.} \quad \sum_{j=1}^{\hat{n}} \hat{x}_{ij} \le \hat{s}_i \qquad i=1,2,\ldots,\hat{m}$$

$$\sum_{i=1}^{\hat{m}} \hat{x}_{ij} \ge \hat{d}_j \qquad j=1,2,\ldots,\hat{n}$$

$$\hat{x}_{ij} \ge 0 \qquad i=1,\ldots,\hat{m}; \quad j=1,\ldots,\hat{n}$$

where

$$\hat{\zeta}_{ij}(\hat{x}_{ij}) = \begin{cases} \hat{f}_{ij} + \hat{v}_{ij}\hat{x}_{ij} & \text{if } \hat{x}_{ij}>0 \\ 0 & \text{otherwise} \end{cases}$$

and

$$\sum_{i=1}^{\hat{m}} \hat{s}_i = \sum_{j=1}^{\hat{n}} \hat{d}_j .$$

Here the $\hat{s}_i$ correspond to the maximum available supply at each supply node or source, and $\hat{d}_j$ is the amount of demand at each demand point or sink. The problem is to find the least-cost way to supply the demand. In general, it might occur that there is excess supply, i.e.

$$\sum_{i=1}^{\hat{m}} \hat{s}_i > \sum_{j=1}^{\hat{n}} \hat{d}_j .$$

However, such a case can always be converted to the above formulation by adding an artificial sink to absorb the excess supply at zero cost. Thus equality of supply and demand will be assumed throughout this dissertation.

Like the more general fixed charge network problem, the importance of the fixed charge transportation problem as a significant special case has been recognized for many years. This significance derives from the facts that many important applications problems take this simple form, and that the very special structure of the problem constraints permits many computational simplifications.

Though the above formulation of the fixed charge transportation problem is the standard one, it will be convenient in the analysis which follows to treat this problem as a special case of the formulation FCNP. The necessary conversion of a transportation problem to a network problem through addition of a super-source and a super-sink is well known and need not be detailed here. However, it should be noted how the required upper bounds on variables with fixed charges are assigned since the above formulation has all arcs uncapacitated. Such bounds are obtained by observing that the flow from source i to sink j cannot exceed $\min\{\hat{s}_i, \hat{d}_j\}$ in any optimal solution. Thus this minimum may be used as an artificial upper bound on arc flows.

1.2.2 The Warehouse Location Problem

Another special class of fixed charge network problems which has received extensive independent attention are the *facilities or warehouse location problems.* In one of many possible forms, the warehouse location problem is stated

$$\min. \sum_{i=1}^{\hat{m}} \sum_{j=1}^{\hat{n}} \hat{v}_{ij}\hat{x}_{ij} + \sum_{i=1}^{\hat{m}} \hat{f}_i\hat{w}_i$$

$$\text{s.t.} \sum_{j=1}^{\hat{n}} \hat{x}_{ij} \le \hat{s}_i\hat{w}_i \qquad i=1,\ldots,\hat{m}$$

$$\sum_{i=1}^{\hat{m}} \hat{x}_{ij} \ge \hat{d}_j \qquad j=1,\ldots,\hat{n}$$

$$\hat{x}_{ij} \ge 0 \qquad i=1,\ldots,\hat{m}; \quad j=1,\ldots,\hat{n}$$

$$\hat{w}_i = 0 \text{ or } 1 \quad i=1,\ldots,\hat{m}.$$

where

$$\sum_{i=1}^{\hat{m}} \hat{s}_i > \sum_{j=1}^{\hat{n}} \hat{d}_j.$$

As with the transportation problem, $\hat{s}_i$ and $\hat{d}_j$ are interpreted as the amount of supply at a potential warehouse or source, and the demand at a city or sink, respectively. In this case, however, fixed charges are incurred, not on the transportation arcs themselves, but in constructing or opening sources. Thus total supply is assumed to exceed total demand in order to exclude the trivial case where all warehouses must be opened.

As with the other cases discussed, warehouse location problems have received independent attention because of the widespread applicability of the formulation and the computational simplicity of its problem structure. Large numbers of factory, warehouse and service facility location analyses can be described by the very simple constraint structure presented above.

As with the fixed charge transportation problem, it will be convenient in most of the analysis which follows to treat warehouse location problems as special cases of the formulation FCNP. Once again the necessary conversion by addition of a super-source and a super-sink is well known. In this case, however, the fixed charge arcs are the circularizing arcs connecting the super-source with the individual sources.

1.3 Plan of the Dissertation

The chapters of the dissertation which follow this brief introduction present an attempt to systematically identify computationally relevant structures of problems FCP and FCNP, and to propose algorithms to exploit these structures. In Chapter II the analysis begins with a brief review of the current literature on methods related to the group theoretic formulation of mixed-integer programs, and on algorithms for the exact solution of fixed charge problems.

Problem structures identified as part of the research of this dissertation are presented in the following two chapters. In Chapter III attention is directed to the general problem FCP. A number of results are obtained, and their potential use in computational procedures is indicated. In Chapter IV the analysis is specialized to the case of FCNP. Unique structures of this special case and simplifications of more general structures are discussed.

Chapter V addresses the use of methods of Lagrangean relaxation in group-related procedures for FCP and FCNP. A technique for use of these methods is proposed, and the simplifications implied by the results of Chapters III and IV are indicated.

Chapter VI integrates the previous results in algorithms for FCP and FCNP. Step-by-step solution procedures combining the ideas of Chapters III, IV and V are proposed for both the general and the network cases.

Chapter VII presents empirical investigations of the computational value of a number of the approaches proposed in previous chapters. Results from the solution of a set of randomly-generated fixed charge network problems by an algorithm employing many of the proposals of this dissertation are presented and analyzed. Details of the computational procedures for generating and solving these problems make up Appendix A.

In Chapter VIII the dissertation concludes with a summary of important results and recommendations for future research. Some conjectures are offered as guidelines for future work.

1.4 Notation and Conventions

In order to effectively present observations about the structure of FCP and FCNP, a number of notational conventions will be required. For the convenience of readers, these conventions are summarized below.

1.4.1 General Notation

The general notational scheme adopted for this dissertation is to use upper case Latin letters to represent matrices, lower case Latin letters to indicate column vectors (or scalars if the vector is of dimension one), upper case Greek letters to denote sets, and lower case Greek letters to represent functions. However, occasional deviations from this standard will be required to conform to established tradition (e.g. use of $\sum$ for summation), and the following four symbols will be reserved for special sets and matrices:

I = an identity matrix of appropriate size.

1 = a matrix (or vector) of ones of appropriate size.

0 = a matrix (or vector) of zeroes of appropriate size.

Φ = the empty set.

When the transpose of a matrix (or vector) is required, it will be denoted by a T superscript. Thus M^T is the transpose of the matrix M. Sets of *rows* from a matrix (or vector) will be denoted by enclosing the matrix in brackets and indicating the limiting row numbers. For example, $[M]_k^1$ = the submatrix consisting of rows 1 through k of M. When only a single row of a matrix is required, this convention will be simplified by dropping the redundant superscript, and if no confusion will result the brackets will also be omitted leaving, for example, x_{ij} = the jth row or component of the vector x_i.

1.4.2 Optimal Solutions and Tableaux

Many of the discussions to follow will deal with optimal solutions, bases, and tableaux for various linear programs. All such references will be *with respect to bases of the well-known bounded Simplex procedure.* When it is desired to speak of the part of a solution vector, cost vector, bound vector or matrix associated with the basic variables, nonbasic variables, etc., *the usual rearrangement of rows and columns will be assumed, and identifying superscripts will be attached to sub-matrices.* Specifically, the superscript B will denote the basic part of a matrix, N the nonbasic part, U the part with nonbasic variables at their upper bounds, and L the part with nonbasic variables at their lower bounds.

The discussions to follow will also require reference to numerous sub-problems. To identify the relations between such problems, a bar over the name of a problem will denote the continuous relaxation of the problem, i.e. the same problem with any congruence constraints relaxed. Elements of the optimal solution, optimal Simplex tableau, etc. for such a continuous relaxation will be similarly denoted by bars over the names of the elements. Conversely, the derivative of a problem obtained by *adding* constraints will be indicated by specifying the additional constraints immediately after the name of the problem. Finally, the function nu (ν) will be used to denote the value of an optimal solution to the problem given as its argument, and the function beta (β) will denote the best currently available lower bound on this optimal solution value. In particular, the infeasible and unbounded cases in optimization problems will be handled implicitly by the conventions

$$\nu\ (\text{any infeasible problem}) = +\infty.$$

$$\nu\ (\text{any unbounded problem}) = -\infty.$$

Examples of the above tableau and problem conventions include the following:

$\overline{FCP}$ = the continuous relaxation of problem FCP, i.e. FCP without the constraints $y \equiv 0 \bmod u_1$.

A_1^B = the part of the *original* matrix A_1 corresponding to the basic variables in an optimal solution to $\overline{FCP}$.

$\bar{A}_1^B$ = the part of the *updated* tableau of A_1 corresponding to the basic variables in an optimal solution to $\overline{FCP}$, i.e., an identity matrix.

c_1^N = the part of the cost vector c_1 corresponding to the nonbasic x_{ij} in an optimal solution to $\overline{FCP}$ or $\overline{FCNP}$.

u_1^L = the part of the upper bound vector u_1 corresponding to the nonbasic x_{1j} at their lower bounds in an optimal solution to $\overline{FCP}$ or $\overline{FCNP}$.

ℓ_2^U = the part of the lower bound vector ℓ_2 corresponding to the nonbasic x_{2j} at their upper bounds in an optimal solution to $\overline{FCP}$ or $\overline{FCNP}$.

$\nu(FCP)$ = the value of an optimal solution to problem FCP.

$\beta(FCP)$ = the best currently available lower bound on $\nu(FCP)$.

$\nu(\overline{FCP}\text{: } y \equiv 0 \bmod u_1)$ = the value of an optimal solution to problem $\overline{FCP}$ with the added constraints $y \equiv 0 \bmod u_1$, i.e. $\nu(FCP)$.

CHAPTER II

LITERATURE SURVEY

This chapter presents a brief summary of previously reported research related to the subject matter of this dissertation. However, integer programming techniques and fixed charge problems have been the object of a vast amount of research over the past two decades, and any review of this literature will necessarily be highly circumscribed. Thus, the following sections will address only the two categories of research most relevant to the analysis which follows, i.e. methods related to the group theoretic approach to integer programming and algorithms for exact solution of fixed charge problems. Any other literature which may be relevant to particular topics of the dissertation will be cited directly in the discussions of those topics.

2.1 Literature of Group Theoretic and Related Methods

Consider the general linear integer program

$$
\text{(IP)} \qquad \begin{aligned} \text{min.} \quad & h^T z \\ \text{s.t.} \quad & Mz = w \\ & z \geq 0 \\ & z \equiv 0 \bmod 1. \end{aligned}
$$

It is well known that an equivalent statement of IP can be made in terms of an optimal Simplex tableau for its continuous relaxation $\overline{IP}$. In this equivalent form the problem is stated

$$\min \quad (\bar{h}^N)^T z^N \tag{2-1}$$

$$\text{s.t.} \quad \bar{M}^N \quad z^N \leq \bar{w} \tag{2-2}$$

(EIP)

$$\bar{M}^N \quad z^N \equiv \bar{w} \bmod 1 \tag{2-3}$$

$$z^N \geq 0 \tag{2-4}$$

$$z^N \equiv 0 \bmod 1. \tag{2-5}$$

In 1965, Gomory [40] demonstrated that if the constraints (2-2) of EIP are relaxed, the resulting problem can be shown to possess many interesting properties from the point of view of group theory in abstract algebra. In particular, he showed that the fractional parts of the columns of the constraints (2-3), i.e. the fractional parts of $\bar{w}$ and the columns of $\bar{M}^N$, form an Abelian group under modulo 1 addition. Thus EIP with constraints (2-2) removed became known as the *group problem associated with IP* (GP(IP)), and methods related to this formulation are referred to as the *group theoretic approach to integer programming*.

One method for using this result proposed by Gomory, and later developed by Hu [61], Shapiro [82,83], Gorry and Shapiro [45], Hefley and Thomas [52], and others, was to solve GP(IP) as a knapsack problem over the elements of the group connected with the constraints (2-3).

If the resulting solution also satisfies (2-2), then it gives an optimal solution for IP. Otherwise the group problem is resolved for the second best solution, etc. until a solution is found which satisfies (2-2). For some all-integer cases of IP promising results have been reported (e.g. Gorry and Shapiro [45]). However, it appears this approach can not be efficiently extended to the mixed-integer case, i.e. the case where some components of z are not subject to congruence requirements.

2.1.1 Penalty-Oriented Group Theory

A somewhat different application of Gomory's theory, developed more recently by Gomory and Johnson [41,42,43,44,65,66], proceeds from the fact that GP(IP) is a relaxation of the equivalent problem EIP. Thus the set of feasible solutions for GP(IP) includes the set of feasible solutions for IP, and $\nu\big(GP(IP)\big)$ is a lower bound on $\nu(IP)$.

Principally, Gomory and Johnson have dealt with an even more relaxed version of EIP, *one row group problems* formed by (2-1), (2-4), (2-5) and one row from (2-3). For such relaxations of both the all-integer and the mixed-integer cases, they have developed methods for generating inequality constraints on z^N which must be satisfied by every solution to the one row problem. In the best case, these constraints are faces of the convex hull of solutions to the one row group problems.

Such *valid inequalities* from the one row group problems can be used in several ways. First, they are valid cutting planes for the overall problem IP. Some development of this approach is found in Glover [35].

An approach of more direct interest in this dissertation is the

the use of valid inequalities in developing penalties for a branch-and-bound approach to IP. In [41,42,43,44,65,66] Gomory, Johnson and Spielberg suggest a procedure using the fact that the value of an optimal solution to a one row group problem is an estimate of the additional cost or penalty for forcing the basic variable corresponding to the row to take on a value which satisfies its congruence constraint. When the one row group problem cannot be efficiently solved, several valid inequalities may be generated from it and the linear program defined by (2-1), (2-4) and these constraints can be solved instead. If such inequalities closely bound the convex hull of solutions to the one row group problem, the value of an optimal solution to such a linear program will approximately equal the value of an optimal solution to the one row group problem. In any case, the value of an optimal solution to this linear program provides a lower bound on the penalty for bringing the corresponding basic variable into conformance with its congruence constraint.

In the work presented by Gomory , Johnson and Spielberg, and in application of the work by Kennington [68] (see Section 2.2.2.3), these penalties from the one row group problems are used in two ways. First, they provide a lower bound on the value of any solution to the full integer program and can thus be used in a branch-and-bound procedure to limit the number of cases which must be explicitly investigated. In addition, since they provide estimates of the penalty for making any particular basic variable satisfy its congruence constraint, the penalties can be used in the branching rules of the branch-and-bound procedure by indicating which integer variable should be the next to be restricted. Computational results in [66] and [68] indicate such approaches are very promising.

2.1.2 Lagrangean Relaxation Theory

A more comprehensive approach to integer programming, which is related to these group theoretic methods, was first suggested by Everett [23] under the name "generalized Lagrange multipliers." Consider the mathematical program

$$(P) \qquad \begin{aligned} \min \quad & \hat{h}^T\hat{z} \\ \text{s.t.} \quad & \hat{M}\hat{z} \ge \hat{w} \\ & \hat{z} \in \Omega, \end{aligned}$$

where Ω is an arbitrary set, presumably containing integrality or other difficult constraints on $\hat{z}$. Everett observed that for all $\hat{u} \ge 0$, the relaxed problem

$$(P_{\hat{u}}) \qquad \begin{aligned} \min \quad & \hat{h}^T\hat{z} + \hat{u}^T(\hat{w}-\hat{M}\hat{z}) \\ \text{s.t.} \quad & \hat{z} \in \Omega \end{aligned}$$

provides a lower bound on P in the sense that

$$\nu(P) \ge \nu(P_{\hat{u}}).$$

Moreover, when the best possible $\hat{u}$ is chosen, i.e. $\hat{u}$ such that

$$\nu(P_{\hat{u}}) = \max_{\hat{v} \geq 0} \nu(P_{\hat{v}}), \tag{2-6}$$

it may occur that

$$\nu(P_{\hat{u}}) = \nu(P). \tag{2-7}$$

Ullman and Newhauser [78] quickly observed that (2-7) can be satisfied for integer programs only in the trivial case where $\nu(P) = \nu(\bar{P})$. However, the use of $\nu(P_{\hat{u}})$ as a lower bound on $\nu(P)$ remains a valid procedure for all integer programs.

Everett proposed no efficient procedure for finding a $\hat{u}$ satisfying (2-6), but in 1966 a method was presented by Brooks and Geoffrion [11]. Their technique employs Dantzig-Wolfe decomposition [18] to calculate an optimal $\hat{u}$. Sub-problems for the procedure consist of minimizations of a linear objective function subject to the constraints

$$\hat{z} \in \Omega.$$

Use of Lagrangean techniques in connection with the group theoretic approach to integer programming was first suggested by Shapiro and Fisher [25,84]. They proposed including the constraints (2-2) in the objective function of GP(IP) to produce the bounding problem

$$\min \quad (\bar{h}^N)^T z^N + \hat{u}^T(\bar{w}-\bar{M}^N z^N)$$

$$\text{s.t.} \quad \bar{M}^N z^N \equiv \bar{w} \bmod 1 \qquad (2\text{-}8)$$

$$\left(GP(IP)_{\hat{u}}\right) \qquad z^N \geq 0 \qquad (2\text{-}9)$$

$$z^N \equiv 0 \bmod 1. \qquad (2\text{-}10)$$

Using $\Omega = \{z^N : z^N$ satisfies (2-8), (2-9) and (2-10)}, the above theory is applicable, and an optimal $\hat{u}$ [in the sense of (2-6)] can be calculated by the procedure of Brooks and Geoffrion. Sub-problems in the decomposition are the original GP(IP) with varying objective functions. If an optimal $\hat{u}$ is used, Shapiro [84] demonstrates that an improved bound on the value of an optimal solution to IP is obtained, i.e.

$$\nu\left(GP(IP)\right) \leq \nu\left(GP(IP)_{\hat{u}}\right),$$

and the inequality may be strict.

More recently Fisher and Shapiro [26], and Geoffrion [31] have observed that this method for the group theoretic approach, and all other generalized Lagrange multiplier schemes are part of a much larger set of methods derived from Lagrangean duality in nonlinear programming. Geoffrion proposes the generic name *Lagrangean relaxation methods* for this larger class of techniques, and that term has been adopted for this dissertation. Among the other techniques Geoffrion, Fisher and Shapiro show to be part of the class of Lagrangean relaxation methods are Held and Karp's approach to the traveling salesman problem [53], and Fisher

[24] and Fisher and Schrage's [27] scheduling procedures. All these methods employ bounds derived from Lagrangean dual problems like (2-6) in seeking a solution to a primal problem like P.

2.2 Literature of Exact Solution of Fixed Charge Problems

Because of the number of important applications presented in Chapter I which can be formulated as fixed charge problems, the development of solution procedures for general or special fixed charge problems has been a major segment of research in integer programming methods for the past two decades. This research can be classified into two distinct categories according to whether the procedures yield *approximate* or *exact* optimal solutions for the particular fixed charge problem being studied.

Approximate methods [14,15,19,88,95] have been based primarily on two important theoretical properties of FCP. The first is the 1954 result of Hirsch and Dantzig [58] that the objective function of a fixed charge problem is a special case of a concave function, and thus that an optimal solution for any fixed charge problem will occur at an extreme-point of the convex set defined by

$$A_1x_1 + A_2x_2 = b \qquad (2\text{-}11)$$

$$u_1 \geq x_1 \geq 0 \qquad (2\text{-}12)$$

$$u_2 \geq x_2 \geq \ell_2. \qquad (2\text{-}13)$$

This property has led many researchers to develop extreme-point search methods for obtaining approximate solutions to FCP.

A second property concerns the continuous relaxation $\overline{\text{FCP}}$. Balinski [4] observed for the fixed charge transportation problem, and Gray [46] extended for general fixed charge problems the result that an optimal solution to $\overline{\text{FCP}}$ can be obtained without explicit consideration of all its variables or constraints. It is sufficient to consider only the reduced problem

$$
\text{(RP)} \quad
\begin{aligned}
&\min \ c_1^T x_1 + c_2^T x_2 \\
&\text{s.t. } A_1 x_1 + A_2 x_2 = b \\
&\qquad u_1 \ge x_1 \ge 0 \\
&\qquad u_2 \ge x_2 \ge \ell_2 .
\end{aligned}
$$

Because all components of c_s in FCP are strictly positive, every optimal solution to the $\overline{\text{FCP}}$ must have $x_1 = y$, $s = 0$. Thus $\nu(\text{RP}) = \nu(\overline{\text{FCP}})$, and an approximate solution for FCP can be obtained by rounding to the solution

$$
\begin{aligned}
y &= \bar{x}_1 \\
s &= u_1 - \bar{x}_1 \\
x_1 &= \bar{x}_1 \\
x_2 &= \bar{x}_2 ,
\end{aligned}
$$

where $\begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \end{pmatrix}$ is an optimal solution to RP.

The solution procedures developed in this dissertation are exact methods, i.e. they produce an exact optimum solution to FCP or FCNP. Thus the literature of exact solution procedures is the most relevant to the research reported herein, and no further detail will be presented on approximate methods. For more complete summaries of the approximate solution literature, see [12,28,46,51,68,71,88,91].

2.2.1 General Fixed Charge Problems

Four researchers are known to have proposed exact solution methods for the general fixed charge problem. The first was Murty [76] who suggested an extreme-point ranking procedure based on Hirsch and Dantzig's result that an optimal solution will occur at an extreme-point of the convex set defined by (2-11), (2-12) and (2-13). The method searches such extreme-points in decreasing order of variable cost, i.e. in decreasing order of value of the objective function

$$v^T x_1 + c_2^T x_2. \tag{2-14}$$

A pivoting scheme is provided for moving from the least cost to the next least cost extreme-point, etc., and the procedure terminates when bounds on the fixed charges indicate that no lower ranked extreme-point could produce an improved solution to the overall fixed charge problem. Murty suggests the procedure will work best on nondegenerate problems, but no computational experience for general fixed charge problems has been reported. Experience with fixed charge transportation problems, reported in [46,76,77], suggests Murty's procedure may be effective for problems with the dimension of y less than 100.

In 1967 Gray [46] reported a decomposition approach for fixed charge problems. The method uses Hillier's bound-and-scan technique [57] to develop specific values of the y vector to be investigated. The corresponding version of $\overline{FCP}$ with y fixed at the assigned value is then solved. A set of clever bounds on the fixed and variable costs of an optimal solution to FCP, and limitations on the number of positive components of x_1 implied by Hirsch and Dantzig's result are used to restrict the number of y vectors which must be explicitly evaluated. Computational experience for both a general algorithm and a specialized procedure for the fixed charge transportation problem is presented in [46]. The method appears effective for general problems with y dimensions up to 30 and transportation problems with y's of up to 50 components.

Steinberg [89] presented a direct branch-and-bound method for general fixed charge problems in 1969. In effect, his procedure enumerates the possible values of y by fixing certain components y_j and testing if any setting of the remaining components could provide an optimal solution for FCP. Rules for testing such partial solutions include restrictions like Gray's on the number of positive x_{1j} and a cost bound obtained by minimizing (2-14) subject to (2-11), (2-12), and (2-13), and adding a lower bound on fixed charges for components of y not presently fixed. Computational success is reported for general fixed charge problems with y dimensions up to 30.

Jones and Soland [67] also presented a branch-and-bound procedure for general fixed charge problems in 1969. In fact their research addressed fixed charge formulations far more general than FCP, involving nonlinear constraints and objective functions and "multi-level fixed charge" components, i.e. activities for which different fixed charges are applicable at different activity levels. For the case of problems like FCP, however, Jones and Soland's method reduces to a fixing selected components of y and solving the corresponding version of RP to obtain a cost bound. If the value of an optimal solution to such a RP exceeds the value of a known feasible solution for FCP, no setting of the remaining components of y can produce an optimal solution and consideration of the partial solution can be terminated. No problem with more than 15 multi-level fixed charge components was solved.

2.2.2 Fixed Charge Network Problems

Even though a great deal of research has been devoted to solution of the warehouse location and fixed charge transportation cases, no instance is known in the literature of an exact solution procedure for general fixed charge network problems. The only relevant material found as a part of this research is work by Zavarei and Frish [97] which demonstrates that every fixed charge network can be formulated as a fixed charge transportation problem at the expense of increasing the number of nodes and arcs. Thus, at least theoretically, any fixed charge transportation algorithm can be viewed as an algorithm for general fixed charge network problems.

2.2.2.1 Warehouse Location Problems. In contrast to the situation for general network problems, a considerable amount of attention has been devoted to solution of the special case defined as the warehouse location problem in Section 1.2.2. The most important exact solution developments will be summarized in the following discussion, but more complete reviews are found in [12,46,71].

The first important exact solution method for the warehouse location problem was reported by Effroymson and Ray [21] in 1966. Their procedure, which is applicable only to the case where supplies at each warehouse are assumed infinite, is a branch-and-bound technique depending heavily on solution of a version of $\overline{FCNP}$ at each step to develop bounds for eliminating partial y solutions. However, the fact that $\overline{FCNP}$ can be solved trivially in this uncapacitated case is exploited in a set of post-optimality operations which improve the quality of the bounds. Criteria for determining that free y_j must take on given values in completions of the current partial solution are also tested, and if any new y_j are fixed, $\overline{FCNP}$ is reoptimized with the additional y restrictions enforced. Computationally, the authors report solving a problem with the dimension of y equal to 50.

Spielberg [86,87] presented several procedures which use essentially the same bounds as Effroymson and Ray, but a different enumeration procedure. Instead of fixing some components of y and attempting to determine that no setting of the remaining components can produce an optimal solution, his method systematically investigates a series of completely fixed y vectors. A number of sequences of y vectors were tried, but none proved clearly superior, and computational times were comparable to Effroymson and Ray's.

Ellwein [22] applied a specialized version of Benders' decomposition procedure [10] to capacitated warehouse location problems. Taking advantage of Gray's bounds, Effroymson and Ray's opening and closing rules, and bounds produced by linear sub-problems of the Benders' method, Ellwein's algorithm implicitly enumerates candidate vectors y until an optimal solution is identified. Computational success is reported for y's of dimension up to 25.

Bulfin and Unger [12,13] also reported a procedure using some of the previous results in the context of Benders' decomposition algorithm [10]. An implicit enumeration procedure like Geoffrion's [29] is used in a master integer problem, with the development of new partial solutions partially controlled by solution of Benders' continuous sub-problems. Ellwein's bounds are employed in the enumeration procedure to reduce the number of partial solutions explicitly investigated. Computational experience in [12] indicates the procedure can efficiently solve capacitated problems with y's of dimension up to 25. Experience reported in [13] for a separate version of the algorithm adapted for the uncapacitated case, shows solution of problems with y's as large as 50 in a few seconds.

2.2.2.2 Fixed Charge Transportation Problems. As was the case for warehouse location problems, fixed charge transportation problems have been the subject of extensive research. Thus, only the most important developments can be summarized here. More complete literature surveys are provided in [28,46,51,68,91].

The first important exact solution method for fixed charge transportation problems was developed by Speilberg [85] in 1964. The method is a direct implementation of Benders' partitioning [10] on the problem, with a branch-and-bound method used to solve integer sub-problems. Speilberg states that the method is efficient for y's as large as 150, but discouraging for any larger problems.

As mentioned above, the procedures developed by Murty [76] and Gray [46] for more general problems have been specialized to the fixed charge transportation problem. However, since no important differences in approach are used for the transportation case their methods will not be repeated here.

The method developed by Frank [28] builds on Murty's approach. Murty's ranking of extreme-points in terms of variable cost is used along with a ranking of fixed costs obtained from a modification of the defender algorithm (see Bellmore and Ratliff [9]). Three different schemes for combining these two rankings were investigated, but no problems with y's larger than 16 were solved.

The first known group theoretic investigation of any fixed charge problem is the work of Tompkins [91]. He identified a number of properties of the Abelian group associated with fixed charge transportation problems. A branch-and-bound procedure was developed which took advantage of some of these properties to reduce the number of solutions explicitly evaluated, but no computational results have been presented.

Hefley and Thomas [52] have reported experimenting with a more direct group theoretic approach. They derive and solve the group problem for a fixed charge transportation problem as a knapsack problem over the associated group. Simplifications derived from Hefley's methods of group theoretic decomposition [50] are exploited to reduce the difficulty of solving the group problem. Problems with y's as large as 24 are reported solved by this procedure.

The most recent group theoretic approach to fixed charge problems is the work of Kennington [68]. Since Kennington's research provided the starting point for much of the analysis reported in this dissertation, it is discussed in some detail in the following section.

2.2.2.3 Kennington's Group Theoretic Approach. Kennington's [68] approach to the fixed charge transportation problem is a penalty-oriented group theoretic approach using the principles presented in Section 2.1.1. Thus a branch-and-bound algorithm is used with penalties developed from group problems providing cost bounds and guiding the selection of new branching variables.

Kennington formulates the problem as an all-integer program, i.e. requires $x_1 \equiv 0 \bmod 1$ in addition to the other constraints of FCNP, and demonstrates that the resulting group problem congruence constraints will contain only 0, +1 and -1 coefficients. A large number of simplifications in the group penalty approach then follow from this special problem structure. Among these are the following:

1. The group problem associated with a fixed charge transportation problem can be efficiently constructed from the optimal tableau of the reduced problem, RP. Thus the efficiency of solving RP instead of $\overline{\text{FCNP}}$ at each step in the branch-and-bound algorithm can be extended to group theoretic approaches.

2. In contrast to the usual case, one row group problems for the fixed charge transportation problem can easily be solved exactly. Thus one row group penalties can be derived without first resorting to the generation of a set of valid inequalities.

3. As an additional aid in using the one row group penalties for branching rules, it is possible to interpret such penalties as costs of moving y_j "up" to u_{1j} or "down" to 0 instead of simply to some value satisfying the congruence constraint.

4. One row group penalties for the fixed charge transportation problem are exactly equal to those developed from different approaches by Tomlin [90].

5. The constraint matrix of the group problem associated with a fixed charge transportation problem is totally unimodular.

Computational experience for Kennington's algorithm is presented in [68]. The results show efficient solution of fixed charge transportation problems with y's of dimension up to 100.

CHAPTER III

GROUP RELATED STRUCTURE OF FIXED CHARGE PROBLEMS

In this chapter a number of structural characteristics related to the group problem for general fixed charge problems are investigated. Most of Kennington's results for the fixed charge transportation problem (see Section 2.2.2.3) are shown to generalize to all fixed charge problems, and new concepts involving penalty problems stronger than one row group problems are presented. Before proceeding with these results, however, the general solution framework to which the results are relevant will be briefly reviewed.

Recall from Section 1.4 the conventions

β (any minimization problem)	= the best currently available lower bound on the value of an optimal solution to the problem,
ν (any unbounded problem)	= $-\infty$,
ν (any infeasible problem)	= $+\infty$, and let
ν (the best known solution to FCP)	= ν^*.

Then Figure 1 presents a brief flow chart of the branch-and-bound approach to fixed charge problems which provides the motivation for the research of this dissertation. The principal steps in the procedure can be summarized as follows:

<u>*Step 0.*</u> Place the whole problem FCP in the *candidate list* (i.e. in the set of restricted versions of FCP which might still yield

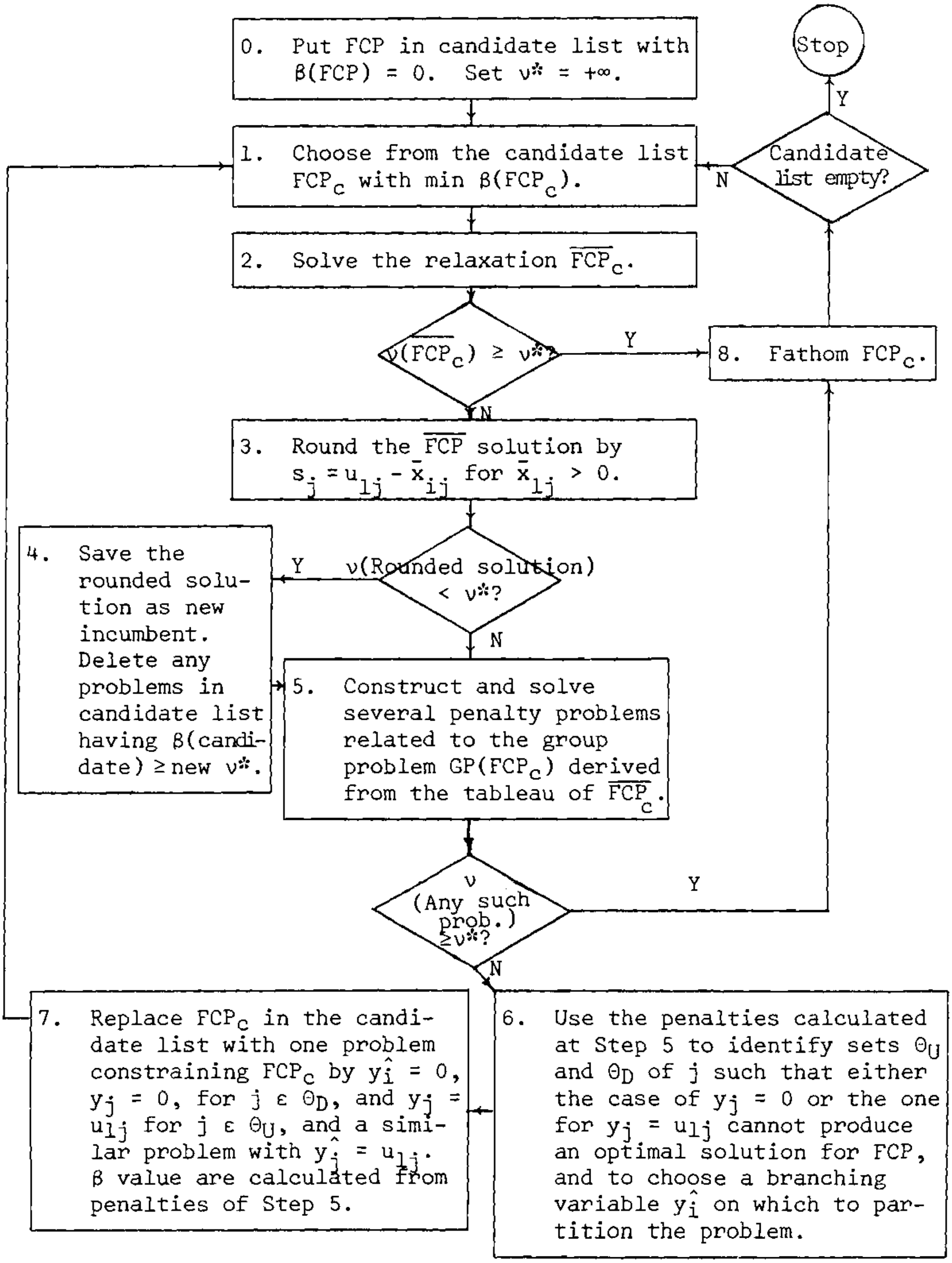

Figure 1. Flow Chart of General Branch-and-Bound Approach

an optimal solution to the full problem). Set $\beta(FCP) = 0$ and $v^* = +\infty$, and proceed to Step 1.

Step 1. Choose as the current candidate, FCP_c, the element of the candidate list satisfying

$$\beta\{(FCP_c) = \min \{\beta(FCP_{c'}): \ FCP_{c'} \text{ in candidate list}\},$$

and proceed to Step 2.

Step 2. Solve the continuous relaxation of FCP_c, i.e. $\overline{FCP}_c$. If $v(\overline{FCP}_c) \geq v^*$, proceed to Step 8 because no *completion* of FCP_c (i.e. no setting of the y_j not assigned values in FCP_c) can produce a solution to FCP with value less than that of a known solution. If $v(\overline{FCP}_c) < v^*$, proceed to Step 3.

Step 3. Create a feasible solution for FCP by rounding "up" the optimal solution to $\overline{FCP}_c$, i.e. by setting

$$s_j = \begin{cases} u_{1j} - \bar{x}_{1j} & \text{if } \bar{x}_{1j} > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$y = s + \bar{x}_1$$

$$x_1 = \bar{x}_1$$

$$x_2 = \bar{x}_2,$$

where $\bar{x}_1$ and $\bar{x}_2$ are the optimal values of x_1 and x_2 in the solution of

$\overline{FCP}_c$. If the value of this rounded solution is less than ν^*, proceed to Step 4. Otherwise go to Step 5.

Step 4. A new *incumbent solution* has been found, i.e. the rounded solution to $\overline{FCP}_c$ provides a feasible solution to FCP with value less than any solution found so far. Save this incumbent as a possible optimal solution, and eliminate from the candidate list any problems with β value greater than or equal to the value of the new incumbent. If the new $\nu^* = -\infty$, stop; FCP is unbounded. Otherwise, proceed to Step 5.

Step 5. Attempt to obtain a better lower bound on $\nu(FCP_c)$ than $\nu(\overline{FCP}_c)$ by constructing and solving several penalty problems related to the group problem $GP(FCP_c)$ derived from the optimal tableau of $\overline{FCP}_c$. If the value of the optimal solutions to any of these penalty problems is greater than or equal to ν^*, proceed to Step 8. Each problem is a relaxation of FCP_c, and thus has optimal solution value less than or equal to $\nu(FCP_c)$. If all penalty problem solutions have value less than ν^*, proceed to Step 6.

Step 6. Use the penalty information of Step 5 to identify elements of the sets

$$\Theta_U = \{j: \text{ penalties show } \nu(FCP_c\text{: } y_j = 0) \geq \nu^*\}$$

$$\Theta_D = \{j: \text{ penalties show } \nu(FCP_c\text{: } y_j = u_{1j}) \geq \nu^*\}.$$

Also use the group-related penalties of Step 5 to choose a new *branching variable* $y_{\hat{i}}$ on which to *partition* the problem, i.e. choose a new $y_{\hat{i}}$ to fix in FCP_c such that $\hat{i} \notin \Theta_U \cup \Theta_D$. Then go to Step 7.

Step 7. Replace FCP_c in the candidate list by two more restricted problems. One is defined by FCP_c with the additional constraint that the branching variable $y_{\hat{i}} = 0$, $y_j = 0$ for $j \varepsilon \Theta_D$ and $y_j = u_{1j}$ for $j \varepsilon \Theta_U$. The other problem is identical except that $y_{\hat{i}}$ is restricted to equal its upper bound. β values for these two new candidates are as obtained from the penalty problems of Step 5. Next proceed to Step 1 to select a new FCP_c.

Step 8. *Fathom* FCP_c, i.e. eliminate FCP_c from the candidate list because no completion of it can produce a feasible solution to FCP with value less than v^*. If the candidate list is now empty, stop; if an incumbent solution exists, it is an optimal solution for FCP, and otherwise FCP is infeasible. If the candidate list is not empty, proceed to Step 1 to select a new FCP_c.

From Figure 1 and the above discussion it is evident that in the context of such a branch-and-bound procedure relevant problem structure and computational simplifications include the following:

1. Efficient procedures for solving continuous problems $\overline{FCP}_c$.

2. Convenient methods for constructing $GP(FCP_c)$ from the optimal solution to $\overline{FCP}_c$.

3. Usable characterizations of penalty problems most likely to give the highest cost bounds.

4. Efficient algorithms for construction and solution of penalty problems.

5. Improved procedures for using penalty information in selecting branching variables.

The remainder of this dissertation presents a number of results relating to such considerations.

3.1 Solution of the Continuous Relaxation

As formulated in Section 1.1, the problem FCP is equivalent to

$$\text{(FCP)}\quad \begin{array}{lllr}
\min & c_1^T x_1 + c_2^T x_2 + c_s^T s & & (3\text{-}1)\\
\text{s.t.} & A_1 x_1 + A_2 x_2 & = b & (3\text{-}2)\\
& -Ix_1 + Iy - Is & = 0 & (3\text{-}3)\\
& u_1 \ge y & \ge 0 & (3\text{-}4)\\
& u_1 \ge x_1 & \ge 0 & (3\text{-}5)\\
& u_1 \ge s & \ge 0 & (3\text{-}6)\\
& u_2 \ge x_2 & \ge \ell_2 & (3\text{-}7)\\
& y \equiv 0 \bmod u_1, & & (3\text{-}8)
\end{array}$$

with $c_s > 0$. This section investigates procedures for simplified solution of the continuous relaxation $\overline{\text{FCP}}$, i.e. the version of FCP without constraints (3-8).

3.1.1 Construction of a Solution from the Reduced Problem

As indicated in Section 2.2, Balinski [4] and Gray [46] have shown that $v(\overline{\text{FCP}})$ is equal to the value of an optimal solution to the *reduced problem*

$$\text{(RP)} \qquad \begin{aligned} \min \quad & c_1^T x_1 + c_2^T x_2 \\ \text{s.t.} \quad & A_1 x_1 + A_2 x_2 = b \\ & u_1 \geq x_1 \geq 0 \\ & u_2 \geq x_2 \geq \ell_2. \end{aligned}$$

Thus the value of an optimal solution to $\overline{FCP}$ can be obtained without explicit consideration of all its constraints and variables. Through a series of lemmas, some elaborations on this result will now be developed.

<u>3.1.1.1 Lemma.</u> If vectors x_1^B and x_2^B form a basis for RP, then these vectors together with the vector y form a basis for $\overline{FCP}$.

Proof. The indicated vectors will form a basis for $\overline{FCP}$ if the corresponding columns in the constraints (3-2) and (3-3) form a linearly independent set. Consider a linear combination (z_1,z_2,z_y) of these columns such that

$$\begin{pmatrix} A_1^B \\ -I \\ 0 \end{pmatrix} z_1 + \begin{pmatrix} A_2^B \\ \\ 0 \end{pmatrix} z_2 + \begin{pmatrix} 0 \\ \\ I \end{pmatrix} z_y = 0.$$

Now it must be that $z_1 = z_2 = 0$ because $\{x_1^B,x_2^B\}$ is a basis for RP and so the columns (A_1^B,A_2^B) are linearly independent. But then it must also be true that $z_y = 0$ because no other combination of unit vectors can produce the zero vector. Thus the only linear combination of the columns corresponding to x_1^B, x_2^B and y which yields the zero vector has all zero coefficients, and the columns are linearly independent.

Q.E.D.

<u>3.1.1.2 Lemma.</u> The updated Simplex tableau for $\overline{FCP}$ corresponding to basis vectors x_1^B, x_2^B and y is (after renumbering of rows and columns as indicated in Section 1.4.2) given by

	x_1^B	x_1^N	x_2^B	x_2^N	y	s	RHS
	0	$(\hat{c}_1^N)^T$	0	$(\hat{c}_2^N)^T$	0	c_s^T	–
x_1^B	I	$\hat{A}_1^N$	0	$\hat{A}_2^N$	0	0	$\hat{b}$
x_2^B	0		I				
y	0	$[\hat{A}_1^N]^1_{k^B}$	0	$[\hat{A}_2^N]^1_{k^B}$	I	-I	$[\hat{b}]^1_{k^B}$
		-I		0			0

where k^B is the dimension of x_1^B and the updated tableau of RP corresponding to the basis vectors x_1^B and x_2^B is given by

	x_1^B	x_1^N	x_2^B	x_2^N	RHS
	0	$(\hat{c}_1^N)^T$	0	$(\hat{c}_2^N)^T$	-
x_1^B	I	$\hat{A}_1^N$	0	$\hat{A}_2^N$	$\hat{b}$
x_2^B	0		I		

Proof. The basis matrix corresponding to x_1^B, x_2^B and y in $\overline{FCP}$ is of the form

$$\left(\begin{array}{cc|c} A_1^B & A_2^B & 0 \\ \hline -I & 0 & \\ 0 & 0 & I \end{array}\right).$$

The inverse of this matrix can be calculated by the well-known formula for inversion by minors as

$$\left(\begin{array}{c|c} \begin{pmatrix} A_1^B & A_2^B \end{pmatrix}^{-1} & 0 \\ \hline \left[\begin{pmatrix} A_1^B & A_2^B \end{pmatrix}^{-1}\right]^{1}_{k^B} & \\ 0 & I \end{array}\right).$$

Direct multiplication of this inverse times the original constraint matrix of $\overline{FCP}$ produces the updated constraint matrix indicated in the Lemma when it is recognized that (A_1^B, A_2^B) is the corresponding basis matrix for RP. For example, consider the lower part of the columns corresponding to x_1^N in the constraint matrix of $\overline{FCP}$. Multiplication of the appropriate submatrices yields

$$\begin{pmatrix} \left[\left(A_1^B \;,\; A_2^B\right)^{-1}\right]^1_{k^B} & \vdots \\ \hline 0 & I \end{pmatrix} \begin{pmatrix} A_1^N \\ \hline 0 \\ -I \end{pmatrix} = \begin{pmatrix} [\hat{A}_1^N]^1_{k^B} + 0 \\ \hline 0 \quad - I \end{pmatrix} = \begin{pmatrix} [\hat{A}_1^N]^1_{k^B} \\ \hline -I \end{pmatrix}.$$

Calculation of the updated cost row for nonbasic variables is similar. In terms of the updated constraint matrix the costs are

$$\begin{pmatrix} c_1^N \\ c_2^N \\ c_s \end{pmatrix} - \begin{pmatrix} \hat{A}_1^N & \hat{A}_2^N & 0 \\ \hline [\hat{A}_1^N]^1_{k^B} & [\hat{A}_2^N]^1_{k^B} & -I \\ -I & 0 & \end{pmatrix}^T \begin{pmatrix} c_1^B \\ c_2^B \\ 0 \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} c_1^N \\ c_2^N \end{pmatrix} - \begin{pmatrix} \hat{A}_1^N & \hat{A}_2^N \end{pmatrix}^T \begin{pmatrix} c_1^B \\ c_2^B \end{pmatrix} \\ \hline c_s \end{pmatrix} = \begin{pmatrix} \hat{c}_1^N \\ \hat{c}_2^N \\ c_s \end{pmatrix}.$$

Q.E.D.

<u>3.1.1.3 Lemma.</u> Let the values of x_1 and x_2 corresponding to the basis $\{x_1^B, x_2^B\}$ in RP be $\hat{x}_1$ and $\hat{x}_2$, respectively. Then the values of x_1, x_2, y and s corresponding to the basis $\{x_1^B, x_2^B, y\}$ in $\overline{FCP}$ are given by

$x_1 = y = \hat{x}_1$, $x_2 = \hat{x}_2$, $s = 0$.

Proof. First consider the nonbasic variables in $\overline{FCP}$. The components of $\hat{x}_1^N$ and $\hat{x}_2^N$ must exactly equal the upper or lower bounds on x_1^N and x_2^N because they were part of a basic solution to RP. Similarly, the lower bound 0 provides a value which can correspond to a basic solution for $\overline{FCP}$. Thus it remains only to show that the values of basic variables implied by these nonbasic values are those given in the statement of the Lemma. Using the updated tableau of Lemma 3.1.1.2, the basic variables of $\overline{FCP}$ can be calculated

$$\begin{pmatrix} x_1^B \\ \\ x_2^B \end{pmatrix} = \hat{b} - \hat{A}_1^N \hat{x}_1^N - \hat{A}_2^N \hat{x}_2^N - 0 = \begin{pmatrix} \hat{x}_1^B \\ \\ \hat{x}_2^B \end{pmatrix},$$

where the last equality follows because the center expression is identical to the one for the values of basic variables in RP. Similarly,

$$y = \begin{pmatrix} [\hat{b}]^1{}_{k^B} \\ \\ 0 \end{pmatrix} - \begin{pmatrix} [\hat{A}_1^N]^1{}_{k^B} \\ \\ -I \end{pmatrix} \hat{x}_1^N - \begin{pmatrix} [\hat{A}_2^N]^1{}_{k^B} \\ \\ 0 \end{pmatrix} \hat{x}_2^N = \begin{pmatrix} \hat{x}_1^B \\ \\ \hat{x}_1^N \end{pmatrix}.$$

Q.E.D.

<u>3.1.1.4 Theorem.</u> Let $\{x_1^B, x_2^B\}$ be a basis which satisfies Simplex optimality criteria for RP, and let $x_1 = \bar{x}_1$, $x_2 = \bar{x}_2$ be the corresponding

optimal basic solution. Then $\{x_1^B, x_2^B, y\}$ is an optimal basis for $\overline{FCP}$ yielding the optimal basic solution $x_1 = y = \bar{x}_1$, $x_2 = \bar{x}_2$, $s = 0$.

Proof. That $\{x_1^B, x_2^B, y\}$ is a basis for $\overline{FCP}$ yielding the indicated solution follows from Lemmas 3.1.1.1 and 3.1.1.3. It remains only to show that this basis is optimal for $\overline{FCP}$. By Lemma 3.1.1.2, the adjusted cost row of the $\overline{FCP}$ tableau corresponding to the indicated basis is given by

x_1^B	x_1^N	x_2^B	x_2^N	y	s
0	$(\bar{c}_1^N)^T$	0	$(\bar{c}_2^N)^T$	0	c_s^T

,

where $\bar{c}_1^N$ and $\bar{c}_2^N$ are extracted directly from the optimal tableau of RP. $\bar{c}_1^N$ and $\bar{c}_2^N$ must satisfy Simplex optimality criteria for the solution $x_1^N = \bar{x}_1^N$, $x_2^N = \bar{x}_2^N$ in $\overline{FCP}$ because they were optimal in RP. Moreover, $c_s > 0$ implies that Simplex optimality criteria are also satisfied for the lower-bounded vector s. Thus Simplex optimality criteria hold for all nonbasic variables, and the theorem is proved.

Q.E.D.

<u>3.1.1.5 Corollary</u>. For $\overline{FCP}$ and RP as defined above, $\nu(\overline{FCP}) = \nu(RP)$.

Proof. If there is an optimal solution to RP, there must be an optimal basic solution satisfying Simplex optimality criteria. Let $x_1 = \bar{x}_1$, $x_2 = \bar{x}_2$ be such a basic solution. Then by Theorem 3.1.1.4, $x_1 = y = \bar{x}_1$, $x_2 = \bar{x}_2$, $s = 0$ is an optimal solution to $\overline{FCP}$. Thus

$$v(\overline{FCP}) = c_1^T\bar{x}_1 + c_2^T\bar{x}_2 + 0^T\bar{x}_1 + c_s^T(0) = c_1^T\bar{x}_1 + c_2^T\bar{x}_2 = v(RP).$$

Q.E.D.

The principal implication of the results presented in this section is that solution of the reduced problem RP can not only provide the value of an optimal solution for the continuous relaxation $\overline{FCP}$, but an optimal basis and the associated Simplex tableau as well. All such elements of an optimal basic solution for $\overline{FCP}$ can be obtained by direct extraction from corresponding elements of the solution of RP, and no additional computations are required.

3.1.2 Alternative Constructions

Section 3.1.1 developed one construction for creating an optimal basic solution to $\overline{FCP}$ from an optimal basic solution to RP. Clearly, if $x_1 = \bar{x}_1$, $x_2 = \bar{x}_2$ is an optimal solution to RP, then $x_1 = y = \bar{x}_1$, $x_2 = \bar{x}_2$, $s = 0$ must be the corresponding optimal solution for $\overline{FCP}$ because $c_s > 0$. However, more than one basis for $\overline{FCP}$ may be able to provide this solution and satisfy Simplex optimality criteria.

Consider first the constructed part of the $\overline{FCP}$ basis corresponding to an element x_{1j}^B, i.e. a component of x_1 which was basic in the optimal solution to RP. Assuming that solution to RP is non-degenerate,[1]

$$u_{1j}^B > \bar{x}_{1j}^B > 0.$$

But then the optimal value of the corresponding y component, $\bar{y}_j$, must

[1] See Appendix B for further discussion of basic solutions in bounded Simplex procedures.

satisfy

$$u^B_{1j} > \bar{y}_j > 0,$$

and y_j must be basic in the solution of $\overline{\text{FCP}}$. Thus, at least in the non-degenerate case, y_j must always be basic in $\overline{\text{FCP}}$ whenever x_{1j} is basic in RP.[1]

Even in the non-degenerate case, however, alternatives are possible for the part of the constructed basis corresponding to x^N_1, the part of x_1 which was nonbasic in the optimal solution to RP. Under some conditions, the y_i corresponding to such x^N_{1j} may be replaced in the basis by either x^N_{1j} or s_i. The following theorem enumerates the possibilities.

<u>3.1.2.1 Theorem.</u> Let $\{x^B_1, x^B_2\}$ be an optimal basis for RP which satisfies Simplex optimality and yields the optimal tableau

	x^B_1	x^N_1	x^B_2	x^N_2	RHS
	0	$(\bar{c}^N_1)^T$	0	$(\bar{c}^N_2)^T$	-
x^B_1	I	$\bar{A}^N_1$	0	$\bar{A}^N_2$	$\bar{b}$
x^B_2	0		I		

.

[1]It should be noted that alternative bases are possible if $x^B_{1j} = 0$ or $x^B_{1j} = u^B_{1j}$, but such cases have been omitted in the interest of brevity.

Further, let y^B, s^B and c_s^B be the portions of the y, s and c_s vectors corresponding to x_1^B, and let y^N, s^N and c_s^N be the portions corresponding to x_1^N. Then x_1^B, x_2^B, y^B, together with a constructed vector z^N form an optimal basis for $\overline{FCP}$ which satisfies Simplex optimality criteria if and only if z^N is created according to the following rules:

(i) If $x_{1j}^N = 0$ in the solution to RP, then

$$z_j^N = \begin{cases} y_j^N \quad \text{or } x_{1j}^N & \text{if } (\bar{c}_{1j}^N - c_{sj}^N) < 0 \\ y_j^N,\ s_j^N \text{ or } x_{1j}^N & \text{if } (\bar{c}_{1j}^N - c_{sj}^N) = 0 \\ y_j^N \quad \text{or } s_j^N & \text{if } (\bar{c}_{1j}^N - c_{sj}^N) > 0. \end{cases}$$

(ii) If $x_{1j}^N = u_{1j}^N$ in the solution to RP, then

$$z_j^N = y_j^N \text{ or } x_{1j}^N.$$

Proof. By Lemma 3.1.1.2 the optimal tableau for $\overline{FCP}$ corresponding to the basis $\{x_1^B, x_2^B, y^B, y^N\}$ is given by

	x_1^B	x_1^N	x_2^B	x_2^N	y^B	y^N	s^B	s^N	RHS
	0	$(\bar{c}_1^N)^T$	0	$(\bar{c}_2^N)^T$	0	0	$(c_s^B)^T$	$(c_s^N)^T$	-
x_1^B	I	$\bar{A}_1^N$	0	$\bar{A}_2^N$	0	0	0	0	$\bar{b}$
x_2^B	0		I						
y^B	0	$[\bar{A}_1^N]^1_{k^B}$	0	$[\bar{A}_2^N]^1_{k^B}$	I	0	-I	0	$[\bar{b}]^1_{k^B}$
y^N	0	-I	0	0	0	I	0	-I	0

(3-15)

Theorem 3.1.1.4 demonstrated that Simplex optimality criteria are satisfied for this basis, and thus $z_j^N = y_j^N$ is a valid choice under all conditions.

If pivot is performed to replace y^N in this basis by s^N, the resulting updated Simplex tableau is given by

	x_1^B	x_1^N	x_2^B	x_2^N	y^B	y^N	s^B	s^N	RHS
	0	$(\bar{c}_1^N - c_s^N)^T$	0	$(\bar{c}_2^N)^T$	0	$(c_s^N)^T$	$(c_s^B)^T$	0	-
x_1^B	I	$\bar{A}_1^N$	0	$\bar{A}_2^N$	0	0	0	0	$\bar{b}$
x_2^B	0		I						
y^B	0	$[\bar{A}_1^N]^1_{k^B}$	0	$[\bar{A}_2^N]^1_{k^B}$	I	0	-I	0	$[\bar{b}]^1_{k^B}$
s^N	0	I	0	0	0	-I	0	I	0

(3-16)

Note that the adjusted cost for y^N in this tableau is the positive vector c_s^N. Thus, Simplex optimality criteria can be satisfied only if $y^N = 0$ in the optimal solution to $\overline{FCP}$. But then $x_1^N = 0$ because $y^N = x_1^N$ in any optimal solution to $\overline{FCP}$. Next, observe that the adjusted cost for x_1^N in the tableau is $(\bar{c}_1^N - c_s^N)$. For $x_1^N = 0$, this value can satisfy Simplex optimality criteria only if it is non-negative. Therefore it can be concluded that $z_j^N = s_j^N$ provides a basis satisfying Simplex optimality criteria if and only if $x_{1j}^N = 0$ and $(\bar{c}_{1j}^N - c_{sj}^N) \geq 0$.

Similarly, if a pivot is performed to replace s^N in the basis by x_1^N, the resulting updated Simplex tableau is given by

	x_1^B	x_1^N	x_2^B	x_2^N	y^B	y^N	s^B	s^N	RHS
	0	0	0	$(\bar{c}_2^N)^T$	0	$(\bar{c}_1^N)^T$	c_s^B	$-(\bar{c}_1^N - c_s^N)^T$	
x_1^B	I	0	0	$\bar{A}_2^N$	0	$\bar{A}_1^N$	0	$-\bar{A}_1^N$	$\bar{b}$
x_2^B	0		I						
y^B	0	0	0	$[\bar{A}_2^N]^1_{k^B}$	I	$[\bar{A}_1^N]^1_{k^B}$	-I	$-[\bar{A}_1^N]^1_{k^B}$	$[\bar{b}]^1_{k^B}$
x_1^N	0	I	0	0	0	-I	0	I	0

(3-17)

Examination of the adjusted cost row for this case will show that Simplex optimality criteria will be satisfied whenever $-(\bar{c}_1^N - c_s^N) \geq 0$. Two situations can produce this result. If $x_{1j}^N = 0$, $-(\bar{c}_{1j}^N - c_{sj}^N) \geq 0$ must be explicitly required. However, if $x_{1j}^N = u_{1j}^N$, then $\bar{c}_{1j}^N \leq 0$ by optimality of RP. Since $c_{sj} > 0$, it will always be true that $-(\bar{c}^N_{1j}-c^N_{sj}) > 0$. Therefore it can be concluded that $z^N_j = x^N_{1j}$ produces a basis satisfying Simplex optimality criteria for $\overline{FCP}$ if and only if either $x^N_{1j} = 0$ in the optimal solution for RP and $(\bar{c}^N_{1j}-c^N_{sj}) \leq 0$, or $x^N_{1j} = u^N_{1j}$ in the optimal RP solution.

Q.E.D.

3.2 Group Formulation of the Problem

As briefly indicated in Section 2.1, the group problem associated with an integer program IP [i.e. GP(IP)] is a relaxed version of the equivalent form of the original problem obtained by expressing the problem in terms of an optimal Simplex tableau for its continuous relaxation $\overline{IP}$. Non-negativity constraints on the basic variables in the optimal solution to $\overline{IP}$ are relaxed, but all other requirements of the original problem are retained. The problem is expressed in terms of the nonbasic variables in the optimal solution of $\overline{IP}$, and thus GP(IP) can be viewed as the problem of finding a minimum cost perturbation of the nonbasic variables which will satisfy all congruence constraints of IP while conforming to non-negativity on the nonbasic variables in some optimal solution to $\overline{IP}$.

In this section the group problem associated with FCP will be carefully developed. Before proceeding to this development, however, a slight extension of Gomory's original definition of the group problem must be presented.

As developed by Gomory [40] and summarized in Section 2.1, the group formulation is established only for integer programming problems with zero lower bounds and no implicitly handled upper bounds on decision variables. Moreover, the formulation is expressed in terms of elements of the optimal tableau associated with the ordinary Simplex method. The following are the concepts necessary to generalize this group formulation to the case of FCP, i.e. to problems with implicit upper and lower bounds which have been solved by bounded Simplex procedures:

1. The group problem is viewed as the problem of finding the minimum cost perturbation of the optimum values of nonbasic variables in $\overline{FCP}$ which satisfies all congruence constraints of FCP while conforming to lower bounds on nonbasic variables which were lower-bounded in the optimal $\overline{FCP}$ solution and to upper bounds on nonbasic variables which were upper-bounded in the optimal $\overline{FCP}$ solution. Thus the decision variables are not the nonbasic variables themselves, but slack or perturbation variables which measure how far in the feasible direction from their $\overline{FCP}$ optimum values the nonbasic variables are changed in satisfying the congruence constraints of FCP.

2. Both upper and lower bounds on basic variables in the optimal $\overline{FCP}$ solution are relaxed in the group problem.

3. Upper bounds on nonbasic variables which were lower-bounded in the optimal $\overline{FCP}$ solution and lower bounds on nonbasic variables which were upper-bounded are also relaxed. Thus the perturbation or change variables have no upper limits.

4. To obtain the constraints and objective function of the group problem from the optimal tableau of an upper-bounded Simplex procedure, express the original nonbasic variables in terms of perturbation

variables, and substitute. This has the effect of making columns for perturbations of nonbasic variables identical to the corresponding columns in the optimal bounded Simplex tableau for $\overline{FCP}$ when the variables were lower-bounded in the optimal $\overline{FCP}$ solution, and -1 times the corresponding bounded Simplex column when the variables were upper-bounded in the continuous solution. The right-hand-side of the main congruence constraints of the group problem then becomes, not the adjusted right-hand-side of the bounded Simplex tableau, but the optimal $\overline{FCP}$ values of the basic variables themselves.

For details and justifications of these concepts see Appendix B.

3.2.1 Formulation and Construction of Group Problems

The results of Section 3.1 demonstrate how optimal bounded Simplex tableaux for $\overline{FCP}$ can be constructed from corresponding tableaux for RP, and the concepts presented above indicate how group problems associated with FCP can be derived from optimal bounded Simplex tableaux for $\overline{FCP}$. Thus, group problems for FCP can easily be constructed from optimal bounded Simplex tableaux for RP.

However, Theorem 3.1.2.1 shows that a large number of optimal tableaux for $\overline{FCP}$ can be generated from a particular optimal tableau for RP by varying the membership of the constructed $\overline{FCP}$ basis. Thus a number of different group problems could result from a given optimal solution to RP.

In this section relations between these alternative group problems are investigated. For notational simplicity, however, only three cases will be treated. The cases correspond respectively to choosing all components of the vector y^N, all components of the vector s^N, and all components of the vector x_1^N in forming the changeable part of the basis for $\overline{FCP}$. Corresponding $\overline{FCP}$ tableaux are (3-15), (3-16) and (3-17). Any possibility admitted by Theorem 3.1.2.1 is merely a row-by-row mix of the results for these three cases.

Consider first the case of the basis $\{x_1^B, x_2^B, y^B, y^N\}$, where $\{x_1^B, x_2^B\}$ is an optimal basis for RP. Subdivide the associated nonbasic vectors into upper and lower-bounded parts as indicated in Section 1.4.2, and define the following perturbation variables

$$\begin{aligned} \Delta x_1^L &= x_1^L - 0 & & \\ \Delta x_1^U &= u_1^U - x_1^U & \Delta s^B &= s^B - 0 \\ \Delta x_2^L &= x_2^L - \ell_2^L & \Delta s^N &= s^N - 0. \\ \Delta x_2^U &= u_2^U - x_2^U & & \end{aligned} \tag{3-18}$$

Then the basic variables for this case can be expressed in terms of these *perturbation variables* and the tableau (3-15) as

$$\begin{pmatrix} x_1^B \\ x_2^B \end{pmatrix} = \begin{pmatrix} \bar{x}_1^B \\ \bar{x}_2^B \end{pmatrix} - (\bar{A}_1^N) \begin{pmatrix} \Delta x_1^L \\ -\Delta x_1^U \end{pmatrix} - (\bar{A}_2^N) \begin{pmatrix} \Delta x_2^L \\ -\Delta x_2^U \end{pmatrix} \tag{3-19}$$

$$\begin{pmatrix} y^B \\ y^N \end{pmatrix} = \begin{pmatrix} \bar{y}^B \\ \bar{y}^N \end{pmatrix} - \begin{pmatrix} [\bar{A}_1^N]^1{}_{k^B} \\ -I \end{pmatrix} \begin{pmatrix} \Delta x_1^L \\ -\Delta x_1^U \end{pmatrix} - \begin{pmatrix} [\bar{A}_2^N]^1{}_{k^B} \\ 0 \end{pmatrix} \begin{pmatrix} \Delta x_2^L \\ -\Delta x_2^U \end{pmatrix} + \begin{pmatrix} \Delta s^B \\ \Delta s^N \end{pmatrix} \quad (3\text{-}20)$$

where $\bar{x}_1^B$, $\bar{x}_2^B$, $\bar{y}^B$ and $\bar{y}^N$ are the optimal values of x_1^B, x_2^B, y^B and y^N respectively. Substituting for basic variables according to these relationships, and for nonbasic variables according to (3-18), we obtain the following equivalent form of FCP:

$$\min \quad (\bar{c}_1^N)^T \begin{pmatrix} \Delta x_1^L \\ -\Delta x_1^U \end{pmatrix} + (\bar{c}_2^N)^T \begin{pmatrix} \Delta x_2^L \\ -\Delta x_2^U \end{pmatrix} + \begin{pmatrix} c_s^B \\ c_s^N \end{pmatrix}^T \begin{pmatrix} \Delta s^B \\ \Delta s^N \end{pmatrix} + \nu(\overline{FCP}) \quad (3\text{-}21)$$

$$\text{s.t.} \begin{pmatrix} [\bar{A}_1^N]^1{}_{k^B} \\ -I \end{pmatrix} \begin{pmatrix} \Delta x_1^L \\ -\Delta x_1^U \end{pmatrix} + \begin{pmatrix} [\bar{A}_2^N]^1{}_{k^B} \\ 0 \end{pmatrix} \begin{pmatrix} \Delta x_2^L \\ -\Delta x_2^U \end{pmatrix} - \begin{pmatrix} \Delta s^B \\ \Delta s^N \end{pmatrix} \equiv \begin{pmatrix} \bar{y}^B \\ \bar{y}^N \end{pmatrix} \bmod \begin{pmatrix} u_1^B \\ u_1^N \end{pmatrix} \quad (3\text{-}22)$$

$$\Delta s^B,\ \Delta s^N,\ \Delta x_1^L,\ \Delta x_1^U,\ \Delta x_2^L,\ \Delta x_2^U \geq 0 \quad (3\text{-}23)$$

(EFCP)

$$\begin{pmatrix} u_1^B \\ u_2^B \end{pmatrix} \geq \begin{pmatrix} \bar{x}_1^B \\ \bar{x}_2^B \end{pmatrix} - (\bar{A}_1^N) \begin{pmatrix} \Delta x_1^L \\ -\Delta x_1^U \end{pmatrix} - (\bar{A}_2^N) \begin{pmatrix} \Delta x_2^L \\ -\Delta x_2^U \end{pmatrix} \geq \begin{pmatrix} 0 \\ \ell_2^B \end{pmatrix} \quad (3\text{-}24)$$

$$\begin{pmatrix} u_1^B \\ \\ u_1^N \end{pmatrix} \geq \begin{pmatrix} \bar{y}^B \\ \\ \bar{y}^N \end{pmatrix} - \begin{pmatrix} [\bar{A}_1^N]^1{}_{k^B} \\ \\ -I \end{pmatrix} \begin{pmatrix} \Delta x_1^L \\ -\Delta x_1^U \end{pmatrix} - \begin{pmatrix} [\bar{A}_2^N]^1{}_{k^B} \\ \\ 0 \end{pmatrix} \begin{pmatrix} \Delta x_2^L \\ -\Delta x_2^U \end{pmatrix} + \begin{pmatrix} \Delta s^B \\ \\ \Delta s^N \end{pmatrix} \geq 0 \quad (3\text{-}25)$$

$$u_1^B \geq \Delta s^B, \qquad u_1^N \geq \Delta s^N \qquad (3\text{-}26)$$

$$u_1^L \geq \Delta x_1^L, \qquad u_1^U \geq \Delta x_1^U \qquad (3\text{-}27)$$

$$(u_2^L - \ell_2^L) \geq \Delta x_2^L, \quad (u_2^U - \ell_2^U) \geq \Delta x_2^U. \qquad (3\text{-}28)$$

Several observations can be made about this equivalent form. First, note that if the constraints (3-22) are relaxed, an optimal solution to EFCP is

$$\Delta s^B,\ \Delta s^N,\ \Delta x_1^L,\ \Delta x_1^U,\ \Delta x_2^L,\ \Delta x_2^U = 0.$$

This must be true because such a solution satisfies all constraints except (3-22), and optimality of $\overline{\text{FCP}}$ assures the objective function coefficients of Δx_1^L and Δx_2^L are non-negative, and that those of Δx_1^U and Δx_2^U are non-positive. Thus, since $c_s > 0$, no other solution satisfying (3-23) could produce a smaller value of the objective function (3-21).

Next observe that if the constraints (3-22) are included, the all-zero solution will be a feasible solution only when $\bar{y} \equiv 0 \bmod u_1$, i.e. when an optimal solution to $\overline{\text{FCP}}$ is also an optimal solution to FCP.

Moreover, if $\bar{y} \not\equiv 0 \bmod u_1$, the all-zero solution would continue to be infeasible if all constraints were relaxed except (3-22) and (3-23). Thus minimizing (3-21) subject to (3-22) and (3-23) will produce a generally non-zero solution, and since c_s is positive, $\bar{c}_1^L$ and $\bar{c}_2^L$ are non-negative and $\bar{c}_1^U$ and $\bar{c}_2^U$ are non-positive, the value of this optimal solution will be greater than or equal to $v(\overline{FCP})$.

Finally, note that since the value of an optimal solution to EFCP is equal to v(FCP), the value of an optimal solution to any relaxation of EFCP provides a lower bound on v(FCP). In particular, the value of an optimal solution to any relaxation of EFCP which includes at least (3-22) and (3-23) provides a lower bound on v(FCP) which is in general greater than $v(\overline{FCP})$. Thus the following bounding problem is of interest.

3.2.1.1 Definition. The *group problem for FCP associated with the basis* $\{x_1^B, x_2^B, y^B, y^N\}$ is the optimization problem defined by (3-21), (3-22), and (3-23), and denoted $GP_1(FCP)$.

Development of the group problems for the bases $\{x_1^B, x_2^B, y^B, s^N\}$ and $\{x_1^B, x_2^B, y^B, x_1^N\}$ proceeds in exactly the same way whenever these bases are optimal for $\overline{FCP}$. By defining the perturbation variables

$$\Delta y^L = y^L - 0, \quad \Delta y^U = y^U - 0, \quad \Delta y^N = y^N - 0 \tag{3-29}$$

and using the tableaux (3-16) and (3-17), equivalent forms of FCP could be stated. Relaxation of bounds on basic variables in these equivalent forms produces the following group problems.

3.2.1.2 Definition. The *group problem for FCP associated with the basis* $\{x_1^B, x_2^B, y^B, s^N\}$ is the optimization problem defined by

$$\min \quad (\bar{c}_1^N - c_s^N)^T \begin{pmatrix} \Delta x_1^L \\ -\Delta x_1^U \end{pmatrix} + (\bar{c}_2^N)^T \begin{pmatrix} \Delta x_2^L \\ -\Delta x_2^U \end{pmatrix} + \begin{pmatrix} c_s^B \\ c_s^N \end{pmatrix}^T \begin{pmatrix} \Delta s^B \\ \Delta y^N \end{pmatrix} + \nu(\overline{FCP}) \qquad (3\text{-}30)$$

$$\text{s.t.} \quad [\bar{A}_1^N]^1_{k^B} \begin{pmatrix} \Delta x_1^L \\ -\Delta x_1^U \end{pmatrix} + [\bar{A}_2^N]^1_{k^B} \begin{pmatrix} \Delta x_2^L \\ -\Delta x_2^U \end{pmatrix} - \Delta s^B \equiv \bar{y}^B \bmod u_1^B \qquad (3\text{-}31)$$

$$\Delta s^B,\ \Delta y^N,\ \Delta x_1^L,\ \Delta x_1^U,\ \Delta x_2^L,\ \Delta x_2^U \geq 0 \qquad (3\text{-}32)$$

$$\Delta y^N \equiv 0 \bmod u_1^N, \qquad (3\text{-}33)$$

and denoted $GP_2(FCP)$.

3.2.1.3 Definition. The *group problem for FCP associated with the basis* $\{x_1^B, x_2^B, y^B, x_1^N\}$ is the optimization problem defined by

$$\min \quad (\bar{c}_2^N)^T \begin{pmatrix} \Delta x_2^L \\ \\ -\Delta x_2^U \end{pmatrix} + (\bar{c}_1^N)^T \begin{pmatrix} \Delta y^L \\ \\ -\Delta y^U \end{pmatrix} + (c_s^B)^T \Delta s^B + (c_s^N - \bar{c}_1^N)^T \Delta s^N + \nu(\overline{FCP}) \tag{3-34}$$

$$\text{s.t.} \quad [\bar{A}_2^N]^1_{k^B} \begin{pmatrix} \Delta x_2^L \\ \\ -\Delta x_2^U \end{pmatrix} + [\bar{A}_1^N]^1_{k^B} \begin{pmatrix} \Delta y^L \\ \\ -\Delta y^U \end{pmatrix} - \Delta s^B - [\bar{A}_1^N]^1_{k^B}(\Delta s^N) \equiv \bar{y}^B \bmod u_1^B \tag{3-35}$$

$$\Delta s^B,\ \Delta s^N,\ \Delta y^L,\ \Delta y^U,\ \Delta x_2^L,\ \Delta x_2^U \geq 0 \tag{3-36}$$

$$\Delta y^L \equiv 0 \bmod u_1^L,\ \Delta y^U \equiv 0 \bmod u_1^U, \tag{3-37}$$

and denoted $GP_3(FCP)$.

3.2.2 Relations Between Alternative Formulations

The three formulations of the group problem for FCP presented in the previous section share several properties. Among these are the following:

1. All three group problems can be constructed from the optimal bounded Simplex tableau of the reduced problem RP by direct copying of rows from that tableau. The only computation necessary is the changing of signs on some columns.

2. The value of an optimal solution to any of the three group problems provides a lower bound on $\nu(FCP)$ which will be generally greater than $\nu(\overline{FCP})$.

These similarities do not imply, however, that the three problems are equivalent in their suitability for use in a branch-and-bound procedure like the one discussed at the beginning of this chapter. For example, the next theorem demonstrates that the first two are unequal in their bounding effectiveness.

3.2.2.1 Theorem. For any $GP_1(FCP)$ and $GP_2(FCP)$ as defined above which correspond to Simplex optimal solutions to $\overline{FCP}$, $\nu\big(GP_1(FCP)\big) \geq \nu\big(GP_2(FCP)\big)$. Moreover, any parallel relaxation of the two problems obtained by imposing only part of the constraints (3-31) will similarly have the value of an optimal solution to the relaxation of $GP_1(FCP)$ greater than or equal to the value of an optimal solution to the analogous relaxation of $GP_2(FCP)$.

Proof. First observe that it will always be optimal to have $\Delta y^N = 0$ in $GP_2(FCP)$. Any other solution will incur unnecessary cost because $c_s^N > 0$. Now suppose $GP_1(FCP)$ is relaxed by replacing the constraints (3-22) with those of (3-31), i.e. ignoring the last part of the constraints (3-22). Then for this relaxed version of $GP_1(FCP)$, it will always be optimal to have $\Delta s^N = 0$ because $c_s^N > 0$. Thus the set of solutions which could be optimal for $GP_2(FCP)$ is essentially equal to the set of solutions which could be optimal for such a relaxed form of $GP_1(FCP)$ and is defined by (3-31) and

$$\Delta s^B,\ \Delta x_1^L,\ \Delta x_1^U,\ \Delta x_2^L,\ \Delta x_2^U \geq 0. \tag{3-38}$$

$$\Delta y^N,\ \Delta s^N = 0. \tag{3-39}$$

Now consider the objective functions. Over the set of solutions just described, the two objective functions are identical except for the coefficient of the perturbations of x_1^N. For $GP_1(FCP)$ this coefficient is $\bar{c}_1^N$, but for $GP_2(FCP)$ it is $(\bar{c}_1^N - c_s^N)$. Since $c_s^N > 0$,

$$\bar{c}_1^N > \bar{c}_1^N - c_s^N ,$$

and by Theorem 3.1.2.1 it is possible to construct $GP_2(FCP)$ only when $(\bar{c}_1^N - c_s^N) \geq 0$ and all nonbasic x_{1j} are lower-bounded. Thus if $GP_2(FCP)$ can be constructed, the value of the objective function of $GP_1(FCP)$ will be greater than or equal to the value of the objective function of $GP_2(FCP)$ for any solution satisfying (3-31), (3-38) and (3-39). Certainly such a relationship will continue to hold when the described relaxation of $GP_1(FCP)$ is replaced by the full problem, and so $\nu\big(GP_1(FCP)\big) \geq \nu\big(GP_2(FCP)\big)$.

The argument for the second part of the theorem is exactly analogous. Relaxation of some of the constraints (3-31) in both $GP_1(FCP)$ and $GP_2(FCP)$ would not change any of the above conclusions, and the value of an optimal solution for the relaxation of $GP_1(FCP)$ would still be greater than or equal to the value of an optimal solution for the corresponding relaxation of $GP_2(FCP)$ whenever $GP_2(FCP)$ can be constructed.

Q.E.D.

Theorem 3.2.2.1 demonstrates that the formulation $GP_2(FCP)$ has little attraction in a branch-and-bound procedure for FCP. Generally better bounds will be obtained by use of the formulation $GP_1(FCP)$, and no additional computational effort is required.

Unfortunately, it does not appear possible to develop such a dominance relationship between $GP_1(FCP)$ and $GP_3(FCP)$. For some solutions it appears one would produce higher bounds, and for others the reverse might occur.

However, $GP_1(FCP)$ and $GP_3(FCP)$ do differ in another important aspect. The constraints (3-37) impose requirements on the perturbation solution space which are difficult to deal with computationally. Instead of perturbations moving along a continuum as in $GP_1(FCP)$, they must advance in discrete jumps to satisfy (3-37). Thus any penalty problem derived from $GP_3(FCP)$ is considerably more difficult to solve than a corresponding one from $GP_1(FCP)$.

Because of this computational difficulty with $GP_3(FCP)$ and the weaker bounding properties of $GP_2(FCP)$, *all remaining discussion in this dissertation will address the formulation $GP_1(FCP)$.* To simply notation, the subscript will be omitted, and this formulation referred to simply as GP(FCP). In addition, relaxations of $GP_1(FCP)$ involving only selected rows of (3-22) will be denoted $GP(\Gamma)$ where Γ is the set of indices of all enforced rows.

3.3. Penalties Derived from One-Row Group Problems

While the value of an optimal solution to GP(FCP) might very well provide a very good bound on $\nu(FCP)$, the congruence constraints of (3-22) make GP(FCP) very difficult to solve. Thus Gomory and Johnson [43] have proposed that even more relaxed penalty problems be derived from GP(FCP). In particular, they have suggested solving penalty problems consisting of (3-21), (3-23) and one row of (3-22), i.e. GP(i).

In this section, such one row penalties will be studied in the FCP context. To simplify the notation define d_1, d_2, $a_1(i)$, $a_2(i)$ from the tableau of (3-15) as follows:

$$d_1 = \begin{pmatrix} \bar{c}_1^L \\ -\bar{c}_1^U \end{pmatrix} \tag{3-40}$$

$$d_2 = \begin{pmatrix} \bar{c}_2^L \\ -\bar{c}_2^U \end{pmatrix} \tag{3-41}$$

$$a_1(i) = \begin{bmatrix} [\bar{A}_1^L & -\bar{A}_1^U]^1_{k}{}^B \\ -I & 0 \\ 0 & I \end{bmatrix}_i \tag{3-42}$$

$$a_2(i) = \begin{bmatrix} [\bar{A}_2^L & -\bar{A}_2^U]^1_{k}{}^B \\ 0 & 0 \\ 0 & 0 \end{bmatrix}_i . \tag{3-43}$$

The vectors d_1 and d_2 are then the objective function coefficients of GP(FCP), adjusted in sign according to whether the corresponding variable was upper-bounded or lower-bounded. Similarly, $a_1(i)$ and $a_2(i)$ are the coefficients of one row of the constraints (3-22) after adjustment of the sign.

3.3.1 Simple One-Row Group Penalties

With this notation the *one-row group problem associated with the ith row of GP(FCP)* is given by

$$\min \quad (d_1)^T \Delta x_1^N + (d_2)^T \Delta x_2^N + c_{si} \Delta s_i + \nu(\overline{FCP}) \tag{3-44}$$

$$\big(GP(i)\big) \quad \text{s.t. } a_1(i)^T \Delta x_1^N + a_2(i)^T \Delta x_2^N - s_i \equiv \bar{y}_i \bmod u_{1i} \tag{3-45}$$

$$\Delta x_1^N,\ \Delta x_2^N,\ \Delta s_i \geq 0. \tag{3-46}$$

The problem can be interpreted as that of finding the minimum cost perturbation of the optimal solution to $\overline{FCP}$ which will cause y_i to satisfy its congruence requirement. The quantity

$$\nu\big(GP(i)\big) - \nu(\overline{FCP})$$

is thus an estimate of the additional cost or penalty for making y_i conform to $y_i \equiv 0 \bmod u_{1i}$.

As indicated in Section 2.1.1, even such simple one row group problems are often difficult to solve, and a series of approximation methods have been developed (see [43,44,65,66]). In the FCP case, however, the simple one row group problems can easily be solved exactly.

Since all perturbation variables are continuous, GP(i) reduces to two linear knapsack problems, and the solution presented in the next theorem applies.

<u>3.3.1.1 Theorem.</u> For GP(i) as defined above,

$$\nu\big(GP(i)\big) = \nu(\overline{FCP}) + \min\{\rho^D(i),\rho^U(i)\}$$

where

$$\rho^D(i) = \bar{y}_i \min \begin{Bmatrix} \min\left\{\dfrac{d_{1j}}{a_{1j}(i)} \;:\; a_{1j}(i) > 0\right\} \\ \\ \min\left\{\dfrac{d_{2j}}{a_{2j}(i)} \;:\; a_{2j}(i) > 0\right\} \end{Bmatrix} \tag{3-47}$$

$$\rho^U(i) = (u_{1i}-\bar{y}_i) \min \begin{Bmatrix} \min\left\{\dfrac{d_{1j}}{-a_{1j}(i)} \;:\; a_{1j}(i) < 0\right\} \\ \\ \min\left\{\dfrac{d_{2j}}{-a_{2j}(i)} \;:\; a_{2j}(i) < 0\right\} \\ \\ c_{si} \end{Bmatrix}. \tag{3-48}$$

Proof. Suppose (3-45) in GP(i) is replaced by

$$a_1(i)^T \Delta x_1^N + a_2(i)^T \Delta x_2^N - \Delta s_i = w.$$

Then the resulting problem is a linear knapsack problem, and an optimal solution can be obtained by well-known minimum ratio procedure. In particular, if $w = \bar{y}_i$, the value of a solution is given by

$$\nu(\overline{FCP}) + \rho^D(i) \tag{3-49}$$

where $\rho^D(i)$ is defined in (3-47). Moreover, if $w > \bar{y}_i$, (3-49) provides a lower bound on the value of an optimal solution to the knapsack problem. Similarly, if $w = \bar{y}_i - u_{1i}$, the value of a solution to the knapsack problem is given by

$$\nu(\overline{FCP}) + \rho^U(i), \tag{3-50}$$

and (3-50) provides a lower bound on solution values for all problems with $w \leq \bar{y}_i - u_{1i}$.

By feasibility of $\overline{FCP}$,

$$u_{1i} \geq \bar{y}_i \geq 0.$$

Thus to satisfy (3-45) in GP(i) either

$$a_1(i)^T \Delta x_1^N + a_2(i)^T \Delta x_2^N - \Delta s_i \geq \bar{y}_i \tag{3-51}$$

or

$$a_1(i)^T \Delta x_1^N + a_2(i)^T \Delta x_2^N - \Delta s_i \le \bar{y}_i - u_{1i}. \qquad (3\text{-}52)$$

But by the argument just presented, the minimum cost solutions for these two cases are given by (3-49) and (3-50). Thus the value of an optimal solution to GP(i) will be the minimum of the values for the two cases. Q.E.D.

In addition to having simple exact solutions, the one-row problems GP(i) possess another convenient property. Since GP(i) reduces to two linear programs, one which corresponds to forcing y_i "down" to zero and the other which corresponds to forcing y_i "up" to u_{1i}, it is possible to interpret the functions $\rho^D(i)$ and $\rho^U(i)$. These quantities correspond to the penalties for forcing y_i down and up, respectively. In general group penalty approaches (see [43,44,65,66]) it is possible to obtain only an estimate of the penalty for forcing y_i to satisfy its congruence constraint. For the FCP case, however, this argument shows that penalties for attaining specific values of y_i can be identified. Such specific information is very useful in defining the sets Θ_D and Θ_U, and choosing the branching variable $y_{\hat{i}}$ in a branch-and-bound procedure like the one presented at the beginning of this chapter.

3.3.2 Relation to Tomlin's Penalties

In [90] J. A. Tomlin proposed two other penalty approaches for estimating the cost of forcing an integer variable up or down to the next integral value. One set of penalties is based on simple duality concepts, and the other derives from Gomory's mixed-integer cuts [37]. For perturbation variables which must be integer, i.e. are subject to congruence requirements, Tomlin's two penalties are different. For the case where all perturbation variables are continuous, however, both penalties are the same. For FCP's with optimal $\overline{\text{FCP}}$ solution obtained as in Theorem 3.1.1.4, the penalties are given by

$$\tau^D(i) = \bar{y}_i \min \left\{ \begin{array}{l} \min \left\{ \dfrac{d_{1j}}{a_{1j}(i)} \quad : \quad a_{1j}(i) > 0 \right\} \\ \\ \min \left\{ \dfrac{d_{2j}}{a_{2j}(i)} \quad : \quad a_{2j}(i) > 0 \right\} \end{array} \right\} \qquad (3\text{-}53)$$

$$\tau^U(i) = (u_{1i} - \bar{y}_i) \min \left\{ \begin{array}{l} \min \left\{ \dfrac{d_{1j}}{-a_{1j}(i)} \quad : \quad a_{1j}(i) < 0 \right\} \\ \\ \min \left\{ \dfrac{d_{2j}}{-a_{2j}(i)} \quad : \quad a_{2j}(i) < 0 \right\} \\ \\ \quad c_{si} \end{array} \right\}, \qquad (3\text{-}54)$$

where $\tau^D(i)$ = the penalty for forcing the ith integer variable down to the next value satisfying $y_i \equiv 0 \bmod u_{1i}$,

$\tau^U(i)$ = the penalty for forcing the ith integer variable up to the next value satisfying $y_i \equiv 0 \bmod u_{1i}$.

Direct inspection of these expressions will show that Tomlin's penalties are identical to the simple one-row group penalties for FCP. This observation is summarized in the following theorem.

3.3.2.1 Theorem. For the problem FCP with optimal $\overline{FCP}$ tableau constructed as in Theorem 3.1.1.4, the simple one-row group penalties $\rho^D(i)$ and $\rho^U(i)$ satisfy

$$\rho^D(i) = \tau^D(i)$$

$$\rho^U(i) = \tau^U(i)$$

where $\tau^D(i)$ and $\tau^U(i)$ are the down and up penalties proposed by Tomlin.

3.3.3 Improving Penalties by Considering Upper Bounds

In order for GP(FCP) to conform to the mathematical group theory results developed by Gomory [40], it was necessary to relax the constraints (3-24) through (3-28) of EFCP in formulating the problem. Since Theorem 3.3.1.1 has demonstrated that simple one-row relaxations of GP(FCP) are exceptionally easy to solve, however, it seems logical to consider reimposing some of the constraints lost in the GP(FCP) formulation in order to improve the bounds obtained. The more difficult problem of considering constraints (3-24) and (3-25) will be discussed later in the dissertation, but a simple scheme for including the upper bounds on perturbation variables [i.e. (3-26), (3-27) and (3-28)] is presented here.

Define the upper limits

$$o_2^N = u_2^N - \ell_2^N. \qquad (3\text{-}55)$$

Then the *bounded group problem BGP(Γ) associated with the GP(Γ)* is given by

$$\min \quad (d_1)^T \Delta x_1^N + (d_2)^T \Delta x_2^N + c_s^T \Delta s + \nu(\overline{FCP})$$

$$\text{s.t.} \quad a_1(i)^T \Delta x_1^N + a_2(i)^T \Delta x_2^N - \Delta s_i \equiv \bar{y}_i \bmod u_{1i} \quad \text{for } i \in \Gamma$$

(BGP(Γ))

$$u_1^N \geq \Delta x_1^N \geq 0$$

$$o_2^N \geq \Delta x_2^N \geq 0$$

$$u_1 \geq \Delta s \geq 0.$$

As with GP(Γ), this problem can be interpreted as that of finding the minimum cost perturbation of the optimal solution to $\overline{FCP}$ which will cause y_i to satisfy the constraint $y_i \equiv 0 \bmod u_{1i}$ for all $i \in \Gamma$. However, in BGP(Γ) the constraints (3-26), (3-27) and (3-28) of EFCP are also enforced.

Even with the additional constraints, the one-row cases BGP(i) are almost as easy to solve as GP(i). The next theorem specifies the technique for obtaining an optimal solution.

<u>3.3.3.1 Theorem.</u> For BGP(i) as defined above

$$\nu\big(BGP(i)\big) = \nu(\overline{FCP}) + \min\{\rho^D(i), \rho^U(i)\}$$

where $\rho^D(i)$ is the value of an optimal solution to the problem

$$\min \quad (d_1)^T \Delta x_1^N + (d_2)^T \Delta x_2^N + c_{si} \Delta s_i \tag{3-56}$$

$$\text{s.t.} \quad a_1(i)^T \Delta x_1^N + a_2(i)^T \Delta x_2^N - \Delta s_i = \bar{y}_i \tag{3-57}$$

$$u_1^N \geq \Delta x_1^N \geq 0 \tag{3-58}$$

$$o_2^N \geq \Delta x_2^N \geq 0 \tag{3-59}$$

$$u_{1i} \geq \Delta s_i \geq 0 \tag{3-60}$$

and $\rho^U(i)$ is the value of an optimal solution to a similar problem with the right-hand-side of (3-57) replaced by $(\bar{y}_i - u_{1i})$.

Proof. As with Theorem 3.3.1.1 the version of BGP(i) consisting of (3-56), (3-58), (3-59), (3-60) and the constraint

$$a_1(i)^T \Delta x_1^N + a_2(i)^T \Delta x_2^N - \Delta s_i = w$$

is a linear knapsack problem with optimal solution value increasing monotonically with $|w|$. Moreover, any solution to BGP(i) satisfying (3-45) will also satisfy either (3-51) or (3-52). Thus a least cost solution will always occur when the inequalities in these expressions are replaced by equalities, and the value of an optimal solution to BGP(i) will be the minimum of the two equality cases.

Q.E.D.

From Theorem 3.3.3.1 it can be seen that BGP(i), like GP(i), reduces to two linear knapsack problems for which efficient minimum ratio solution schemes are available. Moreover, the "two case" solution procedure still permits interpretation of the functions $\rho^D(i)$ and $\rho^U(i)$ as the penalties for forcing y_i "down" to 0 and "up" to u_{1i}, respectively. Thus, by solving BGP(i) instead of GP(i) generally higher penalties can be achieved with relatively little computational cost and no loss of other desirable properties.

3.4 Valid Inequalities for Fixed Charge Problems

In the work of Gomory and Johnson [42,43,44] on group theoretic approaches to integer programming the penalty problem ideas discussed in the previous section are closely related to the concept of a valid inequality or cut. In this section a portion of their theory of valid inequalities will be applied to derive some results for FCP.

3.4.1 Restatement of the Theory of Valid Inequalities

Before turning to application of the Gomory-Johnson theory to results for FCP, it is necessary to restate some of the important definitions and results of the theory in terms of the notation and requirements of FCP. The discussion of this section provides such a review, with each definition and theorem being a direct restatement of more general results in [43].

3.4.1.1 Definition [43,p.30]. A *valid inequality* for any relaxation of the equivalent form EFCP is an inequality t defined by

$$t_1^T \Delta x_1^N + t_2^T \Delta x_2^N + t_s^T \Delta s \geq 1 \qquad (3\text{-}61)$$

which is satisfied by every feasible solution $\{\Delta x_1^N, \Delta x_2^N, \Delta s\}$ to that relaxation and has

$$t_1, t_2, t_s \geq 0. \tag{3-62}$$

<u>3.4.1.2 Definition [43,p.30]</u>. A valid inequality t is a *minimal valid inequality* if there exists no other valid inequality w for the same problem satisfying $t > w$, where $t > w$ is defined to mean

$$t_{1j} \geq w_{1j}$$

$$t_{2j} \geq w_{2j}$$

$$t_{sj} \geq w_{sj}$$

for all j, and strict inequality holds in at least one case.

Suppose addition of inequalities is defined to mean coefficient by coefficient addition, and multiplication of an inequality by a scalar is defined to mean multiplication of each coefficient by the scalar. Then clearly the set of valid inequalities for a particular problem is a convex set because an inequality

$$(p)t + (1-p)w$$

constructed from two valid inequalities t and w would obviously satisfy

(3-61) and (3-62) for $1 \geq p \geq 0$. Moreover, it then makes sense to speak of the extreme valid inequalities described in the following definition.

3.4.1.3 Definition [43,p.31]. An *extreme valid inequality* for a given problem is a valid inequality for the problem which cannot be expressed as a convex combination of two distinct valid inequalities for that problem.

The relation between minimal and extreme valid inequalities is given by the following theorem.

3.4.1.4 Theorem [43,p.31]. The extreme valid inequalities for a given problem are minimal valid inequalities for the problem.

Proof. Let t be an extreme valid inequality for some problem. Then if t is not minimal, there exists another valid inequality w such that $w < t$ in the sense described in Definition 3.4.1.2. Consider the inequality

$$\big(t + (t-w)\big).$$

Since t is a valid inequality and (t-w) is non-negative in every component, this inequality satisfies (3-61) and (3-62) and so is a valid inequality. Moreover, this inequality is not identical to w because that would imply $t = w$. But $t = \frac{1}{2} w + \frac{1}{2} \big(t + (t-w)\big)$, which implies t can be expressed as a convex combination of two distinct valid inequalities and contradicts extremality of t. Thus t must be a minimal valid inequality. Q.E.D.

Before applying the above theorem and definitions to some observations about FCP, it is useful to gain some intuition about the different types of valid inequalities. Assume for this purpose that the perturbation solution space of a problem associated with EFCP has only two dimensions. Then the three possibilities are illustrated in Figure 2.

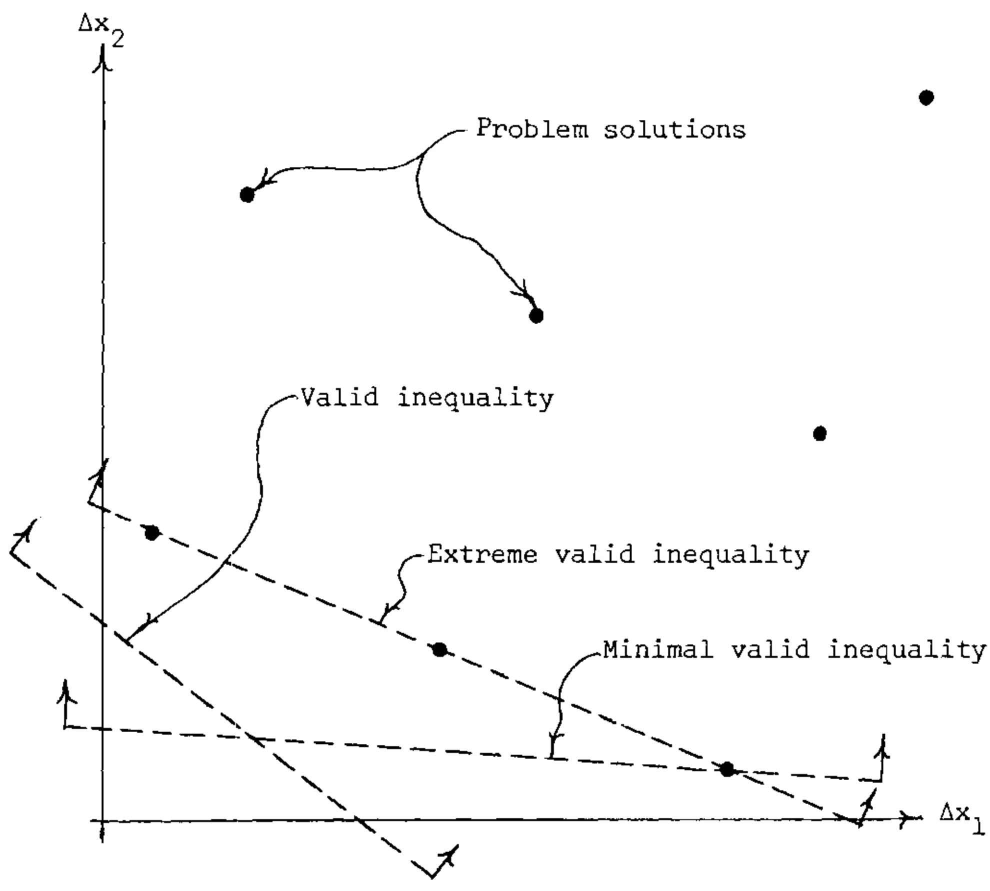

Figure 2. Illustrations of Types of Valid Inequalities

The heavy dots in this figure represent points which satisfy the problem of interest (e.g. GP(FCP)). A valid inequality for this problem is any inequality with positive slope, i.e. with non-negative coefficients, which excludes none of the solution points. The inequality becomes minimal if it touches the convex hull of solution points for the problem, and extreme if it forms a face of that convex hull. Thus in terms of defining the solution space for a particular problem, extreme valid inequalities are the most preferred, followed by minimal valid inequalities and then ordinary valid inequalities. If the entire set of extreme valid inequalities were available, an exact solution could be provided by linear programming for any problem involving the minimization of a non-negative objective function over the indicated solution points because such a solution would occur at the intersection of extreme valid inequalities.

3.4.2 Strength of Gomory Cuts as Valid Inequalities

In 1960, Gomory [37] proposed a method for deriving cutting planes for mixed-integer programs. One such cutting plane can be derived from each infeasible row i of the constraints (3-22), and takes the form

$$\left(GC(i)\right) \qquad \left(g_1(i)\right)^T \Delta x_1^N + \left(g_2(i)\right)^T \Delta x_2^N + \left(g_s(i)\right)^T \Delta s \geq 1$$

where

$$g_{kj}(i) = \begin{cases} \dfrac{a_{kj}(i)}{\bar{y}_i} & \text{if} \quad a_{kj}(i) \geq 0 \text{ and } k=1,2 \\ \\ \dfrac{a_{kj}(i)}{\bar{y}_i - u_{1i}} & \text{if} \quad a_{kj}(i) < 0 \text{ and } k=1,2 \end{cases} \tag{3-63}$$

$$g_{sj}(i) = \begin{cases} \dfrac{-1}{\bar{y}_i - u_{1i}} & \text{if} \quad j=i \\ \\ 0 & \text{otherwise} \end{cases} \tag{3-64}$$

and $a_{1j}(i)$, $a_{2j}(i)$ are as defined in Section 3.3.

The purpose of generating such inequalities is to define the solution space of the original mixed-integer problem in a way which could be dealt with by linear programming. However, the next theorem demonstrates that the GC(i) can be shown to be special cases of valid inequalities.

3.4.2.1 Theorem. Each GC(i) which corresponds to a row i of (3-22) not satisfied by the solution Δx_1^N, Δx_2^N, $\Delta s = 0$ is a valid inequality for EFCP, or any relaxation of EFCP involving at least (3-21), (3-23) and the same row i of (3-22).

Proof. It is sufficient to show GC(i) is a valid inequality for the one-row group problem GP(i) because an inequality which is valid for a given problem will certainly be valid for any other problem with feasible solution set contained in that of the first problem.

For the GP(i) case observe that for any infeasible row i, it must be true that $u_{1i} > \bar{y}_i > 0$. Thus the definitions (3-63) and (3-64) clearly imply all coefficients of GC(i) are finite and non-negative.

Next recall that any solution to GP(i) must satisfy either (3-51) or (3-52). In the case where (3-51) is satisfied it must certainly hold that

$$\sum_{\{j:a_{1j}(i)\geq 0\}} a_{1j}(i)\Delta x_{1j}^N + \sum_{\{j:a_{2j}(i)\geq 0\}} a_{2j}(i)\Delta x_{2j}^N \geq \bar{y}_i. \qquad (3\text{-}65)$$

Similarly, if a solution satisfies (3-52),

$$\sum_{\{j:a_{1j}(i)<0\}} a_{1j}(i)\Delta x_{1j}^N + \sum_{\{j:a_{2j}(i)<0\}} a_{2j}(i)\Delta x_{2j}^N - \Delta s_i \leq \bar{y}_i - u_{1i}. \qquad (3\text{-}66)$$

Dividing (3-65) by $\bar{y}_i$ and (3-66) by $(\bar{y}_i - u_{1i})$ produces two expressions which must be greater than or equal to one. Since either the situation of (3-65) or the situation of (3-66) must occur, it is certainly true that the sum of the two such expressions must be at least one. Thus

$$\left(g_1(i)\right)^T \Delta x_1^N + \left(g_2(i)\right)^T \Delta x_2^N + \left(g_s(i)\right)^T \Delta s \geq 1$$

for all solutions $\{\Delta x_1^N, \Delta x_2^N, \Delta s\}$ of GP(i), and GC(i) is a valid inequality for GP(i). Q.E.D.

In order for GC(i) to be a valid cutting plane for a given problem it is necessary that it be a valid inequality for the problem. Thus Theorem 3.4.2.1 could be shown for any mixed-integer problem, and in fact, the argument given in the proof is exactly that of Gomory in [37]. The next theorem demonstrates, however, that Gomory cuts have unusual significance for the FCP case.

3.4.2.2 Theorem. Each GC(i) which corresponds to a row i of (3-22) not satisfied by the solution Δx_1^N, Δx_2^N, $\Delta s = 0$ is an extreme valid inequality for GP(FCP) and any relaxation of GP(FCP) for which GC(i) is a valid inequality.

Proof. It is sufficient to show that GC(i) is an extreme valid inequality for GP(FCP) since any inequality which is an extreme valid inequality for a given problem is certainly an extreme valid inequality in any relaxation of the problem for which it is valid. Moreover, Theorem 3.4.2.1 implies GC(i) is a valid inequality for GP(FCP). Thus it is only necessary to show that the GC(i) are extreme for GP(FCP).

Suppose, by contradiction, that some GC(i) derived from an infeasible row of (3-22) is not extreme. Then there exist two distinct valid inequalities w and z such that

$$GC(i) = (p)w + (1-p)z \tag{3-67}$$

for some p satisfying $1 > p > 0$.

Now it must be true that $w_{sj} = z_{sj} = 0$ for $j \neq i$. This follows because $g_{sj}(i) = 0$ for $j \neq i$, and two non-negative quantities can produce a convex combination equal to 0 only if they are both zero. Suppose, however, that $w_{si} < g_{si}(i) < z_{si}$, and consider the solution

$$\Delta\hat{x}_1^N = 0, \quad \Delta\hat{x}_2^N = 0, \quad \Delta\hat{s} = u_1 - \bar{y}. \tag{3-68}$$

Clearly, (3-68) provides a feasible solution to GP(FCP) because all perturbation variables are non-negative and the indicated value of Δs will satisfy all the constraints (3-22). But

$$w_1^T\Delta\hat{x}_1^N + w_2^T\Delta\hat{x}_2^N + w_s^T\Delta\hat{s} = w_{si}(u_{1i}-\bar{y}_i) < g_{si}(i)(u_{1i}-\bar{y}_i) = 1,$$

which contradicts the validity of w. The reverse case where $z_{si} < g_{si}(i) < w_{si}$ is exactly analogous.

Thus, if two distinct valid inequalities w and z do exist which satisfy (3-67), there must exist some pair (k,j) with k=1 or 2 and either

$$w_{kj} < g_{kj}(i) < z_{kj} \tag{3-69}$$

or

$$z_{kj} < g_{kj}(i) < w_{kj}. \tag{3-70}$$

Without loss of generality, assume (3-69) holds, and consider the solution

$$\Delta\hat{x}^N_{hn} = \begin{cases} \dfrac{1}{g_{hn}(i)} & \text{if } h=k,\ n=j \\ 0 & \text{otherwise} \end{cases} \tag{3-71}$$

$$\Delta\hat{s}_n = \begin{cases} 0 & \text{if } n = i \\ \left(\dfrac{a_{kj}(n)}{g_{kj}(i)} - \bar{y}_n\right) \bmod u_{1n} & \text{if } n \neq i. \end{cases} \tag{3-72}$$

To show this solution is feasible for GP(FCP), observe first that all values of $\Delta\hat{x}^N_1$, $\Delta\hat{x}^N_2$ and $\Delta\hat{s}$ are clearly non-negative. Moreover, for row i of (3-22)

$$a_1(i)^T\Delta\hat{x}^N_1 + a_2(i)^T\Delta\hat{x}^N_2 - \Delta s_i = \frac{a_{kj}(i)}{g_{kj}(i)} = \bar{y}_i \text{ or } \bar{y}_i - u_{1i},$$

and either of these values is congruent to $\bar{y}_i$ mod u_{1i}. Similarly, for row $n \neq i$

$$a_1(n)^T\Delta\hat{x}^N_1 + a_2(n)^T\Delta\hat{x}^N_2 - \Delta\hat{s}_n = \frac{a_{kj}(n)}{g_{kj}(n)} - \left(\frac{a_{kj}(n)}{g_{kj}(n)} - \bar{y}_n\right) \bmod u_{1n}$$

$$\overset{\bmod u_{1n}}{\equiv} \frac{a_{kj}(n)}{g_{kj}(n)} - \frac{a_{kj}(n)}{g_{kj}(n)} + \bar{y}_n$$

$$\overset{\bmod u_{1n}}{\equiv} \bar{y}_n.$$

Thus the solution of (3-71) and (3-72) satisfies the constraints (3-22) and is feasible for GP(FCP).

But for this solution (remembering $w_{sh} = 0$ for $h \neq i$),

$$w_1^T \Delta \hat{x}_1^N + w_2^T \Delta \hat{x}_2^N + w_s^T \Delta \hat{s} = \frac{w_{kj}}{g_{kj}(i)} < \frac{g_{kj}(i)}{g_{kj}(i)} = 1$$

which violates validity of w. Therefore no distinct valid inequalities satisfying (3-67) can exist, and GC(i) is an extreme valid inequality for GP(FCP).

Q.E.D.

The implication of Theorem 3.4.2.2 is that Gomory cuts are extreme valid inequalities or faces of the convex hull of solutions to GP(FCP). This is a somewhat surprising result because experience with the use of Gomory cuts in integer programming has generally been disappointing computationally (see e.g. [49]). For the FCP case, however, the above theorem shows that the Gomory cuts are as strong as any possible inequality for GP(FCP). The GC(i) may not be the only faces of the convex hull of solutions to GP(FCP), but by Theorem 3.4.1.4 there can be no valid inequality t for GP(FCP) which satisfies t < GC(i).

A possible explanation for this apparent contradiction between observed and theoretical strength of Gomory cuts lies in the fact that Theorem 3.4.2.2 would not hold if GP(FCP) were augmented by certain of the contraints of EFCP which are relaxed in the GP(FCP) formulation. The next section investigates this phenomenon for the case of the constraints (3-25).

3.4.3 Gomory Cuts in Either-Or Problems

Consider the *either-or problem EOP(Γ)* derived from the optimal tableau (3-15) in the form

$$\min \quad d_1^T \Delta x_1^N + d_2^T \Delta x_2^N + c_s^T \Delta s + \nu(\overline{FCP}) \tag{3-73}$$

$$\text{s.t.} \quad a_1(i)^T \Delta x_1^N + a_2(i)^T \Delta x_2^N - \Delta s_i = \bar{y}_i \text{ or } \bar{y}_i - u_{1i} \text{ for } i \in \Gamma \tag{3-74}$$

$$\{EOP(\Gamma)\} \qquad \Delta x_1^N,\ \Delta x_2^N,\ \Delta s \geq 0. \tag{3-75}$$

Clearly, such a problem is a valid relaxation of FCP because it consists of GP(Γ) with some of the constraints (3-25) reimposed. A solution to GP(Γ) is also a solution to EOP(Γ) if it not only satisfies the rows of (3-22) corresponding to $i \in \Gamma$ but also satisfies those rows in such a way that the implied values of the y_i yield $0 \leq y_i \leq u_{1i}$.

The extremality of Gomory cuts implied by Theorem 3.4.2.2 does not always hold for EOP(Γ). Consider, for example, the case where constraints of EOP(Γ) are given by

$$\Delta x_{21} - \Delta s_1 = 5 \text{ or } -4 \tag{3-76}$$

$$\Delta x_{21} - \Delta s_2 = 8 \text{ or } -12 \tag{3-77}$$

$$\Delta x_{21},\ \Delta s_1,\ \Delta s_2 \geq 0, \tag{3-78}$$

and those of GP(Γ) are

$$\Delta x_{21} - \Delta s_1 \equiv 5 \bmod 9$$

$$\Delta x_{21} - \Delta s_2 \equiv 8 \bmod 20$$

$$\Delta x_{21},\ \Delta s_1,\ \Delta s_2 \geq 0.$$

The Gomory cut for (3-76) is

$$\frac{1}{5}\,\Delta x_{21} + \frac{1}{4}\,\Delta s_1 \geq 1. \qquad (3\text{-}79)$$

However, inspection will show that if Δx_{21} is positive in any solution to (3-76), (3-77) and (3-78),

$$\Delta x_{21} \geq 8 \quad \text{and} \quad \Delta s_1 \geq 3.$$

Thus the inequality

$$\frac{1}{32}\,\Delta x_{21} + \frac{1}{4}\,\Delta s_1 \geq 1 \qquad (3\text{-}80)$$

is a valid inequality for the above EOP because it eliminates no feasible solution. But (3-80) has smaller coefficients than (3-79) which implies (3-79) is not minimal. Therefore, by Theorem 3.4.1.4, the Gomory cut (3-79) is also not an extreme valid inequality for the above EOP.

From this example it can be concluded that Gomory cuts will not always be extreme for EOP's of at least two rows. However, the next theorem demonstrates that extremality does hold for the one-row problems EOP(i).

3.4.3.1 Theorem. Each GC(i) which corresponds to a row i of (3-22) not satisfied by the solution Δx_1^N, Δx_2^N, $\Delta s = 0$ is an extreme valid inequality for the corresponding problem EOP(i).

Proof. By Theorem 3.4.2.1 any such GC(i) is a valid inequality for GP(i). Since every solution to EOP(i) is a solution to GP(i), it must therefore be true that GC(i) is a valid inequality for EOP(i).

To show that GC(i) is an extreme valid inequality for EOP(i) it is only necessary to retrace the argument in the proof of Theorem 3.4.2.2. The existence of valid inequalities w and z satisfying (3-67) would continue to be contradicted by the constructed solutions (3-68) and (3-71)-(3-72) because these solutions also satisfy the one-row either-or problem EOP(i).

Q.E.D.

To understand why Theorem 3.4.3.1 cannot be generalized even to the two-row case, it is necessary to understand the essential property of GP(FCP) that made (3-68) and (3-71)-(3-72) feasible solutions. In an intuitive sense this property might be described as "independence" of the rows of (3-22), i.e. the constraints of (3-22) can each be satisfied independently by choosing an appropriate value of Δs_i. Such Δs_i may be highly infeasible for EOP(Γ)'s, but they will produce a feasible solution for GP(Γ). Proceeding from these intuitive ideas, a sufficient condition to achieve this "independence" in two-row EOP's will now be developed.

3.4.3.2 Definition. Row i is said to be *disconnected* in EOP(i,j) if there exists a pair (k,n) such that

$$a_{kn}(i) > 0 \quad \text{and} \quad a_{kn}(j) = 0. \tag{3-81}$$

Similarly, when no such pair exists, i is said to be *connected* in EOP(i,j).

The property of being disconnected can best be understood as a "two way independence." When both the slack Δs_i, which has a negative coefficient, and some component Δx_{kn} with a positive i coefficient can be chosen independently of any row j considerations, row i of EOP(i,j) can always be satisfied. The next theorem demonstrates the expected implication of this fact on the Gomory cuts.

3.4.3.3 Theorem. If row i is disconnected in EOP(i,j), then GC(j) is an extreme valid inequality for EOP(i,j).

Proof. Validity of GC(j) follows from the fact that every solution to EOP(i,j) solves GP(i,j). Thus, since Theorem 3.4.2.1 assures GC(j) is a valid inequality for such GP(i,j), it is certainly a valid inequality for EOP(i,j).

Now suppose, by contradiction, that GC(j) can be expressed in terms of two distinct valid inequalities w and z as shown in (3-67), i.e. that GC(j) is not an extreme valid inequality for EOP(i,j). By argument in the proof of Theorem 3.4.2.2, it must be true that

$$w_{ht} = z_{ht} = g_{ht}(j)$$

for all pairs (h,t) with $a_{ht}(j) = 0$, and also that

$$w_s = z_s = g_s(j).$$

Thus there must be some pair (h,t) for which $a_{ht}(j) \neq 0$ and $w_{ht} \neq z_{ht}$.

Let (h',t') be such a pair, Δx_{kn} be the variable with coefficients as in (3-81), and

$$w_{h't'} < g_{h't'}(i) < z_{h't'}.$$

Then, consider the solution

$$\Delta\hat{x}^N_{ht} = \begin{cases} \dfrac{1}{g_{ht}(j)} & \text{if } h=h',\ t=t' \\ \dfrac{\bar{y}_i}{a_{kn}(j)} - \dfrac{a_{h't'}(i)}{a_{kn}(j)g_{h't'}(j)} & \text{if } h=k,\ t=n,\ a_{h't'}(i) < 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\Delta\hat{s}_j = 0$$

$$\Delta\hat{s}_i = \begin{cases} u_{1i} - \bar{y}_i + \dfrac{a_{h't'}(i)}{g_{h't'}(j)} & \text{if } a_{h't'}(i) \geq 0 \\ \\ 0 & \text{otherwise.} \end{cases}$$

All components of this solution are non-negative by the facts that $a_{kn}(j) > 0$ and $u_{1i} > \bar{y}_i > 0$. Moreover, direct substitution shows that this solution satisfies the constraints (3-74) for both row i and row j. Thus the solution is a feasible solution for EOP(i,j), but

$$w_1^T \Delta \hat{x}_1^N + \hat{w}_2^T \Delta \hat{x}_2^N + w_s^T \Delta \hat{s} = \frac{w_{h't'}}{g_{h't'}(j)} < \frac{g_{h't'}(j)}{g_{h't'}(j)} = 1$$

which contradicts the validity of w.

Q.E.D.

The intuitive principle of "two way independence" discussed above, together with the theoretical result of Theorem 3.4.3.3 suggest that connectedness may be a measure of the degree of interaction between pairs of rows from EOP(FCP). When i is connected in EOP(i,j), or better yet, with both i and j are connected in EOP(i,j), it seems reasonable to conclude that there is a relatively high degree of interaction between the two rows. The application of such a criterion in a branch-and-bound procedure is discussed in the next section.

3.5 Two-Row Penalty Problems

In Section 3.3.3 it was proposed that the basic method of Gomory and Johnson [43] for deriving penalties in a branch-and-bound procedure from one-row group problems GP(i), be extended to solving problems BGP(i), where the latter problems include the constraints of GP(i) and upper bounds on the perturbation variables. In Section 3.4 the addition of the constraints (3-25) to group problems to obtain either-or problems was discussed. If the principles of including bounds on perturbation variables and of limiting solutions to those satisfying the either-or case were combined, the result would seem to be even stronger penalty problems, i.e. even better bounds on $\nu(FCP)$.

Theorem 3.3.3.1 demonstrates, however, that such would not be the case if only one row is considered. In the one-row case, any optimal solution to GP(i) or BGP(i) automatically satisfies the constraints of EOP(i) as well. Thus

$$\nu\big(GP(i)\big) = \nu\big(EOP(i)\big)$$

and

$$\nu\big(BGP(i)\big) = \nu\big(EOP(i)\text{: (3-58),(3-59) and (3-60)}\big).$$

From this dilemma it can be concluded that problems with at least two rows would need to be solved in order for the effects of either-or limitations to be reflected in penalties. However, solution of an either-or problem with k rows by the method of Theorem 3.3.3.1 involves solving 2^k linear programs in order to decide which right-hand-side provides the minimum value. Thus the computational difficulty of such penalty problems increases rapidly with the number of rows considered.

For these reasons, two-row penalty problems were selected for detail consideration in the research reported in this dissertation. The following sections investigate the strength of two-row problems and discuss methods for using such problems in a branch-and-bound scheme like the one discussed at the beginning of this chapter.

3.5.1 Strength of Two-Row Problems

Define the *bounded either-or problem BEOP(Γ)* by

$$\min \quad d_1^T \Delta x_1^N + \quad d_2^T \Delta x_2^N + c_s^T \Delta s + \nu(\overline{FCP}) \tag{3-82}$$

$$\text{s.t.} \quad a_1(i)^T \Delta x_1^N + a_2(i)^T \Delta x_2^N - \Delta s_i = \bar{y}_i \text{ or } \bar{y}_i - u_{1i} \quad \text{for } i \epsilon \Gamma \tag{3-83}$$

$\big(BEOP(\Gamma)\big)$

$$u_1^N \geq \Delta x_1^N \geq 0, \quad o_2^N \geq \Delta x_2^N \geq 0, \quad u_1 \geq \Delta s \geq 0. \tag{3-84}$$

Then the two-row cases BEOP(i,j) clearly satisfy

$$\nu\big(BEOP(i,j)\big) \geq \nu(\overline{FCP})$$

$$\nu\big(BEOP(i,j)\big) \geq \max\{\nu\big(GP(i)\big), \nu\big(GP(j)\big)\}$$

$$\nu\big(BEOP(i,j)\big) \geq \max\{\nu\big(BGP(i)\big), \nu\big(BGP(j)\big)\}$$

$$\nu\big(BEOP(i,j)\big) \geq \max\{\nu\big(BEOP(i)\big), \nu\big(BEOP(j)\big)\}$$

$$\nu\big(BEOP(i,j)\big) \geq \nu\big(GP(i,j)\big)$$

$$\nu\big(BEOP(i,j)\big) \geq \nu\big(BGP(i,j)\big)$$

$$\nu\big(BECP(i,j)\big) \geq \nu\big(EOP(i,j)\big)$$

because the set of feasible solutions to BEOP(i,j) is included in the set of feasible solutions to each of the problems on the right-hand side of the inequalities.

To see that in contrast to the one-row case, all the above inequalities can hold as strict inequalities, consider the BGP(1,2)

$$\begin{aligned} \min \quad & \Delta x_{21} + 3\Delta x_{22} + 1000\Delta s_1 + 1000\Delta s_2 + 500 \\ \text{s.t.} \quad & \Delta x_{21} + \Delta x_{22} - \Delta s_1 = 5 \text{ or } -4 \\ & 2\Delta x_{21} + 2\Delta x_{22} - \Delta s_2 = 8 \text{ or } -12 \end{aligned}$$

$$3 \geq \Delta x_{21} \geq 0, \quad 11 \geq \Delta x_{22} \geq 0, \quad 9 \geq \Delta s_1 \geq 0, \quad 20 \geq \Delta s_2 \geq 0.$$

For this example the solutions to the penalty problems listed above are given by

$$\nu\left(\text{BEOP}(1,2)\right) = 2509$$

$$\nu(\overline{\text{FCP}}) = 500$$

$$\nu\left(\text{GP}(1)\right) = 505$$

$$\nu\left(\text{GP}(2)\right) = 504$$

$$\nu\left(\text{BGP}(1)\right) = 509$$

$$\nu\left(\text{BGP}(2)\right) = 506$$

$$\nu\big(\text{BEOP}(1)\big) = 509$$

$$\nu\big(\text{BEOP}(2)\big) = 506$$

$$\nu\big(\text{GP}(1,2)\big) = 514$$

$$\nu\big(\text{BGP}(1,2)\big) = 536$$

$$\nu\big(\text{EOP}(1,2)\big) = 2505.$$

Thus $\nu\big(\text{BEOP}(1,2)\big)$ provides a better bound on the value of its FCP than any other one or two-row combination of the group-related penalty schemes so far discussed.

Note also that the results for this example show that each type of two-row penalty problem can provide a strictly better bound than either of its one-row relaxations. Thus, while not as strong as BEOP(i,j), the two-row problems GP(i,j), BGP(i,j) and EOP(i,j) are each superior to corresponding one-row problems. Moreover, the BEOP(i,j) and EOP(i,j) are relatively easy to solve. It is only necessary to solve four two-row linear programs with different right-hand-sides to identify the optimal solution.

3.5.2 Use of Two-Row Problems in a Branch-and-Bound Procedure

At least two different schemes might be devised for using two-row problems like EOP(i,j) and BEOP(i,j) in the branch-and-bound procedure discussed at the beginning of this chapter. One would solve at least one two-row problem for each row i still free at a given stage of the branch-and-bound algorithm. The results of these penalty problems would then be used to bound completions of the current candidate solution, to identify elements of the sets Θ_D and Θ_U, and to guide selection of a branching variable $y_{\hat{i}}$.

A different scheme would be to use two-row problems only as a

[illegible] procedure (e.g. solution of

CHAPTER IV

SPECIAL STRUCTURES OF THE FIXED CHARGE NETWORK PROBLEM

As indicated in Section 1.2, the *fixed charge network problem* is the special case of FCP defined by

$$
\text{(FCNP)} \quad
\begin{aligned}
\min \quad & c_1^T x_1 + c_2^T x_2 + c_s^T s & \quad (4\text{-}1)\\
\text{s.t.} \quad & E_1 x_1 + E_2 x_2 = 0 & \quad (4\text{-}2)\\
& -Ix_1 + Iy - Is = 0 & \quad (4\text{-}3)\\
& u_1 \ge y \ge 0 & \quad (4\text{-}4)\\
& u_1 \ge x_1 \ge 0 & \quad (4\text{-}5)\\
& u_1 \ge s \ge 0 & \quad (4\text{-}6)\\
& u_2 \ge x_2 \ge \ell_2 & \quad (4\text{-}7)\\
& y \equiv 0 \bmod u_1, & \quad (4\text{-}8)
\end{aligned}
$$

where (E_1,E_2) is the node-arc incidence matrix of a directed network. In this chapter simplifications of the analysis of Chapter III resulting from the very special structure of the constraints in FCNP will be presented. In particular, a number of the results of Chapter III will be interpreted for the network case, and computational procedures for constructing and solving penalty problems will be presented which make extensive use of the special properties of FCNP.

4.1 Review of Graph Theoretic Interpretations of Continuous Network Problems

Because E_1 in (4-2) is a node-arc incidence matrix, each of its columns has exactly one -1 and one +1 non-zero entry. For a given column j let that -1 entry be in row i, and consider replacing row i in (4-2) by the difference of its current entries and those of row j in (4-3). This transformation revises columns for x_{1j}, y_j and s_j in FCNP, with all other columns remaining unchanged. In particular, the revised columns for x_{1j}, y_j and s_j contain exactly one +1 and one -1 non-zero entry. Thus, if such a transformation were performed for each column j, the effect would be to convert the constraints (4-2) and (4-3) into a node-arc incidence matrix.

The implication of this observation is that any problem FCNP or $\overline{\text{FCNP}}$ can be interpreted as a minimal cost flow problem in a single commodity network. For a given network defined by (E_1,E_2) the corresponding network for FCNP is constructed by replacing each arc x_{1j} by a three-arc set as shown in Figure 3.

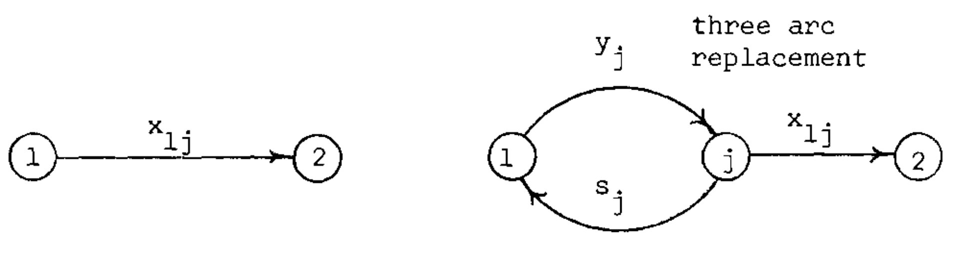

Figure 3. Construction of the FCNP Network from the Underlying Network

All constraints and properties of FCNP can then be interpreted in terms of this expanded network. In particular, the problem $\overline{\text{FCNP}}$ is a continuous minimum cost flow problem on this expanded network, and the vast theory of such problems is applicable.

One of the principal ideas developed in Chapter III was that an optimal solution, optimal tableau, and associated group and other penalty problems for the continuous relaxation $\overline{\text{FCP}}$ of any problem FCP could be easily constructed from the reduced problem RP. In particular, rows of the group or other penalty problems associated with $\overline{\text{FCP}}$ can be extracted directly from corresponding rows of the optimal tableau for RP.

For $\overline{\text{FCNP}}$, the corresponding reduced problem is

(RNP)

$$\min \quad c_1^T x_1 + c_2^T x_2 \tag{4-9}$$

$$\text{s.t.} \quad E_1 x_1 + E_2 x_2 = 0 \tag{4-10}$$

$$u_1 \geq x_1 \geq 0 \tag{4-11}$$

$$u_2 \geq x_2 \geq \ell_2 \tag{4-12}$$

Thus RNP is a continuous minimum cost flow problem on the original problem network.

One set of theory and procedures for network problems like $\overline{\text{FCNP}}$ and RNP is the work of Dantzig [16], Johnson [63], Glover, Klingman and Kearny [36], Langley [75], Kennington and Langley [69], and others on graph theoretic interpretations of such problems. Later in the chapter, this graph theoretic approach will be exploited in procedures for FCNP. Before proceeding to those discussions, however, a number of relevant definitions and results from the graph theoretic approach will be reviewed. In the interest of brevity, proofs of most of the results will be omitted, and the reader referred to more complete discussions in the original works. Only intuitive discussions and examples will be given here, and discussions will be limited to the (sometimes special) cases which can occur in RNP and $\overline{\text{FCNP}}$.

4.1.1 Description of a Basis

As defined above, the constraint sets of the minimum cost flow problems for $\overline{\text{FCNP}}$ and RNP are not of full rank because the sum of the rows is a zero vector. Thus, in order to talk of a basis for these problems, it will be useful to think of adding an identity matrix to E_2 and appending appropriate components to x_2. The new components of x_2 can be thought of as one-ended artificial arcs pointing into each node.

In the terminology of the graph theoretic approach to network problems, a *graph* is a collection of arcs and nodes associated with some network; a *cycle* is a connected set of two-ended arcs of the graph which touches nodes in such a way that every node is touched by exactly two arcs; a *tree* is a connected set of two-ended arcs which contains no cycles; and a *forest* is a set of trees. A forest is said to *span* a graph if each node is touched by exactly one tree. If a one-ended arc is added to each tree so that the number of nodes is equal to the number of arcs, the one-ended arc is called a *root*, and the tree is said to be a *rooted tree*. A collection of such rooted trees is a *rooted forest*, and a rooted forest which spans a network is a *rooted spanning forest*.

In terms of these definitions, the fundamental result on which the graph theoretic approach to network flow problems is based can be stated as follows:

4.1.1.1 Theorem [16,p.356]. The decision and artificial arcs associated with any basis of a network flow problem like RNP form a rooted spanning forest for the network.

Define the node of a rooted tree touched by the root as the *base* of the tree. Then the importance of Theorem 4.1.1.1 derived from the fact that by systematically searching from the base of each tree in the spanning forest associated with a basis for a problem like RNP, it is possible to reach all basic arcs and all nodes without cycling. As the following sections will demonstrate, this implies that all Simplex operations can be performed on the basis of a network problem with a minimum of effort.

To avoid dealing with a number of special cases in these discussions, it will be convenient to think of adding an artificial node to the network of RNP called the *base of the forest*. Artificial arcs will then run from the base of the forest to the base of each tree.

Also, it will be useful to define the direction *up* in a tree as away from the base of the forest, and *down* as toward the base. Similarly, nodes or arcs will be said to be *above* a given node or arc in a tree if they can be reached by proceeding up the tree from the given node or arc.

An example of a basis forest is given in Figure 4. The bases of the three trees are nodes 1, 7 and 3, with roots x_{28}, x_{27}, and x_{29}, respectively. Moving from node 4 to node 7 is proceeding down in the forest, and the part of the tree above arc x_{29} is the branch consisting of x_{23}, x_{22}, x_{12}, x_{21} and nodes 2, 3, 5 and 6 and 8.

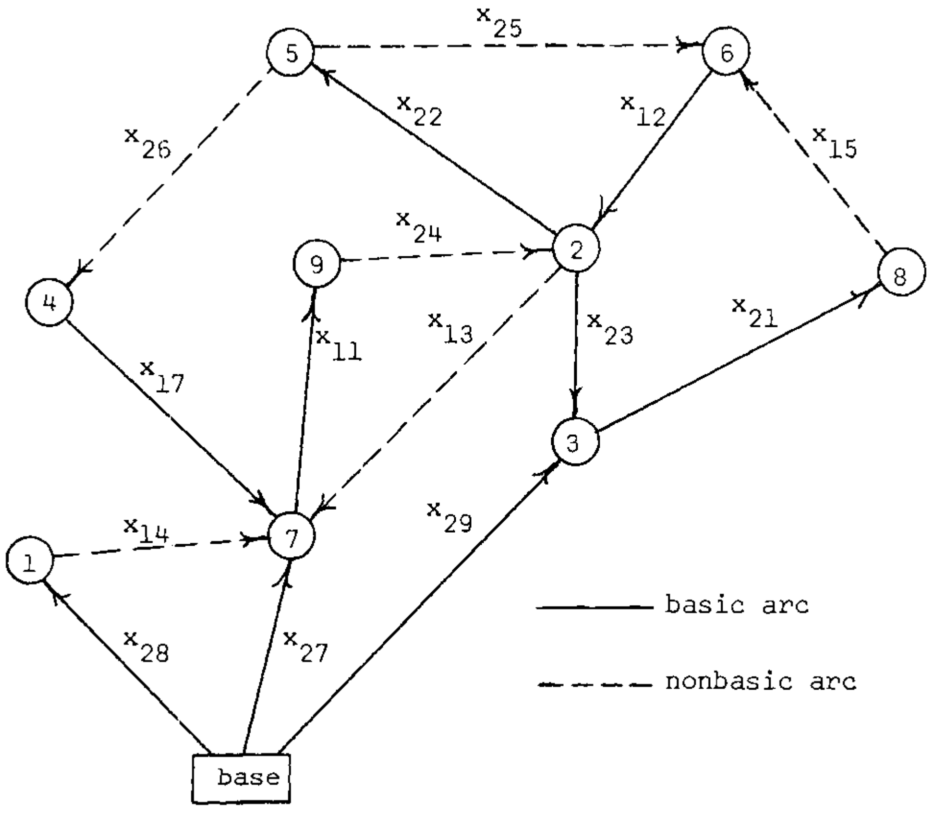

Figure 4. Example of a Basis Forest for RNP

4.1.2 Columns of the Simplex Tableau

One important step in any Simplex-type algorithm is to generate columns of the updated Simplex tableau, i.e. representations of the effect on basic variables of changes in the value of nonbasic variables. For the case of a network problem, Theorem 4.1.1.1 greatly simplifies this process. Since any basis is a rooted spanning forest, the introduction of any nonbasic arc into the forest causes a cycle. Thus a flow change on a nonbasic arc effects exactly those basic arcs along its cycle, and such basic arcs will be the only ones with non-zero entries in the updated tableau column corresponding to the nonbasic arc.

Consider, for example, the case of x_{15} in Figure 4. Basic arcs with non-zero entries in the updated tableau column of x_{15} would be x_{12}, x_{23} and x_{21}. Similarly, the column of x_{26} would have non-zero entries for x_{22}, x_{23}, x_{17}, and the artificial variables x_{27} and x_{29}. A precise statement of these principles is given in the following theorem.

4.1.2.1 Theorem [75, Section 3.3]. For a nonbasic, lower-bounded arc in a network problem define the forward direction as the direction of flow of the arc, and the reverse direction as the opposite of the forward direction. For a nonbasic, upper-bounded arc, define the forward and reverse directions exactly oppositely. Then in the updated Simplex tableau associated with a particular basis forest for the network problem, the column associated with a given nonbasic variable will have a -1 coefficient for every basic variable which points in the forward direction along the cycle introduced by adding the nonbasic arc to the basis forest, a +1 coefficient for every basic variable which points in the reverse direction along this cycle, and 0 coefficients for all other basic variables.

4.1.3 Rows of the Basis Inverse

Another important element of Simplex operations is the inverse of the basis matrix. It is well known that the updated Simplex tableau for a given basis can be obtained by multiplying the original tableau by the basis inverse.

In the network case the basis inverse, like many other Simplex elements, can be generated in terms of the basis forest representation. In particular, note that a row of the basis inverse $[(E_1^B,E_2^B)^{-1}]_i$ must satisfy

$$[(E_1^B,E_2^B)^{-1}]_i(E_1^B,E_2^B) = (0,\ldots,0,1,0,\ldots,0) \qquad (4\text{-}13)$$

where the 1 of the vector on the right is in the *ith* position. Remember, however, that every column of the basis (E_1^B,E_2^B) contains at most one +1 and one -1, with all other entries 0. Thus the entries in a row of the basis inverse corresponding to a given basis forest can be constructed by proceeding up the trees in the forest and calculating the entries for each node so that the difference of the entries at the ends of each basic arc will satisfy (4-13).

For example, consider the row of the inverse corresponding to arc x_{23} in Figure 4. If the entry for the base of the forest is fixed at 0, then entries for all nodes not above x_{23} in the forest must also be 0 in order to obtain 0 differences on the right-hand side of (4-13). Similarly, the entries for nodes 2, 5 and 6 must be equal in order to obtain 0 differences in (4-13). But the entry for node 2 must be one less than that of node 3 in order to produce the single 1 in (4-13). Thus, the row of the basis inverse corresponding to x_{23} is given by

node	base	1	2	3	4	5	6	7	8	9
entry	0	0	-1	0	0	-1	-1	0	0	0 .

A more formal statement of these principles is given in the following theorem.

4.1.3.1 Theorem [68, p.55]. The row of the basis inverse corresponding to any basic arc of RNP will have +1 entries for all nodes above the arc in the basis forest if the arc is directed away from the base of the forest and -1 entries for all nodes above the arc in the basis forest if the arc is directed toward the base of the forest. All nodes not above the arc in the basis forest will have 0 entries.

4.1.4 Rows of the Simplex Tableau

From the results of Chapter III, it is clear that an important element of any group-theory-based algorithm is the ability to generate rows of the optimal Simplex tableau for RNP. In particular, it is necessary to produce the parts of rows of the optimal tableau corresponding to nonbasic variables.

A constructive method for generating such rows is given by Theorem 4.1.3.1. After generating the row of the basis inverse for a given basis variable, it is only necessary to subtract the entries for the end nodes of a nonbasic arc to calculate the entry in the updated Simplex tableau for that arc. In Figure 4, for example, the column corresponding to x_{15} in the updated Simplex tableau would have the entry

$$+1\ (-1)\ -1\ (0) = -1$$

in the row of x_{23}.

For some of the development which follows, however, it will be useful to have a more intuitive concept of the rows of the updated Simplex tableau. Recall the principle of Section 4.1.2 that the only basic arcs affected by the introduction of a given nonbasic arc into a basis forest are those along the cycle the nonbasic arc produces in the forest. This implies that the row of the updated tableau for a given basic arc will contain non-zero entries for nonbasic arcs only when the basic arc is part of the cycles corresponding to those nonbasic arcs.

Membership of a basic arc in the cycle corresponding to a nonbasic arc can be characterized in terms of the branch of the basis forest above the basic arc. If a nonbasic arc touches exactly once in this branch, then the basic arc must be part of the cycle associated with the nonbasic arc because a path from any part of the basis forest to the branch above an arc must include that arc. Similarly, if a nonbasic arc touches twice in the branch above a basic arc, or does not touch at all, then the basic arc is not part of the cycle associated with the nonbasic arc.

For example, consider the row of the optimal tableau for the basis in Figure 4 corresponding to arc x_{29}. Arc x_{26} touches only once in the branch of the tree above x_{29}, and thus x_{29} is part of the cycle corresponding to x_{26}. On the other hand, x_{15} touches twice in the branch above x_{29}, and x_{14} does not touch at all. Thus x_{29} is not a part of the cycles for either x_{15} or x_{14}.

A formalization of these principles is given in the following theorem.

4.1.4.1 Theorem [75, Section 3.4]. Let x_{ij} be a basic arc in a network problem and $\Psi(x_{ij})$ be the set of nodes above x_{ij} in the corresponding basis forest. Then any nonbasic arc flowing from node w to node z will have an entry in the updated Simplex tableau row for x_{ij} as follows:

(i) If $w \in \Psi(x_{ij})$ and $z \notin \Psi(x_{ij})$, the entry is -1 if the nonbasic arc is lower-bounded and x_{ij} is oriented away from the base of the forest, or if the nonbasic arc is upper-bounded and x_{ij} is oriented toward the base of the forest, and the entry is +1 otherwise.

(ii) If $w \notin \Psi(x_{ij})$ and $z \in \Psi(x_{ij})$, the entry is +1 if the nonbasic arc is lower-bounded and x_{ij} is oriented away from the base of the forest, or if the nonbasic arc is upper-bounded and x_{ij} is oriented toward the base of the forest, and the entry is -1 otherwise.

(iii) In all other cases the entry is 0.

4.1.5 Labeling of the Basis Forest

The previous four sections have shown how important components of the optimal basic solution for problems like RNP can be generated from the basis forest associated with an optimal solution. In order to actually perform the procedures described, however, a set of labels which define the basis forest would be required.

In particular, review of the processes described in the previous sections will show that labels must support at least three functions. First, it must be possible to identify the bases of all trees. Second, labels must facilitate searching down the forest from a given point to identify the cycle associated with a nonbasic arc. Finally, labels must permit moving up basis trees to construct basis inverses, etc.

Johnson [64], Langley [75], and Kennington and Langley [69], Glover Klingman and Kearny [36], and others have proposed a number of labeling schemes which will permit such functions. The one which appears to be the most satisfactory for FCNP is given in the following definition.

4.1.5.1 Definition. The *basis label* of a node w in a network problem like RNP is $\{\delta(w),\mu(w),\gamma(w),\alpha(w)\}$ where

$\delta(w)$ = the number of the node directly below w in the basis forest (0 if the node is the base of the forest).

$\mu(w)$ = the number of a node directly above w in the basis forest (0 if no such node exists).

$\gamma(w)$ = the number of a node z such that $\delta(w) = \delta(z)$ and $\gamma(\hat{z}) \neq z$ for all $\hat{z} \neq w$ satisfying $\delta(w) = \delta(\hat{z})$ (0 if no such node exists).

$\alpha(w)$ = the number of the arc connecting w and $\delta(w)$.

The components of the basis label are referred to as the *down node, up node, right node* and *down arc,* respectively.

Figure 5 provides one of several such sets of basis labels for the basis forest in the example of Figure 4. For example, the down node label of node 4 is 7 because 7 in the next closer node to the base of the forest, and the down arc label of node 4 is 17 because arc 17 connects nodes 4 and 7. The chain of nodes immediately above node 3 begins with node 2, which is the up node from 3, and proceeds right to node 8.

By using the basis labels of Definition 4.1.5.1, all the network functions mentioned in the previous sections can be performed. Bases of trees are identified by those nodes having down node equal to 0.

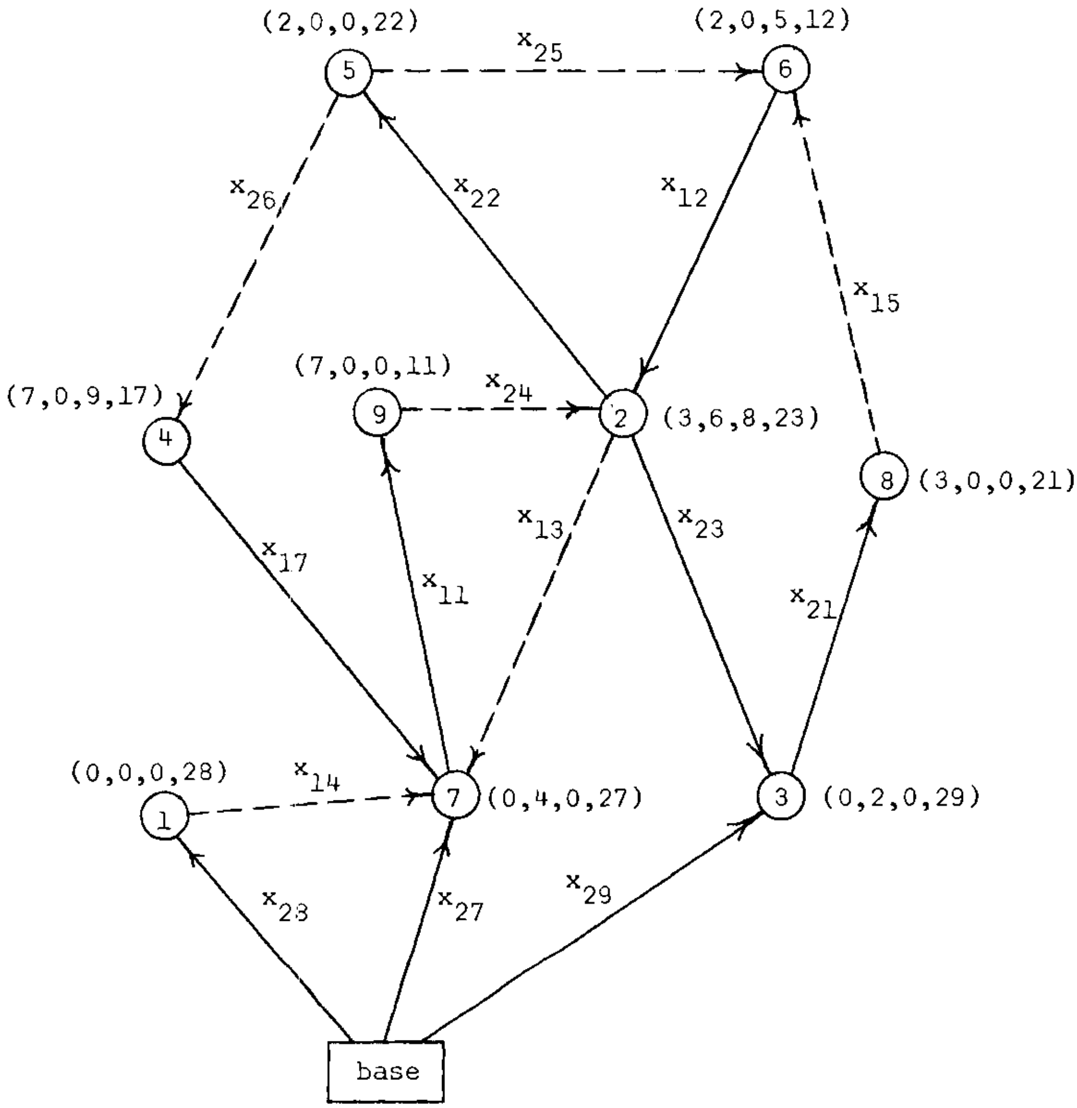

Figure 5. Basis Labels for the RNP Example

The basic arcs in the cycle associated with a non-basic arc can be identified by proceeding down from both ends of the nonbasic arc to the base of the forest. All down arcs encountered exactly once in this search are members of the cycle. To identify the part of a tree above a given node, first follow up node labels from that node. When a 0 up node label is encountered, follow the right node label one step and proceed up again. If both up node and right node labels are 0, proceed down one node and then right. All desired nodes have been searched when the original node is re-encountered.

For the example of Figure 5, the cycle associated with nonbasic arc x_{15} is identified by proceeding down from both its ends. Down arcs from the node 6 end are x_{12}, x_{23} and x_{29}. Those from the 8 end are x_{21} and x_{29}. Thus the cycle is x_{12}, x_{23} and x_{21}.

To identify the branch of the tree above arc x_{29}, begin at node 3. Then successively move up to node 2, up to node 6, right to node 5, down to node 2, right to node 8, and back to node 3. The nodes in the branch are thus nodes 2, 3, 5, 6 and 8, and the arcs in the branch are the down arcs from nodes 2, 5, 6 and 8.

One final observation about the labels of Definition 4.1.5.1 is that the functions they support are not only the ones necessary for generating updated tableaux in group problems, but also the ones required to perform pivots of the primal Simplex method.[1] Thus when a Simplex algorithm using such labels is employed to solve RNP, the labels for an optimal basis forest will automatically be available for group analyses at the completion of the Simplex procedure.

---

[1]See Section A.2.2 of Appendix A for discussion of how the steps of a Simplex pivot are performed with the labels.

4.2 Group-Related Problems in Terms of the Group Theoretic Description

Analogously to Section 3.3, define

$$d_1 = \begin{pmatrix} \bar{c}_1^L \\ -\bar{c}_1^U \end{pmatrix}$$

$$d_2 = \begin{pmatrix} \bar{c}_2^L \\ -c_2^U \end{pmatrix}$$

$$e_1(i) = \begin{bmatrix} [\bar{\bar{E}}_1^L & -\bar{\bar{E}}_1^U]^1_{k^B} \\ -I & 0 \\ 0 & I \end{bmatrix}$$

$$e_2(i) = \begin{bmatrix} [\bar{\bar{E}}_2^L & -\bar{\bar{E}}_2^U]^1_{k^B} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

where $\bar{c}_1^L$, $\bar{c}_1^U$, $\bar{c}_2^L$, $\bar{c}_2^U$, $\bar{\bar{E}}_1^L$, $\bar{\bar{E}}_1^U$, $\bar{\bar{E}}_2^L$, $\bar{\bar{E}}_2^U$ and k^B are derived from the optimal tableau of $\overline{\text{FCNP}}$ as indicated in Section 1.4. Then group problems GP(Γ) for FCNP can be stated

$$\min \quad d_1^T \Delta x_1^N \;+\; d_2^T \Delta x_2^N + c_s^T \Delta s + \nu(\overline{FCNP}) \tag{4-14}$$

$$\text{s.t.} \quad e_1(i)^T \Delta x_1^N + e_2(i)^T \Delta x_2^N - \Delta s_i \equiv \bar{y}_i \bmod u_{1i} \text{ for } i \in \Gamma \tag{4-15}$$

$(GP(\Gamma))$

$$\Delta x_1^N,\ \Delta x_2^N,\ \Delta s \geq 0. \tag{4-16}$$

When the constraints (4-15) are replaced by

$$e_1(i)^T \Delta x_1^N + e_2(i)^T \Delta x_2^N - \Delta s_i = \bar{y}_i \text{ or } \bar{y}_i - u_{1i} \quad \text{for } i \in \Gamma \tag{4-17}$$

the either-or problem EOP(Γ) is obtained, and if the constraints

$$u_1^N \geq \Delta x_1^N, \quad o_2^N \geq \Delta x_2^N, \quad u_1 \geq \Delta s \tag{4-18}$$

are added to GP(Γ) and EOP(Γ), the bounded problems BGP(Γ) and BEOP(Γ) result. In this section interpretations and simplifications of these group-related problems will be developed in terms of the graph-theoretic ideas presented in Section 4.1.

4.2.1 Interpretation of the Group-Related Problems

To interpret the meaning of these group-related problems on the network of FCNP, it is first necessary to construct the basis for $\overline{FCNP}$ implied by Theorem 3.1.1.4. Given a basis forest for RNP, the corresponding forest for $\overline{FCNP}$ is constructed by replacing each arc corresponding to a component of x_1 by a three-arc set with an artificial node as shown in Figure 3. In accordance with Theorem 3.1.1.4, the arc x_{1j} is basic only if it was basic for RNP, the arc y_j is always basic, and the arc s_j is always nonbasic.

Figure 6 shows this full basis forest for the case of Figure 4. Arc x_{17}, for example, has been replaced by the three-arc set x_{17}, y_7 and s_7. The new x_{17} is basic because it was basic in Figure 4, y_7 is basic because all y_j are basic in the solution of Theorem 3.1.1.4, and s_7 in nonbasic because all s_j are nonbasic in the Theorem 3.1.1.4 solution. The construction for x_{14} is similar, except that only y_4 is basic because x_{14} was nonbasic in Figure 4.

Recall that the introduction of any nonbasic arc into this basis forest produces a cycle. Moreover, the forward direction of flow around this cycle is the direction of flow on the nonbasic arc if it is lower-bounded and the opposite if the arc is upper-bounded.

With these concepts in mind, review of the definitions of perturbation variables in (3-18) will show that increasing a perturbation variable from 0 to an amount h is exactly equivalent to introducing a flow change of size h in the forward direction around the cycle corresponding to the nonbasic arc with which the perturbation variable is associated. For example, if x_{26} is lower-bounded in the case of Figure 6, then setting Δx_{26} = h is equivalent to introducing a flow change of size h around the cycle $\{x_{26}, y_7, x_{17}, x_{27}, x_{29}, x_{23}, x_{22}\}$. This additional flow will add to the existing flow along basic arcs oriented in the forward direction and subtract from the flow on those oriented in the reverse direction.

Next observe that the rows of the full group problem GP(FCNP) and the corresponding either-or problem EOP(FCNP) require that the

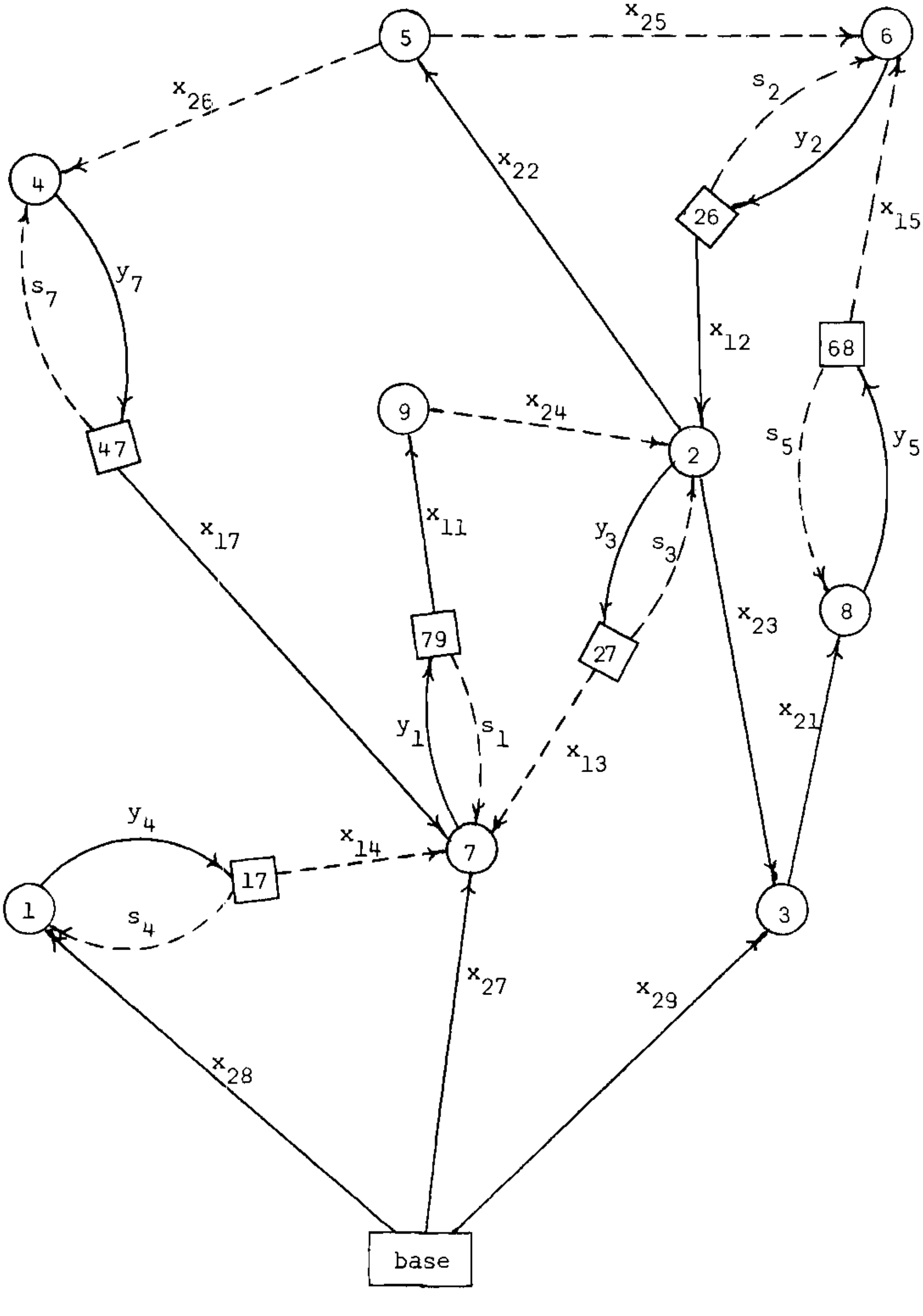

Figure 6. Basis Forest for $\overline{\text{FCNP}}$ Corresponding to the RNP Example

flows along all arcs y_j must be forced into either congruence with u_{1j} or equality with 0 or u_{1j}. Thus the *ith* constraint of (4-15) or (4-17) can be interpreted as requiring that flow changes be introduced in the forward direction around cycles for nonbasic arcs which include y_i in such a way that the flow along y_i is congruent to u_{1i} in the GP(FCNP) case, and equal to 0 or u_{1i} in the EOP(FCNP) case.

Consider, for example, the case of y_7 in Figure 6. This arc is part of cycles corresponding to x_{26} and s_7. Thus the seventh row of (4-15) would require that flow changes in the forward direction be introduced on either the cycle for x_{26}, or the cycle for s_7, or both in such a way as to make the flow on y_7 congruent to u_{17}. The corresponding row of (4-17) requires the stronger result that the flow on y_7 equal either 0 or u_{17}.

When the bounding constraints (4-18) are also considered, flow adjustments around cycles corresponding to nonbasic arcs are further restricted. The additional constraints require that in seeking to satisfy (4-15) or (4-17), flow changes on cycles for nonbasic arcs are not so large as to render infeasible the flows on the nonbasic arcs themselves.

Finally, the above interpretation can be used to show why even BEOP(FCNP) is not equivalent to FCNP. The constraints of BEOP(FCNP) require that all constraints on arcs y_j be satisfied, and that implied values of flows on nonbasic arcs remain feasible. However, it might still be true that the effect of flow changes around various cycles is to force some basic x_{1j} or x_{2j} to take on infeasible flows.

In the case of Figure 6, for example, recall that a flow change might need to be introduced around the cycle of x_{26} in order to make y_7 satisfy (4-17). In BEOP(FCNP) this flow change would be restricted to one which keeps x_{26} feasible and makes y_7 satisfy (4-17). However, it could be that the change would also make arc x_{27}, x_{29}, x_{23} or x_{22} take on an infeasible flow, i.e. a flow which did not satisfy its upper or lower bound.

4.2.2 The Group Tree and Reduced Inverse

By taking advantage of the interpretation of the previous section, a number of computational simplifications can be obtained. First, observe the cycles associated with the slack arcs s_j in Figure 6. Each such cycle contains only one basic arc, i.e. y_j. Thus, in finding a set of flow changes around cycles which will satisfy one of the group-related problems defined above, cycles for the s_j could easily be handled implicitly. Only the more complex cycles of nonbasic x_{ij} require explicit attention.

With this observation, it becomes possible to again deal with group-related problems in terms of the reduced problem RNP. All cycles for nonbasic x_{ij} can be just as effectively traced on the basis forest for RNP as on the forest for $\overline{\text{FCNP}}$, and nonbasic s_j can be handled separately.

For the example of Figure 6 this simplification reduces the problem once again to Figure 4. The following definition leads to an even further reduction.

4.2.2.1 Definition. A *macro-node* of a basis forest for RNP is a single node used to replace any maximal set of ordinary nodes in the forest of RNP which are connected by a tree of basic arcs drawn entirely from the vector x_2. The reduced version of the basis forest obtained by replacing the nodes of each macro-node, and the arcs which connect them, by the macro-node is called the *group tree* for RNP.

To illustrate the concepts of macro-nodes and group trees, consider again the case of Figure 4. Figure 7 shows the group tree which results from the basis forest of that example when all nodes are collapsed into macro-nodes.

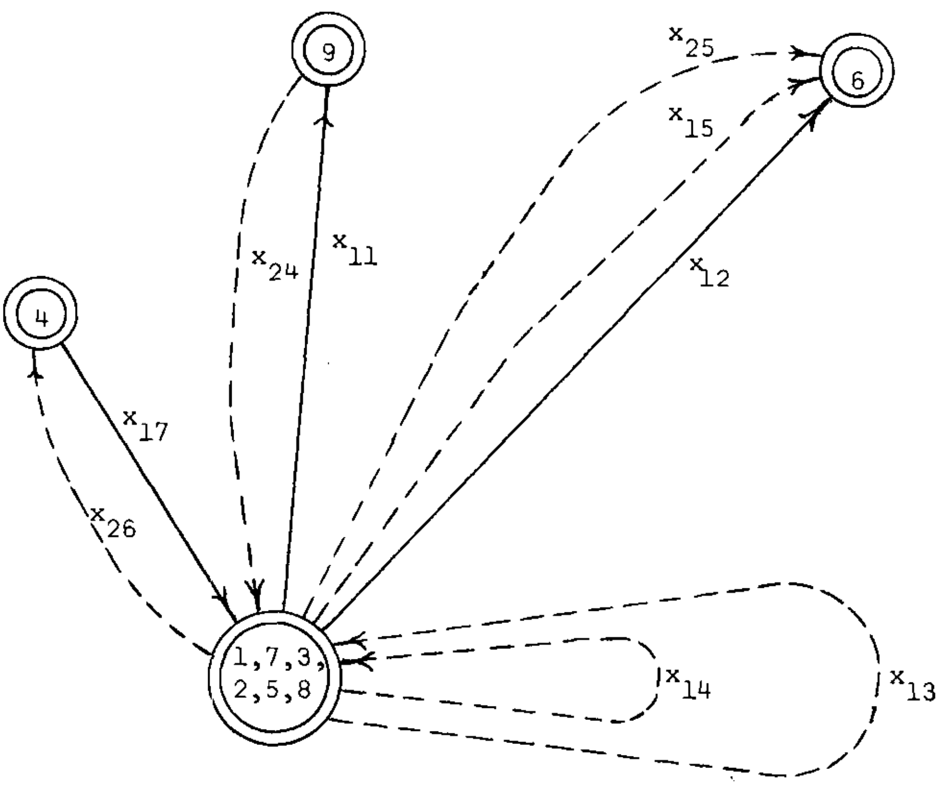

Figure 7. Example of a Group Tree for RNP

For example, the nodes 1, 7, 3, 2, 5 and 8, and the tree of basic arcs connecting them, have been replaced by a single macro-node. This is possible, because all such arcs are components of x_2. Nodes 4, 9 and 6 could not be collapsed because they are connected to the rest of the forest by basic components of x_1. Note that it is proper to refer to the reduced version of the basis forest as a tree because roots are always components of x_2 and thus the bases of all trees in the ordinary basis forest will always be a part of the same macro-node.

Several other observations about the group tree can be made. First, note that one interpretation of a macro-node is that of a collapsed segment of cycles associated with a nonbasic arc. All such cycles are still visible in the group tree, but components x_{2j} of the cycles have been absorbed into macro-nodes so that the effects on basic x_{1j} are highlighted. The values of the basic x_{1j} are, of course, the flows of interest in group-related problems because they correspond directly to the values of the y_j.

Next, note that the cycles in the forest of RNP corresponding to two different nonbasic arcs which connect the same macro-nodes in the group tree will contain exactly the same basic arcs x_{1j}. For example, in the case of Figure 4, the full cycles for x_{25} and x_{15} are $\{x_{22},x_{12}\}$ and $\{x_{21},x_{23},x_{12}\}$, respectively. Both involve only the component x_{12} of x_1, and thus both have exactly the same appearance in the group tree.

Finally, observe that any nonbasic arcs connecting nodes in the same macro-node contain no components of x_1 in their cycles. Thus they affect no basic arcs y_j (except possibly ones associated with the nonbasic x_{1j} themselves) and have little effect in group-related problems.

Examples of such arcs are x_{14} and x_{13} of Figure 7. Flows along these arcs might be changed to correct y_4 or y_3, but changes can have no effects on y_j not directly related to the nonbasic arcs themselves. Thus these nonbasic arcs can largely be disregarded in group-related problems for FCNP.

The above observations about the group tree can be developed more formally in terms of the properties of a basis inverse for RNP presented in Section 4.1.3. The next several results provide such a formalization.

4.2.2.2 Theorem. The columns of a basis inverse for RNP corresponding to any two nodes w and z which are part of the same macro-node in the group tree of RNP will have equal entries in all rows associated with components of x_1^B.

Proof. Consider some component x_{hk} of x_1^B, and let $\Psi(x_{hk})$ be the set of nodes in the branch above x_{hk} in the basis forest for RNP, and $\tilde{\Psi}(x_{hk})$ be the set containing all other nodes of RNP. Now nodes w and z are connected by a chain of components of x_2^B because they are in the same macro-node of the group tree for RNP. Thus x_{hk} cannot be a part of the chain between w and z and so either

$$w \text{ and } z \in \Psi(x_{hk})$$

or

$$w \text{ and } z \in \tilde{\Psi}(x_{hk}).$$

By Theorem 4.1.3.1 this implies w and z have identical entries in the row of the basis inverse corresponding to x_{hk}.

Q.E.D.

<u>4.2.2.3 Corollary</u>. Any nonbasic arc connecting two nodes of RNP which are part of the same macro-node in the group tree for RNP will have zero entries in each of the first k^B rows of the group-related constraints (4-15) and (4-17), i.e. in all rows corresponding to components of x_1^B.

Proof. The entry for a nonbasic variable in any of the first k^B rows of (4-15) or (4-17) can be obtained by multiplying an appropriate row of the basis inverse for RNP by the original tableau column for the non-basic variable. But that original tableau column has only a +1 in the row associated with one node touched by the arc and a -1 in the row associated with the other touched node. Since by Theorem 4.2.2.2 the components of the row of the basis inverse opposite these non-zero entries must be equal, it follows that the result of the vector multiplication will be 0.

Q.E.D.

<u>4.2.2.4 Corollary</u>. Define the forward direction for a nonbasic arc of RNP as in Section 4.2.1, i.e. equal to the direction of flow on the arc when it is lower-bounded and opposed to the direction of flow when the arc is upper-bounded. Further let x_{nk} and $x_{n'k'}$ be two nonbasic arcs of RNP connecting, in the same direction, nodes which are members of the same two macro nodes, i.e. either both arcs have forward direction leading from two nodes in the same macro-node to two nodes which are both in

a different macro-node, or the same is true in the reverse direction. Then the entries in the first k^B rows of (4-15) and (4-17) corresponding to Δx_{nk} and $\Delta x_{n'k'}$ will be equal.

Proof. Follows from Theorem 4.2.2.2 by direct multiplication of the appropriate columns for nonbasic variables and rows of the basis inverse as in the proof of Corollary 4.2.2.4.

Q.E.D.

The importance of Corollary 4.2.2.4 lies in its implication that a *reduced basis inverse* for RNP, which contains one row for each component of x_1^B and one column for each macro-node, is all that is required to generate the non-trivial elements of the constraints (4-15) and (4-17). Thus the group-related problems GP(Γ) and EOP(Γ) can be easily constructed if the following labels are identified.

4.2.2.5 Definition. For each node w in the basis forest for RNP,

$\eta(w)$ = the number of the macro-node to which w belongs.

For each macro-node z in the group tree for RNP,

$\lambda(i,z)$ = the element of the reduced basis inverse for RNP associated with the *ith* component of x_1^B and the macro-node z.

Moreover, the labels of the basis forest for RNP presented in Section 4.1.5 make it very easy to construct such group tree labels. It is only necessary to trace up the forest for RNP, recording a macro-node change whenever a basic component of x_1 is encountered and setting elements of the reduced inverse according to Theorem 4.1.3.1. The details of such a procedure are provided in the following algorithm.

4.2.2.6 Algorithm. Let $(\delta(n),\mu(n),\gamma(n),\alpha(n))$ be the labels of an optimal basis forest for RNP as defined in Definition 4.1.5.1. Then the labels $\eta(n)$, and $\lambda(i,k)$ can be obtained as follows:

Step 0. Set the next available macro-node $\hat{k} = 1$ and $\lambda(i,k) = 0$ for all i and k.

Step 1. Scan sequentially the nodes until a new tree base (i.e. a node n with $\delta(n)=0$) is found. If none is found, stop; the algorithm is complete. Otherwise, set the arc index set $\Lambda = \Phi$, the current node n' = the number of the node which is the new base, $\eta(n') = 1$, and the current macro-node $k' = 1$, and go to Step 2.

Step 2. Proceed up by letting $n = \mu(n')$. If $n = 0$ go to Step 5. Otherwise, proceed to Step 3 if $\alpha(n)$ is a component of x_1 and to Step 4 if it is a component of x_2.

Step 3. Let $\Lambda = \Lambda \cup \{\alpha(n)\}$ if $\alpha(n)$ is oriented away from the base of the forest, and $\Lambda = \Lambda \cup \{-\alpha(n)\}$ if $\alpha(n)$ is oriented toward the base of the forest. Also let $\hat{k} = \hat{k} + 1$, $k' = \hat{k}$, $\lambda(i,k') = +1$ for all i in Λ such that $i > 0$, and $\lambda(-i,k') = -1$ for all i in Λ such that $i < 0$. Then go to Step 4.

Step 4. Set $n' = n$ and $\eta(n') = k'$. Then go to Step 2.

Step 5. Proceed right by setting $n = \gamma(n')$. If $n = 0$ go to Step 6. Otherwise remove $\pm\alpha(n')$ from Λ if $\alpha(n')$ is a component of x_1, set $k' = \eta(\delta(n'))$, and then go to Step 3 if $\alpha(n)$ is a component of x_1, and to Step 4 if it is a component of x_2.

Step 6. Proceed down by setting $n = \delta(n')$. If $n = 0$, go to Step 1. Otherwise remove $\pm\alpha(n')$ from Λ if $\alpha(n')$ is a component of x_1, set $n' = n$ and $k' = \eta(n')$, and go to Step 5.

Table 1 illustrates the algorithm for the case of Figure 4 with labels as in Figure 5.

Table 1. Steps in Algorithm 4.2.2.6 for Example Problem RNP

Algorithm Step	Variables Assigned Values	Algorithm Step	Variables Assigned Values
0	$\hat{k}=1$, all $\lambda(i,k)=0$	2	$n=0$
1	$\Lambda=\Phi$, $n'=1$, $\eta(1)=1$, $k'=1$	5	$n=0$
2	$n=0$	6	$n=3$, $n'=3$, $k'=1$
5	$n=0$	5	$n=0$
6	$n=0$	6	$n=0$
1	$\Lambda=\Phi$, $n'=3$, $\eta(3)=1$, $k'=1$	1	$\Lambda=\Phi$, $n'=7$, $\eta(7)=1$, $k'=1$
2	$n=2$	2	$n=4$
4	$n'=2$, $\eta(2)=1$	3	$\Lambda=\{-17\}$, $\hat{k}=3$, $k'=3$, $\lambda(17,3)=-1$
2	$n=6$	4	$n'=4$, $\eta(4)=3$
3	$\Lambda=\{-12\}$, $\hat{k}=2$, $k'=2$, $\lambda(12,2)=-1$	2	$n=0$
4	$n'=6$, $\eta(6)=2$	5	$n=9$, $\Lambda=\Phi$, $k'=1$
2	$n=0$	3	$\Lambda=\{+11\}$, $\hat{k}=4$, $k'=4$, $\lambda(11,4)=+1$
5	$n=5$, $\Lambda=\Phi$, $k'=1$	4	$n'=9$, $\eta(9)=4$
4	$n'=5$, $\eta(5)=1$	2	$n=0$
2	$n=0$	5	$n=0$
5	$n=0$	6	$n=7$, $\Lambda=\Phi$, $n'=7$, $k'=1$
6	$n'=2$, $k'=1$	5	$n=0$
5	$n=8$, $k'=1$	6	$n=0$
4	$n'=8$, $\eta(8)=1$	1	Stop

4.3 Unimodularity of the Group-Related Problems

A matrix M is said to be *totally unimodular* if the determinant of every square sub-matrix of M is equal to +1, 0 or -1. In this section it will be demonstrated that the main constraint matrix of the group-related problems GP(Γ), BGP(Γ), EOP(Γ) and BEOP(Γ) possesses this convenient property.

4.3.1 Review of Unimodularity Theory

In order to prove total unimodularity for the group-related problems of FCNP, two results from the theory of unimodular matrices are required. They are stated in the following lemmas.

4.3.1.1 Lemma (Heller and Tompkins [56]). Every basis of the matrix M has determinant ±1 if the following are satisfied.

(i) Every column of M has at most two non-zero entries.

(ii) Every element of M is either 0, +1 or -1.

(iii) The rows of M can be partitioned into two sets Ω_1 and Ω_2 such that the rows for non-zero entries with like signs in any column are members of different sets, and rows for non-zero entries with opposing signs in any column are members of the same set.

Proof. Let k be the rank of M. Then the proof proceeds by induction on k. For k = 1 the lemma follows trivially from property (ii). Now assume the lemma holds for k < n and consider a n by n basis $\hat{M}$ of M. Every column of $\hat{M}$ must have at least one non-zero entry because $\hat{M}$ is a basis matrix. Moreover, if any column of $\hat{M}$ contained only one non-zero entry +1 or -1, then expansion by minors on that element would prove the lemma because the minor of the element must have determinant +1, 0 or -1 under the inductive hypothesis. Thus the lemma can only fail to be true if $\hat{M}$ has exactly two non-zero entries in each column. For this case property (iii) implies the sum of the rows of $\hat{M}$ which are members of Ω_1 is exactly equal to the sum of the rows of $\hat{M}$ which are members of Ω_2. But this implies the rank of $\hat{M}$ is less than n which contradicts the fact that $\hat{M}$ is a basis matrix.

Q.E.D.

4.3.1.2 Lemma (Dantzig and Veinott [17]). If every basis of the matrix (M,I) has determinant +1 or -1, then M is totally unimodular.

Proof. Consider some square sub-matrix $\hat{M}$ of M. If $\hat{M}$ is singular then $\det(\hat{M}) = 0$ and the lemma follows trivially. If $\hat{M}$ is non-singular then a basis of the form

$$\begin{pmatrix} \hat{M} & 0 \\ \hat{N} & I \end{pmatrix}$$

can be constructed from the columns of (M,I). But expansion by minors shows the determinant of this basis matrix is equal to $\det(\hat{M})$, and so $\det(\hat{M}) = \pm 1$.

Q.E.D.

4.3.2 Results for Group-Related Problems

With these two lemmas it is possible to prove the following theorem.

<u>4.3.2.1 Theorem</u>. Let R(Γ) be the matrix of coefficients for the left-hand side of (4-15) or (4-17). Then R(Γ) is totally unimodular.

Proof. Every R(Γ) will obviously be totally unimodular if R(FCNP) where $\Gamma = \{1,2,\ldots,n_1\}$ is totally unimodular, and Lemma 4.3.1.2 demonstrates R(FCNP) will be totally unimodular if every basis sub-matrix of $\left(R(FCNP),I\right)$ has determinant +1 or -1. Moreover, every column of R(FCNP) is by definition +1 or -1 times a column of the optimal Simplex tableau for $\overline{FCNP}$. Since the absolute value of the determinant of a matrix does not change if signs are reversed on some columns, it follows that the theorem will be true if every basis of the optimal Simplex tableau for $\overline{FCNP}$ has determinant +1 or -1.

Thus, consider a basis matrix $\bar{R}$ from the optimal $\overline{FCNP}$ tableau. Recall from Section 4.1 that the original tableau for $\overline{FCNP}$ can be rearranged so that it becomes a node-arc incidence, followed by an identity matrix; let R' be the basis matrix of this rearranged original tableau corresponding to $\bar{R}$ in the optimal tableau; and let B' be the basis matrix of the rearranged original tableau which produces the optimal $\overline{FCNP}$ solution. The original tableau for $\overline{FCNP}$ trivially satisfies the requirements of Lemma 4.3.1.1, and so each of the basis matrices R' and B' has determinant ±1. But this implies $(B')^{-1}$ has determinant ±1. Thus,

$$\det(\bar{R}) = \det\left((B')^{-1}R'\right) = \det\left((B')^{-1}\right)\det(R') = +1 \text{ or } -1.$$

Q.E.D.

One use of this result is given by the following corollary.

4.3.2.2 Corollary. In two rows i and j drawn from the constraint matrix R(i,j), either columns from the set

$$\left\{\begin{pmatrix}+1\\-1\end{pmatrix},\begin{pmatrix}-1\\+1\end{pmatrix}\right\} \tag{4-19}$$

or columns from the set

$$\left\{\begin{pmatrix}+1\\+1\end{pmatrix},\begin{pmatrix}-1\\-1\end{pmatrix}\right\} \tag{4-20}$$

may be present, but not both.

Proof. If any combination of a column from the set of (4-19) and a column from the set of (4-20) were present in the same problem, then the 2 by 2 sub-matrix of these two columns would have determinant ±2. For example,

$$\det\begin{pmatrix}+1 & -1\\+1 & +1\end{pmatrix} = +2.$$

But by Theorem 4.3.2.1, every square sub-matrix of the constraints R(i,j) must have determinant +1, 0 or -1. Thus no combination of elements of the sets of (4-19) and (4-20) is possible. Q.E.D.

4.4 Generation and Solution of Penalty Problems

Sections 3.3 and 3.5 developed a variety of one and two-row penalty problems for FCP. In Section 3.5.1 relations between these problems were investigated, and it was demonstrated that the *bounded either-or problems* BEOP(i) and BEOP(i,j) were the strongest one and two-row problems, respectively.

For the case of FCNP, these strongest problems are defined as follows:

BEOP(i) = the problem defined by (4-14), (4-16), (4-18), and the *ith* row of (4-17).

BEOP(i,j) = the problem defined by BEOP(i) with the addition of the *jth* row of (4-17).

Taking advantage of the analysis in previous sections of this chapter, procedures for constructing and solving these penalty problems for FCNP will now be presented. Moreover, since all the other one and two-row penalty problems of Chapter III are relaxations of some BEOP, the following constructions also provide the necessary methods for obtaining GP's, BGP's and EOP's.

4.4.1 Generation of Penalty Problems

Assume that an optimal solution $\{\bar{x}_1,\bar{x}_2,\bar{y},\bar{s}\}$ has been obtained for $\overline{\text{FCNP}}$ by the Simplex method of Appendix A.2.2 and the construction of Theorem 3.1.1.4. Let $\pi(n)$ be the optimal Simplex multipliers obtained with this optimal solution. Finally, assume that Algorithm 4.2.2.6 has been executed so that the functions $\eta(n)$ and $\lambda(i,k)$ are available. Then the elements of the problems BEOP(i) and BEOP(i,j) can be generated by the following procedures.

4.4.1.1 Objective Function. The objective function (4-14) is derived directly from the updated cost row of the optimal tableau for RNP. Calculating these entries according to the well-known formula,

$$d_{ij} = |c_{ij} + \pi(k_1) - \pi(k_2)|,$$

where the arc corresponding to x_{ij} runs from node k_1 to node k_2. The coefficients c_{sj} of the Δs_j are given in the original statement of the problem and thus require no calculation.

4.4.1.2 Right-Hand-Sides. The right-hand-sides of the constraints (4-17) are derived directly from the optimal $\overline{\text{FCNP}}$ flows $\bar{y}$. For the *ith* row, the values are $\bar{y}_i$ and $\bar{y}_i - u_{1i}$.

4.4.1.3 Upper Bounds on Perturbation Variables. In each case, the upper bounds on perturbation variables (4-18) are obtained as the difference of the upper and lower bounds on the non-basic variable corresponding to the perturbation variable.

4.4.1.4 Constraint Rows Up to Row k^B. For a row n of (4-17) corresponding to a component of x_1^B, i.e. numbered less than or equal to k^B, the main coefficients of the left-hand side are derived by multiplication of the original columns and row n of the reduced inverse as follows:

$e_{ij}(n) = \lambda(n,\eta(k_2)) - \lambda(n,\eta(k_1))$ if the arc corresponding to x_{ij} runs from node k_1 to node k_2 and x_{ij} is nonbasic lower-bounded in the optimal solution to $\overline{\text{FCNP}}$.

$e_{ij}(n) = \lambda\left(n,\eta(k_1)\right) - \lambda\left(n,\eta(k_2)\right)$ if the arc corresponding to x_{ij} runs from node k_1 to node k_2 and x_{ij} is nonbasic upper-bounded in the optimal solution to $\overline{FCNP}$.

The coefficient of Δs_n is -1.

4.4.1.5 Constraint Rows Below Row k^B. For a row n of (4-17) corresponding to a component of x_1^N, i.e. numbered above k^B, coefficients of the left-hand side are all zero except for a -1 on Δs_n, and either a -1 or a +1 on Δx_{1n} according to whether x_{1n} is nonbasic lower-bounded or nonbasic upper-bounded in the optimal solution to $\overline{FCNP}$.

4.4.2 Solution of One-Row Problems

Recall from the discussion of Section 3.5 that the one-row problems BEOP(i) and BGP(i) have identical optimal solutions. Thus, Theorem 3.3.3.1 provides the outline for a procedure to solve any BEOP(i) of FCNP, i.e. BEOP(i) can be solved by constructing two linear knapsack problems and solving them by the well-known minimum ratio procedure.

However, the minimum ratio method is simplified in the network case by the fact (implied by Theorem 4.3.2.1) that every coefficient of row i is +1, 0 or -1. The down and up penalties $\rho^D(i)$ and $\rho^U(i)$ can thus be calculated by ranking the coefficients of the objective function (4-14) in ascending order, and proceeding sequentially up this list. The cost of each perturbation variable with a +1 coefficient in row i which is encountered in this search is evaluated at the minimum of its upper bound and the amount of $\bar{y}_i$ not yet fulfilled, and the result is added into a total which becomes $\rho^D(i)$. Similarly, the cost of each perturbation variable with a -1 coefficient in row i which is encountered in this search is evaluated at the minimum of its upper bound and the amount of $(u_{1i}-\bar{y}_i)$ not yet fulfilled, and the result is added to a total which becomes $\rho^U(i)$. The optimal solution value for BEOP(i) is then given by

$$\nu\big(\mathrm{BEOP}(i)\big) = \nu(\overline{\mathrm{FCNP}}) + \min\{\rho^D(i),\rho^U(i)\}.$$

4.4.3 Solution of Two-Row Problems

For the two-row problems BEOP(i,j) of FCNP, four two-row linear programs must be solved to calculate $\nu\big(\mathrm{BEOP}(i,j)\big)$. Each of the four corresponds to a different combination of the two possible right-hand-sides of each row.

Unfortunately, it does not appear that a simple list search scheme like the one presented in the previous section can be devised to solve these two-row linear problems.[1] Thus some iterative procedure like the Simplex method would probably be required to solve the linear programs associated with BEOP(i,j).

However, a great simplification in the structure of such two-row linear programs is provided by Corollary 4.3.2.2. Since columns of any BEOP(i,j) with two non-zero entries must be drawn from either the set of (4-19) or that of (4-20), any of the linear programs for BEOP(i,j) can be converted into a three-node minimum cost flow problem by multiplying the first row by +1 or -1, creating a third row by multiplying -1 times the sum of the second row and the revised first row, and adding

---

[1]For the unbounded case of EOP(i,j), however, a search method is presented in Appendix A.2.5.

arcs corresponding to the revised right-hand-sides of the linear program.

Consider for example the BEOP(1,2) constraints

$$\Delta x_{21} - \Delta x_{22} + \Delta x_{23} \qquad - \Delta s_1 \qquad = 4 \text{ or } -5$$

$$\Delta x_{21} - \Delta x_{22} \qquad + \Delta x_{24} \qquad - \Delta s_2 = 7 \text{ or } -11$$

$$5 \geq \Delta x_{21} \geq 0,\ 7 \geq \Delta x_{22} \geq 0,\ 12 \geq \Delta x_{23} \geq 0$$

$$4 \geq \Delta x_{24} \geq 0,\ 9 \geq \Delta s_1 \geq 0,\ 18 \geq \Delta s_2 \geq 0.$$

For the case of the linear program with right-hand-sides 4 and 7, multiplication of the first row by -1, calculation of a third row as the negative sum of the two existing rows, and addition of variables z_1 and z_2 to carry the right-hand-sides, produces the constraint set:

$$-\Delta x_{21} + \Delta x_{22} - \Delta x_{23} \qquad + \Delta s_1 \qquad + z_1 \qquad = 0$$

$$+\Delta x_{21} - \Delta x_{22} \qquad + \Delta x_{24} \qquad - \Delta s_2 \qquad - z_2 = 0$$

$$+ \Delta x_{23} - \Delta x_{24} - \Delta s_1 + \Delta s_2 - z_1 + z_2 = 0$$

$$5 \geq \Delta x_{21} \geq 0, \quad 7 \geq \Delta x_{22} \geq 0, \quad 12 \geq \Delta x_{23} \geq 0, \quad 4 \geq \Delta x_{24} \geq 0$$

$$9 \geq \Delta s_1 \geq 0, \quad 18 \geq \Delta s_2 \geq 0, \quad 4 \geq z_1 \geq 4, \quad 7 \geq z_2 \geq 7.$$

Since every column of this constraint set has exactly one +1 and one -1 this linear program can now be interpreted as the network problem depicted in Figure 8. A labeling procedure like the one described in

Appendix A.2.2 could then be used to obtain an optimal solution.

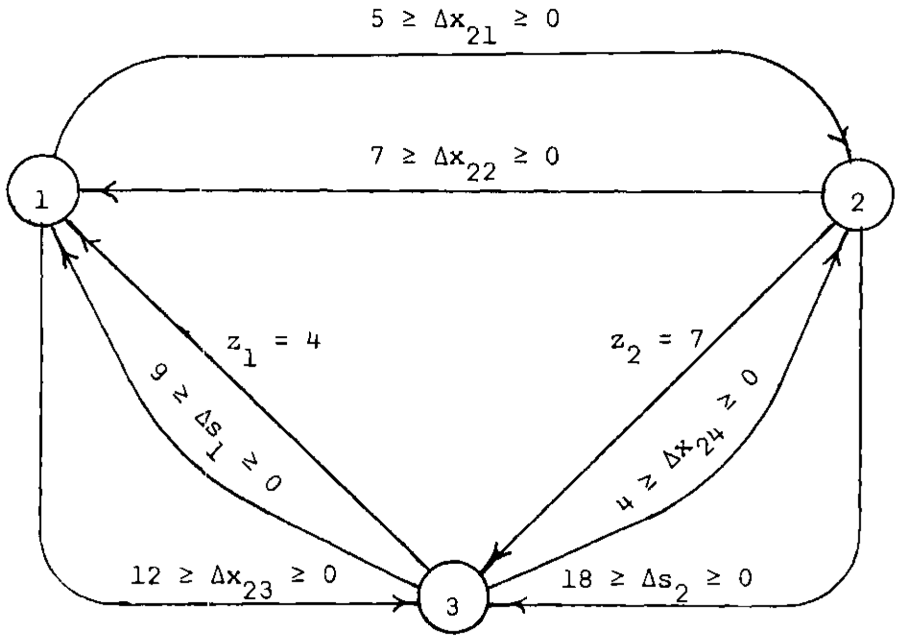

Figure 8. Network Problem Corresponding to One Linear Program of an Example BEOP(i,j)

CHAPTER V

LAGRANGEAN STRENGTHENING OF PENALTY PROBLEMS

The equivalent problem EFCP of Section 3.2.1 can be restated

$$\min \quad d_1^T \Delta x_1^N + d_2^T \Delta x_2^N + c_s^T \Delta s + \nu(\overline{FCP}) \tag{5-1}$$

$$\text{s.t.} \quad F_1 \Delta x_1^N + F_2 \Delta x_2^N + F_s \Delta s \geq 1 \tag{5-2}$$

$$(\text{EFCP}) \qquad \begin{pmatrix} u_1^B \\ u_2^B \\ u_1 \end{pmatrix} \geq \begin{pmatrix} \bar{x}_1^B \\ \bar{x}_2^B \\ \bar{y} \end{pmatrix} - H_1 \Delta x_1^N - H_2 \Delta x_2^N - H_s \Delta s \geq \begin{pmatrix} 0 \\ \ell_2^B \\ 0 \end{pmatrix} \tag{5-3}$$

$$\Delta x_1^N \geq 0, \quad \Delta x_2^N \geq 0, \quad \Delta s \geq 0 \tag{5-4}$$

$$u_1^N \geq \Delta x_1^N, \quad o_2^N \geq \Delta x_2^N, \quad u_1 \geq \Delta s \tag{5-5}$$

$$[H_1 \Delta x_1^N + H_2 \Delta x_2^N + H_s \Delta s]_{m+n_1}^{m+1} \equiv \bar{y} \bmod u_1, \tag{5-6}$$

where the matrices H_1, H_2 and H_s are derived from an optimal tableau for the reduced problem RP by

$$H_1 = \begin{pmatrix} \bar{A}_1^L & -\bar{A}_1^U \\ \hdashline [\bar{A}_1^L & -\bar{A}_1^U]^{1}_{k}{}^{B} \\ -I & 0 \\ 0 & I \end{pmatrix}, \quad H_2 = \begin{pmatrix} \bar{A}_2^L & -\bar{A}_2^U \\ \hdashline [\bar{A}_2^L & -\bar{A}_2^U]^{1}_{k}{}^{B} \\ & \\ \multicolumn{2}{c}{0} \end{pmatrix}, \quad H_s = \begin{pmatrix} 0 \\ \hdashline -I \end{pmatrix},$$

d_1, d_2, Δx_1^N, Δx_2^N, Δs and o_2^N are defined as in Chapter III, and the matrices F_1, F_2 and F_s define a set of redundant constraints (5-2), each row of which is a valid inequality[1] for the problem defined by (5-1), (5-4), (5-5) and (5-6). In the strongest one or two-row problems so far proposed, (5-1) and the constraints (5-4) and (5-5) were combined with rows $m + i$ and $m + j$ of (5-3), and rows i and j of (5-6) to produce BEOP(i,j). Section 3.5 demonstrated that $\nu\big(\text{BEOP}(i,j)\big)$ provides a generally stronger bound on $\nu(\text{FCP})$ than any of the other one and two-row group-related penalty problems defined in Chapter III.

In this chapter an extension of BEOP(i,j) providing partial consideration of the remainder of the constraints (5-3) and (5-6) will be developed by using the methods of Lagrangean relaxation presented in Section 2.1.2. These remaining constraints, which appear computationally difficult to include explicitly in penalty problems, will thus be dealt with implicitly by including them in the objective function of the penalty problems.

Specifically, let p, q and r be non-negative Lagrange multiplier

---

[1]See Section 3.4 for a definition of valid inequalities.

vectors of appropriate dimension corresponding to the constraints of the right inequality of (5-3), the left inequality of (5-3), and (5-2), respectively. Then the penalty problems of interest will be one and two-row relaxations of the *Lagrangean either-or problem*

(LEOP(FCP))

$$\min \quad d_1^T \Delta x_1^N + d_2^T \Delta x_2^N + c_s^T \Delta s + v(\overline{FCP}) \tag{5-7}$$

$$+ p^T \left(\begin{pmatrix} 0 \\ \ell_2^B \\ 0 \end{pmatrix} - \begin{pmatrix} \bar{x}_1^B \\ \bar{x}_2^B \\ \bar{y} \end{pmatrix} + H_1 \Delta x_1^N + H_2 \Delta x_2^N + H_s \Delta s \right)$$

$$- q^T \left(\begin{pmatrix} u_1^B \\ u_2^B \\ u_1 \end{pmatrix} - \begin{pmatrix} \bar{x}_1^B \\ \bar{x}_2^B \\ \bar{y} \end{pmatrix} + H_1 \Delta x_1^N + H_2 \Delta x_2^N + H_s \Delta s \right)$$

$$+ r^T \left(1 - F_1 \Delta x_1^N - F_2 \Delta x_2^N - F_s \Delta s \right)$$

$$\text{s.t.} \quad [H_1 \Delta x_1^N + H_2 \Delta x_2^N + H_s \Delta s]_i = \bar{y}_i \text{ or } \bar{y}_i - u_{1i}, \tag{5-8}$$

$$i = m+1, \ldots, m+n_1$$

$$u_1^N \geq \Delta x_1^N \geq 0, \quad o_2^N \geq \Delta x_2^N \geq 0, \quad u_1 \geq \Delta s \geq 0. \tag{5-9}$$

Note that both the bound constraints (5-3) and the valid inequalities (5-2) have been included in the objective function (5-7). If all rows of (5-8) were always enforced, there would be no advantage in including the redundant constraints (5-2). However, when one or two-row relaxations of LEOP(FCP) are solved, having the valid inequalities in the objective function is equivalent to partial enforcement of the relaxed rows of (5-8).

Note also that the bounded either-or problems can be thought of as special cases of Lagrangean either-or problems where all Lagrange multipliers are set to zero. Thus, since it is only necessary that the multipliers be non-negative for $\nu\left(\text{LEOP(FCP)}\right)$ to be a lower bound on ν(FCP), a better choice of the multipliers might be possible.

5.1 Alternative Multiplier Calculations

While a proper choice of Lagrange multipliers might make relaxations of LEOP(FCP) attractive penalty problems, the development of good multipliers is not straightforward. Some alternative calculation approaches are analyzed in the following discussion.

5.1.1 Optimal Multipliers

For a given constraint set (5-2), the optimal choice for p, q and r are $\bar{p}$, $\bar{q}$ and $\bar{r}$ where

$$\nu\left(\text{LEOP(FCP)}: p=\bar{p},q=\bar{q},r=\bar{r}\right) = \max_{\hat{p},\hat{q},\hat{r}\geq 0} \{\nu\left(\text{LEOP(FCP)}: p=\hat{p},q=\hat{q},r=\hat{r}\right)\}. \quad (5\text{-}10)$$

Such multipliers yield the maximum bound on ν(FCP) which can be produced from LEOP(FCP). Moreover, if (5-2) is a sufficiently exact representation of the convex hull of solutions to (5-6), the next theorem demonstrates that this bound is stronger than one obtained from $\nu(\text{BEOP(FCP)})$.

<u>5.1.1.1 Theorem.</u> Let Γ be a subset of the integers $m + 1$ through $m + n_1$ and the problems BEOP(Γ) and LEOP(Γ) be the versions of BEOP(FCP) and LEOP(FCP) where row j of (5-8) is enforced if $j \in \Gamma$. Also let F_1, F_2 and F_s be such that the constraints (5-2) and (5-9) define the convex hull of feasible solutions to BEOP(Γ) or LEOP(Γ). Then, if $\bar{p}$, $\bar{q}$ and $\bar{r}$ are as in (5-10), $\nu(\text{BEOP}(\Gamma)) \le \nu(\text{LEOP}(\Gamma\text{: } p=\bar{p}, q=\bar{q}, r=\bar{r})$, and if every optimal solution to BEOP(Γ) violates (5-3),

$$\nu(\text{BEOP}(\Gamma)) < \nu(\text{LEOP}(\Gamma\text{: } p=\bar{p}, q=\bar{q}, r=\bar{r})).$$

Proof.

$$\nu(\text{BEOP}(\Gamma)) = \nu\Big(\min \;\; d_1^T \Delta x_1^N + d_2^T \Delta x_2^N + c_s^T \Delta s \quad \text{s.t. } \Delta x_1^N, \Delta x_2^N, \Delta s \text{ satisfy (5-2) and (5-9)}\Big)$$

because the minimum of a linear objective function over a bounded set will occur at an extreme point of the convex hull of that set. Thus obviously

$$\nu(\text{BEOP}(\Gamma)) \le \nu\Big(\min \;\; d_1^T \Delta x_1^N + d_2^T \Delta x_2^N + c_s \Delta s \quad \text{s.t. } \Delta x_1^N, \Delta x_2^N, \Delta s \text{ satisfy (5-2), (5-3) and (5-9)}\Big), \tag{5-11}$$

and the inequality will be strict if every optimal solution to BEOP(Γ)

violates (5-3). Now, the problem on the right in (5-11) is a linear program and thus has value of an optimal solution equal to that of its dual,

$$\max_{\{p,q,r\geq 0\}} \left\{ \min_{\substack{\{\Delta x_1^N, \Delta x_2^N, \Delta s \\ \text{satisfying} \\ (5\text{-}4) \text{ and } (5\text{-}5)\}}} \begin{bmatrix} (5\text{-}7) \text{ evaluated at} \\ p,q,r,\Delta x_1^N, \Delta x_2^N, \Delta s \end{bmatrix} \right\}$$

which is exactly LEOP(Φ: $p=\bar{p},q=\bar{q},r=\bar{r}$). Clearly $\nu\big(\text{LEOP}(\Phi\text{: } p=\bar{p},q=\bar{q},r=\bar{r})\big) \leq \nu\big(\text{LEOP}(\Gamma\text{: } p=\bar{p},q=\bar{q},r=\bar{r})\big)$. Thus,

$$\nu\big(\text{BEOP}(\Gamma)\big) \leq \nu\big(\text{LEOP}(\Phi : p=\bar{p},q=\bar{q},r=\bar{r})\big) \leq \nu\big(\text{LEOP}(\Gamma\text{: } p=\bar{p},q=\bar{q},r=\bar{r})\big)$$

and the first inequality is strict when every optimal solution to BEOP(Γ) violates (5-3).

Q.E.D.

In [25], Shapiro and Fisher proposed a method for obtaining similar optimal Lagrange multipliers for the case of all-integer programs. As applied to FCP their method would deal with the Lagrangean problem

$$\min \quad d_1^T \Delta x_1^N + d_2^T \Delta x_2^N + c_s^T \Delta s + \nu(\overline{FCP}) \tag{5-12}$$

$$+ p^T \left[\begin{pmatrix} 0 \\ \ell_2^B \\ 0 \end{pmatrix} - \begin{pmatrix} \bar{x}_1^B \\ \bar{x}_2^B \\ \bar{y} \end{pmatrix} + H_1 \Delta x_1^N + H_2 \Delta x_2^N + H_s \Delta s \right]$$

$$- q^T \left[\begin{pmatrix} u_1^B \\ u_2^B \\ u_1 \end{pmatrix} - \begin{pmatrix} \bar{x}_1^B \\ \bar{x}_2^B \\ \bar{y} \end{pmatrix} + H_1 \Delta x_1^N + H_2 \Delta x_2^N + H_s \Delta s \right]$$

$$- t^T \begin{pmatrix} u_1^N - \Delta x_1^N \\ o_2^N - \Delta x_2^N \\ u_1 - \Delta s \end{pmatrix}$$

$$\text{s.t.} \quad [H_1 \Delta x_1^N + H_2 \Delta x_2^N + H_s \Delta s]_{m+n_1}^{m+1} \equiv \bar{y} \bmod u_1 \tag{5-13}$$

$$\Delta x_1^N \geq 0, \quad \Delta x_2^N \geq 0, \quad \Delta s \geq 0. \tag{5-14}$$

The explicit constraints are thus those of GP(FCP), with all other constraints being moved to the objective function.

The method these researchers used to find p, q and t which maximized the value of an optimal solution to the above problem is a variant of Dantzig-Wolfe decomposition [18]. Sub-problems consisting of minimizing a linear objective function over the constraints (5-13) and (5-14) are solved repeatedly in order to obtain optimal p, q and t from a master problem.

For the cases considered by Fisher and Shapiro, such a calculation procedure may be feasible. In the case of FCP, however, solution of GP(FCP) even once would be an extensive process. Thus it is not computationally feasible to consider repeated solution of GP(FCP) as a subproblem in a procedure for finding optimal Lagrange multipliers.

5.1.2 Continuous Program Multipliers

A much simpler method proposed by Geoffrion [31] for certain integer programming problems obtains Lagrange multipliers from the dual solution to the continuous analogs of integer programs. In the context of LEOP(FCP) this method would relax (5-2) and (5-6) in EFCP and solve the resulting $\overline{\text{EFCP}}$. Dual multipliers for the constraints of (5-3) then provide values for p and q, and r is fixed at 0.

Recall, however, that an optimal solution to $\overline{\text{EFCP}}$ is provided by $\Delta x_1^N, \Delta x_2^N$, $\Delta s = 0$ because EFCP is derived from an optimal tableau for $\overline{\text{FCP}}$. Moreover, the constraints (5-3) correspond to bounds on the basic variables in $\overline{\text{FCP}}$. Thus, if the solution to $\overline{\text{FCP}}$ was nondegenerate, these constraints cannot be binding in the all-zero solution to $\overline{\text{EFCP}}$, and the dual multipliers for the constraints will be 0. Therefore, at least in the nondegenerate case, if the simple method of obtaining Lagrange multipliers from the dual of $\overline{\text{EFCP}}$ is attempted, all multipliers will be zero, and LEOP(FCP) will reduce to BEOP(FCP).

5.1.3 Obtaining Multipliers with Gomory Cuts

The alternative proposed in this dissertation falls between the optimal case of Section 5.1.1 and the trivial case of Section 5.1.2. Specifically, it is proposed that the set of Gomory cuts obtained from the optimal Simplex tableau for $\overline{FCP}$ be used as the constraint set (5-2). Values for the Lagrange multipliers can then be derived from the dual solution to the linear program

$$\min \quad d_1^T x_1^N + d_2^T \Delta x_2^N + c_s^T \Delta s + \nu(\overline{FCP}) \tag{5-15}$$

$$\text{s.t.} \quad G_1 \Delta x_1^N + G_2 \Delta x_2^N + G_s \Delta s \geq 1 \tag{5-16}$$

$$(\widetilde{FCP}) \qquad \begin{pmatrix} u_1^B \\ u_2^B \\ u_1 \end{pmatrix} \geq \begin{pmatrix} \bar{x}_1^B \\ \bar{x}_2^B \\ \bar{y} \end{pmatrix} - H_1 \Delta x_1^N - H_2 \Delta x_2^N - H_s \Delta s \geq \begin{pmatrix} 0 \\ \ell_2^B \\ 0 \end{pmatrix} \tag{5-17}$$

$$u_1^N \geq \Delta x_1^N \geq 0, \quad o_2^N \geq \Delta x_2^N \geq 0, \quad u_1 \geq \Delta s \geq 0, \tag{5-18}$$

where the constraints (5-16) are the Gomory cuts. In particular, optimal dual multipliers for the constraints of the right-hand inequality of (5-17) provide a value for p, dual multipliers for the left-hand inequality of (5-17) provide q, and those of the constraints (5-16) provide r.

This Gomory cut approach has a number of advantages over the other two schemes presented. First, in comparison to repeated solving of GP(FCP) required by Fisher and Shapiro's method, imposition of Gomory cuts on a linear program is relatively simple computationally. In addition, no Gomory cut is satisfied by the solution

$$\Delta x_1^N,\ \Delta x_2^N,\ \Delta s = 0.$$

Thus, it is likely that values of p, q and r obtained from the dual solution to FCP will not all be zero as can occur with the method of Section 5.1.2.

Unfortunately, it does not appear possible to prove that Lagrange multipliers $\tilde{p}$, $\tilde{q}$ and $\tilde{r}$ obtained from $\widetilde{\widetilde{FCP}}$ will always yield

$$\nu\big(\text{BEOP}(\Gamma)\big) \le \nu\big(\text{LEOP}(\Gamma\text{: } p=\tilde{p}, q=\tilde{q}, r=\tilde{r})\big). \tag{5-19}$$

However, Theorem 5.1.1.1 demonstrated that if the constraints (5-2) and (5-9) define the convex hull of solutions to BEOP(Γ), then (5-19) holds. Even with the bounds (5-9), the Gomory cuts certainly do not fully define the convex hull of solutions to BEOP(Γ). However, the results of Section 3.4 demonstrate that the Gomory cut derived from row i, i.e. GC(i), always provides a face of convex hull of solutions to GP(Γ) if $i \in \Gamma$, and under some conditions also provides a face for EOP(Γ). Moreover, it may be true that GC(i) is violated by some solutions to BEOP(Γ) when $i \notin \Gamma$. Thus, in some aspects the full set of Gomory cuts may be stronger than the convex hull of solutions to BEOP(Γ) because some BEOP(Γ) solutions may not satisfy (5-16).

5.1.3.1 Simplifications. If this Gomory cut method is used for constructing penalty problems LEOP(i,j) in a branch-and-bound procedure like the one summarized at the beginning of Chapter III, several simplifications can be noted. First, it is obviously true that

$$\nu(\overline{FCP}) \le \nu(\widetilde{\overline{FCP}}) \le \nu(FCP).$$

Thus the problem $\widetilde{FCP}$ itself is a penalty problem, and if the value of an optimal solution to $\widetilde{FCP}$ exceeds that of some known feasible solution to FCP, the current candidate problem can be fathomed.

Next, note that it is not really p, q and r that are required in LEOP(i,j), but the objective function constant and coefficients of Δx_1^N, Δx_2^N, Δs. If these coefficients are denoted by $\tilde{z}$, $\tilde{d}_1$, $\tilde{d}_2$ and $\tilde{c}_s$, respectively, then they are expressed by

$$\tilde{z} = \nu(\overline{FCP}) + \tilde{p}^T \begin{pmatrix} 0 \\ \ell_2^B \\ 0 \end{pmatrix} - \tilde{q}^T \begin{pmatrix} u_1^B \\ u_2^B \\ u_1 \end{pmatrix} - (\tilde{p}-\tilde{q})^T \begin{pmatrix} \bar{x}_1^B \\ \bar{x}_2^B \\ \bar{y} \end{pmatrix} + \tilde{r}^T 1$$

$$\tilde{d}_1 = d_1 + (\tilde{p}-\tilde{q})^T H_1 - \tilde{r}^T G_1$$

$$\tilde{d}_2 = d_2 + (\tilde{p}-\tilde{q})^T H_2 - \tilde{r}^T G_2$$

$$\tilde{c}_s = c_s + (\tilde{p}-\tilde{q})^T H_s - \tilde{r}^T G_s$$

where $\tilde{p}$, $\tilde{q}$ and $\tilde{r}$ are the optimal dual solution to $\widetilde{FCP}$. Comparison of these values to the formulation of $\widetilde{FCP}$ will demonstrate that the required objective function values are exactly the updated cost row of an optimal Simplex tableau for $\widetilde{FCP}$. Thus, if a Simplex procedure is used to solve $\widetilde{FCP}$, $\tilde{p}$, $\tilde{q}$ and $\tilde{r}$ need not be identified explicitly. The objective function for LEOP can be obtained directly from the optimal tableau of $\widetilde{FCP}$.

Finally, recall that $\nu\big(\text{LEOP}(\Gamma)\big)$ provides a lower bound on $\nu(\text{FCP})$ whenever p, q and r are non-negative. Thus, if a dual Simplex procedure is used to impose the Gomory cuts in $\widetilde{FCP}$, it is not necessary that the process be carried to optimality. The values of p, q and r at any stage of the dual process will be non-negative, and so the value of the dual solution at each Simplex iteration will provide a lower bound on $\nu(\text{FCP})$. If this bound reaches the value of a known feasible solution to FCP, the process can be terminated.

<u>5.1.3.2 Complication</u>. Unfortunately, one complication of the computational procedures developed in Chapter III will also result from the use of $\widetilde{FCP}$. It does not appear possible to devise a general procedure for constructing a solution to $\widetilde{FCP}$ from a solution for the reduced problem RP. Thus, if the computational simplicity of RP is to be exploited, it would be necessary to first solve RP, then construct the full Simplex solution for $\overline{FCP}$, and lastly impose the Gomory cuts on this full version of $\overline{FCP}$ in order to solve $\widetilde{FCP}$.

5.2 Imposing Gomory Cuts by Decomposition

An additional computational complication arises when the linear program RP or $\overline{FCP}$ has a special structure which can be exploited by a special version of the Simplex method. For such cases, $\widetilde{FCP}$ may not possess this same special structure, and thus computational efficiency would be lost if $\widetilde{FCP}$ were used to calculate Lagrange multipliers.

An example of this case is FCNP. In Chapter IV it was demonstrated that RNP and $\overline{FCNP}$ are suitable to solution by a very efficient graph-theoretic version of the primal Simplex method. Such a solution procedure is applicable because every basis of a minimum cost flow network problem can be shown to contain no cycles. A similar property does not hold for the problem $\widetilde{FCNP}$. If the Gomory cuts were directly imposed on the network, the special basis structure would be lost.

This quandary suggests use of a procedure which indirectly imposes the Gomory cuts on problems like $\overline{FCNP}$. In particular, a decomposition method is required which solves the easier problems RP or $\overline{FCP}$ as a single sub-problem to a master problem constrained by the Gomory cuts.

An important requirement on such a decomposition method would be that it maintain dual feasibility for $\widetilde{FCP}$, i.e. that intermediate values of p, q and r provide valid Lagrange multipliers for LEOP(Γ). Such a property makes it possible to take advantage of the simplifications presented in Section 5.1.3.1. In particular, it would permit stopping the procedure before an optimal solution to $\widetilde{FCP}$ had been reached without sacrificing the bounding value of LEOP(Γ).

One published decomposition method which has this property is Balas' Infeasibility Pricing method [1]. The remainder of this section describes the application of this method to the FCP case.

5.2.1 An Infeasibility Pricing Method

The general concept of the Infeasibility Pricing method, as proposed by Balas [1] and specialized to the network case by Bazaraa [8], is to treat sub-problems as Lagrangean relaxations of the full linear program of interest. Successively better and better Lagrange multipliers for the linking constraints are produced until the optimal solutions of all sub-problems corresponding to some multipliers also satisfy all linking constraints.

At each iteration the multipliers are improved in three steps. First, a master problem is solved which seeks to obtain a perturbation of the current optimal solutions for the sub-problems which will satisfy the linking constraints. If one is found, the procedure terminates. Otherwise, the dual solution of the master problem provides a direction of improvement on the current Lagrange multipliers for the linking constraints.

However, it may happen that any movement in this direction will cause Simplex optimality (i.e. dual feasibility) requirements to be lost in the sub-problems. Thus, a second step is required to adjust the optimal bases of the sub-problems so that infinitesimally small movements in the desired direction will not disturb Simplex optimality.

Finally, the third step is to move as far as possible in the desired direction. A limit is provided by the maximum movement which retains Simplex optimality for the sub-problems.

In the $\widetilde{FCP}$ case this approach consists of finding optimal Lagrange multipliers for placing the Gomory cuts in the objective function of $\overline{FCP}$. This modified version of $\overline{FCP}$ is the single sub-problem. At each iteration of the algorithm, a master problem is solved in an attempt to find a perturbation of the current optimal solution of the sub-problem which will satisfy the Gomory cuts. If none is found, the objective function of the sub-problem is revised and the sub-problem is reoptimized. The process continues until an objective function is obtained which will yield an optimal solution to the sub-problem which satisfies the Gomory cuts.

For more complete development of the general procedure see [1] and [8]. In particular, a proof of the convergence of the Infeasibility Pricing method is given in [1].

5.2.1.1 Notation. Before proceeding to a detailed statement of the implementation of the Infeasibility Pricing method for solving $\widetilde{FCP}$, it will be useful to define certain deviations from the standard notation of this dissertation. In particular, *the B and N superscripts will be used in this discussion only to refer to the segments of various vectors and matrices corresponding to parts of* Δx_1^N*,* Δx_2^N *and* Δs *which were basic in the most recent solution of the sub-problem.* Thus, the superscripts B and N associated with the basis of $\overline{FCP}$ will be dropped, and, for example, Δx_1^B and Δx_1^N will refer to the parts of Δx_1^N in $\widetilde{FCP}$ which were respectively basic and nonbasic in the last sub-problem. Similarly, a new superscript 0 will be used to denote the parts of vectors and matrices corresponding to nonbasic variables in the most recent solution of the sub-problem which had adjusted costs equal to zero, e.g. Δs^0 is the part of Δs which was nonbasic in the most recent sub-problem and had adjusted costs of 0. Since the superscript B, N and 0 parts of Δs and Δx_1 will no longer be directly associated with x_1^B and x_1^N, it will also be necessary to denote the upper bounds on Δs and Δx_1 by distinct symbols o_s and o_1. Finally, the convention will be adopted that every solution value, tableau part, etc. taken from the most recent sub-problem will be marked with a tilda (˜).

5.2.1.2 Algorithm. With these notational conventions, an Infeasibility Pricing method for $\widetilde{\text{FCP}}$ can be defined as follows:

Step 0. From the optimal Simplex tableau of $\overline{\text{FCP}}$ construct the *sub-problem* consisting of the Lagrangean relaxation of FCP with Lagrange multipliers for the Gomory cuts equal to 0, i.e.

$$\min. \quad \tilde{d}_1^T \Delta x_1 + \tilde{d}_2^T \Delta x_2 + \tilde{c}_s^T \Delta s + \nu(\overline{\text{FCP}}) \tag{5-20}$$

(SP)

$$\text{s.t.} \quad \begin{pmatrix} u_1 \\ u_2 \\ o_s \end{pmatrix} \geq \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{y} \end{pmatrix} - H_1 \Delta x_1 - H_2 \Delta x_2 - H_s \Delta s \geq \begin{pmatrix} 0 \\ \ell_2 \\ 0 \end{pmatrix} \tag{5-21}$$

$$o_1 \geq \Delta x_1 \geq 0, \quad o_2 \geq \Delta x_2 \geq 0, \quad o_s \geq \Delta s \geq 0 \tag{5-22}$$

where $\tilde{d}_1 = d_1$, $\tilde{d}_2 = d_2$, $\tilde{c}_s = c_s$. No optimization is necessary since an optimal solution to this problem is $\Delta\tilde{x}_1 = 0$, $\Delta\tilde{x}_2 = 0$, $\Delta\tilde{s} = 0$.

Thus, proceed immediately to Step 1.

Step 1. From the final Simplex tableau of the most recent SP, construct the following *master problem.*

$$\min \quad 1^T f \tag{5-23}$$

$$\text{s.t.} \quad (G_1^B, G_2^B, G_s^B)\left[\begin{pmatrix}\Delta\tilde{x}_1^B\\ \Delta\tilde{x}_2^B\\ \Delta\tilde{s}^B\end{pmatrix} - \tilde{H}_1^0(\Delta x_1^0-\Delta\tilde{x}_1^0) - \tilde{H}_2^0(\Delta x_2^0-\Delta\tilde{x}_2^0) - \tilde{H}_s^0(\Delta s^0-\Delta\tilde{s}^0)\right]$$

$$(\text{MP}) \qquad + G_1^0(\Delta x_1^0-\Delta\tilde{x}_1^0) + G_2^0(\Delta x_2^0-\Delta\tilde{x}_2^0) + G_s^0(\Delta s^0-\Delta\tilde{s}^0) \tag{5-24}$$

$$+ G_1^U(\Delta\tilde{x}_1^U) \quad + G_2^U(\Delta\tilde{x}_2^U) \quad + G_s^U(\Delta\tilde{s}^U) + f \geq 1$$

$$\begin{pmatrix}o_1^B\\ o_2^B\\ o_s^B\end{pmatrix} \geq \begin{pmatrix}\Delta\tilde{x}_1^B\\ \Delta\tilde{x}_2^B\\ \Delta\tilde{s}^B\end{pmatrix} - \tilde{H}_1^0(\Delta x_1^0-\Delta\tilde{x}_1^0) - \tilde{H}_2^0(\Delta x_2^0-\Delta\tilde{x}_2^0) - \tilde{H}_s^0(\Delta s^0-\Delta\tilde{s}^0) \geq \begin{pmatrix}0\\ 0\\ 0\end{pmatrix} \tag{5-25}$$

$$o_s^0 \geq \Delta s^0 \geq 0, \quad o_1^0 \geq \Delta x_1^0 \geq 0, \quad o_2^0 \geq \Delta x_2^0 \geq 0, \quad f \geq 0. \tag{5-26}$$

Let the optimal dual multipliers for the constraints (5-24) have the value $\hat{r}$, and the optimal value of f be $\hat{f}$. Then go to Step 2.

Step 2. Introduce into the basis of SP all nonbasic variables corresponding to components of

$$\begin{bmatrix} \tilde{d}_1 - (h)\hat{r}^T G_1 \\ \tilde{d}_2 - (h)\hat{r}^T G_2 \\ \tilde{c}_s - (h)\hat{r}^T G_s \end{bmatrix}$$

which cannot satisfy Simplex optimality criteria in SP for any positive value of the scalar h, i.e.

$$\begin{aligned}
&\Delta x^0_{1j} \text{ such that } \Delta\tilde{x}^0_{1j} = 0 \quad \text{and } \hat{r}^T G_1 > 0 \\
&\Delta x^0_{1j} \text{ such that } \Delta\tilde{x}^0_{1j} = o^0_{1j} \text{ and } \hat{r}^T G_1 < 0 \\
&\Delta x^0_{2j} \text{ such that } \Delta\tilde{x}^0_{2j} = 0 \quad \text{and } \hat{r}^T G_2 > 0 \\
&\Delta x^0_{2j} \text{ such that } \Delta\tilde{x}^0_{2j} = o^0_{2j} \text{ and } \hat{r}^T G_2 < 0 \\
&\Delta s^0_j \text{ such that } \Delta\tilde{s}^0_j = 0 \quad \text{and } \hat{r}^T G_s > 0 \\
&\Delta s^0_j \text{ such that } \Delta\tilde{s}^0_j = o^0_{sj} \text{ and } \hat{r}^T G_s < 0.
\end{aligned}$$

Next, make any additional pivots necessary to restore optimality in SP, let the final values of the adjusted costs be d_1', d_2' and c_s', and let the value of the optimal solution to SP be $\nu'(SP)$. If $\nu'(SP)$ is greater than or equal to the value of a known feasible solution to FCP, stop; the current candidate problem can be fathomed. Otherwise, if $\hat{f} = 0$, an optimal solution for $\widetilde{FCP}$ has been obtained, and d_1', d_2', c_s' are the desired objective function coefficients for LEOP(Γ)'s. If neither of these stopping criteria is met, proceed to Step 3.

Step 3. Choose a scalar $\tilde{h}$ so that

$$\tilde{h} = \sup \left\{ h \geq 0 : \begin{pmatrix} d_1' - (h)\hat{r}^T G_1 \\ d_2' - (h)\hat{r}^T G_2 \\ c_s' - (h)\hat{r}^T G_s \end{pmatrix} \text{satisfies Simplex optimality in SP} \right\},$$

i.e. move as far as possible in the direction $\hat{r}$ without violating dual feasibility in SP. If $\tilde{h} = +\infty$, stop; $\widetilde{FCP}$ is infeasible, and the current candidate problem can be fathomed. Otherwise update the objective function for SP by

$$\begin{pmatrix} d_1 \\ d_2 \\ c_s \end{pmatrix} = \begin{pmatrix} d_1' - (\tilde{h})\hat{r}^T G_1 \\ d_2' - (\tilde{h})\hat{r}^T G_2 \\ c_s' - (\tilde{h})\hat{r}^T G_s \end{pmatrix},$$

and go to Step 1.

5.2.2 Simplifications in the Network Case

When the above Infeasibility Pricing algorithm is applied to the case of $\widetilde{FCNP}$, a number of simplifications result. The next several subsections detail some of these.

5.2.2.1 Solving Sub-Problems. By a direct change of variables back to the original ones of FCNP, it is clear that the sub-problem SP can be solved on the original $\overline{FCNP}$ network. The only difference will be changes in the objective function. At any iteration of the Infeasibility Pricing algorithm, the values of the $\widetilde{FCNP}$ variables Δx_1, Δx_2, and Δs can be obtained as the absolute difference between the current flow on arcs which were nonbasic in the optimal solution of $\overline{FCNP}$, and the optimal $\overline{FCNP}$ flows on the arcs.

5.2.2.2 Generating Gomory Cuts. Suppose that the algorithm of Section 4.2.2.6 was executed after an optimal solution to $\overline{FCNP}$ had been obtained. Then elements of the Gomory cuts (defined in (3-63) and (3-64)) can be generated for a variable in MP corresponding to an arc from node w_1 to node w_2 by using the reduced inverse entries $\lambda(i,z)$ as follows:

$$g_{kj}(i) = \begin{cases} \tilde{\lambda}/\bar{y}_i & \text{if } \tilde{\lambda} \geq 0 \text{ and } \bar{x}_{kj} \text{ is lower-bounded} \\ -\tilde{\lambda}/\bar{y}_i & \text{if } \tilde{\lambda} \leq 0 \text{ and } \bar{x}_{kj} \text{ is upper-bounded} \\ \tilde{\lambda}/(\bar{y}_i - u_{1i}) & \text{if } \tilde{\lambda} < 0 \text{ and } \bar{x}_{kj} \text{ is lower-bounded} \\ -\tilde{\lambda}/(\bar{y}_i - u_{1i}) & \text{if } \tilde{\lambda} > 0 \text{ and } \bar{x}_{kj} \text{ is upper-bounded} \end{cases}$$

where $\tilde{\lambda} = \lambda(i,\eta(w_2)) - \lambda(i,\eta(w_1))$.

$$g_{sj}(i) = \begin{cases} -1/(\bar{y}_i - u_{1i}) & \text{if } j=i \\ 0 & \text{otherwise.} \end{cases}$$

Thus the columns of MP which involve segments of the Gomory cut matrices G_1, G_2 and G_s could be rapidly constructed at the beginning of an MP execution, or regenerated as needed during the solution of MP by a revised Simplex algorithm.

5.2.2.3 Generating Updated Tableau Columns from SP. In a similar way, the elements of MP involving segments of the updated sub-problem matrices $\tilde{H}_1$, $\tilde{H}_2$ and $\tilde{H}_s$ can be generated rapidly from the network of $\overline{\text{FCNP}}$. Theorem 4.1.2.1 provides a method for constructing any column of the updated tableau corresponding to a nonbasic arc in a network problem by scanning the cycle formed in the basis forest by the given nonbasic arc. Only one slight change in this approach is required to generate the columns of $\tilde{H}_1$, $\tilde{H}_2$ or $\tilde{H}_s$ used in MP. If an arc was nonbasic in both the optimal solution to $\overline{\text{FCNP}}$ and that of the current SP, then it is only necessary to redefine the forward direction in Theorem 4.1.2.1 to be the direction of flow on the arc if the arc was lower-bounded *in the optimal solution to* $\overline{\textit{FCNP}}$, and the direction opposed to direction of flow on the arc if the arc was upper-bounded *in the* $\overline{\textit{FCNP}}$ *solution.* Columns of $\tilde{H}_1$, $\tilde{H}_2$ and $\tilde{H}_s$ for variables which were nonbasic in $\overline{\text{FCNP}}$ but basic in SP could also be generated easily, but they are not required for MP.

CHAPTER VI

GROUP-RELATED PENALTY ALGORITHMS

In the previous three chapters a number of methods were proposed for improving the efficiency of branch-and-bound solution procedures for fixed charge problems which use group-theory-related penalty problems in obtaining bounds and branching rules. A general outline of such an approach was presented at the beginning of Chapter III, and numerous schemes for constructing, augmenting and solving group-related penalty problems make up the remainder of Chapter III, and all of Chapters IV and V.

In this chapter, an integration of these discussions will be provided by the detailed statement of algorithms for FCP and FCNP which are based on the results of the previous analysis. For the convenience of the reader, notation terminology and step numbers will be kept consistent with the branch-and-bound method presented at the beginning of Chapter III and summarized in Figure 1.

6.1 Algorithm for General Fixed Charge Problems

In terms of the terminology of Chapters III and V, an algorithm for general fixed charge problems in the form of FCP is given below. Details of Steps 5 and 6 are also summarized in Figure 9.

Step 0. Place FCP in the candidate list, set $\beta(\text{FCP}) = 0$ and $\nu^* = +\infty$, and go to Step 1.

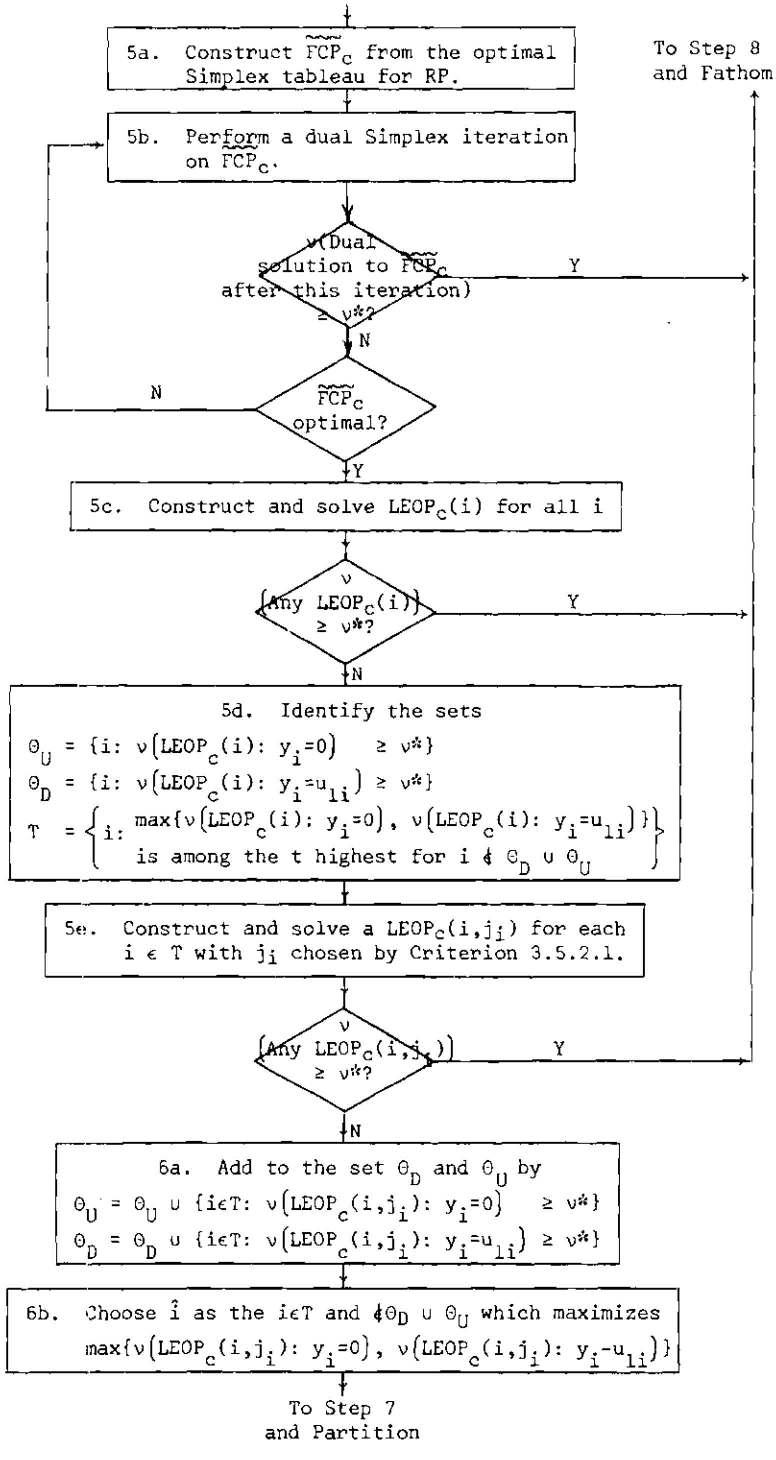

Figure 9. Flow Chart of Details in Steps 5 and 6 of Algorithm for FCP

Step 1. Choose as the current candidate, FCP_c, the element of the candidate list satisfying

$$\beta(FCP_c) = \min\{\beta(FCP_{c'}): FCP_{c'} \text{ in candidate list}\},$$

and proceed to Step 2.

Step 2. Solve the continuous relaxation of FCP_c, i.e. $\overline{FCP}_c$, by solving the corresponding reduced problem RP_c and constructing a solution to $\overline{FCP}_c$ according to the rules of Theorem 3.1.1.4. If $\nu(\overline{FCP}_c) \geq \nu^*$, proceed to Step 8 because no completion of FCP_c can produce a solution to FCP with value less than that of the incumbent. If $\nu(\overline{FCP}_c) < \nu^*$, proceed to Step 3.

Step 3. Create a feasible solution for FCP by rounding "up" the optimal solution to $\overline{FCP}_c$, i.e. by setting

$$s_j = \begin{cases} u_{1j} - \bar{x}_{ij} & \text{if } \bar{x}_{ij} > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$y = s + \bar{x}_1$$

$$x_1 = \bar{x}_1$$

$$x_2 = \bar{x}_2,$$

where $\bar{x}_1$ and $\bar{x}_2$ are the optimal values of x_1 and x_2 in the solution of $\overline{FCP}_c$. If the value of this rounded solution is less than ν^*, proceed to

Step 4. Otherwise, go to Step 5a.

Step 4. A new incumbent solution has been found. Save this incumbent as a possible optimal solution to FCP, and eliminate from the candidate list any problems with β value greater than or equal to the value of the new incumbent. If the new $\nu^* = -\infty$, stop; FCP is unbounded. Otherwise, proceed to Step 5a.

Step 5a. Construct the problem $\widetilde{FCP}_c$ associated with $\overline{FCP}_c$ from the optimal tableau for RP_c. The optimal tableau for $\overline{FCP}_c$ is obtained according to Lemma 3.1.1.2, and Gomory cuts are derived from this optimal tableau according to the definition in Section 3.4.2. Next, go to Step 5b.

Step 5b. Perform a dual Simplex pivot on $\widetilde{FCP}_c$, and calculate the value of the dual solution to $\widetilde{FCP}_c$ at the completion of this pivot. If this dual solution value is greater than or equal to ν^*, go to Step 8 because no completion of FCP_c can provide a better solution than the incumbent. Otherwise, if optimality of $\widetilde{FCP}_c$ has been attained, go to Step 5c, and if optimality has not been reached, repeat Step 5b.

Step 5c. Construct and solve all one-row Lagrangean either-or problems associated with $\overline{FCP}_c$, i.e. all $LEOP_c(i)$. The objective function for these problems is given by the adjusted cost row of the optimal tableau for $\widetilde{FCP}_c$, and the constraints can be constructed from the optimal tableau for RP_c according to (3-42) and (3-43) in Section 3.3.1. The problems are solved by the minimum ratio procedure of Theorem 3.3.3.1.

If the value of an optimal solution to any of these $LEOP_c(i)$ is greater than or equal to ν^*, go to Step 8 because no completion of FCP_c can have a better optimal solution than the incumbent. Otherwise, go to Step 5d.

Step 5d. Using the "down" and "up" penalties obtained in the solution of the $LEOP_c(i)$, i.e. the values $\nu(LEOP(i)\colon y_i=0)$ and $\nu(LEOP(i)\colon y_i=u_{1i})$, define the sets

$$\Theta_U = \{i\colon \nu(LEOP_c(i)\colon y_i=0) \geq \nu^*\}$$

$$\Theta_D = \{i\colon \nu(LEOP_c(i)\colon y_i=u_{1i}) \geq \nu^*\}$$

$$T = \left\{ i\colon \begin{array}{l} \max\{\nu(LEOP_c(i)\colon y_i=0),\ \nu(LEOP_c(i)\colon y_i=u_{1i})\} \\ \text{is among the } t \text{ highest values for } i \notin \Theta_D \cup \Theta_U \end{array} \right\}.$$

Then go to Step 5e.

Step 5e. Construct and solve a two-row Lagrangean either-or problem $LEOP_c(i,j_i)$ for each $i \in T$. The problems are constructed exactly as in Step 5c, with j_i being chosen according to the criterion of Section 3.5.2.1. If the value of an optimal solution to any of these penalty problems is greater than or equal to ν^*, go to Step 8 because no completion of FCP_c can provide a better solution than the incumbent. Otherwise, proceed to Step 6a.

Step 6a. Add to the sets Θ_D and Θ_U by

$$\Theta_U = \Theta_U \cup \{i \in T\colon \nu(LEOP_c(i,j_i)\colon y_i=0) \geq \nu^*\}$$

$$\Theta_D = \Theta_D \cup \{i \in T\colon \nu(LEOP_c(i,j_i)\colon y_i=u_{1i}) \geq \nu^*\}$$

Then go to Step 6b.

Step 6b. Choose the branching variable $y_{\hat{i}}$ so that $\hat{i}$ is the $i \in T$ and $\notin \Theta_D \cup \Theta_U$ which maximizes

$$\max\{\nu\big(LEOP_c(i,j_i)\colon y_i=0\big),\ \nu\big(LEOP_c(i,j_i)\colon y_i=u_{1i}\big)\}.$$

Then go to Step 7.

Step 7. Replace FCP_c in the candidate list by two more restricted problems. One is defined by FCP_c with the additional constraints that $y_{\hat{i}} = 0$, $y_j = 0$ for $j \in \Theta_D$, and $y_j = u_{1j}$ for $j \in \Theta_U$. The second problem is identical to the first except that $y_{\hat{i}}$ is restricted to equal $u_{1\hat{i}}$. β values for the two problems are $\nu\big(LEOP_c(\hat{i},j_{\hat{i}})\colon y_{\hat{i}}=0\big)$ and $\nu\big(LEOP_c(\hat{i},j_{\hat{i}})\colon y_{\hat{i}}=u_{1\hat{i}}\big)$, respectively. Next, go to Step 1.

Step 8. Fathom FCP_c, i.e. eliminate FCP_c from the candidate list because no completion of it can produce a feasible solution to FCP with value less than that of the incumbent solution ν^*. If the candidate list is now empty, stop. If an incumbent solution exists, it is an optimal solution for FCP, and otherwise FCP is infeasible. If the candidate list is not empty, proceed to Step 1.

6.2 Algorithm for Fixed Charge Network Problems

In terms of the terminology of Chapters III, IV and V, an algorithm for fixed charge network problems in the form of FCNP is given below. Details of Steps 5 and 6 are also summarized in Figure 10.

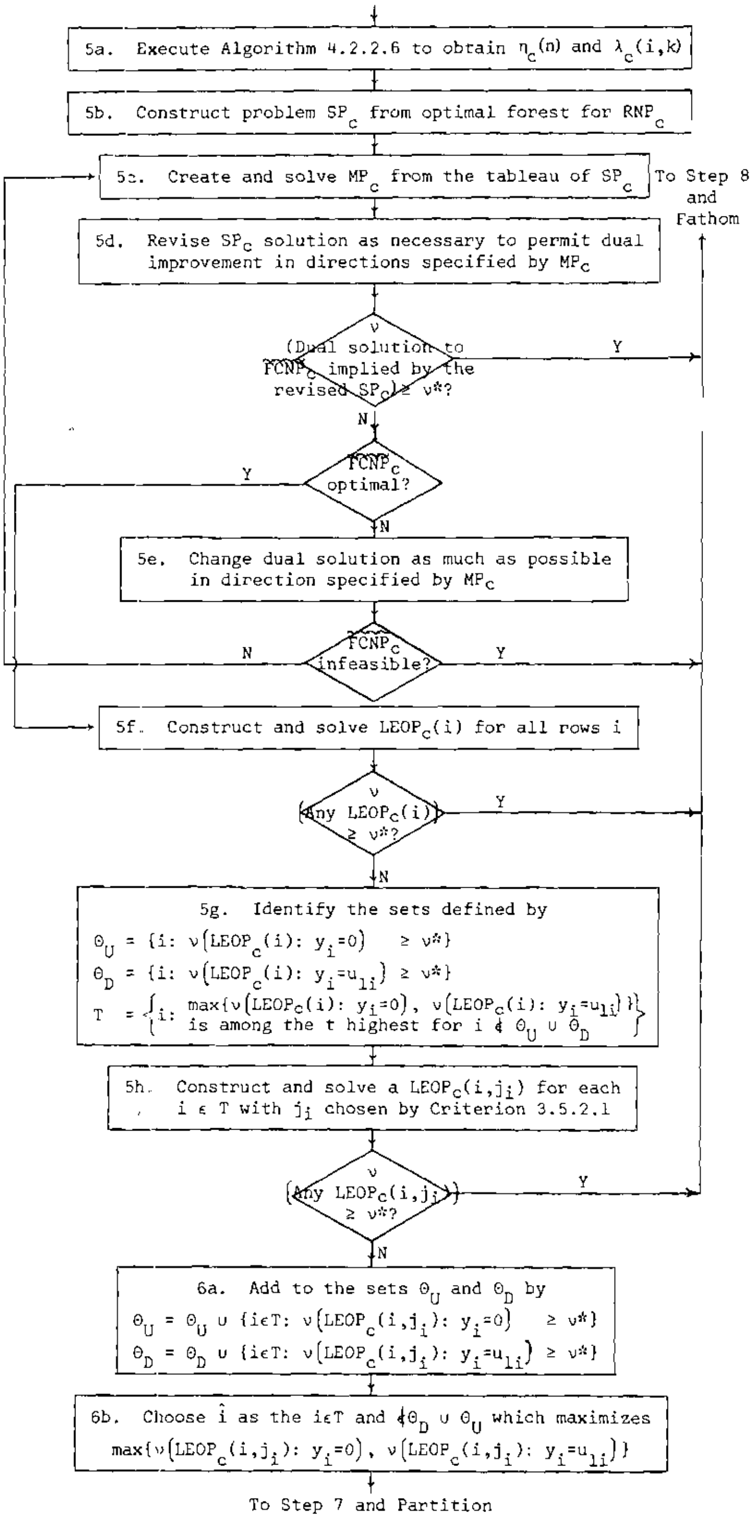

Figure 10. Flow Chart of Details in Steps 5 and 6 of Algorithm for FCNP

Step 0. Place FCNP in the candidate list, set $\beta(\text{FCNP}) = 0$ and $\nu^* = +\infty$, and go to Step 1.

Step 1. Choose as the current candidate, FCNP_c, the element of the candidate list satisfying

$$\beta(\text{FCNP}_c) = \min\{\beta(\text{FCNP}_{c'}): \text{FCNP}_{c'} \text{ in candidate list}\},$$

and proceed to Step 2.

Step 2. Solve the continuous relaxation of FCNP_c, i.e. $\overline{\text{FCNP}}_c$, by solving the corresponding reduced problem RNP_c with a graph theory-oriented algorithm and constructing a solution to $\overline{\text{FCNP}}_c$ according to the rules of Theorem 3.1.1.4. If $\nu(\overline{\text{FCNP}}_c) \geq \nu^*$, proceed to Step 8 because no completion of FCNP_c can produce a solution to FCNP with value less than that of the incumbent. If $\nu(\overline{\text{FCNP}}_c) < \nu^*$, proceed to Step 3.

Step 3. Create a feasible solution for FCNP by rounding "up" the optimal solution to $\overline{\text{FCNP}}_c$, i.e. setting

$$s_j = \begin{cases} u_{1j} - \bar{x}_{1j} & \text{if } \bar{x}_{1j} > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$y = s + \bar{x}_1$$

$$x_1 = \bar{x}_1$$

$$x_2 = \bar{x}_2,$$

where $\bar{x}_1$ and $\bar{x}_2$ are the optimal values of x_1 and x_2 in the solution of $\overline{FCNP}_c$. If the value of this rounded solution is less than ν^*, proceed to Step 4. Otherwise, go to Step 5a.

Step 4. A new incumbent solution has been found. Save this incumbent as a possible optimal solution to FCNP, and eliminate from the candidate list any problems with β values greater than or equal to the value of the new incumbent. If the new $\nu^* = -\infty$, stop; FCNP is unbounded. Otherwise, proceed to Step 5a.

Step 5a. Execute Algorithm 4.2.2.6 to identify the macro-node assignments $\eta_c(n)$, and the reduced basis inverse entries $\lambda_c(i,k)$ from the optimal basis forest for RNP_c. Then go to Step 5b.

Step 5b. Construct problem SP_c from the optimal basis forest for RNP_c. One additional node and two additional arcs are added for each component of x_1 as in Figure 3, and basis labels are adjusted to correspond to the $\overline{FCNP}_c$ solution described in Theorem 3.1.1.4. Then go to Step 5e.

Step 5c. Create and solve the Infeasibility Pricing master problem MP_c from SP_c. Gomory cuts and SP_c tableau columns are derived as described in Section 5.2.2, and solution is by any Simplex procedure. Next, go to Step 5d.

Step 5d. Revise the optimal solution of SP_c as described in Step 2 of Algorithm 5.2.1.2 to permit dual solution improvement in the direction obtained from the dual multipliers of the Gomory cuts in the last MP_c. If the value of the dual solution for $\widetilde{FCNP}_c$ implied by this revised SP_c solution is greater than or equal to ν^*, go to Step 8 and fathom $FCNP_c$. If infeasibility in the Gomory cuts was forced to zero in the last solution to MP_c, go to Step 5f. Otherwise, go to Step 5e.

Step 5e. Change the dual solution to SP_c as much as possible in the direction obtained from the last MP_c according to the procedure outlined in Step 3 of Algorithm 5.2.1.2. If there is no limit on the amount of change, i.e. $\widetilde{\overline{FCNP}}_c$ is infeasible, proceed to Step 8 and fathom. Otherwise go to Step 5c.

Step 5f. Construct and solve all one-row Lagrangean either-or problems associated with $\overline{FCNP}_c$, i.e. all $LEOP_c(i)$. The objective function for these problems is given by the adjusted cost row of the final solution for SP_c, and constraints are generated from the $\eta_c(n)$ and $\lambda_c(i,k)$ as specified in Section 4.4.1. The problems are solved by the minimum ratio procedure of Section 4.4.2.

If the value of an optimal solution to any of these $LEOP_c(i)$ is greater than or equal to ν^*, go to Step 8 and fathom. Otherwise, go to Step 5g.

Step 5g. Using the "down" and "up" penalties obtained in the solution of the $LEOP_c(i)$, i.e. the values $\nu\left(LEOP_c(i)\colon\ y_i=0\right)$ and $\nu\left(LEOP_c(i)\colon\ y_i=u_{1i}\right)$, define the sets

$$\Theta_U = \{i\colon\ \nu\left(LEOP_c(i)\colon\ y_i=0\right) \geq \nu^*\}$$

$$\Theta_D = \{i\colon\ \nu\left(LEOP_c(i)\colon\ y_i=u_{1i}\right) \geq \nu^*\}$$

$$T = \left\{ i: \begin{array}{l} \max\{\nu\big(LEOP_c(i): y_i=0\big),\ \nu\big(LEOP_c(i): y_i=u_{1i}\big)\} \\ \text{is among the t highest values for } i \notin \Theta_U \cup \Theta_D \end{array} \right\}$$

Then go to Step 5h.

Step 5h. Construct and solve a $LEOP_c(i,j_i)$ for each $i \in T$, with j_i chosen by Criterion 3.5.2.1. The problems are constructed as in Section 4.4.1 and solved as four network problems on a graph constructed as in Section 4.4.3.

If the value of an optimal solution to any of these penalty problems is greater than or equal to ν^*, go to Step 8 and fathom. Otherwise, proceed to Step 6a.

Step 6a. Add to the sets Θ_U and Θ_D by

$$\Theta_U = \Theta_U \cup \{i \in T: \nu\big(LEOP_c(i,j_i): y_i=0\big) \geq \nu^*\}$$

$$\Theta_D = \Theta_D \cup \{i \in T: \nu\big(LEOP_c(i,j_i): y_i=u_{1i}\big) \geq \nu^*\}$$

Then go to Step 6b.

Step 6b. Choose the branching variable $y_{\hat{i}}$ so that $\hat{i}$ is the $i \in T$ and $\notin \Theta_D \cup \Theta_U$ which maximizes

$$\max\ \{\nu\big(LEOP_c(i,j_i): y_i=0\big),\ \nu\big(LEOP_c(i,j_i): y_i=u_{1i}\big)\}.$$

Then go to Step 7.

Step 7. Replace $FCNP_c$ in the candidate list by two more restricted problems. One is defined by $FCNP_c$ with the additional constraints that $y_{\hat{i}} = 0$, $y_j = 0$ for $j \epsilon \Theta_D$, and $y_j = u_{1j}$ for $j \epsilon \Theta_U$. The second problem is identical to the first except that $y_{\hat{i}}$ is restricted to equal $u_{1\hat{i}}$. β values for the two problems are $\nu\left(LEOP_c(\hat{i}, j_{\hat{i}})\colon\ y_{\hat{i}}=0\right)$ and $\nu\left(LEOP_c(\hat{i}, j_{\hat{i}})\colon\ y_{\hat{i}}=u_{1\hat{i}}\right)$, respectively. Next, go to Step 1.

Step 8. Fathom $FCNP_c$, i.e. eliminate $FCNP_c$ from the candidate list because no completion of it can produce a feasible solution to FCNP with value less than that of the incumbent solution. If the candidate list is now empty, stop. If an encumbent solution exists, it is an optimal solution for FCNP, and otherwise FCNP is infeasible. If the candidate list is not empty, proceed to Step 1.

6.3 Justification of the Algorithms

Justification of the above algorithms involves both showing the validity of the bounds on ν(FCP) and ν(FCNP) calculated in various steps of the procedures and demonstrating the convergence of the overall branch-and-bound scheme implicit in both algorithms. The validity of bounds used in the algorithms has been extensively treated in Chapters III, IV and V, and need not be demonstrated again here. Moreover, convergence of the overall branch-and-bound scheme is guaranteed by the observations that

1. The algorithms must terminate after processing of a finite number of candidate problems because there are only a finite number of settings for the components of y, and no repetitions are permitted.

2. The algorithms must produce an optimal solution or demonstrate that none exists because the only settings of components of y which are not explicitly pursued are those which bounding demonstrates cannot produce a better solution than some known one.

Thus it can be concluded that each of the above algorithms will produce an optimal solution or demonstrate that none exists in a finite number of steps.

CHAPTER VII

COMPUTATIONAL ANALYSIS

Chapters III, IV and V presented a number of group-related approaches to fixed charge problems in the form of FCP and FCNP, and Chapter VI integrated these proposals into branch-and-bound algorithms. The approaches chosen for inclusion in the algorithms of Chapter VI were those which theoretical analysis had indicated would be the most promising in the sense that they would most expedite a branch-and-bound procedure like the one presented at the beginning of Chapter III.

In this Chapter theoretical analysis of the value of various approaches is complemented by empirical investigation. A series of experiments are reported which analyze whether advantages suggested by theoretical results are actually observed in representative test problems.

7.1 Description of Experiments

The approach selected to accomplish such an empirical analysis is a classical factoral experimental design. A number of different group-related solution approaches which could be tested within available computer resources were applied to randomly-generated test problems possessing all combinations of the properties previous researchers have indicated most affected computational efficiency of algorithms for fixed charge problems.

In particular, a version of the algorithm of Section 6.2 was used to generate and solve fixed charge network problems in manners specified by the following factors:

1. *Type of problem* - whether the problem is a general FCNP (GNP), a fixed charge transportation problem (FCTP),[1] or a warehouse location problem (WLP).[1]

2. *Size of y* - the number of arcs in the problem with fixed charges (code 0 = 20, code 1 = 50, code 2 = 75, code 3 = 100).

3. *Relative size of fixed costs* - whether the fixed costs in a problem are small or large relative to variable costs (code 1 = small, i.e. fixed charges make up less than 5% of the value of an optimal solution; code 2 = large, i.e. fixed charges make up 15-30% of the value of an optimal solution).

4. *Solution method* - the combination of group-related techniques used in solution of the problem (code 0 = use no group-related techniques; code 1 = use only the EOP(i); code 2 = use only the BEOP(i); code 3 = use the BEOP(i) and randomly chosen EOP(i,j); code 4 = use the BEOP(i) and EOP(i,j) chosen by Criteria 3.5.2.1).

Details of the generation and solution routines employed are given in Appendix A.

It was initially planned to test all combinations of the above factors at the indicated level codes. However, preliminary testing revealed that structures of GNP's, FCTP's and WLP's were so different that results for different solution procedures could not be compared across problem types. In addition, early results showed that problems with the dimension of y greater than 50 could not be solved within reasonable time limits without some penalty techniques being used.

Thus two replications of six separate factoral experiments

---

[1]See Section 1.2 for definition of these special cases.

were actually performed. The three principal experiments focused separately on the GNP, FCTP and WLP cases. Each case was tested in all combinations of large y sizes (codes 1, 2 and 3), relative fixed costs, and group-related solution methods. In addition, three special experiments were run to analyze the impact on smaller GNP's, FCTP's, and WLP's of eliminating all group-related solution schemes.

Each combination of factors used in the experiments is presented in the list of Table 2. Parameters and results for each combination are then summarized in Tables 3, 4, 5 and 6.

7.2 Analysis of Experimental Results

General inspection of the results in Tables 3, 4, 5 and 6 suggests that relatively large fixed charge network problems can be solved in 10 to 15 minutes by any of the group-related penalty methods of Chapters III and IV. Averages for every problem type, y size and fixed charge pattern reported in those tables are within such reasonable computational boundaries, yet the problems with 100 fixed charge arcs are as large or larger than any FCNP's previously reported solved efficiently.

In order to more precisely determine the effect of various factors on the experimental results, statistical analysis of variance was applied to the results. The exact statistical assumptions underlying the analysis of variance could not be verified in the relatively unstructured domain of randomly-generated optimization problems, but since all the problem factors other researchers have indicated had significant effects on computational results were included in the experiments, the

Table 2. Definition of Test Problem Options

Code	No. Fix Chg. Arcs	Relative Size Fix Costs	Group-Related Penalty Problems in Solution Method
010	20	Small	None
011	20	Small	EOP(i)
020	20	Large	None
021	20	Large	EOP(i)
111	50	Small	EOP(i)
112	50	Small	BEOP(i)
113	50	Small	BEOP(i) and randomly chosen EOP(i,j)
114	50	Small	BEOP(i) and criteria chosen EOP(i,j)
121	50	Large	EOP(i)
122	50	Large	BEOP(i)
123	50	Large	BEOP(i) and randomly chosen EOP(i,j)
124	50	Large	BEOP(i) and criteria chosen EOP(i,j)
211	75	Small	EOP(i)
212	75	Small	BEOP(i)
213	75	Small	BEOP(i) and randomly chosen EOP(i,j)
214	75	Small	BEOP(i) and criteria chosen EOP(i,j)
221	75	Large	EOP(i)
222	75	Large	BEOP(i)
223	75	Large	BEOP(i) and randomly chosen EOP(i,j)
224	75	Large	BEOP(i) and criteria chosen EOP(i,j)
311	100	Small	EOP(i)
312	100	Small	BEOP(i)
313	100	Small	BEOP(i) and randomly chosen EOP(i,j)
314	100	Small	BEOP(i) and criteria chosen EOP(i,j)
321	100	Large	EOP(i)
322	100	Large	BEOP(i)
323	100	Large	BEOP(i) and randomly chosen EOP(i,j)
324	100	Large	BEOP(i) and criteria chosen EOP(i,j)

Table 3. Summary of General Network Test Problems

Problem Options	No. Nodes	No. Arcs	Fix Chg. Arcs	Average Part Solution Fix Chg.	Average No. Candidate Problems Solved	Average Solution Time (sec)
111	34	159	50	.010	6	2.2
112	34	159	50	.034	14	4.5
113	34	159	50	.048	17	6.2
114	34	159	50	.010	3	1.5
121	34	159	50	.067	10	3.2
122	34	159	50	.203	23	8.6
123	34	159	50	.055	6	1.8
124	34	159	50	.198	8	4.2
211	50	238	75	.024	13	6.8
212	50	238	75	.033	19	12.6
213	50	238	75	.039	24	16.2
214	50	238	75	.031	14	16.9
221	50	238	75	.074	12	8.3
222	50	238	75	.072	6	5.0
223	50	238	75	.094	22	14.6
224	50	238	75	.076	7	7.2
311	66	317	100	.026	18	21.6
312	66	317	100	.024	33	35.0
313	66	317	100	.041	16	22.1
314	66	317	100	.029	16	22.8
321	66	317	100	.277	38	50.8
322	66	317	100	.148	30	34.9
323	66	317	100	.182	28	26.1
324	66	317	100	.098	8	11.4

Table 4. Summary of Transportation Test Problems

Problem Options	No. Nodes	No. Arc	Fix Chg. Arcs	Average Part Solution Fix Chg.	Average No. Candidate Problems Solved	Average Solution Time (sec)
111	20	69	50	.020	3	.5
112	20	69	50	.019	22	2.2
113	20	69	52	.017	6	.8
114	20	69	52	.018	7	1.0
121	20	69	52	.189	332	43.8
122	20	69	52	.218	52	5.7
123	20	69	52	.227	94	11.2
124	20	69	52	.208	104	15.7
211	24	98	75	.020	14	3.2
212	24	98	75	.016	18	3.8
213	24	98	75	.022	11	2.2
214	24	98	75	.024	20	4.6
221	24	98	75	.202	652	146.2
222	24	98	75	.202	276	53.0
223	24	98	75	.230	262	76.6
224	24	98	75	.188	180	51.1
311	26	125	100	.022	33	10.6
312	26	125	100	.021	12	3.9
313	26	125	100	.023	34	10.4
314	26	125	100	.024	31	14.4
321	26	125	100	.230	1358	557.0
322	26	125	100	.186	548	159.4
323	26	125	100	.206	459	150.1
324	26	125	100	.154	279	151.2

Table 5. Summary of Warehouse Location Test Problems

Problem Options	No. Nodes	No. Arc	Fix Chg. Arcs	Average Part Solution Fix Chg.	Average No. Candidate Problems Solved	Average Solution Time (sec)
111	62	311	50	.048	5	4.3
112	62	311	50	.030	6	4.2
113	62	311	50	.032	6	5.9
114	62	311	50	.038	4	5.0
121	62	311	50	.166	70	51.5
122	62	311	50	.208	249	242.1
123	62	311	50	.142	18	15.6
124	62	311	50	.142	149	187.0
211	92	466	75	.034	16	26.0
212	92	466	75	.054	13	19.4
213	92	466	75	.040	16	26.3
214	92	466	75	.039	14	23.4
221	92	466	75	.139	161	312.9
222	92	466	75	.156	71	128.6
223	92	466	75	.164	233	478.4
224	92	466	75	.154	77	187.7
311	122	621	100	.032	16	37.8
312	122	621	100	.037	10	26.6
313	122	621	100	.036	8	22.2
314	122	621	100	.037	14	45.0
321	122	621	100	.146	216	663.5
322	122	621	100	.138	149	494.4
323	122	621	100	.130	258	803.6
324	122	621	100	.138	160	521.6

analysis of variance procedure was considered adequate for indicating the importance of various effects.

Table 6. Summary of Special Analysis Problems

Problem Type	Prob. Options	Nodes	Arcs	Fix Chg. Arcs	Average Part Solution Fix Chg.	Average No. Candidate Problems Solved	Average Solution Time (sec)
General	010	18	69	20	.021	39	1.4
General	011	18	69	20	.024	5	.4
General	020	18	69	20	.128	226	5.0
General	021	18	69	20	.381	7	.5
Transportation	010	14	33	20	.015	450	6.9
Transportation	011	14	33	20	.023	4	.2
Transportation	020	14	33	20	.202	1542	29.6
Transportation	021	14	33	20	.140	12	.4
Warehouse	010	20	45	20	.018	2536	32.3
Warehouse	011	20	45	20	.010	3	.1
Warehouse	020	20	45	20	.100	1098	20.2
Warehouse	021	20	45	20	.075	6	.2

The response variable selected for analysis of variance calculations is the number of candidate problems explicitly investigated in solving a FCNP. This variable was chosen because it appeared to give the most accurate measure of the true impact of different group-related penalty procedures. Solution times are also very important, but the effect of the various penalty procedures on solution times is clouded by the programming efficiency of routines to execute the penalty procedures.

Results of the analysis of variance for this response variable are given in Tables 7 through 12. The significance of various

Table 7. Analysis of Variance for General Test Problems

Effect	Sum of Squares	Degrees of Freedom	Mean Square	F-ratio[1]
Size of y	1,325	2	662	4.80 ***
Relative fixed cost	3	1	3	.022
Solution method	889	3	296	2.15 *
Size-cost interaction	242	2	121	.877
Size-method interaction	717	6	120	.870
Cost-method interaction	210	3	70	.507
Error	4,155	30	138	1.00

Table 8. Analysis of Variance for Transportation Test Problems

Effect	Sum of Squares	Degrees of Freedom	Mean Square	F-ratio[1]
Size of y	578,314	2	289,157	2.97 **
Relative fixed cost	1,603,814	1	1,603,814	16.5 ***
Solution method	646,824	3	215,608	2.21 *
Size-cost interaction	503,331	2	251,666	2.58 **
Size-method interaction	219,089	6	36,515	.375
Cost-method interaction	654,060	3	218,020	2.24 *
Error	2,921,737	30	97,391	1.00

[1]Single * denotes significant at α = .25 level; ** denotes significant at α = .10 level, and *** denotes significant at α = .05.

Table 9. Analysis of Variance for Warehouse Location Test Problems

Effect	Sum of Squares	Degrees of Freedom	Mean Square	F-ratio[1]
Size of y	13,697	2	6,848	.890
Relative fixed cost	235,760	1	235,760	30.6 ***
Solution method	2,543	3	847	.110
Size-cost interaction	11,244	2	5,622	.731
Size-method interaction	53,701	6	8,950	1.16
Cost-method interaction	2,815	3	938	.122
Error	230,759	30	7,692	1.00

Table 10. Analysis of Variance for General Problems in Special Analysis

Effect	Sum of Squares	Degrees of Freedom	Mean Square	F-ratio[1]
Relative fixed cost	17,860	1	17,860	1.13
Solution method	32,004	1	32,004	2.02 *
Cost-method interaction	17,133	1	17,133	1.08
Error	63,308	4	15,827	1.00

[1]Single * denotes significance at α = .25 level; ** denotes significance at α = .10 level, and *** denotes significance at α = .05.

Table 11. Analysis of Variance for Transportation Problems in Special Analysis

Effect	Sum of Squares	Degrees of Freedom	Mean Square	F-ratio[1]
Relative fixed cost	605,000	1	605,000	55.1 ***
Solution method	1,955,288	1	1,955,288	178.2 ***
Cost-method interaction	548,528	1	548,528	53.3 ***
Error	43,896	4	10,974	1.00

Table 12. Analysis of Variance for Warehouse Location Problems in Special Analysis

Effect	Sum of Squares	Degrees of Freedom	Mean Square	F-ratio[1]
Relative fixed cost	1,030,330	1	1,030,330	2.02 *
Solution method	6,572,125	1	6,572,125	12.9 ***
Cost-method interaction	1,037,520	1	1,037,520	2.03 *
Error	2,044,017	4	511,004	1.00

[1]Single * denotes significance at α = .25 level; ** denotes significance at α = .10 level, and *** denotes significance at α = .05.

factors implied by the results is discussed in the next several subsections.

7.2.1 Size of y Effects

Since the possible number of candidate problems increases exponentially with the size of the y vector, it could be expected that the number of candidate problems actually solved would also be greatly affected by the size of y. Analysis of variance results for GNP's and FCTP's generally confirm this expectation. Both Table 7 and Table 8 show fairly significant size effects.

It is interesting, however, that the same significance is not observed in the results for WLP's. Great variations in the response were observed at all sizes of y.

7.2.2 Relative Fixed Cost Effects

Most previously reported research on fixed charge problems has also indicated that computational efficiency is highly effected by the relative size of the fixed and variable costs. If fixed costs are small, $\nu(\overline{FCNP})$ provides a good estimate of $\nu(FCNP)$, and only a few candidates need to be explicitly explored. When fixed costs are high, however, numerous possibilities for y must be investigated.

Experimental results for FCTP's and WLP's strongly confirm this previous experience. Relative fixed cost effects appear very significant in both Table 8 and Table 9.

However, results for GNP's show the relation between fixed and variable costs is relatively insignificant. A possible explanation of this phenomenon is that higher fixed costs in GNP's tend only to force all flows along arcs without fixed charges. Thus, the value of a $v(\overline{FCNP})$ as a bound on $v(FCNP)$ is not diminished as fixed charges increase.

7.2.3 Solution Method Effects

The experimental factor of greatest interest to the research of this dissertation is the effect of changing the solution procedure used. Any group-related techniques shown to be significantly superior would provide suitable focuses for future research and applications.

Results in Tables 7 through 12 do show significant solution method effects except in the case of large WLP's. The most outstanding of these effects is the difference between the no-group analysis and one-row analysis methods. Even for the relatively small case of 20 fixed charge arcs, results in Tables 10 through 12 indicate significant improvements are obtained by using at least some group-related penalties.

Unfortunately, the differences among group-related techniques are not so clear from experimental results. Both GNP's and FCTP's showed some significant effects of variation in solution methods, but differences between particular methods cannot be statistically verified.

The clearest pattern arises in the FCTP case presented in Figure 11. This graph shows the mean number of candidate problems investigated as a function of problem size and solution method. The results strongly imply that simple one-row group penalties are inferior to other techniques. Some advantage is apparently gained by

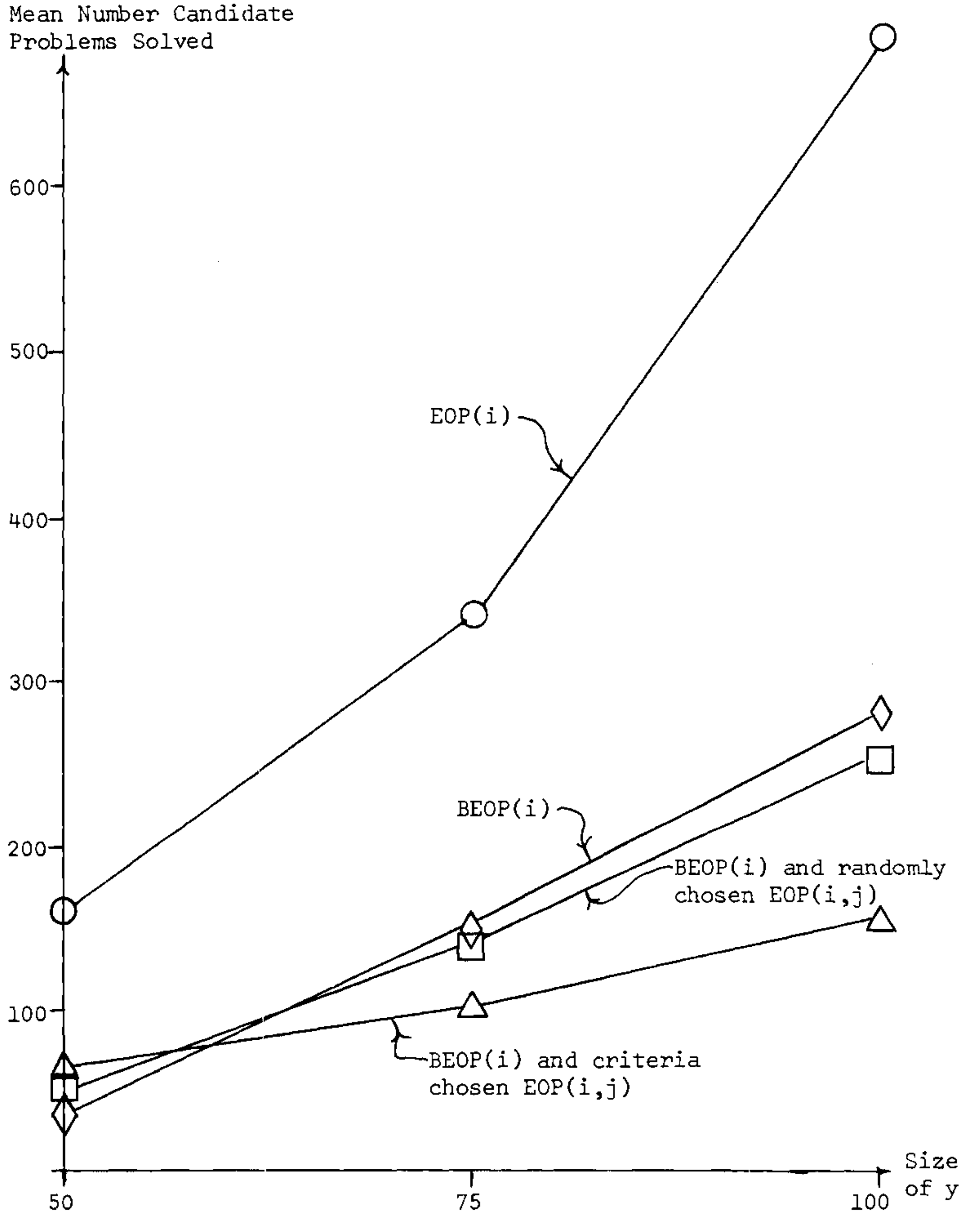

Figure 11. Mean Numbers of Candidate Problems Solved for Transportation Problems by Size of y and Solution Method

using two-row approaches, but the greatest single improvement derives from consideration of the bounds on perturbation variables.

The GNP pattern of Figure 12 is somewhat more confused because performance of the solution methods varies widely. However, the approach of using both one-row and criteria-selected two-row problems appears to be the most effective method. Since results of Figure 11 also show this method to be the most effective, it appears safe to conclude the method has some promise for efficient solution of FCNP's.

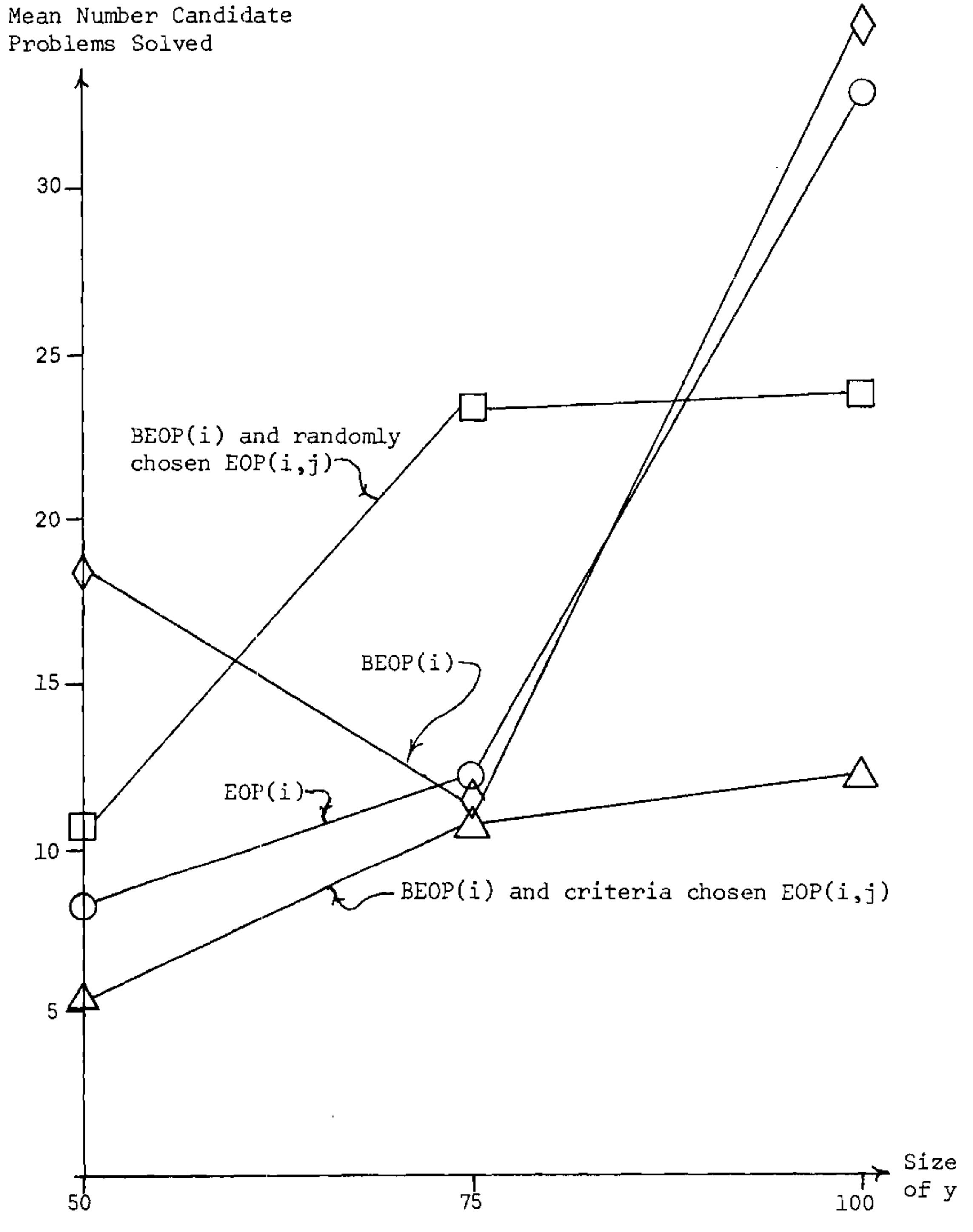

Figure 12. Mean Numbers of Candidate Problems Solved for General Test Problems by Size of y and Solution Method

CHAPTER VIII

CONCLUSIONS AND RECOMMENDATIONS

The principal objective of the research reported in this dissertation was to identify effective techniques for exploiting the special structures of fixed charge problems in group-theory-related branch-and-bound solution procedures. A number of such techniques were developed in Chapters III, IV and V, and computational experience was reported in Chapter VII. The principal results of those chapters can be summarized as follows:

1. *Extension of Kennington's Work to General Fixed Charge Problems.* The great majority of results for fixed charge transportation problem in [68] were shown to generalize to any fixed charge linear program. In particular, the solution of the continuous analog of FCP and all group-related penalty problems can be simply constructed from the optimal solution to a reduced linear program, one-row group penalty problems can easily be solved exactly because all congruence-constrained variables can be made basic, and one-row group penalties are exactly equal to those of Tomlin [90] and thus can be interpreted as having an "up" or "down" orientation.

2. *Investigation of Interactions Between Rows of the Group Problem for FCP.* Some understanding of the interactions between rows of the group problem associated with FCP can be obtained by investigating the conditions under which individual rows can be independently

satisfied. In this connection, the Gomory cuts for FCP can be shown to be faces of the convex hull of solutions to the associated group problem because of a limited independence between rows, and the property of connectedness is related to the degree of independence.

3. *Specialization to the Network Case.* The group problem for a fixed charge network problem can be interpreted as the problem of finding an optimal set of flow adjustments around cycles in the network corresponding to nonbasic arcs. Using this interpretation, the full problem network can be collapsed for purposes of group-related analysis into a reduced network on macro-nodes. Moreover, the constraints of the group-related problems can be shown to be totally unimodular.

4. *Lagrangean Enforcement of Bounds on Basic Variables.* A computationally feasible method for including constraints relaxed in penalty problems in the objective function via Lagrange multipliers can be obtained by imposing Gomory cuts on the optimal continuous solution to FCP and using appropriate dual variable values for the multipliers. Moreover, this approach can be extended to the fixed charge network case without sacrificing the special network structure by a decomposition approach.

5. *Computational Experience with Fixed Charge Network Problems.* Randomly selected fixed charge network problems with as many as 100 fixed charge variables can be efficiently solved by a group-related branch-and-bound algorithm. Moreover, the value of group-related penalty approaches can be statistically verified.

At the completion of this research, however, a number of possibilities for fixed charge problems still require investigation. Among these are the following:

1. *Additional Computational Experience.* The relatively promising computational results for fixed charge network problems which were presented in Chapter VII suggest the more complex computational procedures omitted from initial experiments are worthy of testing. Specifically, implementations of the group-related penalty procedures proposed for general fixed charge problems and of the Lagrangean techniques presented in Chapter VII should be pursued.

2. *Improved Selection Criteria.* The results in Chapter VII indicate some advantage of two-row penalty problems selected according to Criteria 3.5.2.1, but greater improvement might derive from a better criterion. Additional research into measures of interaction between rows of group problems appears warranted.

3. *Alternative Group Problems.* Section 3.2 defined several group problems which could be derived from a given solution to the reduced problem RP. The alternative involving congruence constraints on perturbation variables was not pursued in this dissertation because of the difficulty of solving the resulting penalty problems. However, this alternative group problem could be approached by the approximate techniques of Gomory and Johnson [43,44,65]. Improved penalties might derive from dealing with this alternative group formulation and using such approximate methods.

4. *Graph of the Group Problem.* In Section 4.4.3 it was

demonstrated that two-row group-related penalty problems could be viewed as minimum cost flow problems on a particular graph. However, Tutte [92,93] has shown that such a graph can be constructed for many totally unimodular matrices. Since Theorem 4.3.2.1 demonstrated that the constraint matrix of the group problem for every fixed charge network problem is totally unimodular, it may thus be possible to identify an underlying graph for penalty problems of more than two rows.

More generally, certain of the principles presented in this research on fixed charge problems may be applicable to other integer and mixed-integer programs. Any class of problems for which a feasible solution to the full problem can always be constructed from a solution to its continuous relaxation would probably possess most of the properties demonstrated in Chapter III because some "independence" of integer variables would have to be present, i.e. it would have to be possible to perturb the continuous solution to satisfy congruence constraints one by one. Similarly, the Gomory cut approach for deriving Lagrange multipliers which was presented in Chapter V is applicable to improving group problems for any mixed-integer program. In fact, any set of valid inequalities could be used in the suggested manner to obtain Lagrange multipliers for the bounds on basic variables.

APPENDIX A

ALGORITHM USED IN COMPUTATIONAL EXPERIMENTS

In Chapter VII some computational experiments with fixed charge network problems were reported. All those experiments were run with a version of the algorithm of Section 6.2 implemented in FORTRAN on the Georgia Institute of Technology's Univac 1108.

For the convenience of other researchers, an outline of this algorithm is presented in this Appendix. A first section summarizes the algorithm in a step-by-step fashion, and later sections add comments on some techniques not detailed elsewhere in the dissertation and offer some computational observations. The notation of Chapter VI and the step numbers of Figure 1 are used throughout.

A.1 Statement of the Algorithm

In terms of the notation of Chapters III through VI, and the step numbers of Figure 1, the fixed charge network algorithm used in experimentation can be stated:

Step 0a. Generate a random problem FCNP according to input specifications and construction techniques outlined in Section A.2.1. Then go to Step 0b.

Step 0b. Solve RNP using the algorithm summarized in Section A.2.2, and save the optimal solution and basis forest for restart solution of later sub-problems as described in Section A.2.3. Then, go to

Step 0c.

Step 0c. Initialize all pointers in the candidate list defined in Section A.2.4, and place FCNP in the candidate list with $\beta(\text{FCNP}) = \nu(\text{RNP})$. Also, set $\nu^* = +\infty$, $\text{FCNP}_c = \text{FCNP}$ and go to Step 3.

Step 1. Select as FCNP_c the lowest cost bound element of the candidate list chain defined in Section A.2.4. Then go to Step 2a.

Step 2a. Use the saved optimal solution to RNP to create a basic feasible solution for RNP_c according to the method outlined in Section A.2.3. Then, go to Step 2b.

Step 2b. Starting from the solution and basis forest constructed in Step 2a, solve RNP_c. If $\nu(\text{RNP}_c) \geq \nu^*$, go to Step 8 and fathom. Otherwise, go to Step 3.

Step 3. Create a feasible solution for FCNP by rounding "up" the optimal solution to $\overline{\text{FCNP}}_c$, i.e. by setting

$$s_j = \begin{cases} u_{1j} - \bar{x}_{1j} & \text{if } \bar{x}_{1j} > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$y = s + \bar{x}_1$$

$$x_1 = \bar{x}_1$$

$$x_2 = \bar{x}_2,$$

where $\bar{x}_1$ and $\bar{x}_2$ are the optimal values of x_1 and x_2 in the solution to RNP_c. If the value of this rounded solution is less than ν^*, go to

Step 4a. Otherwise, go to Step 5a.

Step 4a. Save the rounded $\overline{FCNP}_c$ solution as the new incumbent solution. If the new ν^* is greater than the β value of every candidate in the candidate list, go to Step 5a. If the new $\nu^* = -\infty$, stop; FCNP is unbounded. Otherwise go to Step 4b.

Step 4b. Scan up the candidate chain defined in Section A.2.4 until a candidate is found with β value at least equal to ν^*. Eliminate from further consideration all problems in the part of the chain beginning with that candidate. Then, go to Step 5a.

Step 5a. Go to Step 8 and fathom if the optimal solution to $\overline{FCNP}_c$ satisfies all congruence constraints on y. Otherwise, go to Step 5b if group-related penalties are to be used for this test, and to Step 6a if not.

Step 5b. Execute Algorithm 4.2.2.6 to identify the macro-node assignments $\eta_c(n)$, and the reduced basis inverse entries $\lambda_c(i,k)$ from the optimal basis forest for RNP_c. Then go to Step 5c.

Step 5c. Calculate the adjusted cost coefficients of nonbasic arcs in the optimal solution to RNP_c by the method of Section 4.4.1.1 to provide the objective function for group-related penalty problems. Then go to Step 5d.

Step 5d. Construct an increasing cost sequential chain of all arcs which were nonbasic in the optimal solution to $\overline{FCNP}_c$ and which connect two nodes that are not members of the same macro-node. Then, go to Step 5e.

Step 5e. If bounds on one-row problems are to be considered in this test solve all BEOP(i), and otherwise solve all EOP(i). In either case, the method of solution is the one specified in Section 4.4.2 with nonbasic arcs being considered in the order of the chain of Step 5d.

If the value of the optimal solution to any of these one-row problems is greater than or equal to ν^*, go to Step 8 and fathom. Otherwise, go to Step 5f.

Step 5f. Using the "up" and "down" penalties obtained in the solution of the one-row problems of Step 5e, define the sets

$$\theta_U = \left\{ i: \begin{array}{l} \text{the "down" (i.e. } y_i=0\text{) case of the } ith \\ \text{problem in Step 5e had value} \geq \nu^* \end{array} \right\}$$

$$\theta_D = \left\{ i: \begin{array}{l} \text{the "up" (i.e. } y_i=u_{1i}\text{) case of the } ith \\ \text{problem in Step 5e had value} \geq \nu^* \end{array} \right\}$$

If no two-row group-related penalty problems are to be used in this test, go to Step 6b. Otherwise, go to Step 5g.

Step 5g. Using the results of Step 5e and 5f define the sets

$$T = \left\{ i: \begin{array}{l} \text{the maximum of the value of the "up" and} \\ \text{the "down" case in the } ith \text{ problem of Step} \\ \text{5e is among the t greatest values for such} \\ \text{maxima among } i \notin \Theta_U \cup \Theta_D. \end{array} \right\}$$

Then, go to Step 5h.

Step 5h. Construct and solve an EOP(i,j_i) for each $i \in T$. The problems are constructed as in Section 4.4.1 and solved according to the minimum ratio procedure of Section A.2.5. The second row j_i is chosen randomly if criteria are not to be used in this test, and otherwise as the j which maximizes $\omega_1(j) + \omega_2(j)$ where,

$$\omega_1(j) = \begin{cases} 1 & \text{if } x_{1j} \text{ was nonbasic in the optimal solution to } RNP_c \\ 2 & \text{if } x_{1j} \text{ was basic in the optimal optimal solution to } RNP_c \text{ and either } \bar{x}_{1j} = 0 \text{ or } \bar{x}_{1j} = u_{1j} \\ 3 & \text{otherwise.} \end{cases}$$

$$\omega_2(j) = \begin{cases} 1 & \text{if either row i or row j is connected in EOP(i,j), but not both.} \\ 2 & \text{if both rows i and j connected in EOP(i,j)} \\ 0 & \text{otherwise.} \end{cases}$$

If the value of the optimal solution to any of these penalty problems is greater than or equal to ν^*, go to Step 8 and fathom. Otherwise, proceed to Step 6c.

Step 6a. Choose the branching variable $y_{\hat{i}}$ randomly, set $\Theta_D = \Theta_U = \Phi$, and let

$$\beta(FCNP_c: y_{\hat{i}}=0) = \beta(FCNP_c: y_{\hat{i}}=u_{1\hat{i}}) = \nu(\overline{FCNP_c}).$$

Then, go to Step 7.

Step 6b. Choose the branching variable $y_{\hat{i}}$ so that $\hat{i}$ is the i which maximizes for $i \notin \Theta_U \cup \Theta_D$ the maximum of the values of the "up" and the "down" cases in the *ith* problem solved in Step 5e. Let

$$\beta(FCNP_c\colon y_{\hat{i}}=0) \quad = \text{the "down" value for } \hat{i} \text{ in Step 5e.}$$

$$\beta(FCNP_c\colon y_{\hat{i}}=u_{1\hat{i}}) = \text{the "up" value for } \hat{i} \text{ in Step 5e.}$$

Then, go to Step 7.

Step 6c. Add to the set Θ_U and Θ_D by

$$\Theta_U = \Theta_U \cup \{i \epsilon T\colon \nu\big(EOP_c(i,j_i)\colon y_i=0\big) \quad \geq \nu^*\}$$

$$\Theta_D = \Theta_D \cup \{i \epsilon T\colon \nu\big(EOP_c(i,j_i)\colon y_i=u_{1i}\big) \geq \nu^*\}.$$

Then go to Step 6d.

Step 6d. Choose the branching variable $y_{\hat{i}}$ so that $\hat{i}$ is the $i \epsilon T$ and $\notin \Theta_U \cup \Theta_D$ which maximizes

$$\max\{\nu\big(EOP_c(i,j_i)\colon y_i=0\big),\ \nu\big(EOP_c(i,j_i)\colon y_i=u_{1i}\big)\}.$$

Let

$$\beta(FCNP_c\colon y_{\hat{i}}=0) \quad = \nu\big(EOP_c(\hat{i},j_{\hat{i}})\colon y_{\hat{i}}=0\big)$$

and

$$\beta(FCNP_c\colon y_{\hat{i}}=u_{1\hat{i}}) = \nu\big(EOP_c(\hat{i},j_{\hat{i}})\colon y_{\hat{i}}=u_{1\hat{i}}\big).$$

Then, go to Step 7.

Step 7. Remove $FCNP_c$ from the candidate list chain defined in Section A.2.4 and create two new candidate problems. One is defined by $FCNP_c$ with the additional constraints that $y_{\hat{i}} = 0$, $y_{\hat{j}} = 0$ for $j \epsilon \Theta_D$, and $y_{\hat{j}} = u_{1\hat{j}}$ for $j \epsilon \Theta_U$. The second problem is identical to the first except that $y_{\hat{i}}$ is restricted to equal $u_{1\hat{i}}$. β values for the two problems are as defined in Step 6.

After the two problems have been created, scan up the candidate chain to insert the problems in proper cost bound sequence. Then, proceed to Step 1.

Step 8. Fathom $FCNP_c$, i.e. eliminate $FCNP_c$ from the candidate list chain. If the candidate list is now empty, stop. If an incumbent solution exists, it is optimal solution for FCNP, and otherwise FCNP is infeasible. If the candidate list is not empty, proceed to Step 1.

A.2 Detailed Description of Important Components

The majority of the procedures included in the above algorithm are either self-explanatory or documented in the main text of this dissertation at the points indicated by references in the statement of the algorithm. However, certain computational techniques not directly connected with the theory of this dissertation do require additional explanation. The following sub-sections provide such detail on five important components of the above algorithm.

A.2.1 Generation of Random Problems

The method used to generate random problems at Step 0a of the above algorithm is a variation of the scheme proposed by Klingman, Napier and Stutz in [72]. Input cards are read which specify certain parameters of the desired network and a seed for a pseudo-random number generator. A FORTRAN routine then creates a feasible network problem satisfying the specifications in the input cards.

Specific items in the input card are defined in Table 13. Before outlining how these parameters are used to create a test network problem, however, it will be useful to define the term *random partitioning*. A random partitioning of a given quantity into sub-quantities associated with the elements of some finite set is achieved by

1. Obtaining a prorata share by dividing the total quantity by the number of elements in the set.
2. Randomly dividing the prorata share for each element of the set into two parts, one added to the sub-quantity for that element and the other added to the sub-quantity for a randomly chosen element.
3. Adding any remaining part of the total quantity due to rounding in previous steps to the sub-quantity for a randomly chosen element.

With this definition and the parameters of Table 13, the problem generating routine can be outlined as shown below. Costs for generated arcs are calculated as specified in Table 14.

Step i. Randomly partition the total supply over all sources (pure and transshipment source nodes).

Step ii. If any pure transportation nodes are to be included, randomly partition them over all sources.

Table 13. Input Parameters for Problem Generation

Parameter Name	Definition
Problem type	Whether the problem is a general fixed charge network problem, a fixed charge transportation problem, or a warehouse location problem.
Pure sources	The number of nodes which are to be pure sources or supply points.
Pure sinks	The number of nodes which are to be pure sinks or demand points.
Transshipment sources	The number of nodes which are to be both supply points and transshipment points.
Transshipment sinks	The number of nodes which are to be both demand points and transshipment points.
Pure transshipment	The number of nodes which are to be purely transshipment points.
Arcs	The total number of arcs in the network.
Fixed charge arcs	The total number of arcs with fixed charges in the network.
Variable cost	The upper and lower limits on variable costs for arcs in the network.
Prorata fixed cost range	The upper and lower limits on prorata fixed costs for fixed charge arcs in the network.
Upper bound range	The upper and lower limits on upper bounds for arcs in the network.
Total supply	The total supply for all sources in the network.
Percent excess supply	The fraction by which total demand is less than total supply.
Seed	The seed for the pseudo-random number generator of the generating routine.

Table 14. Cost Calculations for Generated Arcs

Arc Type	Problem Case	Variable Cost	Pro Rata Fixed Cost[1]
Super-source to source	General FCNP, fixed charge arc[2]	Random within input range	Random within input range
	General FCNP, non-fixed charge arc[2]	Random within input range	None
	Fixed charge transportation problem	0	None
	Warehouse location problem	0	Random within input range
Sink to super-sink	All cases	0	None
Super-sink to super-source	All cases	0	None
Non-circularizing	General FCNP, fixed charge arc[2]	Random within input range	Random within input range
	General FCNP, non-fixed charge arc[2]	Random within input range	None
	Fixed charge transportation problem	Random within input range	Random within input range
	Warehouse location problem	Random within input range	None

[1]Note that pro rata, not full fixed costs are generated. Thus the full fixed charge on a given arc is correlated with its upper bound.

[2]In general fixed charge network problems, it is randomly decided whether or not an arc should have a fixed charge according to the average density of fixed charge arcs implied by input parameters.

Step iii. Create arcs to make a chain between each source and any pure transshipment nodes allocated to it. Lower bounds for the arcs are zero, and upper bounds are assigned by the formula

$$\begin{matrix}\text{upper} \\ \text{bound}\end{matrix} = \max \left\{ \begin{matrix}\text{lower limit on upper} \\ \text{bounds specified on input}\end{matrix}, \quad \begin{matrix}\text{supply allocated} \\ \text{to this chain}\end{matrix} \right\} .$$

If no pure-transshipment nodes are assigned to a given source its chain consists solely of the supply node itself.

Step iv. Connect each chain to a randomly chosen set of sinks (pure and transshipment sink nodes). Bounds for such arcs are assigned exactly as in Step iii.

Step v. Calculate the preliminary demands for each sink by randomly partitioning the supply on each chain over the sinks connected to the chain.

Step vi. If total supply is not to equal total demand, reduce the calculated demand at each sink by an amount equal to the percentage of excess specified on input.

Step vii. Build circularizing arcs connecting each source to a super-source, each sink to a super-sink, and the super-sink to the super-source. Bounds on these arcs are given by

$$\text{lower bound} = \begin{cases} 0 & \text{if arc connects super-source to source} \\ \text{Demand at sink} & \text{if arc connects sink to super-sink} \\ \text{Total demand} & \text{if arc connects super-sink to super-source} \end{cases}$$

$$\text{upper bound} = \begin{cases} \text{Supply at source} & \text{if arc connects super-source to source} \\ \text{Total supply} & \text{if arc connects sink to super-sink} \\ \text{Total supply} & \text{if arc connects super-sink to super-source.} \end{cases}$$

Step viii. Build the remaining fixed charge and non-fixed charge arcs required to reach the total numbers specified on input. Lower bounds are 0, and upper bounds are chosen randomly within the range specified on input.

Step ix. If the problem is a transportation problem or a warehouse location problem, revise the upper bounds on arcs connecting sources to sinks to conform to the formula

$$\text{upper bound} = \min \left\{ \begin{matrix}\text{supply at} \\ \text{arc's source}\end{matrix}, \quad \begin{matrix}\text{demand at} \\ \text{arc's sink}\end{matrix} \right\}.$$

A.2.2 Solution of Network Problems

The continuous network problems RNP and RNP_c in the algorithm of Section A.1 were solved by a primal Simplex procedure which takes advantage of the rooted spanning forest representation of a basis. In addition to the concepts presented in Section 4.1, the important theoretical result on which the procedure is based is that it is only necessary to change $\pi(n)$ and basis labels in the part of the basis forest above the outgoing arc to execute a Simplex pivot. For a proof of this result see Langley [75].

Table 15 presents the information stored about each arc and each node. Using this information, a Simplex procedure for RNP or RNP_c can be outlined as follows:

Step i. Create an initial basic solution by setting the flow on each arc equal to its lower bound and introducing an artificial arc to balance flows at each node. Artificial arcs are treated as having the very large positive cost $\hat{m}$. Next, go to Step ii.

Step ii. Search all nonbasic arcs until one is located which is lower-bounded with negative adjusted cost or upper-bounded with positive adjusted cost. If no such arc is found, stop; the algorithm is completed. Otherwise define the located arc to be the *incoming arc* for the current Simplex pivot.

Step iii. Use the down node and down arc labels to search down from both ends of the incoming arc until the cycle formed in the basis forest by the incoming arc has been identified. Define the forward direction around this cycle as the direction of flow on the arc, if the arc is lower-bounded, and the opposing direction if the arc is upper-bounded. Choose as the *outgoing arc* for this Simplex pivot the basic arc in this cycle which allows the minimum flow change where permissible

Table 15. Information Stored for Network Algorithm

Name	Associated With	Definition
Begin node	arcs	The number of the node where the arc begins.
End node	arcs	The number of the node where the arc ends.
Lower bound	arcs	The lower bound on flow on the arc.
Upper bound	arcs	The upper bound on flow on the arc.
Fixed cost	arcs	The pro rata fixed charge associated with the arc.
Variable cost	arcs	The variable cost associated with the arc.
Flow	arcs	The current flow on the arc.
Basis status	arcs	Whether the arc is currently basic, non-basic lower-bounded or nonbasic upper-bounded.
Down node $(\delta(n))$	nodes	The number of the node immediately below the labeled node in the basis forest (0 if none).
Up node $(\mu(n))$	nodes	The number of a node immediately above the labeled node in the basis forest (0 if none).
Right node $(\gamma(n))$	nodes	The number of another node sharing the same down node with the labeled node (0 if none).
Down arc $(\alpha(n))$	nodes	The number of the arc connecting the labeled node to its down node.
Dual variable $(\pi(n))$	nodes	The current value of the dual multiplier for the labeled node.

change for arcs in the forward direction is the difference of their upper bounds and their current flows, and permissible change for arcs in the reverse direction is the difference of their current flows and their lower bounds. If no such arc exists because the flow change around the cycle is unrestricted, stop; the current problem is unbounded. Otherwise, go to Step iv.

Step iv. Use the down node, up node and right node labels to search through the part of the tree above the outgoing arc and revise basis labels as required to "disconnect" this part of the forest at the outgoing arc and "connect" it at the incoming arc. Then go to Step v.

Step v. Use the revised basis labels to search the part of the forest previously above the outgoing arc and revise the dual variables $\pi(n)$ as required to restore complementary slackness, i.e. as required to make adjusted costs equal to zero along all basic arcs. Then go to Step ii.

A.2.3 Restart of the Network Algorithm

Execution of an algorithm like the one presented in Section A.1 requires solution of many reduced problems RNP_c derived from candidate problems $FCNP_c$. Recognizing that each $\overline{FCNP}_c$ is merely $\overline{FCNP}$ with the values of some y_j constrained to particular values, the strategy of beginning the solution of each $\overline{FCNP}_c$ from the optimal solution to $\overline{FCNP}$ was adopted for the computational experimentation of this dissertation.

To see how this can be accomplished with the reduced problems RNP_c, recall that the cost coefficients c_{1j} on variables with fixed charges include two components. Specifically,

$$c_{1j} = v_j + c_{sj}$$

where v_j = the variable cost on the arc associated with x_{1j}.

c_{sj} = the pro rata fixed cost on the arc associated with x_{1j}.

Thus, when a given y_j is fixed in $FCNP_c$ these coefficients can be represented by

$$c_{1j} = \begin{cases} v_j + \hat{m} & \text{if } y_j \text{ is fixed at } 0 \\ v_j + 0 & \text{if } y_j \text{ is fixed at } u_{1j}, \end{cases}$$

where $\hat{m}$ is a very large constant. Moreover, this representation implies that only the objective function changes when RNP is replaced by RNP_c. Thus, the optimal basic solution to RNP is a basic feasible solution to RNP_c.

In particular, the above observations demonstrate that the optimal flows and basis forest from RNP may be used as a starting point for RNP_c. However, the dual solution defined by the $\pi(n)$ labels must be changed to obtain complementary slackness for the starting network of RNP_c, i.e. to make the adjusted costs of all basic arcs equal to 0. Thus, the procedure adopted for this dissertation saves all flows and basis labels at Step 0b of Algorithm A.1, but starting $\pi(n)$ are recalculated for each RNP_c. The details of Step 2a in Algorithm A.1 can be stated as follows:

Step i. Restore all arc flows and basis labels to their optimal RNP status and determine c_1 for RNP_c. Then go to Step ii.

Step ii. If all trees in the RNP_c basis forest have been labeled, stop; the algorithm is completed. Otherwise, locate the next unlabeled tree, and let n = the node number of its base. Next, set $\pi(n) = \hat{m}$ and go to Step iii.

Step iii. Proceed up by letting $n' = \mu(n)$. If $n' = 0$, go to Step iv. Otherwise, set

$$\pi(n') = \begin{cases} \pi(n) + \text{the cost of } \alpha(n') & \text{if } \alpha(n') \text{ is oriented away from the base of the tree.} \\ \pi(n) - \text{the cost of } \alpha(n') & \text{if } \alpha(n') \text{ is oriented toward the base of the tree.} \end{cases}$$

Next, set n = n' and repeat Step iii.

Step iv. Proceed right by letting $n' = \gamma(n)$. If $n' = 0$, go to Step v. Otherwise, set

$$\pi(n') = \begin{cases} \pi\big(\delta(n')\big) + \text{the cost of } \alpha(n') & \text{if } \alpha(n') \text{ is oriented away from the base of the tree.} \\ \pi\big(\delta(n')\big) - \text{the cost of } \alpha(n') & \text{if } \alpha(n') \text{ is oriented toward the base of the tree.} \end{cases}$$

Next, set n = n' and go to Step iii.

Step v. Proceed down by letting $n' = \delta(n)$. If n' is the base of the current tree, go to Step ii. Otherwise, set n = n' and go to Step iv.

A.2.4 Handling and Storage of the Candidate List

In order to efficiently store the possibly large number of candidate problems awaiting investigation at various times in the execution of the algorithm of Section A.1, a doubly chained list technique was adopted. More specifically, the portion of core storage allocated for candidate problems was divided into seven-word segments packed on a bit-by-bit basis with the items listed in Table 16.

Table 16. Items Stored on Candidate Problems

Item Name	Definition
Chain successor pointer	The number of the next segment in the chain to which this candidate list area segment currently belongs.
Cost bound	The best available lower bound on the value of any completion of the problem stored in this segment of the candidate list area.
Variable fixed statuses	Indicators for each $i = 1,2,...,n_1$ of whether y_i is fixed at value 0, fixed at value u_{1i} or free in the problem stored in this segment of the candidate list area.

At any point in the solution of a problem, these segments were connected in two chained lists. The first connected the segments containing information about candidate problems which were still members of the candidate list. These segments were linked in ascending order of cost bound. The second chained list connected segments of the candidate list area in core which were available for storage of newly created candidate problems.

In order to fully explain the handling of these lists, define the following variables:

$\hat{b}$ = the segment number of the pending candidate problem with the lowest cost bound.

$\hat{t}$ = the segment number of the pending candidate problem with the highest cost bound.

$\hat{n}$ = the segment number of the next segment available for a newly created candidate.

$\beta(k)$ = the cost bound of the problem in segment number k.

$\psi(k)$ = the chain successor pointer of segment number k.

The following sections describe the processing of these variables at each relevant step of the algorithm of Section A.1.

A.2.4.1 Initialization at Step 0c. The steps in initializing the candidate list chains a Step 0c are the following:

Step i. Set $\psi(k) = k + 1$ for all segments k in the candidate list area of core memory.

Step ii. Place the problem FCNP in segment number one of the candidate list area, and set $\hat{b} = \hat{t} = 1$, $\hat{n} = 2$.

A.2.4.2 Selecting at Step 1. Because the candidate list chain is in sequence by cost bound, the candidate problem selected for investigation is always the one stored in segment $\hat{b}$.

A.2.4.3 Testing for Fathoming at Step 4a. The test of whether a new incumbent solution is as low as any member of the candidate list is merely to check if $\beta(\hat{t}) \leq \nu^*$.

A.2.4.4 Eliminating Candidates at Step 4b. If at least one candidate can be eliminated because its cost bound is as large as the new incumbent solution value, the following procedure is used to update the candidate list area chains.

Step i. Let $k = \hat{b}$ (i.e. the segment number of the problem which just produced a new incumbent solution). Then, go to Step ii.

Step ii. Advance up the candidate chain by letting $k' = k$, $k = \psi(k')$. If $\beta(k) \geq v^*$, the desired break point in the chain has been reached, so proceed to Step iii. Otherwise, repeat Step ii.

Step iii. Transfer all segments above k' in the candidate chain to the available segments chain by setting $\psi(\hat{t}) = \hat{n}$ and $\hat{n} = k$. Next, make k' the new end of the candidate chain by setting $\hat{t} = k'$. Then stop; the chain update is completed.

A.2.4.5 Adding Candidates at Step 7. The steps for replacing the current candidate problem with two new candidates are as follows:

Step i. Let $i_1 = \hat{b}$ and store information about the new candidate with the lower cost bound in segment i_1. Next, let $i_2 = \hat{n}$ and store information about the new candidate with the higher cost bound in segment i_2. Then update pointers by setting $\hat{b} = \psi(\hat{b})$, $\hat{n} = \psi(\hat{n})$, and go to Step ii.

Step ii. Let $k = \hat{b}$. If the new $\beta(i_1) \leq \beta(\hat{b})$ insert i_1 at the bottom of the candidate chain by setting $\psi(i_1) = \hat{b}$, $\hat{b} = i_1$, and go to Step v. Otherwise, go to Step iii.

Step iii. Advance up the candidate chain by setting $k' = k$, $k = \psi(k')$. If $\beta(k) \geq \beta(i_1)$, go to Step iv and insert i_1 in the candidate chain. Otherwise, repeat Step iii.

Step iv. Insert i_1 in the candidate chain by setting $\psi(i_1) = k$, $\psi(k') = i_1$. Then let $k' = i_1$ and go to Step vi.

Step v. If the new $\beta(i_2) \leq \beta(\hat{b})$ insert i_2 at the bottom of the candidate chain by setting $\psi(i_2) = \hat{b}$, $\hat{b} = i_2$, and stop; the procedure is complete. Otherwise, go to Step vi.

Step vi. If $\beta(k) \geq \beta(i_2)$ go to Step vii and insert i_2 in the candidate chain. Otherwise, advance up the candidate chain by setting $k' = k$, $k = \psi(k')$ and repeat Step vi.

Step vii. Insert i_2 in the candidate chain by setting $\psi(i_2) = k$, $\psi(k') = i_2$. Then, stop; the procedure is complete.

A.2.4.6 Fathoming at Step 8. Removal of a candidate from future consideration requires removing $\hat{b}$ from the candidate chain and placing it in the available segment chain by setting $k = \hat{b}$, $\hat{b} = \psi(\hat{b})$, $\psi(k) = \hat{n}$, and $\hat{n} = k$.

A.2.5 Solving Two-Row Problems EOP(i,j)

Though considerably more complicated than the one-row case, two-row linear programs derived from the penalty problems EOP(i,j) can be solved by an extension of the minimum ratio approach for one-row linear knapsack problems when the constraint matrices are totally unimodular. Recall from Corollary 4.3.2.2 that total unimodularity implies only certain combinations of columns can appear in two-row group-related problems. Either columns of the form $(+1,-1)^T$ and $(-1,+1)^T$, or columns of the form $(+1,+1)^T$ and $(-1,-1)^T$ may be present in a problem, but not both. Thus three possible cases can occur. Either no columns have two non-zero entries, or columns with two non-zero entries have entries of like sign, or columns with two non-zero entries have entries of opposing sign.

Suppose functions $\epsilon_{\bullet}^{\bullet}(i,j,z)$ are defined from two-row problems EOP(i,j) so that for example

$$\epsilon_{-}^{+}(i,j,z) = (z)\begin{Bmatrix}\text{minimum cost coefficient in the} \\ \text{group objective function for a} \\ \text{column having a +1 in row i and} \\ \text{a -1 in row j of EOP(i,j).}\end{Bmatrix}.$$

Then the value of a solution to EOP(i,j) and the associated penalties $\rho_D^D(i,j)$, $\rho_U^D(i,j)$, $\rho_D^U(i,j)$, and $\rho_U^U(i,j)$ can be obtained for the three cases by the expressions given below.

A.2.5.1 Case of No Columns with Two Non-Zero Coefficients.

$$\nu\left(EOP(i,j)\right) = \nu(\overline{FCNP}) + \min\{\rho_D^D(i,j),\rho_U^D(i,j),\rho_D^U(i,j),\rho_U^U(i,j)\}$$

where

$$\rho_D^D(i,j) = \epsilon_0^+(i,j,\bar{y}_i) + \epsilon_+^0(i,j,\bar{y}_j)$$

$$\rho_U^D(i,j) = \epsilon_0^+(i,j,\bar{y}_i) + \epsilon_-^0(i,j,u_{1j}-\bar{y}_j)$$

$$\rho_D^U(i,j) = \epsilon_0^-(i,j,u_{1i}-\bar{y}_i) + \epsilon_+^0(i,j,\bar{y}_j)$$

$$\rho_U^U(i,j) = \epsilon_0^-(i,j,u_{1i}-\bar{y}_i) + \epsilon_-^0(i,j,u_{1j}-\bar{y}_j)$$

A.2.5.2 Case of Columns with Coefficients of Like Sign.

$$\nu\left(EOP(i,j)\right) = \nu(\overline{FCNP}) + \min\{\rho_D^D(i,j),\rho_U^D(i,j),\rho_D^U(i,j),\rho_U^U(i,j)\}$$

where

$$\rho_D^D(i,j) = \min\begin{cases}\epsilon_0^+(i,j,\bar{y}_i) + \epsilon_+^0(i,j,\bar{y}_j)\\ \epsilon_+^+(i,j,\min\{\bar{y}_i,\bar{y}_j\}) + \epsilon_0^+(i,j,\max\{0,\bar{y}_i-\bar{y}_j\}) \\ \qquad + \epsilon_+^0(i,j,\max\{0,\bar{y}_j-\bar{y}_i\})\\ \epsilon_+^+(i,j,\max\{\bar{y}_i,\bar{y}_j\}) + \epsilon_0^-(i,j,\max\{0,\bar{y}_j-\bar{y}_i\}) \\ \qquad + \epsilon_-^0(i,j,\max\{0,\bar{y}_i-\bar{y}_j\})\end{cases}$$

$$\rho_U^D(i,j) = \min\begin{cases}\epsilon_0^+(i,j,\bar{y}_i) + \epsilon_-^0(i,j,u_{1j}-\bar{y}_j)\\ \epsilon_+^+(i,j,\bar{y}_i) + \epsilon_-^0(i,j,u_{1j}-\bar{y}_j+\bar{y}_i)\\ \epsilon_-^-(i,j,u_{1j}-\bar{y}_j) + \epsilon_0^+(i,j,u_{1j}-\bar{y}_j+\bar{y}_i)\end{cases}$$

$$\rho_D^U(i,j) = \min \begin{cases} \epsilon_0^-(i,j,u_{1i}-\bar{y}_i) + \epsilon_+^0(\bar{y}_j) \\ \epsilon_+^+(i,j,\bar{y}_j) + \epsilon_0^-(i,j,u_{1i}-\bar{y}_i+\bar{y}_j) \\ \epsilon_-^-(i,j,u_{1i}-\bar{y}_i) + \epsilon_+^0(i,j,u_{1i}-\bar{y}_i+\bar{y}_j) \end{cases}$$

$$\rho_U^U(i,j) = \min \begin{cases} \epsilon_0^-(i,j,u_{1i}-\bar{y}_i) + \epsilon_-^0(u_{1j}-\bar{y}_j) \\ \epsilon_-^-(i,j,\min\{u_{1i}-\bar{y}_i,u_{1j}-\bar{y}_j\}) + \epsilon_0^-(i,j,\max\{0,u_{1i}-\bar{y}_i-u_{1j}+\bar{y}_j\}) \\ \qquad + \epsilon_-^0(i,j,\max\{0,u_{1j}-\bar{y}_j-u_{1i}-\bar{y}_i\}) \\ \epsilon_-^-(i,j,\max\{u_{1i}-\bar{y}_i,y_{1j}-\bar{y}_j\}) + \epsilon_0^+\{i,j,\max\{0,u_{1j}-\bar{y}_j-u_{1i}+\bar{y}_i\}) \\ \qquad + \epsilon_+^0(i,j,\max\{0,u_{1i}-\bar{y}_i-u_{1j}+\bar{y}_j\}) \end{cases}$$

A.2.5.3 Case of Columns with Coefficients of Opposite Signs.

$$\nu\left(EOP(i,j)\right) = \nu(\overline{FCNP}) + \min\{\rho_D^D(i,j),\rho_U^D(i,j),\rho_D^U(i,j),\rho_U^U(i,j)\}$$

where

$$\rho_D^D(i,j) = \min \begin{cases} \epsilon_0^+(i,j,\bar{y}_i) + \epsilon_+^0(i,j,\bar{y}_j) \\ \epsilon_-^+(i,j,\bar{y}_i) + \epsilon_+^0(i,j,\bar{y}_i+\bar{y}_j) \\ \epsilon_+^-(i,j,\bar{y}_j) + \epsilon_0^+(i,j,\bar{y}_i+\bar{y}_j) \end{cases}$$

$$\rho_U^D(i,j) = \min \begin{cases} \epsilon_0^+(i,j,\bar{y}_i) + \epsilon_-^0(i,j,u_{1j}-\bar{y}_j) \\ \epsilon_-^+(i,j,\min\{\bar{y}_i,y_{1j}-\bar{y}_j\}) + \epsilon_0^+(i,j,\max\{0,\bar{y}_i-u_{1j}+\bar{y}_j\}) \\ \qquad + \epsilon_-^0(i,j,\max\{0,u_{1j}-\bar{y}_j-\bar{y}_i\}) \\ \epsilon_-^+(i,j,\max\{\bar{y}_i,u_{1j}-\bar{y}_j\}) + \epsilon_0^-(i,j,\max\{0,u_{1j}-\bar{y}_j-\bar{y}_i\}) \\ \qquad + \epsilon_+^0(i,j,\max\{0,\bar{y}_i-u_{1j}+\bar{y}_j\}) \end{cases}$$

$$\rho_D^U(i,j) = \min \begin{cases} \epsilon_0^-(i,j,u_{1i}-\bar{y}_i) + \epsilon_+^0(\bar{y}_j) \\ \epsilon_+^-(i,j,\min\{u_{1i}-\bar{y}_i,\bar{y}_j\}) + \epsilon_0^-(i,j,\max\{0,u_{1i}-\bar{y}_i-\bar{y}_j\}) \\ \qquad + \epsilon_+^0(i,j,\max\{0,\bar{y}_j-u_{1i}+\bar{y}_i\}) \\ \epsilon_+^-(i,j,\max\{u_{1i}-\bar{y}_i,\bar{y}_j\}) + \epsilon_0^+(i,j,\max\{0,\bar{y}_j-u_{1i}+\bar{y}_i\}) \\ \qquad + \epsilon_-^0(i,j,\max\{0,u_{1i}-\bar{y}_i-\bar{y}_j\}) \end{cases}$$

$$\rho_U^U(i,j) = \min \begin{cases} \epsilon_0^-(i,j,u_{1i}-\bar{y}_i) + \epsilon_-^0(i,j,u_{1j}-\bar{y}_j) \\ \epsilon_-^+(i,j,u_{1j}-\bar{y}_j) + \epsilon_0^-(i,j,u_{1i}-\bar{y}_i+u_{1j}-\bar{y}_j) \\ \epsilon_+^-(i,j,u_{1i}-\bar{y}_i) + \epsilon_-^0(i,j,u_{1i}-\bar{y}_i+u_{1j}-\bar{y}_j) \end{cases}$$

A.3 Computational Notes

As a by-product of the experiments reported in Chapter VII, information about the computational effectiveness of some of the above techniques was compiled. The following sections briefly analyze this information.

A.3.1 Effect of Macro-Node Analysis

In Step 5d of Algorithm A.1 all arcs were eliminated from penalty problems if they connected two nodes in the same macro-node. Results for the three main experiments reported in Chapter VII show that this test eliminated from consideration an average of 64 per cent of the variables in group problems for general FCNP's, 39 per cent of the variables for fixed charge transportation problems, and 33 per cent of the variables for warehouse location problems. Thus a considerable reduction in the computational effort required to solve one and two-row penalty problems was obtained through macro-node analysis.

A.3.2 Efficiency of Restart Procedure

The restart procedure described in Section A.2.3 proceeds by starting the solution of each problem RNP_c from the optimal solution to RNP. Results from the principal experiments described in Chapter VII showed a considerable savings in computation resulted from this procedure. RNP's associated with general FCNP's required an average of 223 pivots to reach optimality, yet the restarted problems RNP_c required an average of only 26 pivots to reach optimality. Similarly, RNP's associated with fixed charge transportation problems required an average of 114 pivots, while corresponding RNP_c's solved in an average of 16 pivots, and RNP's associated with warehouse location problems required an average of 291 pivots, while corresponding RNP_c's solved in an average of 25 pivots.

A.3.3 Size of the Candidate List

It is often suggested that the candidate list approach detailed in Section A.2.4 requires too much computer storage to be feasible for large problems. If a very large number of candidate problems had to be simultaneously stored, this criticism would be valid. However, results for the main experiments described in Chapter VII show that no more than 26 problems were ever in a candidate list for general FCNP's, and that similar maximums for fixed charge transportation problems and warehouse location problems were 733 and 510, respectively. Thus it appears that candidate lists required for relatively large FCNP's are well within the storage capabilities of modern scientific computers.

APPENDIX B

GROUP PROBLEMS WITH GENERAL UPPER AND LOWER BOUNDS

The *linear mixed-integer program with general upper and lower bounds* can be formulated

$$\min \quad c_1^T x_1 + c_2^T x_2 \tag{B-1}$$

$$\text{s.t.} \quad A_1 x_1 + A_2 x_2 = b \tag{B-2}$$

(MIP)

$$u_1 \geq x_1 \geq \ell_1 \tag{B-3}$$

$$u_2 \geq x_2 \geq \ell_2 \tag{B-4}$$

$$x_1 \equiv 0 \bmod w, \tag{B-5}$$

where c_1, u_1, ℓ_1, w and x_1 are n_1-vectors, c_2, u_2, ℓ_2 and x_2 are n_2-vectors, b is an m-vector, A_1 is an m by n_1-matrix, A_2 is an m by n_2 matrix, and $\ell_1, u_1 \equiv 0 \bmod w$. Gomory and Johnson [40,41,42,43,44,65] have derived a number of important properties for a group problem associated with MIP in the case of $w = 1$, u_1 and u_2 infinite, ℓ_1 and ℓ_2 zero. In this Appendix, their derivation will be extended to the case of any MIP. All notation is as defined in Section 1.4 unless otherwise noted.

For a mixed-integer program in non-negative variables Gomory and Johnson derive the group problem by the following steps:

1. Solve the relaxation of the problem obtained by ignoring integrality constraints.

2. Represent the basic variables in the optimal solution of the relaxed problem in terms of the nonbasic variables by using the optimal *standard* Simplex tableau.

3. Use this representation to restate the objective function entirely in terms of the nonbasic variables.

4. Re-impose the integrality constraints directly on the nonbasic variables, and indirectly on the basic variables through their representation in terms of the nonbasic variables.

5. Relax non-negativity requirements on all basic variables.

Suppose now that the continuous relaxation of MIP, i.e. $\overline{\text{MIP}}$, is solved by a *bounded* Simplex procedure. In order to derive the group problem for MIP it is necessary to find a procedure like the one above to obtain a problem which minimizes a revised objective function subject to equivalent forms of the congruence constraints (B-5), and nonnegativity on variables nonbasic in the optimal $\overline{\text{MIP}}$ solution.

In terms of the optimal bounded Simplex tableau for $\overline{\text{MIP}}$, the variables considered basic in that optimal solution, i.e. x_1^B and x_2^B, can be represented

$$\begin{pmatrix} x_1^B \\ \\ x_2^B \end{pmatrix} = \bar{b} - \bar{A}_1^L x_1^L - \bar{A}_1^U x_1^U - \bar{A}_2^L x_2^L - \bar{A}_2^U \bar{x}_2^U . \tag{B-6}$$

Thus, if the dimension of x_1^B is k^B, an equivalent form of the part of (B-5) referring to x_1^B is given by

$$[\bar{A}_1^L x_1^L + \bar{A}_1^U x_1^U + \bar{A}_2^L x_2^L + \bar{A}_2^U x_2^U]^1_{k^B} \equiv [\bar{b}]^1_{k^B} \bmod w^B. \qquad \text{(B-7)}$$

Also, the remainder of (B-5) is given by

$$x_1^L \equiv 0 \bmod w^L \quad \text{and} \quad x_1^U \equiv 0 \bmod w^U. \qquad \text{(B-8)}$$

Recall that applying the bounded Simplex procedure to a given problem is equivalent to applying the ordinary Simplex procedure to the expanded problem obtained by treating the bound constraints as part of the main constraints, letting the decision variables be unrestricted in sign, and adding two non-negative slack variables for each decision variable.[1] One slack measures the absolute difference between the variable and its upper bound, and the other measures the absolute difference between the variable and its lower bound. In this expanded problem, the decision variables are always basic, and the slacks are basis or nonbasic according to the following rules:

1. If the decision variable was basic in the bounded Simplex solution, both the bound slack associated with its lower bound and the bound slack associated with its upper bound are basic in the full Simplex solution.

2. If the decision variable was nonbasic lower-bounded in the bounded Simplex solution, the bound slack associated with its lower bound is nonbasic at value 0, and the bound slack associated with its upper bound is basic in the full Simplex solution.

3. If the decision variable was nonbasic upper-bounded in the bounded Simplex solution, the bound slack

[1] See for example Hadley [48, Section 11.7].

associated with its upper bound is nonbasic at value 0 and the bound slack associated with its lower bound is basic in the full Simplex solution.

Let the *nonbasic* bound slacks for such cases in $\overline{MIP}$ be defined as follows:

$$\Delta x_1^L = x_1^L - \ell_1^L \qquad \Delta x_1^U = u_1^U - x_1^U \tag{B-9}$$

$$\Delta x_2^L = x_2^L - \ell_2^L \qquad \Delta x_2^U = u_2^U - x_2^U. \tag{B-10}$$

Then an equivalent form of (B-7) and (B-8) is given in terms of these non-negative nonbasic variables by

$$[\bar{A}_1^L(\Delta x_1^L+\ell_1^L) + \bar{A}_1^U(u_1^U-\Delta x_1^U) + \bar{A}_2^L(\Delta x_2^L+\ell_2^L) + \bar{A}_2^U(u_2^U-\Delta x_2^U]^1_{k^B} \equiv [\bar{b}]^1_{k^B} \bmod w^B \tag{B-11}$$

$$(\Delta x_1^L+\ell_1^L) \equiv 0 \bmod w^L \quad \text{and} \quad (u_1^U-\Delta x_1^U) \equiv 0 \bmod w^U. \tag{B-12}$$

Moving all constant terms to the right-hand side of (B-11) yields

$$[\bar{A}_1^L\Delta x_1^L - \bar{A}_1^U\Delta x_1^U + \bar{A}_2^L\Delta x_2^L - \bar{A}_2^U\Delta x_2^U]^1_{k^B} \equiv \bar{x}_1^B \bmod w^B. \tag{B-13}$$

Similarly, moving constant terms to the right-hand side of (B-12) and recalling that $\ell_1^L \equiv 0 \bmod w^L$ and $u_1^U \equiv 0 \bmod w^U$ yields

$$\Delta x_1^L \equiv 0 \bmod w^L \qquad \Delta x_1^U \equiv 0 \bmod w^U. \tag{B-14}$$

Thus the congruence constraints (B-5) can be equivalently represented in terms of the non-negative nonbasic variables defined in (B-9) and (B-10) by (B-13) and (B-14).

Proceeding in an exactly similar way, the objective function of MIP is given in terms of the bounded Simplex nonbasic variables by

$$\min \; (\bar{c}_1^L)^T x_1^L + (\bar{c}_1^U)^T x_1^U + (\bar{c}_2^L)^T x_2^L + (\bar{c}_2^U)^T x_2^U + \bar{z} \qquad \text{(B-15)}$$

where $\bar{z}$ is the adjusted right-hand-side of the cost row in the optimal bounded Simplex tableau for $\overline{\text{MIP}}$. Substituting the definitions (B-9) and (B-10) and simplifying yields the equivalent form

$$\min \; (\bar{c}_1^L)^T \Delta x_1^L - (\bar{c}_1^U)^T \Delta x_1^U + (\bar{c}_2^L)^T \Delta x_2^L - (\bar{c}_2^U)^T \Delta x_2^U + \nu(\overline{\text{MIP}}). \qquad \text{(B-16)}$$

Thus both the congruence constraints and the objective function of MIP can be restated in terms of non-negative, nonbasic variables by using the nonbasic variables for the standard Simplex method in place of those for the bounded Simplex method. The remaining problem in reproducing Gomory and Johnson's derivation is to identify which constraints are equivalent to non-negativity requirements on basic variables in the solution of the ordinary Simplex method. Such constraints must be relaxed in formulating the group problem.

By the rules given above, the basic variables for the full Simplex method include the two bound slacks for any variable considered basic in the bounded Simplex solution to MIP, the upper bound slacks for variables considered lower-bounded in the bounded Simplex solution, and the lower bound slacks for any variables considered upper-bounded in the bounded Simplex solution. Thus non-negativity of the basic variables in the full solution is equivalent to the original constraints

$$u_1^B \geq x_1^B \geq \ell_1^B \qquad u_2^B \geq x_2^B \geq \ell_2^B$$

$$u_1^L \geq x_1^L \qquad u_2^L \geq x_2^L$$

$$x_1^U \geq \ell_1^U \qquad x_2^U \geq \ell_2^L \ .$$

Expressing these constraints in terms of the variables defined in (B-9) and (B-10) yields

$$\begin{pmatrix} u_1^B \\ u_2^B \end{pmatrix} \geq \begin{pmatrix} \bar{x}_1^B \\ \bar{x}_2^B \end{pmatrix} - \bar{A}_1^L \Delta x_1^L + \bar{A}_1^U \Delta x_1^U - \bar{A}_2^L \Delta x_2^L + \bar{A}_2^U \Delta x_2^U \geq \begin{pmatrix} \ell_1^B \\ \ell_2^B \end{pmatrix} \qquad \text{(B-17)}$$

$$(u_1^L - \ell_1^L) \geq \Delta x_1^L,\ (u_1^U - \ell_1^U) \geq \Delta x_1^U,\ (u_2^L - \ell_2^L) \geq \Delta x_2^L,\ (u_2^U - \ell_2^U) \geq \Delta x_2^U. \qquad \text{(B-18)}$$

Thus, the constraints which must be relaxed in the group problem for MIP include upper and lower limits on the variables considered basic in the bounded Simplex solution, as well as upper limits on the nonbasic slack variables defined in (B-9) and (B-10).

In summary then, the group problem for MIP is given by

$$\text{(GP(MIP))} \quad \begin{aligned} \min \quad & (\bar{c}_1^L)^T \Delta x_1^L - (\bar{c}_1^U)^T \Delta x_1^U + (\bar{c}_2^L)^T \Delta x_2^L - (\bar{c}_2^U)^T \Delta x_2^U + \nu(\overline{MIP}) \\ \text{s.t.} \quad & [\bar{A}_1^L \Delta x_1^L - \bar{A}_1^U \Delta x_1^U + \bar{A}_2^L \Delta x_2^L - \bar{A}_2^U \Delta x_2^U]_{k^B}^1 \equiv \bar{x}_1^B \bmod w^B \\ & \Delta x_1^L \equiv 0 \bmod w^L, \quad \Delta x_1^U \equiv 0 \bmod w^U \\ & \Delta x_1^L, \Delta x_1^U, \Delta x_2^L, \Delta x_2^U \geq 0. \end{aligned}$$

GP(MIP) can be viewed as the problem of finding the minimum cost perturbation of the optimal values of variables considered nonbasic in the bounded Simplex solution of $\overline{MIP}$ which satisfies all congruence constraints of MIP while conforming to lower bounds on variables which were lower-bounded in the $\overline{MIP}$ solution and to upper bounds on variables which were upper-bounded in the $\overline{MIP}$ solution. If an optimal perturbation for GP(MIP) also satisfies (B-17) and (B-18), then the implied values of x_1 and x_2 provide an optimal solution to MIP.

BIBLIOGRAPHY

1. Balas, E., "An Infeasibility Pricing Decomposition Method for Linear Programs," *Operations Research,* 14, 847-873, 1966.

2. Balas, E., "Duality in Discrete Programming," Technical Report No. 67-5, Carnegie-Mellon University, 1967.

3. Balas, E., "A Note on the Asymptotic Theory of Integer Programming and the 0-1 Case," Management Sciences Research Report No. 249, Carnegie-Mellon University, 1971.

4. Balinski, M. L., "Fixed Cost Transportation Problems," *NRLQ,* 8, 41-54, 1961.

5. Balinski, M. L., "Integer Programming: Methods, Uses, Computation," *Management Science,* 12, 253-313, 1965.

6. Balinski, M. L., "On Recent Developments in Integer Programming," *Proceedings of the International Symposium on Mathematical Programming,* Princeton University, 1967.

7. Balinski, M. L. and K. Spielberg, "Methods for Integer Programming: Algebraic, Combinatorial, and Enumerative," *Progress in Operations Research,* Volume 3, J. Aronofsky editor, John Wiley and Sons, 1968.

8. Bazarra, M. S., "An Infeasibility Pricing Algorithm for the Multicommodity Minimum Cost Flow Problem," working paper in the School of Industrial and Systems Engineering, Georgia Institute of Technology, 1973.

9. Bellmore, M. and H. D. Ratliff, "Optimal Defense of Multi-Commodity Network," *Management Science,* 18, B147-B185, 1971.

10. Benders, J. F., "Partitioning Procedures for Solving Mixed-Variables Programming Problems," *Numerische Mathematik,* 4, 238-252, 1962.

11. Brooks, R. and A. M. Geoffrion, "Finding Everett's Lagrange Multipliers by Linear Programming," *Operations Research,* 14, 1149-1153, 1966.

12. Bulfin, R. L., *An Algorithm for the Plant Location Problem,* masters thesis in the School of Industrial and Systems Engineering, Georgia Institute of Technology, 1972.

13. Bulfin, R. L. and V. E. Unger, "Computational Experience with an Algorithm for the Locked Box Problem," *Proceedings of 1973 Conference of the ACM,* pp. 16-19, 1973.

14. Cooper, L. and C. Drebes, "An Approximate Solution Method for the Fixed Charge Problem," *NRLQ,* 14, 101-113, 1967.

15. Cooper, L. and A. M. Olson, "Random Perturbations and the MI-MII Heuristics for the Fixed-Charge Problem," Report No. COO-1493-7, Department of Applied Mathematics and Computer Science, Washington University, 1968.

16. Dantzig, G. B., *Linear Programming and Extensions,* Princeton University Press, 1962.

17. Dantzig, G. B., and A. F. Veinott, Jr., "Integral Extreme Points," *SIAM Review,* 10, 371-372, 1968.

18. Dantzig, G. B. and P. Wolfe, "The Decomposition Algorithm for Linear Programs," *Econometrica,* 29, 767-778, 1961.

19. Denzler, D. R., "An Approximative Algorithm for the Fixed Charge Problem," *NRLQ,* 16, 411-416, 1969.

20. Dwyer, P. S., "Use of Completely Reduced Matrices in Solving Transportation Problems with Fixed Charges," *NRLQ,* 13, 239-313, 1966.

21. Efroymson, M. A. and T. L. Ray, "A Branch-Bound Algorithm for Plant Location," *Operations Research,* 14, 361-368, 1966.

22. Ellwein, L. B., *Fixed Charge Location-Allocation Problems with Capacity and Configuration Constraints,* Technical Report No. 70-2, Department of Industrial Engineering, Stanford University, 1970.

23. Everett, H., "Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources," *Operations Research,* 11, 399-471, 1963.

24. Fisher, M. L., "Optimal Solution of Scheduling Problems Using Generalized Lagrange Multipliers and Relaxation," Report 720, Center for Mathematical Studies in Business and Economics, University of Chicago, 1971.

25. Fisher, M. L. and J. F. Shapiro, "Constructive Duality in Integer Programming," working paper OR-008-72, Operations Research Center, Massachusetts Institute of Technology, 1972.

26. Fisher, M. L. and J. F. Shapiro, "Constructive Duality in Discrete Optimization," working paper 625-72, Alfred P. Sloan School of Management, Massachusetts Institute of Technology, 1972.

27. Fisher, M. L. and L. Schrage, "Using Lagrange Multipliers to Schedule Elective Hospital Admissions," University of Chicago Report, 1972.

28. Frank, R. S., *On the Fixed Charge Hitchcock Transportation Problem,* dissertation, The Johns Hopkins University, 1972.

29. Geoffrion, A. M., "An Improved Implicit Enumeration Approach for Integer Programming," *Operations Research,* 17, 437-454, 1969.

30. Geoffrion, A. M., "Duality in Nonlinear Programming: A Simplified Applications-Oriented Development," *SIAM REVIEW*, 13, 1-37, 1971.

31. Geoffrion, A. M., "Lagrangean Relaxation and Its Uses in Integer Programming," working paper no. 195, Western Management Science Institute, University of California-Los Angeles, 1972.

32. Geoffrion, A. M. and R. E. Marsten, "Integer Programming Algorithms: A Survey," *Management Science,* 18, 465-491, 1972.

33. Gilmore, P. C. and R. E. Gomory, "The Theory and Computation of Knapsack Functions," *Operations Research,* 14, 1045-1074, 1966.

34. Glover, F., "Surrogate Constraints," *Operations Research,* 13, 879-919, 1965.

35. Glover, F., "Faces of an Integer Polyhedron for Cyclic Groups--Part I," Department of Statistics and Operations Research, University of Texas, 1968.

36. Glover, F., D. Klingman and D. Kearny, "The Augmented Predecessor Index Method for Locating Stepping Stone Paths and Assigning Dual Prices in Distribution Problems," *Transportation Science,* 6, 171-179, 1972.

37. Gomory, R. E., "An Algorithm for the Mixed Integer Problem," RAND Report P-1885, 1960.

38. Gomory, R. E., "An Algorithm for Integer Solutions to Linear Programs," in *Recent Advances in Mathematical Programming,* R. L. Graves and P. Wolfe editors, McGraw-Hill, 1963.

39. Gomory, R. E., "All-Integer Programming Algorithm," in *Industrial Scheduling,* J. F. Muth and E. L. Thompson editors, Prentice-Hall, 1963.

40. Gomory, R. E., "On the Relation Between Integer and Non-Integer Solutions to Linear Programs," *Proceedings of the National Academy of Sciences,* 53, 260-265, 1965.

41. Gomory, R. E., "Faces of an Integer Polyhedron," *Proceedings of the National Academy of Sciences,* 57, 16-18, 1967.

42. Gomory, R. E., "Some Polyhedra Related to Combinatorial Problems," *Linear Algebra and Its Applications,* 2, 451-558, 1969.

43. Gomory, R. E. and E. L. Johnson, "Some Continuous Functions Related to Corner Polyhedra," *Mathematical Programming,* 3, 23-85, 1972.

44. Gomory, R. E. and E. L. Johnson, "Some Continuous Functions Related to Corner Polyhedra-II," *Mathematical Programming,* 3, 359-389, 1972.

45. Gorry, G. A. and J. F. Shapiro, "An Adaptive Group Theoretic Algorithm for Integer Programming Problems," *Management Science,* 17, 285-306, 1971.

46. Gray, P., "Mixed Integer Programming Algorithms for Site Selection and Other Fixed-Charge Problems Having Capacity Constraints," Technical Report No. 101, Department of Operations Research and Statistics, Stanford University, 1967.

47. Gray, P., "Exact Solution of the Fixed-Charge Transportation Problem," *Operations Research,* 19, 1529-1538, 1971.

48. Hadley, G., *Linear Programming,* Addison-Wesley, 1962.

49. Haldi, J. and L. M. Isaacson, "A Computer Code for Integer Solutions to Linear Problems," *Operations Research,* 13, 945-959, 1965.

50. Hefley, G. L., "Group Programming Decomposition in Integer Programming," THEMIS Report No. 52, Systems Research Center, Industrial and Systems Engineering Department, University of Florida, 1971.

51. Hefley, G. L. and B. O. Bair, "Fixed Charge Transportation Problem: A Survey," presented to the 43rd National Conference of the Operations Research Society of America, 1973.

52. Hefley, G. L. and M. E. Thomas, "Decomposition of the Group Problem in Integer Programming," Report No. IME72-101, University of Iowa, 1973.

53. Held, M. and R. M. Karp, "The Traveling Salesman Problem and Minimum Spanning Trees," *Operations Research,* 18, 1138-1162, 1970.

54. Held, M. and R. M. Karp, "The Traveling Salesman Problem and Minimum Spanning Trees: Part II," *Mathematical Programming,* 1, 6-25, 1971.

55. Heller, I., "On Unimodular Sets of Vectors," in *Recent Advances in Mathematical Programming,* R. L. Graves and P. Wolfe editors, McGraw-Hill, 1963.

56. Heller, I. and C. B. Tompkins, "An Extension of a Theorem of Dantzig's," in *Linear Inequalities and Related Systems,* H. W. Kuhn and A. W. Tucker editors, Princeton University Press, 1956.

57. Hillier, F. S., "An Optimal Bound and Scan Algorithm for Integer Linear Programming," Technical Report No. 3, Department of Industrial Engineering, Stanford University, 1966.

58. Hirsch, W. M. and G. B. Dantzig, "The Fixed Charge Problem," *NRLQ,* 15, 413-424, 1968.

59. Hirsch, W. M. and A. J. Hoffman, "Extreme Varieties, Concave Functions and the Fixed Charge Problem," *Communications on Pure and Applied Mathematics,* 14, 355-369, 1961.

60. Hoffman, A. J. and J. B. Kruskal, "Integral Boundary Points of Convex Polyhedra," in *Linear Inequalities and Related Systems,* H. W. Kuhn and A. W. Tucker editors, Princeton University Press, 1956.

61. Hu, T. C., "Decomposition Algorithm for Shortest Paths in a Network," *Operations Research,* 16, 91-102, 1968.

62. Hu, T. C., *Integer Programming and Network Flows,* Addison-Wesley, 1969.

63. Johnson, E. L., "Programming in Networks and Graphs," ORC Report 65-1, University of California-Berkeley, 1965.

64. Johnson, E. L., "Networks and Basic Solutions," *Operations Research,* 14, 619-623, 1966.

65. Johnson, E. L., "Cyclic Groups, Cutting Planes, and Shortest Paths," presented at the Mathematical Programming Seminar, Mathematics Research Center, University of Wisconsin-Madison, 1972.

66. Johnson, E. L. and K. Spielberg, "Inequalities in Branch and Bound Programming," IBM Research Report RC-3649, 1971.

67. Jones, A. P. and R. M. Soland, "A Branch-and-Bound Algorithm for Multi-Level Fixed-Charge Problems," *Management Science,* 16, 67-76, 1969.

68. Kennington, J. L., *Fixed-Charge Transportation Problem: A Group Theoretic Approach,* dissertation in the School of Industrial and Systems Engineering, Georgia Institute of Technology, 1973.

69. Kennington, J. L., and R. W. Langley, "Transportation Problem: A Primal Procedure," presented to the 43rd National Conference of the Operations Research Society of America, 1973.

70. Khumawala, B. M., "An Efficient Branch and Bound Algorithm for the Warehouse Location Problem," *Management Science,* 18, B718-B731, 1972.

71. Khumawala, B. M. and D. C. Whybark, "A Survey of Facility Location Methods," paper No. 350, Krannert Graduate School of Industrial Administration, Purdue University, 1972.

72. Klingman, D., A. Napier, and J. Stutz, "NETGEN--A Program for Generating Large Scale (Un) Capacitated Assignment, Transportation, and Minimum Cost Flow Problems," Report CS 109, Center for Cybernetic Studies, University of Texas-Austin, 1973.

73. Kuehn, A. A. and J. J. Hamburger, "A Heuristic Program for Locating Warehouses," *Management Science,* 9, 1963.

74. Kuhn, H. W. and W. J. Baumol, "An Approximate Algorithm for the Fixed-Charge Transportation Problem," *NRLQ,* 9, 1-15, 1962.

75. Langley, R. W., *Continuous and Integer Generalized Flow Problems,* dissertation in the School of Industrial and Systems Engineering, Georgia Institute of Technology, 1973.

76. Murty, K. G., "Solving the Fixed Charge Problem by Ranking Extreme Points," *Operations Research,* 16, 268-279, 1968.

77. Murty, K. G., private communication, 1973.

78. Nemhauser, G. L. and Z. Ullman, "A Note on the Generalized Lagrange Multiplier Solution to an Integer Programming Problem," *Operations Research,* 16, 450-452, 1968.

79. Rardin, R. L. and V. E. Unger, "A Surrogate Constraint for 0-1 Mixed-Integer Programs," presented to the 43rd National Conference of the Operations Research Society of America, 1973.

80. Robers, P. and L. Cooper, "A Study of the Fixed Charge Transportation Problem," Report No. COO-1493-9, Department of Applied Mathematics and Computer Science, Washington University, 1969.

81. Saaty, T. L., *Optimization in Integers and Related Extremal Problems,* McGraw-Hill, 1970.

82. Shapiro, J. F., "Dynamic Programming Algorithms for the Integer Programming Problem--I: The Integer Programming Problem Viewed as a Knapsack Type Problem," *Operations Research,* 16, 103-121, 1968.

83. Shapiro, J. F., "Group Theoretic Algorithms for the Integer Programming Problem--II: Extension to a General Algorithm," *Operations Research,* 16, 928-947, 1968.

84. Shapiro, J. F., "Generalized Lagrange Multipliers in Integer Programming," *Operations Research,* 19, 68-76, 1971.

85. Spielberg, K., "On the Fixed-Charge Transportation Problem," *Proceedings of the 19th National Conference of the ACM,* A1.1-1 to A1.1-13, 1964.

86. Spielberg, K., "An Algorithm for the Simple Plant Location Problem with Some Side Conditions," *Operations Research,* 17, 81-111, 1969.

87. Spielberg, K., "Plant Location with Generalized Search Origin," *Management Science,* 15, 1969.

88. Steinberg, D. I., "The Fixed Charge Problem," Report No. COO-1493-15, Department of Applied Mathematics and Computer Science, Washington University, 1969.

89. Steinberg, D. I., "The Fixed Charge Problem," *NRLQ,* 17, 217-236, 1970.

90. Tomlin, J. A., "An Improved Branch-and-Bound Method for Integer Programming," *Operations Research,* 19, 1070-1075, 1971.

91. Tompkins, C. J., *Group Theoretic Structures in the Fixed Charge Transportation Problem,* dissertation in the School of Industrial and Systems Engineering, Georgia Institute of Technology, 1971.

92. Tutte, W. T., *Introduction to the Theory of Matroids,* American Elsevier, 1971.

93. Tutte, W. T., "An Algorithm for Determining Whether a Given Binary Matroid is Graphic," *Proceedings of the American Mathematical Society,* 11, 905-917, 1960.

94. Unger, V. E., "Capital Budgeting and Mixed Zero-One Integer Programming," *AIIE Transactions,* 2, 28-36, 1970.

95. Walker, W., "Adjacent Extreme Point Algorithms for the Fixed-Charge Problem," Technical Report No. 40, Center for Environmental Quality Management, Cornell University, 1968.

96. Wolsey, L. A., "Mixed Integer Programming: Discretization and the Group Theoretic Approach," Technical Report No. 42, Operations Research Center, Massachusetts Institute of Technology, 1969.

97. Zaverei, M. M. and I. T. Frisch, "On the Fixed Charge Problem," *International Journal of Control*, 16, 897-907, 1972.

VITA

Ronald Lee Rardin was born May 3, 1943 in Kansas City, Missouri, and grew up in the Kansas City area. He graduated from the Shawnee-Mission East High School in 1961 and immediately entered the University of Kansas. There Mr. Rardin was awarded a Bachelor of Arts degree "with distinction" (1965), a Master of Public Administration degree (1967), and a number of scholarships and memberships in honorary societies including Phi Beta Kappa. His undergraduate majors were mathematics and political science, and his graduate major was municipal administration.

Mr. Rardin began his professional career as an Administrative Analyst with the Research and Budget Department of the City of Ft. Worth, Texas in 1966. In 1967 he left Texas to become Research Analyst with the Stanford Research Institute in Menlo Park, California. After nearly three years with Stanford Research, Mr. Rardin returned to Texas as the Assistant Data Systems Director of Kimbell, Incorporated.

While in California Mr. Rardin attended Stanford University on a part-time basis, and in 1971 he returned to full-time graduate study in the School of Industrial and Systems Engineering of Georgia Institute of Technology. There he pursued a doctoral program in operations research while working as a Graduate Teaching and Research Assistant. He was awarded the Ph.D. degree in 1974, and immediately accepted a permanent faculty position as an Assistant Professor in the School of Industrial and Systems Engineering.

Mr. Rardin was married in September, 1969, to Miss Blanca Quiroga Prada who is a native of La Paz, Bolivia.