WORKING PAPER
ALFRED P. SLOAN SCHOOL OF MANAGEMENT

AN ADAPTIVE GROUP THEORETIC ALGORITHM FOR INTEGER PROGRAMMING PROBLEMS

318-68

G. Anthony Gorry

and

Jeremy F. Shapiro

MASSACHUSETTS
INSTITUTE OF TECHNOLOGY
50 MEMORIAL DRIVE
CAMBRIDGE, MASSACHUSETTS 02139

Preliminary Draft

AN ADAPTIVE GROUP THEORETIC ALGORITHM FOR INTEGER PROGRAMMING PROBLEMS

318-68

G. Anthony Gorry

and

Jeremy F. Shapiro

Abstract

A prototypical algorithm for solving integer programming problems is presented. The algorithm combines group theoretic methods for finding integer solutions to systems of linear equations under the control of heuristic supervisory procedures. The latter pre-structure the overall problem and guide the search for an optimal solution by organizing subproblems and selecting the appropriate analytical methods to apply to them. Here there is a decided emphasis on the diagnostic facility of the supervisor in order that the various analytic methods may be adapted to the overall problem and to the particular subproblems encountered.

Throughout the paper, the variety and flexibility of the group theoretic methods are emphasized, as well as the potential of heuristic selection and control of these methods.

Table of Contents

Appendices

1. INTRODUCTION

1.1 Introductory Remarks

The purpose of this paper is the construction of a prototypical algorithm for solving integer programming (LP) problems which integrates diverse analytical methods under the control of an "intelligent" supervisory program. The original motivation for our work was the observed anomalous behavior of different IP computer codes in solving test problems. It was seen that there is a great disparity in the performance of existing codes in solving a given problem. Moreover, the performance of a given code on a given problem can depend in a non-trivial way upon the problem definition and the problem solving strategy of the code. For example, the efficiency of Gomory's cutting plane algorithm on a given IP problem depends on the choice of cuts added to the linear programming (LP) problem at each iteration [17].

Thus, it appeared to us that some problem diagnosis was desired in order to fit the proper algorithm to a given problem, and also in order to control certain strategies of the chosen algorithm to effect faster convergence. Additional interest in this problem was derived from the analogies between IP problem diagnosis and other diagnostic problems which have been solved by computer [20,21,22]. A diagnostic model for IP problems seems to have considerable potential as a basis for a flexible, adaptive algorithm. We have also investigated the application of algebraic group theory to the problem of characterizing integer solutions to systems

of linear equations [35,36,37,38]. The plethora of algorithmic procedures suggested by this theory implies that some diagnosis is required to fit the proper procedures to a given problem.

The work presented here, then, was directed at the problem of establishing a conceptual and theoretical framework within which a flexible and adaptive IP algorithm could be developed.[1] Our investigations have convinced us that such an algorithm will combine both heuristic and analytic methods. The latter will be used in the solution of subproblems generated during the attempt to solve the given IP problem. The heuristic methods will be used by a supervisory procedure which organizes and selects subproblems and chooses the appropriate analytical methods to employ in their solution. The manner in which the supervisory procedure structures the IP problem and the way in which it exploits information gained from the solution of subproblems are but two examples of several areas of concern. Another is the diagnostic strategy employed by the supervisor to ascertain the character of a given problem, perhaps opening it up to the application of special purpose algorithms. Similarly, the algorithm should attempt to utilize data from computation on previous IP problems in diagnosing a given problem and updating program parameters.

Our goal, therefore, is to develop some of these heuristic methods as well as to formulate new analytical methods which will facilitate the

---

[1] In this paper, we use the term algorithm both to refer to the overall procedure we develop as well as any of the various sub-algorithms employed.

solution of IP problems. Some important insights have been gained into both of these areas, and they are indicated here. The overall problem, however, is as yet unsolved. Our purpose in this paper is to present our basic approach to the more general problem, and to show how our results to date indicate the potential value of the approach. Certain sections of this preliminary version are less developed than others. It is hoped that the next version of this paper will treat the neglected topics in fuller detail.

1.2 Statement of the Problem

The IP problem is written in initial canonical form

$$\min z = cw \quad (1.1a)$$

$$\text{s.t.} \quad Aw = b^0 \quad (1.1b)$$

$$w_j = 0 \text{ or } 1 \quad j \varepsilon S \quad (1.1c)$$

$$w_j = 0,1,2,\ldots \quad j \varepsilon S^c \quad (1.1d)$$

where c is an (m+n) vector of integers, A is an mx(m+n) matrix of integers, b^0 is an m-vector of integers. We assume A is originally of the form A = (I,A') where I is an mxm identity matrix. The analysis below remains valid even if A is not put into this form. We have done so in order to simplify the discussion of the group theoretic methods in Section 2. A generic column of A is denoted by a_j.

solution of IP problems. Some important [illegible] have gained from both of these areas, and they are indicative [illegible] overall problem, however, is as yet unsolved. Our purpose [illegible] to present our basic approach to the more general problem [illegible] results to date indicate the potential value of the approach [illegible] of this preliminary version [illegible]. It is hoped that the next version of [illegible] in fuller detail.

1.2 Statement of the Problem

The IP problem [illegible] as follows:

[illegible] (1.1a)

s.t. [illegible] (1.1b)

(1.1c) [illegible]

(1.1d) [illegible]

where [illegible]

integers [illegible]

Shortest [illegible]

units [illegible]

[illegible]

The first step is to solve (1.1) as the LP problem

$$\min z = cw \tag{1.2a}$$

$$\text{s.t.} \quad Aw = b^0, \tag{1.2b}$$

$$w \geq 0, \tag{1.2c}$$

$$w_j \leq 1, \quad j\varepsilon S \tag{1.2d}$$

If $S \neq \phi$, the upper bounding variant of the simplex algorithm [9] should be used to take into account implicitly the constraints (1.2d).

Let B denote the optimal basis found by this algorithm and rearrange the columns of A so that $A = (R,B)$; similarly, the vector $c = (c_R, c_B)$. Let x denote the nonbasic variables and y denote the basic variables. We use B to transform (1.1) to

$$\min z = z_0 + \bar{c}x \tag{1.3a}$$

$$\text{s.t.} \quad y = \bar{b}^0 - \bar{R}x \tag{1.3b}$$

$$x_j = 0 \text{ or } 1, j\varepsilon S \tag{1.3c}$$

$$x_j = 0,1,2,\ldots, \; j\varepsilon S^c \tag{1.3d}$$

$$y_i = 0 \text{ or } 1, \; i\varepsilon S \tag{1.3e}$$

$$y_i = 0,1,2,\ldots, \quad i\varepsilon S^c \tag{1.3f}$$

where $z_0 = c_B B^{-1} b^0$, $\bar{b}^0 = B^{-1} b^0$, $\bar{c} = c_R - c_B B^{-1} R$, and $\bar{R} = B^{-1} R$. Note that the vectors $\bar{b}^0$ and $\bar{c}$ are non-negative since B is an optimal LP basis. Moreover, $\bar{b}_i^0 \leq 1$ for $i\varepsilon S$.

A permissible correction x is one for which the constraints (1.3c) and (1.3d) hold. A feasible correction x is a permissible correction such that the resulting y_i from (1.3b) satisfy (1.3e) and (1.3f) hold. An optimal correction x is a feasible correction which solves problem (1.3). Problem (1.3) can be interpreted as: Find a feasible correction x so that the additional cost $\bar{c}x$ to the optimal LP cost z_0 is minimal. Henceforth, the constant z_0 will be omitted from the objective function (1.3a).

We will attempt to solve (1.3) by implicitly testing all feasible corrections in (1.3). The set of all permissible corrections is a tree which can be simply described in a recursive fashion. Starting at K=0, the corrections at level K are the corrections whose sum $\sum_{j=1}^{n} x_j = K$. Let x be an arbitrary correction at level K and let $j_0(x)$, $j(x)$ be defined by

$$j_0(x) = \max \{j \mid x_j > 0\} \tag{1.4a}$$

$$j(x) = \begin{cases} j_0(x) + 1 & \text{if } j_0(x) \varepsilon S \\ j_0(x) & \text{if } j_0(x) \varepsilon S^c \end{cases} \tag{1.4b}$$

The correction x is connected to the level K+1 by continuing x to $x+e_j$ for $j \geq j(x)$, where e_j is the jth unit vector in n-space.[1]

It is convenient at this point to introduce some new notation and terminology which we borrow from [11]. The best solution $\hat{x}(b^0)$ found

---

[1]Here we use the Form of an Optimal Correction Lemma from Section 3.2.

A permissible correction x is one for which [illegible] (1.3c) and (1.3d) hold. A feasible correction [illegible] correction such that the residuals [illegible] from (1.3b) satisfy (1.3a) [illegible] An optimal correction x is a feasible correction [illegible] (1.3). Problem (1.1) can be interpreted [illegible] that the additional [illegible] the original [illegible] Hence forth, the problem (1.1) [illegible] (1.3).

We will attempt [illegible] corrections [illegible] (Theorem [illegible] which can be [illegible] the corrections [illegible] Let x be [illegible] define [illegible]

[illegible] (1.4)

[illegible]

[illegible]

[illegible]

thus far at any point in the implicitly exhaustive search is called the incumbent. The incumbent cost is $\hat{z}(b^0) = \bar{c}\cdot\hat{x}(b^0)$. We say a correction x has been fathomed if it is possible either (1) to discover an optimal continuation $w \geq x$, or (2) to ascertain that no feasible continuation of x has a lower cost than the incumbent cost $\hat{z}(b^0)$. If (1) obtains, then $\hat{x}(b^0) \leftarrow w$, $\hat{z}(b^0) \leftarrow \bar{c}w$ only if $\bar{c}w < \hat{z}(b^0)$. If x is fathomed, then it is clear that the entire subtree beneath x is implicitly tested and hence x is not continued.

The procedure for searching through the tree of enumerated corrections is particularized in Section 3. For our purposes here, it suffices to recognize that any implicitly exhaustive search procedure will consider a sequence $\{x^k\}_{k=0}^{K}$ of (permissible) corrections ($x^0 = (0,\ldots,0)$) which are tested by the algorithm. Unlike Balas' linear search method of implicit enumeration [1,11], there will generally be more than one unfathomed correction being considered by the algorithm at any given time. It is true, however, that only one correction at a time is tested and hence the sequence $\{x^k\}_{k=0}^{K}$.

Consider an arbitrary x^k in the sequence and define the set F_k by

$$F_k = \{j(x^k),\ j(x^k)+1,\ldots,n\} \tag{1.5}$$

F_k is the set of free variables[1] relative to x^k. An optimal continuation of x^k is x^k+u where u is an optimal solution to the subproblem

---

[1] Strictly speaking, F_k is the set of free non-basic variables. The (original) basic variables $y_i = x_{n+i}$ are always free even when they become non-basic during analysis.

thus far at any point in the [illegible] each is called the

incumbent. The incumbent [illegible] say a watermaster

has been informed [illegible] as optimal

continuation [illegible] iteration of

a [illegible] then

[illegible] then it is

clear that [illegible]

a is not continued.

The [illegible]

is particularly [illegible]

remember [illegible] consider

a [illegible]

of [illegible]

success [illegible]

[illegible] it is

thus [illegible]

sequence [illegible]

[illegible]

$$\min \quad z(b^k) = \sum_{j\varepsilon F_k} \bar{c}_j x_j \tag{1.6a}$$

s.t.

$$y = \bar{b}^k - \sum_{j\varepsilon F_k} \bar{a}_j x_j \tag{1.6b}$$

$$x_j = 0 \text{ or } 1, \quad j\varepsilon S \cap F_k \tag{1.6c}$$

$$x_j = 0,1,2,\ldots, \quad j\varepsilon S^c \cap F_k \tag{1.6d}$$

$$y_i = 0 \text{ or } 1, \quad i\varepsilon S \tag{1.6e}$$

$$y_i = 0,1,2,\ldots, i\varepsilon S^c \tag{1.6f}$$

where $\bar{b}^k = B^{-1}b^k = B^{-1}(b^0 - Rx^k)$. Note that (1.6) reduces to (1.3) when $x^k = x^0$.

Consider a sub-sequence $\{x^{k_i}\}_{i=0}^{I}$ with the property that $x^{k_0} = 0$ and x^{k_i} is a continuation of $x^{k_{i-1}}$, $i=1,\ldots,I$; such a sub-sequence is called a path of the tree of enumerated corrections. There is a corresponding (sub) sequence $\{b^{k_i}\}_{i=0}^{I}$ of integer m-vectors in m-space, and thus the path in the tree corresponds to the path

$$p = (b^0, b^{k_1}), \; (b^{k_1}, b^{k_2}), \ldots, \; (b^{k_{I-1}}, b^{k_I})$$

in m-space. As we shall see in Section 2, the performance of the various analytical fathoming methods for each of the x^{k_i} depends upon certain geometric properties of the path p. Moreover, decisions about the procedures to be used in trying to fathom x^{k_I} are made partly on the basis of the information available from analyzing the x^{k_i}, $i=0,\ldots,I-1$.

In much of the analysis below, we will consider problem (1.6) (and others) with a variable integer m-vector $b(b=b^k)$. From the point of view of dynamic programming, b is the state vector, and it should be clear to the reader that there is an intimate connection between discrete dynamic programming and tree search. In Section 2, we discuss briefly the IP problem from the dynamic programming point of view. It is to be emphasized, however, that dynamic programming as it is generally understood is not an efficient procedure for solving the IP problem.

Let $X(b^k)$ denote the set of feasible corrections to (1.6) when $\bar{b}^k = B^{-1}b^k = B^{-1}(b^0 - Rx^k)$:

$$X(b^k) = \{x: \; x_j=0, \; j\notin F_k; \; x_j, \; j\varepsilon F_k \text{ satisfy (1.6c) and (1.6d)}; \; y_i \text{ satisfy (1.6b), (1.6e), and (1.6f)}\} \qquad (1.7)$$

Thus, corresponding to the sequence $\{x^k\}_{k=0}^{K}$ are the sequences of solution sets $\{X(b^k)\}_{k=0}^{K}$ and the optimal costs $\{z(b^k)\}_{k=0}^{K}$.

All of the analytical methods and algorithms discussed in Section 2 to be used in attempting to fathom enumerated corrections have the following common feature. The algorithms do not attempt to solve (1.6) as it is stated. Instead, the minimize $\sum_{j=1}^{n} \bar{c}_j x_j$ over some set $Y \supset X(b^k)$ which is more amenable to description and analysis. Solving the more general problem has two possible benefits. First, if

$$\bar{c}x^k + \min \{ \sum_{j \epsilon F_k} \bar{c}_j x_j : \ x \epsilon Y \} \geq \hat{z}(b^0),$$

then it is clear that x^k is fathomed. On the other hand, if we can find a u* which is optimal in $\min \{ \sum_{j \epsilon F_k} \bar{c}_j x_j : \ x \epsilon Y \}$ such that $x^k + u^*$ is a feasible correction, then x^k is fathomed. Moreover, $\hat{x}(b^0) \leftarrow x^k + u^*$, $\hat{z}(b^0) \leftarrow \bar{c}(x^i + u^*)$ since $\bar{c}(x^k + u^*) \leftarrow \hat{z}(b^0)$ by assumption.

$$\overline{cx}^h + \min \{ \sum_{i \in F_h} \bar{c}_i x_i \mid x \in Y \} ,$$

then it is clear that x^h is fathomed. On the other hand, if we can find a u^* which is optimal in $\min \{ \sum_{j \in F_h} c_j x_j \mid x \in Y \}$ such that $x^h + u^*$ is a feasible completion, then x^h is fathomed. Moreover, $x(h^0) = x^h + u^*$, $z(h^0) = \bar{c}(x^h + u^*)$ since $c(x^h + u^*) = \dots$ by assumption.

1.3 Overview

In Section 2, we will present the theoretical basis for a variety of analytic methods for solving IP problems. These methods are derived in large part from a group theoretic view of the problem. Section 3 is devoted to the problem-solving strategies employed by the supervisory portion of our adaptive group theoretic (AGT) algorithm for IP problems. The major topics of this section are: 1) the manner in which subproblems are selected by the supervisor, and 2) the strategies which are used to analyze a given subproblem. In Section 4, we present the basic AGT algorithm. A discussion of our work with an emphasis on areas for further research and extension is in Section 5. In the appendices to the paper (Sections 6-11) we discuss a number of topics in more detail, and present the details of some sub-algorithms used in the AGT algorithm. In this preliminary version, some of these discussions are incomplete.

2. ANALYTICAL METHODS

2.1 Introduction

The analytical methods to be used in the algorithm are based on the group theoretic methods discussed in [17,18,19,36,37,38,39]. The fundamental idea that we exploit is the following: The set of integer solutions to a system of linear equations is effectively characterized by an equivalent set consisting of the solutions to an equation of elements from a finite abelian group. In [18], Gomory exposed the importance of this idea and specialized the approach to IP problems for which (1) non-negative integer solutions are required, and (2) a best (optimal) solution is chosen from the set of feasible solutions by a linear criterion function.

We remark that the transformation of a combinatorial problem to a problem over an abelian group is a classic technique of number theory. Moreover, once the integer programming problem is studied from this perspective, a wide variety of number theoretic techniques are suggested. The possibilities are, in fact, sufficiently large to make the selection of the particular techniques to be used in analyzing a given problem non-obvious. It is undoubtedly true that as the application of number theory to IP problems progresses, insights will be gained which will lead to better procedures. Nevertheless, the selection of procedures to be used in analyzing a given problem must always depend to some degree on prior assessments of uncertain characteristics of the problem. For

this reason, it appears that a decision theoretic approach to problem analysis is indispensible to the supervision of an efficient (adaptive) algorithm. Our purpose in this section is to expose the variety of number theoretic methods which appear promising for IP problem analysis. Space and time limitations prevent us from treating in full detail all of the topics to be mentioned below.

As we shall see, the group methods have a dynamic as well as a static aspect. They are static in that they provide the fathoming tests for a particular enumerated correction x^k. This is the concern of Section 2.2. On the other hand, the group methods are dynamic because there is interaction between the fathoming tests for each of the corrections in the sequence $\{x^k\}_{k=1}^{K}$. These dynamics are discussed in Section 2.3.

2.2 Group Theoretic Analysis of Subproblems (Static Analysis)

We begin with a discussion of the methods as they are applied statically to a subproblem of the form of (1.6). For ease of exposition, all of the results in the beginning of this subsection are for the group transformation relative to the optimal LP basis B which was used to transform (1.1) to (1.2). It is easily seen that the same results hold if the transformation is relative to any mxm non-singular matrix made up of the columns a_j of A. For additional ease of exposition, we assume a generic m-vector b.

The transformation of problem (1.1) into a group optimization problem relative to B is accomplished as follows. Define[1]

---

[1]We use [a]=integer part of a, or [a] is the largest vector of integers such that $t \leq a$.

this reason, it appears that a definite theoretic approach to graphic analysis is indispensable to the [illegible] topological algorithms. Our purpose in this section is to review the variety of modern theoretic methods which appear [illegible]. Space and time limitations prevent [illegible] all of the topics [illegible] mentioned below.

As we [illegible], the group methods have a [illegible] as well as a static aspect. [illegible] the following theory for a particular [illegible]. This is the concern of Section 2.2. On the other hand, [illegible] graph-theoretic because there is interaction between [illegible] the intra-group in the sequence [illegible] Section 2.3.

2.2 Group Theoretic [illegible]

[illegible]

$$\alpha_j = D\{B^{-1}a_j - [B^{-1}a_j]\}; \quad j=1,\ldots,n \tag{2.1}$$

where $D = |\det B|$. In addition, define

$$\beta = D\{B^{-1}b - [B^{-1}b]\}; \tag{2.2}$$

The set of elements $\{\alpha_j\}_{j=1}^{n}$ with addition modulo D generates a finite abelian group G of order D [17]. This group can be more compactly represented as follows. Given a finite abelian group G, there exist unique positive integers $q_1,\ldots,q_r$, such that $q_1|q_2|\ldots|q_r$ (q_i divides q_{i+1}),

$$D = \prod_{i=1}^{r} q_i$$

and

$$G \simeq Z_{q_1} \oplus \ldots \oplus Z_{q_r} \tag{2.3}$$

where

$$Z_{q_i} = \text{residue class of the integers modulo } q_i.$$

Thus, G is uniquely and isomorphically represented by a collection of D r-tuples of integers

$$(\lambda_1,\ldots,\lambda_r) \text{ where } 0 \le \lambda_i \le q_i - 1 \text{ for } i=1,\ldots,r. \tag{2.4}$$

These r-tuples are ordered lexicographically by the rule $(d_1,\ldots,d_r) < (h_1,\ldots,h_r)$ if $d_{i_1} < h_{i_1}$ where i_1 is the smallest index such that $d_i \neq h_i$.

$$\phi_j = D(B^{-1}a_j - [B^{-1}a_j]) \tag{2.1}$$

where $D = |\det B|$. In addition, define

$$\phi = D(B^{-1}b - [B^{-1}b]) \tag{2.2}$$

The set of elements $\{\phi_j\}_{j=1}^{n}$ with addition modulo D generates a finite abelian group G of order [illegible]. This group can be more compactly represented as follows. Given a finite abelian group G, there exist unique positive integers [illegible] such that [illegible] divides [illegible]

[illegible]

and

[illegible] (2.3)

[illegible]

[illegible] modulo q_i

[illegible]

of G [illegible]

This imposes a linear ordering on the elements of G. Henceforth, we will assume that this isomorphic representation of G is being used and we say $G = \{\lambda_s\}_{s=0}^{D-1}$ where λ_s is the sth r-tuple and $\lambda_0 = \theta$, the identity element. Finally, the original α_j and β are now considered to be r-tuples of the form (2.4) above.

The advantages of the representation (2.3) over other representations are twofold. First, r never exceeds m and almost always is very small relative to m. In fact, for the majority of integer programming problems tested to date, r=1. Relative to other representations, (2.3) is a minimal generating system in the sense that r is the minimal number of cyclic subgroups of G which can be combined by direct summing to give G [10, p. 47]. An algorithmic procedure for achieving (2.3) is programmed and running on the IBM 360/65 at MIT. A second advantage of this representation is that it is unique to a given finite abelian group. It may be possible, therefore, to categorize integer programming problems by their induced group structure and thereby utilize experience gained on one problem in solving a second. We point out, however, that (2.3) may not be the best representation for a given problem. More is said about this below.

A network representation of G is useful in exposing several insights which we shall exploit. The network η involves only the non-basic variables x_j and the basic variables y_i are related to the x_j by the equations (1.3b). The network consists of D nodes, one for each of the elements λ_s. Directed arcs of the form $(\lambda_s - \alpha_j, \lambda_s)$, j=1,...,n, are

drawn to each node λ_s, s=1,...,D. Note that arcs are not drawn to $\lambda_0 \equiv \theta$. (It is convenient to add the node $\lambda_D = (q_1,\ldots,q_r)$, and the arcs[1] $(\lambda_D - \alpha_j, \lambda_D)$ j=1,...,n, to η. The significance of these additions to η is discussed below.) A path μ in η connecting θ to λ_s is backtracked to a solution $x = (x_1,\ldots,x_n)$ where x_j is set equal to the number of times an arc of the form $(\lambda_\ell - \alpha_j, \lambda_\ell)$ is used in μ. The relationship between this group network and a subproblem of the form (1.6) is the following. Every feasible solution to (1.6) corresponds to a path in η connecting θ to β_k in (2.2) for $b = b^k$.

Henceforth, when we speak of the group structure of a particular group, we shall mean the canonical representation (2.3) and the corresponding network. The group structure is to be distinguished from various group (optimization) problems which can be solved for a given group.

In the analysis below, subproblems (1.6) for $b = b^0, b^1,\ldots,b^k,\ldots,b^K$, are transformed into group optimization problems. In addition to the group structure discussed above, there are two quantities which require specification in order to formulate and solve the two group problems derivable from a given subproblem. First, each right hand side b^k is mapped into the group element

$$\beta_k = D\,\{B^{-1}b^k - [B^{-1}b^k]\}$$

which appears as the right hand side in the group constraint equation. The number of distinct right hand sides cannot, of course, exceed D.

Thus, we will formulate and solve the group problems for a generic group element λ_s as the right hand side in the group equation.

Second, for each subproblem there is the set F_k of free variables which can be used in continuing the correction x^k. In terms of the group problems, F_k restricts to $j \varepsilon F_k$ the types of arcs $(\lambda_k - \alpha_j, \lambda_k)$ which can be used in spanning $\alpha_b k$. We let F denote a generic subset of $\{1,2,...,n\}$ corresponding to a set of free variables.

In particular, the constraints of a typical group problem are

$$\sum_{j \varepsilon F} \alpha_j x_j = \lambda_s (\text{mod } D) \tag{2.6}$$

$$x_j \varepsilon S_j, \quad j \varepsilon F \tag{2.7}$$

where the sets S_j are to be specified in one of two ways. First, we let $S_j = N \equiv \{0,1,2,...\}$ $j \varepsilon F$, and define the solution set[1]

$$\Gamma_u(\lambda_s;F) = \{x\text{: x satisfies (2.6) and (2.7) with } S_j = \{0\}, \; j \notin F;\; S_j = N,\; j \varepsilon F\} \tag{2.8}$$

Second, we let $S_j = \{0,1\}$, $j \varepsilon S \cap F$, $S_j = N$, $j \varepsilon S^c \cap F$, and define the

$$\Gamma_v(\lambda_s;F) = \{x\text{: x satisfies (2.6) and (2.7) with } S_j = \{0\}, \; j \notin F;\; S_j = \{0,1\};\; j \varepsilon S \cap F,\; S_j = N,\; j \varepsilon S^c \cap F\} \tag{2.9}$$

With these definitions we can state

---

[1]Notice that F may be restrictive to the point that $\Gamma_u(\lambda_s;F)$ in (2.8) and/or $\Gamma_v(\lambda_s;F)$ in (2.9) are empty. This is to be desired because it indicates that a particular problem has no solution, and hence is fathomed.

Thus, we will formulate and solve the group [illegible] as the a general group element A_g as the right cond[illegible] [illegible].

Second, for each subproblem there is [illegible] free variables which can be used in continuing the [illegible] of the group problems, [illegible] η_j, ℓ_j, which can be used in [illegible] a general indices of (1,2,...,n) corresponding [illegible] [illegible]ations.

In particular, [illegible] group are [illegible]

[illegible]

[illegible]

[illegible]

[illegible]

LEMMA 2.1 For any subproblem (1.6) with $b^k = b^0 - Rx^k$, the following set inclusions hold:

$$\Gamma_u(\beta_k;\ F_k) \supset \Gamma_v(\beta_k;\ F_k) \supset X(b^k) \tag{2.10}$$

Proof: Choose any solution $x \epsilon X(b^k)$. It sufficies to show that $x \epsilon \Gamma_v(\beta_k;F_k)$ since the proof that $x \epsilon \Gamma_u(\beta_k;F_k)$ is almost the same. Since $x \epsilon X(b^k)$, we have from (1.6) that

$$Dy = D\bar{b}^k - \sum_{j \epsilon F_k} D\bar{a}_j x_j$$

$$x_j \epsilon \{0,1\},\ j \epsilon S \cap F_k$$

$$x_j \epsilon N,\quad j \epsilon S^c \cap F_k,$$

and y is a vector of integers satisfying (1.6e) and (1.6f). We substitute for $D\bar{a}_j = DB^{-1}a_j$ and $D\bar{b}^k = DB^{-1}b^k$ from (2.1) and (2.2), the result

$$Dy = \beta_k - D[B^{-1}b] - \sum_{j \epsilon F_k} \{\alpha_j - D[B^{-1}a_j]\}\ x_j,$$

or,

$$D\{y + [B^{-1}b] + \sum_{j \epsilon F_k} [B^{-1}a_j]x_j\} + \sum_{j \epsilon F_k} \alpha_j x_j = \beta_k.$$

$$x_j \epsilon \{0,1\},\ j \epsilon S \cap F_k$$

$$x_j \epsilon N,\ j \epsilon S^c \cap F_k.$$

Reducing both sides of the constraint equation modulo D yields

$$\sum_{j\varepsilon F_k} \alpha_j x_j = \beta_k$$
$$x_j \varepsilon \{0,1\},\ j\varepsilon S \cap F_k$$
$$x_j \varepsilon N,\ \ j\varepsilon S^c \cap F_k$$

which is what we wanted to show.

In words, lemma 2.1 states that each feasible solution to an IP subproblem (1.6) corresponds to a path connecting θ to β_k in the network η. The two group optimization problems induced from the above group structure are[1]

Unconstrained Group Problem:

$$\min z_u(\lambda_s;F) = \sum_{j\varepsilon F_k} \bar{c}_j x_j \quad \text{s.t. } x\varepsilon\Gamma_u(\lambda_s;F) \tag{2.11}$$

Zero-One Group Problem:

$$\min z_v(\lambda_s;F) = \sum_{j\varepsilon F_k} \bar{c}_j x_j \quad \text{s.t. } x\varepsilon\Gamma_v(\lambda_s;F) \tag{2.12}$$

Both of these group problems are shortest route problems in the network η if a cost (length) $\bar{c}_j$ is associated with each arc of the form $(\lambda_\ell - \alpha_j, \lambda_\ell)$. In particular, the Unconstrained Group Problem is the problem of finding an unconstrained shortest route path connecting θ to β_k,

---

[1]If either of problems (2.11) or (2.12) has no feasible solution, we take the objective function value to be $+\infty$.

Reducing both sides of the constraint equation modulo D yields

$$\sum_j \alpha_j x_j = \alpha_0$$

$x_j \in \{0,1\},\ j \in J_1$

$x_j \in Z,\ j \in S'/J_1$

which is what we wanted to show.

In words, lemma 2.2 states that each feasible solution to an IP subproblem (1.5) corresponds to a path connecting 0 to α_0 in the network G. The two group optimization problems induced from the above group structure are

Unconstrained Group Problem:

$$\min z_0(\alpha_0; 0) = \ldots \quad [illegible] \tag{2.14}$$

Zero-One Group Problem:

[illegible]

[illegible]

using arcs of the form $(\lambda_\ell - \alpha_j, \lambda_\ell)$, $j\varepsilon F_k$. The Zero-One Group Problem has the side constraints that arcs of the form $(\lambda_\ell - \alpha_j, \lambda_\ell)$ for $j\varepsilon S \cap F$ can be used at most once. The shortest route paths connecting θ to λ_D are minimal cost circuits in η. The corrections backtracked from these circuits are used in attempting to fathom x^k in (1.6) such that $\bar{b}^k$ is integer but a constraint either of the form (1.6c) or (1.6d) is violated.

Notice that the group problems viewed as shortest route problems are special shortest route problems because the same types of arcs can be shown to each node (save θ). Special algorithms were constructed in [37] and [39] for exploiting this structure. In particular, the Unconstrained Group Problem (2.11) can be solved by the algorithm GTIP1 of [37] and this algorithm is reproduced in Appendix A. Let $u(\lambda_s;F)$ denote the optimal solution to (2.11) found by AGIP1.[1] In addition to solving (2.11) GTIP1 also finds $u(\lambda_p;F)$, $p=0,1,\ldots,D$ for the given set F of free variables. As we shall see, this property is very useful to the dynamic workings of the AGT algorithm.

The Zero-One Group Problem (2.12) can be solved by the algorithm of [39] which is reproduced in Appendix B. Let $v(\lambda_s;F)$ denote the optimal solution to (2.12) found by the algorithm of [39]. This algorithm also has the property that it finds $v(\lambda_p;F)$, $p=0,1,\ldots,D$, for the given set F of free variables.

---

[1]Alternative optima to (2.11) may be important and GTIP1 can be amended to find some of them (See [37]). For simplicity, we assume GTIP1 finds a unique solution to (2.10).

The set inclusions in Lemma 2.1 imply

Corollary 2.1: Let $\tilde{x}$ be an optimal solution to subproblem (1.6). Let $u(\beta_k;F_k)$ and $v(\beta_k;F_k)$ be optimal solutions with costs $z_u(\beta_k;F_k)$ and $z_v(\beta_k;F_k)$. Then

(i) $\bar{c}\tilde{x} \geq \bar{c}v(\beta_k;F_k) \geq \bar{c}u(\beta_k;F_k)$,

and

(ii) If $x^k + u(\beta_k;F_k)$ $(x^k + v(\beta_k;F_k))$ is a feasible correction, then it is an optimal correction.

The implication of Corollary 2.1 is that x^k is fathomed if one of the four tests (2.13), (2.14), (2.15), (2.16) obtains. These tests are:

$$\bar{c}(x^k + u(\beta_k;F_k)) \geq \hat{z}(b^0) \tag{2.13}$$

$$\bar{c}(x^k + v(\beta_k;F_k)) \geq \hat{z}(b^0), \tag{2.14}$$

$$x^k + u(\beta_k;F_k) \text{ is a feasible correction in (1.3)} \tag{2.15}$$

$$x^k + v(\beta_k;F_k) \text{ is a feasible correction in (1.3)} \tag{2.16}$$

If (2.15) (or (2.16)) obtains, then a new incumbent has been found and $\hat{x}(b^0) \leftarrow x^k + u(\beta_k;F_k)$ $(\hat{x}(b^0) \leftarrow x^k + u(\beta_k;F_k))$ $\hat{z}(b^0) \leftarrow \bar{c}\hat{x}(b)$.

It is important to remark at this point that if either of the group problems with $b^k = b^0$ and $F_k = F_0 = \{1,2,\ldots,n\}$ yields a feasible correction in (1.3), then this correction is optimal and no further analysis is needed. In this case, of course, the tree search is not required. The entire analysis and algorithmic construction below are predicated on the assumption that a given integer programming problem (1.1) is a difficult one to solve. By this we mean that the group algorithms applied to (1.1) fail to yield optimal solutions. Moreover, we go further and state that a difficult problem also has the property that the subsequent search (after the algorithms fail) is extensive.

Before turning to a discussion of group transformations with respect to arbitrary (dual feasible) bases, we mention that there are sufficient conditions from [37] on the use of the group optimization algorithms in finding an optimal continuation of a correction x^k. These conditions are

$$\text{(i)}\ \rho_i b^k \geq (D-1)\ (\underset{\bar{a}_{ij} > 0}{\text{maximum}}\ \bar{a}_{ij})\quad i=1,\ldots,m, \tag{2.17a}$$

$$\text{(ii)}\ \rho_i b^k \leq 1 + (D-1)\ (\underset{\bar{a}_{ij} < 0}{\text{minimum}}\ \bar{a}_{ij}),\ i\varepsilon S, \tag{2.17b}$$

where ρ_i is the ith row of B^{-1}, and $\bar{a}_{i_j}$ is the ith component of $\bar{a}_j = B^{-1}a_j$, j=1,...,n. The relations (2.17a) and (2.17b) depend upon the existence of a cyclic unconstrained and zero-on shortest route paths in η . Computational experience has shown that these sufficient conditions are gross overestimates of the "fat" required in $B^{-1}b^k$ and $1-\rho_i b^k$ for the corrections from the group problems to be feasible. Hence, these

conditions are of limited computational use. However, the deviation of these conditions suggests some qualitative results which provide some measure of IP problem characterization. This is discussed in Section 2.3.

As we have previously mentioned, a group transformation such as the one discussed above can be effected with any mxm non-singular basis made up of the columns a_j of A. However, the group optimization algorithms rely heavily on the assumption that the arc costs $\bar{c}_j$ are non-negative and therefore that there exists an acyclic shortest route path in η. For this reason we will confine our attention to bases which are dual feasible. It is important to recognize that a dual feasible basis B_t has a special meaning in the context of solving a subproblem of the form (1.6) for the correction x^k. We are confined to $j \varepsilon F_k \cup \{n+1,\ldots,n+m\}$ for activities a_j from which to form B_t. If all of the activities a_j, $j=1,2,\ldots,n+m$ were considered, then B_t would not necessarily remain dual feasible. Thus, if $\sigma_t = \{j_1,\ldots,j_m\}$ is the set of indices of the column forming B_t, then we can meaningfully define a set $F_k^t = (F_k \cup \{n+1,\ldots,n+m\}) \cap \sigma_t^c$ which contains the non-basic free variables relative to the basis B_t. These are the variables that will be used in solving the group problems derived from B_t.

Strictly speaking, then, the dependence of the matrix B_t on the set of activities from which it was formed should be indicated. We will omit such an explicit notation with the understanding that any dual feasible basis and the ensuing group theoretic analysis will always be relative to some set of free variables. Similarly, we let R_t be the

matrix formed by the activities a_j, $j \varepsilon F_k^t$. The relative cost factors are $\bar{c}_j^t = c_j - c_{B_t} B_t^{-1} a_j \geq 0$, $j \varepsilon F_k^t \cup \sigma_t$. Finally, let ρ_i^t denote the ith row of B_t^{-1}.

The details of using a dual feasible basis in attempting to fathom a correction x^k are given below. The interactions between dual feasible bases and the resulting group problems for a sequence of subproblems are discussed in Subsection 2.3. For convenience, we assume a generic b and a generic set F or free variables (relative to x^k). The resulting set F^t of free variables relative to B_t is $F^t = (F \cup \{n+1,\ldots,n+m\}) \cap \sigma_t^c$.

The elements

$$\alpha_j^t = D_t\{B_t^{-1} a_j - [B_t^{-1} a_j]\}, j \varepsilon F_k^t \tag{2.18a}$$

where $D_t = |\det B_t|$, generate the abelian group G_t of order D_t. (To ensure that the full group of order D_t is spanned, temporarily augment R_t with whatever unit vectors required to have it contain an mxm identity matrix. These augmented unit vectors are omitted once the group representation (2.3) is found.) The group right hand side element corresponding to b is

$$\beta^t = D_t\{B_t^{-1} b - [B_t^{-1} b]\} \tag{2.18b}$$

Using the canonical representation of G_t, we have $G_t = \{\lambda_s^t\}_{s=0}^{D_t-1}$ where each λ_s^t is an r_t-tuple. There is a group network η_t corresponding to G_t, and as before, we augment η_t by the node $\lambda_{D_t}^t = (q_1^t,\ldots,q_r^t)$ and the arcs $(\lambda_{D_t}^t - \alpha_j^t, \lambda_{D_t}^y)$, $j\varepsilon\sigma_t^c$.

Again we are interested in certain paths in the group network $_t$ connecting θ^t to β^t. Because we will be finding shortest route paths from θ^t to λ_s^t, $s=1,\ldots,D_t$, we take λ_s^t as a generic group right hand side. The constraints of a typical group problem for group t are

$$\sum_{j\varepsilon F^t} \alpha_j^t d_j = \lambda_s^t \pmod{D_t} \tag{2.19}$$

$$d_j \varepsilon S_j,\ j\varepsilon F^t \tag{2.20}$$

where the sets S_j are specified as before in one of two ways. First,

$$\psi_u^t(\lambda_s^t;F^t) = \{d:\ d \text{ satisfies (2.19) and (2.20) with } S_j = \{0\},\ j\varepsilon(F^t)^c \cap \sigma_t^c;\ S_j=N,\ j\varepsilon F^t \cap \sigma_t^c\}\ ; \tag{2.21}$$

and second,

$$\psi_v^t(\lambda_s^t;F^t) = \{d:\ d \text{ satisfies (2.19) and (2.20) with } S_j = \{0\},\ j\varepsilon(F^t)^c \cap \sigma_t^c;\ S_j=\{0,1\},\ j\varepsilon S \cap F^t \cap \sigma_t^c,\ S_j=N,\ j\varepsilon S^c \cap F^t \cap \sigma_t^c\} \tag{2.22}$$

In order to express all group solutions in common terms, for any $d \epsilon \psi_u^t(\lambda_s^t;F^t)$ or $d \epsilon \psi_v^t(\lambda_s^t;F^t)$, define x by

$$x_j = \begin{cases} d_j \text{ if } j \epsilon \sigma_t^c \\ \rho_i^t(b-R_t d) \text{ if } j=j_i \epsilon \sigma_t \end{cases} \qquad (2.23)$$

where $\lambda_s^t = \beta^t$ for some b. Thus, the group solution sets in terms of the x_j can be defined as before. Let

$$\Gamma_u^t(\lambda_s^t;\ F^t) = \{x:\ \ x \text{ satisfies (2.23) for some } d \epsilon \psi_u^t(\lambda_s^t;F^t)\} \qquad (2.24)$$

and

$$\Gamma_v^t(\lambda_s^t;\ F^t) = \{x:\ \ x \text{ satisfies (2.23) for some } d \epsilon \psi_v^t(\lambda_s^t;F^t)\} \qquad (2.25)$$

With this background, we have

Corollary 2.2 For any subproblem (1.6) with $b^k = b^0 - Rx^k$, the following set inclusions hold:

$$\Gamma_u^t(\beta_k^t;\ F_k^t) \supset \Gamma_v^t(\beta_k^t;F_k^t) \supset X(b^k).$$

The group optimization problems induced from the above group structure are:

Unconstrained Group Problem t:

$$\min z_u^t(\lambda_s^t;\ F^t) = \sum_{j\varepsilon F^t} \bar{c}_j^t d_j \quad \text{s.t. } d\varepsilon\psi_u^t(\lambda_s^t;F^t) \tag{2.26}$$

Zero-One Group Problem t:

$$\min z_v^t(\lambda_s^t;\ F^t) = \sum_{j\varepsilon F^t} \bar{c}_j^t d_j \quad \text{s.t. } d\varepsilon\psi_v^t(\lambda_s^t;F^k) \tag{2.27}$$

For each optimal solution d to (2.26), let the solution $u^t(\lambda_s^t;F^t)$ denote the resulting correction derived from d by (2.23). Similarly, let $v^t(\lambda_s^t;F^t)$ denote the resulting correction derived from any optimal solution (2.27). Thus, we have the following extension of Corollary 2.1

Corollary 2.3 Let x be an optimal solution to subproblem (1.6) and let B_t be any dual feasible basis in (1.1). Let $u^t(\lambda_k^t;F_k^t)$ and $v^t(\lambda_k^t;F_k^t)$ be derived from optimal solutions to the Unconstrained Group Problem t and the Zero-One Group Problem t. Then

(i) $\bar{c}x \geq \bar{c}u^t(\beta_k^t;F_k^t) \geq \bar{c}v^t(\beta_k^t;F_k^t)$,

and

(ii) if $x^k + u^t(\beta_k^t;F_k^t)$ $(x^k+v^t(\beta_k^t;F_k^t))$ is a feasible correction, then it is an optimal continuation of x^k.

Suppose now that there is a collection $\{B_t\}_{t=1}^T$ of dual feasible

bases ($B_1 = B$) and the corresponding group optimization problems available for attempting to fathom an enumerated correction x^k. It is clear by the above analysis that x^k is fathomed if one of the following four tests obtains:

$$\text{(i)} \quad \max_{t=1,\ldots,T} \quad \bar{c}u^t(\beta_k^t;F_k^t) \geq \hat{z}(b^0) - \bar{c}x^k \tag{2.28}$$

$$\text{(ii)} \quad \max_{t=1,\ldots,T} \quad \bar{c}v^t(\beta_k^t;F_k^t) \geq \hat{z}(b^0) - \bar{c}x^k \tag{2.29}$$

$$\text{(iii)} \quad x^k + u^t(\beta_k^t;F_k^t) \text{ is a feasible correction for some } t \tag{2.30}$$

$$\text{(iv)} \quad x^k + v^t(\beta_k^t;F_k^t) \text{ is a feasible correction for some } t \tag{2.31}$$

If (iii) (or iv) holds for some t, then $\hat{x}(b^0) \leftarrow x^k + u^t(\beta_k^t;F_k^t)$ ($\hat{x}(b^0) \leftarrow x^k + v^t(\beta_k^t;F_k^t)$) and $\hat{z}(b^0) \leftarrow \bar{c}\hat{x}(b^0)$.

It is easy to show that additional group optimization problems can be constructed as follows. Substitute the relative cost factors with respect to any dual feasible basis in the objective functions of (2.26) and (2.27) rather than the $\bar{c}_j^t$ relative to B_t for which the sets $\psi_u^t(\lambda_s^t;F^t)$ and $\psi_v^t(\lambda_s^t;F^t)$ are derived. We omit further mention of these problems but the computational usefulness of this idea will be explored further.

There are two remarks to be made about the group problems (2.26) and (2.27). First, as a rule only one of these problems will be used in

attempting to fathom a given correction x^k. Limited computational experience indicates that (2.27) requires approximately five times the computation required by (2.26). On the other hand, a solution to (2.27) yields a correction which is more likely to lead to a fathoming of x^k. A procedure for choosing between the two group problems is discussed in section 3.3. It suffices here to mention that the result of the decision process is a correction $h^t(\beta_k^t;F_k^t)$ which equals either $u^t(\beta_k^t;F_k^t)$ or $v^t(\beta_k^t;F_k^t)$.

Second, the reader should note that not all group problems t are equally useful in fathoming a given correction x^k. For an arbitrary sub-problem, the most appropriate dual feasible basis to use in trying to fathom the sub-problem is the one that is also optimal feasible; that is, the most appropriate basis is B_t such that

$$B_t^{-1}b^k \geq 0 \tag{2.32}$$

and

$$1-\rho_t^i b^k \geq 0 \text{ for } i\varepsilon S \tag{2.33}$$

Our reasoning here is that the correction $u^t(\beta_k^t;F_k^t)$ or $v^t(\beta_k^t;F_k^t)$ is the one most likely to be feasible in the subproblem x^k since 1) we begin with feasibility in (2.32) and (2.33) and 2) the shortest route paths in η_t yielding these solutions tend to be short. Moreover, the relative cost factors $\bar{c}_j^t$ are the appropriate (at least in the LP sense) ones for the given b^k. We remark in passing that the apparent correction between the group theoric IP results and duality for IP[3] has been largely ignored.

Finally, we present a brief discussion of the application of group theory to Gomory's cutting plane method. The cutting plane method begins

with the LP solution (1.3) of (1.1). When this solution is not integer, it is possible to deduce new constraints (cuts) to add to (1.3) with the properties 1) the current optimal LP solution is infeasible in the augmented problem and 2) every feasible integer solution is feasible in the augmented problem. As demonstrated in [17], there are D distinct cuts (including the mill cut) which can be deduced from an LP solution of the form (1.3). A typical cut to be added is of the form

$$\sum_{j=1}^{n} \alpha_{ij} x_j \geq (\alpha_b 0)_i$$

where x_{ij} and $(\alpha_b 0)_i$ are the ith components of α_j and $\alpha_b 0$ when the representations (2.1) and (2.2) are used.

The collection of cuts forms the same abelian group G of order D with addition (mod D) of the rows in (2.6) with $\lambda_s - \alpha_b 0$ and $F = F^0 = \{1,2,...,n\}$. It has long been observed that not all of the D cuts have equal resolution or strength. Only recently, however, has insight been gained about the identification of strong cuts [19]. A strong cut can be described as follows. Plot in n-space all the solutions $x \varepsilon \Gamma_u(b^0;F_0)$ and form the convex polyhedron which is the convex hull of these points. A cut is strong if it is a face of this convex polyhedron, and such faces can be generated by solving a problem similar to problem (2.11) with $\lambda_s = \alpha_b 0$ and $F=F_0$. The reader is referred to [19] for further details.

Although the theoretical development in [19] may well be of great algorithmic importance, we do not include the cutting plane algorithm as with the LP solution (1.1) of (1.1). When this solution is not integer, it is possible to deduce new constraints (cuts) [illegible] to (1.1) with the properties 1) the current optimal solution is [illegible] in the augmented problem and 2) every feasible integer solution to (1.1) is in the augmented problem. As demonstrated in [illegible] the [illegible] (including the all cut) which can be deduced [illegible] solution [illegible] = (1/3) typical cut to be added is of the form [illegible]

[illegible]

where V_{ij} and (a_j [illegible] representation [illegible] used.

The [illegible] with [illegible] it has [illegible] representation of the [illegible] the (constr[illegible] [illegible] integer [illegible] with [illegible] [illegible] subsequent [illegible]

[illegible]

an analytic method in this version of the AGT algorithm. Instead, we await a more complete theoretical developmentof the relationship between group theory and the cutting plane method.

2.3 Dynamic Group Theoretic Analysis

This section is devoted to a discussion of the dynamic interactions of the group theoretic methods of the previous Section. We shall attempt to describe in a qualitative manner how the subproblems and the various group structures and group problems change and interact as the AGT Algorithm progresses. Alternatively, we can say that our goal is a description of how the computational experience of the AGT Algorithm previous to the analysis of a given subproblem can be related to that analysis. Moreover, we try to relate the analysis of a given subproblem to future computation. A fully specified decision making procedure based on the ideas to be presented here is given in Section 3.3.

There are two fundamental constructs which change as a sequence of subproblems (1.6) are considered. These variable factors are the right hand side b^k and the set of free variables F_k. As we shall see, the dynamics of the AGT Algorithm can be described and analyzed largely by studying and exploiting the changes in b^k and F_k.

Consider, then, the analysis of a subproblem (1.6) derived from an enumerated correction x^k. If x^k is such that $y = b^k$ satisfies (1.6e) and (1.6f), then x^k is fathomed. Otherwise, additional analysis is needed. Before applying group theoretic methods, we make a cursory real-space examination of b^k to ensure that feasibility is attainable. In particular, if either

$$\bar{b}_i^k + \sum_{j \varepsilon F_k \cap S} \max \{0, \bar{a}_{ij}\} + \sum_{j \varepsilon F_k \cap S^c} \max \{0, \bar{a}_{ij} u_j\} < 0 \quad \text{for some } \bar{b}_i^k < 0, \tag{2.40}$$

or

$$\bar{b}_i^k + \sum_{j \in F_k \cap S} \min\{0, \bar{a}_{ij}\} + \sum_{j \in F_k \cap S^c} \min\{0, \bar{a}_{ij}, u_j\} > 1$$

$$\text{for some } \bar{b}_i^k > 1, \; y_i \varepsilon S \qquad (2.41)$$

where u_j is an upper bound on x_j, then x^k is fathomed because feasibility of some $w \geq x^k$ is impossible. This type of fathoming test is the major fathoming test of the algorithm in [11]. It should be clear that (2.40) and (2.41) have weak resolution whenever F_k and/or S^c are large. We include these tests because 1) they provide a measure of real space feasibility not provided by the group methods, and 2) they are computationally cheap since they are additive from problem to problem.

Let us suppose that (2.40) and (2.41) do not lead to a fathoming of x^k and therefore that more sophisticated analysis is in order. Suppose further that (Unconstrained and Zero-one) Group Problems corresponding to dual feasible bases B_t, t=1,...,T, have been formulated and solved prior to the analysis of x^k, and retained. For each t we associate the following:

1) The set $\sigma_t = \{i_1^t, \ldots, i_m^t\}$ of the indices of the activities in B_t;

2) The set $L_t = \{j \mid c_j - c_{B_t} B_t^{-1} a_j < 0\}$ (2.42)

of indices of the infeasible dual rows with respect to B_t (B_t is dual feasible with respect to L_t^c);

3) The group $G_t = \{\lambda_s^t\}_{s=0}^{D_t - 1}$ with the representation (2.3);

4) An optimal group table for an Unconstrained Group Problem (2.21) previously solved with the set of free variables $F^t = U^t$ and $U^t \subseteq \sigma_t^c \cap L_t^c$, and possibly

5) An optimal group table for a Zero-one Problem (2.22) where previously solved with the set of free variables $F^t = V^t$ and $V^t \subseteq \sigma_t^c \cap L_t^c$.

At this time it is important to state that the main fathoming tool for subproblem analysis is considered to be the Unconstrained Group Problem rather than the Zero-one Group Problem. By contrast, the Zero-one Group Problem is used as a second effort method when the Unconstrained Group Problems are judged to be performing poorly as fathoming tests. Of course, if a Zero-one Problem has been computed previous to the analysis of x^k and this previous computation is relevant (in a sense to be explained momentarily), then the zero-one solution is preferred to the unconstrained solution relative to the same set of free variables.

With this background, it can be seen that we attempt to fathom x^k in one of three ways. First, we try to use the stored results described above without recomputation. To do this, it is necessary to ascertain which stored results can be used directly. The stored results which can be used directly are left over from previous subproblem analysis for corrections which dominate x^k in a manner to be described below. Second, if the stored results do not lead to a fathoming of x^k, then we dynamically re-optimize a selected set of unconstrained group problems. Finally, if this fails, we may be willing to find a new dual feasible basis and use the induced group structure and group problem to try to

fathom x^k. As a secondary alternative, it may be worthwhile to resolve a Zero-one Group Problem for a new group. The ensuing paragraphs are a discussion of these considerations.

As stated above, our first concern when attempting to fathom x^k is the application of relevant stored results without recomputation. In particular, a group problem needs to be relevant in two ways. First, the group structure G_t over which the group problem(s) are defined must be derived from a basis B_t which is dual feasible for the LP problem (1.6). Specifically, if B_t is dual feasible with respect to L_t^c, then G_t is a relevant group structure for subproblem x^k only if $(F_k \cup \sigma_1) \cap L_t = \phi$. Let $\tau_1^u \subseteq T$ be the set of relevant group structures in this regard.

The second requirement is on the group problem itself. Given the group structure G_t, a stored Unconstrained Group Problem solution (optimal group table) is relative to some set U^t, $U^t \subseteq \sigma_t^c \cap L_t^c$, of free non-basic variables relative to the basis B_t. For the stored results to be applicable to the analysis of subproblem x^k, it is necessary that $F_k^t \equiv (F_k \cup \sigma_1) \cap \sigma_t^c \subseteq U^t$. If this set inclusion holds, then the Unconstrained Group Problem t relative to U_t is said to <u>dominate</u> the same problem relative to F_k^t. Let $\tau_2^u \subseteq \tau_1^u$ be the set of indices for which (unconstrained) cominance holds. Similarly, a stored zero-one group problem solution relative to V^t is said to <u>dominate</u> the same problem relative to F_k^t if $F_k^t \subseteq V^t$. Let $\tau_1^v \subseteq \tau_2^v$ be the set of indices for which (zero-one) dominance holds.

The usefulness of the stored group problem solutions for $t \in \tau_1^u$, or

$t \epsilon \tau_2^v$ is embodied in

Corollary 2.4 If for subproblem (1.6) with $b^k = b^0 - Rx^k$ and for some t_1 we have $F_k^t \equiv (F_k \cup \sigma_1) \cap \sigma_t^c \subseteq U^t$, then

$$\Gamma_u^t(\beta_k^t;U^t) \geq \Gamma_u^t(\beta_k^t;F_k^t).$$

Similarly, if $F_k^t \leq V^t$, then

$$\Gamma_v^t(\beta_k^t;V^t) \geq \Gamma_v^t(\beta_k^t;F_k^t).$$

As before, these set inclusions imply

Lemma 2.5 Let x be an optimal solution to subproblem (1.6) and let B_t be any dual feasible basis such that $F_k^t \subseteq U^t$. Let $u^t(\beta_k^t;U^t)$ and $v^t(\beta_k^t;V^t)$ be derived by (2.23) from optimal solutions to the Unconstrained and Zero-one Group Problems (2.26) and (2.27). We can conclude

(i) $\bar{c}x \geq \bar{c}u^t(\beta_k^t;F_k^t) \geq \bar{c}u^t(\beta_k^t;U^t)$

(ii) $\bar{c}x \geq \bar{c}v^t(\beta_k^t;F_k^t) \geq \bar{c}v^t(\beta_k^t;V^t)$

(iii) if $x^k + u^t(\beta_k^t;U^t)$ or $x^k + v^t(\beta_k^t;V^t)$ is a easible correction, then it is an optimal continuation of x^k.

Thus, we can fathom x^k if one of the following four tests obtains

by backtracking in the stored solutions for $t\varepsilon\tau_2^u$ or $t\varepsilon\tau_2^v$:

(i) $\overline{cu}^t(\beta_k^t;U^t) \geq \overline{cx}^k - \hat{z}(b)$ (2.43)

(ii) $\overline{cv}^t(\beta_k^t;V^t) \geq \overline{cx}^k - \hat{z}(b)$ (2.44)

(iii) $x^k + u^t(\beta_k^t;U^t)$ is a feasible correction (2.45)

(iv) $x^k + v^t(\beta_k^t;V^t)$ is a feasible correction (2.46)

Note that the Unconstrained and Zero-one Group Problems solved at $x^0 = 0$ for t=1 dominate every enumerated correction x^k.

If the efforts above to use the stored results without recomputation fail, then the next option open to use is to update the Unconstrained Optimal Group Tables for all $t\varepsilon\tau_1^u$ (we ignore for the moment the possibility of updating or recomputing optimal solutions for Zero- ne Group Problems). This updating can begin with the previous Unconstrained Optimal Group Table and therefore the amount of recomputation can be quite small if the commonality between U^t and F_k^t is great (see Section 8). In any case, the updating will require a non-trivial investment of computation time and therefore it is important to make some prior assessment of the relative value of optimal solutions to Unconstrained Group Problems t for $t\varepsilon\tau_1^u$. This value is due to the possible fathoming of x^k and also to the possible fathoming of continuations of x^k if x^k is not fathomed. We point out that we are most desirous of using B_t such that $B_t^{-1} \geq 0$ and $\rho_i^t b_k \leq 1$ for $i = j_s \varepsilon S \cup \sigma_t$. If it can be ascertained that there is some $t\varepsilon\tau$ for which B_t is such an optimal LP basis, then this group structure and Unconstrained

Group Problem is given top ranking. More about this below.

In Section 3.3 we describe how the information collected to date about the performance and relevance of Unconstrained Group Problems, plus some overall problem diagnosis is used to rank that $t\varepsilon\tau_1^u$. It sufficies here to give a qualitative description of the factors taken into consideration in termining the ranking. First, there is the relative and absolute performance of each of the Unconstrained Group Problems in fathoming subproblems previously encountered.

Second, there is the relevance of any given Unconstrained Group Problem to the subproblem x^k. To gain some insight about this, we note that b^k is the end of a path in m-space connecting b^0 to b^k by a sequence of arcs (b^0, b^{k_1}), $(B^{k_1}, b^{k_2}), \ldots, (b^{k_{I-1}}, b^{k_I})$ where $k_I = k$. If B_{t_i} is an optimal LP basis for subproblem b^{k_i}, then the movement in m-space is through a sequence of cones $K^{t_i} = \{b: B_{t_i}^{-1} b \geq 0\}$ in reaching b^k. Since $b^{k_i} = b^{k_{i-1}} + a_{j_{i-1}}$ for some j_{i-1}, the movement can be considered to be fairly smooth and hopefully the path lingers in the cone K^{t_i} for several corrections before moving to a new cone. As for analysis of (1.6) with b^k, it is clear that we are most interested in the cone K^{t_I} and those cones which are adjacent to it. Thus, we can argue qualitatively that the ranking should depend upon the most recent cones penetrated and possibly repenetrated in reaching (1.6) with b^k. The sequence of cones (when they are ascertained) leading to b^k can be recorded easily and the ranking procedure in Section 3.3 depends heavily on inference

from this sequence.

Finally, the ranking procedure considers the potential usefulness of updated group solutions to future subproblem analyses by evaluating the position of x^k in the tree of enumerated solutions.

As a result of the ranking procedure, we assume there is a set $\tau_3^u \epsilon \tau_1^u$ such that the Unconstrained Group Problems are to be updated in some specified order for $t \epsilon \tau_3^u$. In particular, the Unconstrained Group Problem is solved dynamically in the sense that the new solution is derived from the previous one by setting U^t to $(F_k \cup \sigma_1) \cap \sigma_t^c$ (see Section 8). The results of the dynamic reoptimization of the Unconstrained Group Problems for $t \epsilon \tau_3^u$ are used as before in attempting to fathom x^k (see (2.28) and (2.30)). If x^k is still not fathomed, we may choose to find a new group and solve a new group problem. In any event, if x^k is continued the new Unconstrained Group Problem solutions for $t \epsilon \tau_3^u$ will be useful in attempting to fathom its descendants.

Suppose now that the analysis of subproblem x^k has not led to a fathoming of x^k. If it is known that a group problem over the group structure induced by an optimal LP basis for the subproblem has been solved, then the group theoretic analysis is terminated and x^k is continued. Otherwise, we may choose to find this basis and probably solve a new Unconstrained Group Problem derived from it. The details for making this decision are given in Section 3.3. We will consider here the implications of the decision to find an optimal LP basis for (1.6), on the assumption that it is not known whether group analysis with respect to this basis

has been previously performed.

Thus, let B_s be an optimal LP basis for (1.6) which is found by the dual simplex method where the initial dual feasible basis is B_1. If (1.6) is an infeasible LP problem, then x^k is fathomed. Problem (1.6) cannot have an unbounded LP solution because we assume there are upper bounds on each of the variables. Once B_s is found, we compare σ_s to σ_t, t=1,...,T. If $\sigma_s = \sigma_t$ for some t, then B_s is not a new dual feasible basis and the only recourse open is to continue x^k. Suppose that $\sigma_s \neq \sigma_t$, t=1,...,T; in this case, let $B_{T+1} = B_s$, $\sigma_{t+1} = \sigma_s$ and derive the group G_{T+1} with the representation (2.3). Solve the Unconstrained Group Problem (2.26) for B_{T+1} with respect to $U^{T+1} = (F_k \cup \sigma_1) \cap \sigma^c_{T+1}$, and attempt to fathom x^k with the tests (2.28) for t=T+1 or (2.30).

We comment briefly on the rationale for finding a new group structure G_{T+1} when the conditions of the previous paragraph obtain. In add-tion to the obvious benefit of the resulting Unconstrained Group Problem in fathoming x^k, the creation of a new group problem is useful to the fathoming of the possible descendants of x^k. These ideas are illustrated in Figure 1. The cone K^2 is penetrated first by the vector b^{k_1} at which time G_2 and (2.26) with respect to G_2 are formulated and solved. This did not lead to a fathoming of x^{k_1} but the results for G_2 are useful when attempting to fathom x^{k_2} and x^{k_3} which are continuations of x^{k_1}.

If all attempts at fathoming x^k fail, then we consider continuing

x^k to $x^k + e_j$ for $j \geq j(x)$ where $j(x)$ is defined by (1.4) Not all continuations will actually be made for the following reason. For each $x + e_j$ we associate a lower bound value from the relevant group problems. In particular, the lower bound value for $x+e_j$ is

$$\max \{\max_{t \varepsilon t_2} \{\overline{c}x^k + \overline{c}_j + \overline{c}u^t(\beta_k^t - \alpha_j^t; U^t)\},$$

$$\max_{t \varepsilon t_2'} \{\overline{c}x^h + \overline{c}_j + \overline{c}v^t(\beta_k^t - \alpha_j^t; U^t)\},$$

$$\overline{c}x^h + \overline{c}_j + \overline{c}u^{T+1}(\beta_k^{T+1} - \alpha_j^{T+1}; U^{T+1})\} \quad (2.47)$$

where the third term is omitted if a new group T_{T+1} was not generated. The correction x^k is continued to x^k+e_j only if this lower bound value is strictly less than the incumbent value $\hat{z}(b^0)$, If $x^{k'} = x^k+e_j$ is a continuation which is allowed by this rule, then we associate the maximal lower bound value with t for use in plausibility analysis (Section 3.2).

Finally, if it was discovered that B_t was an optimal LP basis for (1.6), then we calculate $B_t^{-1}(b^k-a_j) = B_t^{-1}b^k - B_t^{-1}a_j$. The basis B_t is a feasible and therefore optimal LP basis (1.6) with b^k-a_j if $B_t^{-1}b^k - B_t^{-1}a_j \geq 0$ and $\rho_i^t(b^k-B_t^{-1}a_j) \leq 1$ for $i = i_s \varepsilon S \cap \sigma_t$. If so, then we set $\Delta^{k'}=t$. Notice that the above test and the group identities for all G_T can be updated from x^k to x^k+e_j by addition or subtraction.

3. SEARCH AND FATHOMING PROCEDURES

3.1 Introduction

The AGT algorithm which is presented in Section 4 of this paper employs two basic types of methods in solving a given IP problem. Methods of the first type are called analytical methods, and they are designed for the solution of subproblems generated during the course of solving the given problem. Methods of the second type are heuristic or supervisory methods, and they are used to control the order in which subproblems are attacked as well as the analytic methods which are applied to them. Decisions about subproblem selection and testing are influenced by information obtained from the solution of previous subproblems. Therefore, among the supervisory methods are ones which are directed at diagnostic problems which are encountered during a search for an optimal correction.

In section 2, the analytical methods were discussed. In Sections 3.2 and 3.3, we discuss the supervisory methods associated with subproblem selection and subproblem analysis respectively. Finally, in Section 3.4, we discuss some methods by which the supervisor can structure the IP problem prior to beginning the search. These methods collectively are called pre-search analysis.

3.2 Search Procedures

In this section, we will investigate several search procedures which

implicitly test all corrections to (1.3) for optimality. First we will contrast two fundamentally different algorithms for searching the tree of corrections. These algorithms are a breadth-first algorithm and a depth-first algorithm. Then we will demonstrate the limitations of both these rigid search procedures, and argue for a more flexible strategy. Finally, we will describe the basic search procedure we have developed for our AGT algorithm.

The breadth-first search procedure is so named because an attempt is made to fathom all corrections at level K before any corrections at level K+1 are considered where $K = \sum_{j=1}^{n} x_j$. The breadth-first procedure discussed here incorporates two basic results from [38]:

1) Form of an Optimal Correction Lemma:

Suppose a correction x is not fathomed. Without loss of optimality, we can continue x by the corrections $x + e_j$ for $j=j(x), j(x)+1,\ldots,n$ (See (1.4)).

2) The search can be confined to levels $K = \sum_{j=1}^{n} x_j \leq K^*(\hat{z}(b^0))$ where $K^*(\hat{z}(b^0))$ is retrieved from the solution of the knapsack problem:

$$K^*(\hat{z}(b^0)) = \max \sum_{j=1}^{n} V_j \qquad (3.1a)$$

subject to

$$\sum_{j=1}^{n} (D\cdot\bar{c}_j)V_j \leq D\cdot\hat{z}(b^0) \qquad (3.1b)$$

$$V_j \text{ non-negative integer} \qquad (3.1c)$$

This problem can be solved once for all right hand sides $0,1,\ldots, D\cdot\hat{z}(b)$ by the algorithm of [35].

With these two results in hand, we can describe the breadth-first search procedure as follows. Starting with K=0, attempt to fathom all non-negative integer corrections x such that $\sum_{j=1}^{n} x_j = K$. If a correction x is not fathomed, then the Form of an Optimal Correction Lemma is used to continue x, and all corrections of the form $x + e_j$ for $j=j(x),\ldots,n$ are placed on the (K+1)-list. When all corrections on the K-list have been tested (fathomed or continued), K is indexed to K+1 and the procedure is repeated. If at any level K, the K-list is empty or if $K > K^*(\hat{z}(b^0))$, then the procedure is terminated with the optimal correction $\hat{x}(b^0)$. The exact formulation of this procedure is presented in [38].

Before exploring the breadth-first search in more detail, we turn our attention to a depth-first search procedure to describe its basic operation. As does the breadth-first search described above, the depth-first search employs both the Form of an Optimal Correction Lemma and $K^*(\hat{z}(b))$ to restrict the domain of the search for an optimal correction. The basic operation of the depth-first search proceeds as follows.

The correction x (beginning with x=0) is tested. If x is not fathomed, choose as a new correction to be tested $x + e_{j_*}$ where

$$f(x+e_{j_*}) = \min_{j(x)\leq j\leq n} \{f(x+e_j)\,|\,x+e_j \text{ has not been fathomed}\} \qquad (3.3)$$

for some selection function f. The correction x is fathomed when all

continuations of the form $x + e_j$ for $j\varepsilon(j(x),n)$ have been fathomed. When $x=0$ has been fathomed, the optimal correction $\hat{x}(b^0)$ has been found. Again if the level of a correction x exceeds $K^*(\hat{z}(b^0))$, x is fathomed.

The selection function determines the order in which continuations of a correction x are tested in the event that x is not fathomed directly. The simplest choice for f is $f(x+e_j) = j$. As will be discussed below, a different choice for f can result in more efficient search procedures. The implicit enumeration algorithms of Balas [1] employ such a depth-first search with a selection function which minimizes the total infeasibility in the selected continuation, $x' = x+e_{j_*}$. Observe that such selection functions are equally valid for the ordering of corrections on a K-list in the breadth-first search. Also, in contrast to the breadth-first search. this search may investigate many continuations of a K-level correction before testing another K-level correction.

The introduction of group bounds into the fathoming procedures can improve the efficiency of both the breadth-first and the depth-first search procedures. In spite of this improvement, both procedures remain quite inflexible, each performing a search in a rigidly prescribed manner.

In general terms, the efficiency of a search procedure to obtain an optimal correction to (1.3) can be related to the number of corrections _explicitly_ considered. The use of bounding procedures permits some (hopefully large) fraction of the corrections to be considered _implicitly_. This, of course, is the motivation for improved bounding procedures. Notice, however, that a fundamental determinant in the extent of pruning

in the tree (implicitly eliminating corrections) is the value of the incumbent $\hat{z}(b^0)$. The maximum amount of pruning results when the incumbent is optimal. In general, the closer $\hat{z}(b^0)$ is to the optimal value, the greater the pruning will be. A graphic example of this effect is presented in [34].

Neither the breadth-first search nor the depth-first search discussed above incorporates sufficient means for exploiting this effect. Only a limited (if any) attempt is made to identify paths in the tree which lead to optimal or near optimal corrections.[1] Such corrections are considered only when they are generated by the fixed sequence used by the procedure. The breadth-first search tests <u>all</u> corrections at level K, although one of these corrections may merit investigation in depth first. The depth-first search, on the other hand, pursues a given path to a depth required to find a correction which can be fathomed with only slight regard to potential improvements in the incumbent to be realized on other paths in the tree.

Therefore, we introduce the concept of a <u>plausibility analysis</u> [34] which is intended to increase the likelihood that promising paths in the tree (i.e. those which may lead to an improved incumbent) are explored first by the search procedure. Here we introduce some definitions which we will need for our discussion of plausibility analysis:

---

[1]The use of a selection function corresponds to a limited and local attempt to discover good paths. This is an improvement, but the search still remains quite myopic.

1) A frontal node is a node of the tree of corrections which

   a) has been assigned a value by plausibility analysis

   b) has not been fathomed

   c) has no unfathomed descendants.

Another way to characterize a frontal node is to say that it corresponds to a subproblem which was considered by the supervisor, but has not yet been solved. We will examine this view in more detail below.

2) At any stage in the search, the front is the collection of all the frontal nodes in the tree, and the subproblem list is the collection of all subproblems corresponding to frontal nodes.

3) The select node is the frontal node chosen by plausibility analysis for testing. This amounts to a decision to attempt to solve the corresponding subproblem.

Basically, plausibility analysis operates in the following manner. The nodes in the front are analyzed and the most promising node is designated as the select node.[1] An attempt is made to solve the subproblem corresponding to the select node by fathoming the correction. If the current correction is not fathomed, it is continued as in the breadth-first search. The resulting nodes are added to the front. Their corresponding subproblems are assigned a plausibility value (the measure

---

[1]The select node and its subproblem are deleted from the front and the subproblem list respectively.

of promise) and are added to the subproblem list.[1] In any event, a new select node is chosen and the process is repeated. If the front (or the subproblem list) becomes empty, the search terminates with the optimal $\hat{x}(b^0)$.

In general, the sprouting of the tree from the select node increases the number of frontal nodes. The application of the plausibility analysis to these new frontal nodes may result in a number of decisions. First, the new subproblems may appear less promising than one of those temporarily abandoned at an earlier stage in the search. In this event, the focus of the search will move to a new select node in a different region of the tree.[2] If, on the other hand, one of the new frontal nodes is chosen as the select node, the search continues in the current region of the tree. Finally, during the plausibility analysis of the new frontal node, a new incumbent may be discovered. The new incumbent may obviate the necessity of solving some of the subproblems put aside earlier. In this case, the subproblems in question are removed from the front.

Thus, this <u>multiple-path</u> plausibility analysis can be thought of as developing many subproblems simultaneously. At each stage in the search, plausibility analysis selects the subproblem which it considers the most likely to lead to an improved incumbent. Through the use of plausibility analysis, the supervisor controls the search in an effort to maximize the amount of subproblem pruning which is attained.

---

[1] As will be seen below, plausibility analysis may prune some of these subproblems directly.

[2] Because there is a cost associated with moving from one region of the tree to another, we require the plausibility value of the select node in the new region to exceed those of the local frontal nodes by some minimal value.

It is important to note that the plausibility analysis employed by the supervisor is a hazard-free heuristic. The analysis is heuristic because the measure used to assess the potential value of a given subproblem cannot guarantee an optimal ordering of the subproblems in the search. Indeed, if it could, it would be a direct means for obtaining an optimal correction. At best, it is designed to improve the likelihood that on the average paths leading to good incumbents are explored early in the search. The heuristic is hazard-free, however, in the sense that it will prune an optimal correction only if the incumbent is optimal. It will never prune solutions which are better than the incumbent.

The particular plausibility value assigned to a correction x^k in the search procedure of the AGT algorithm is the greatest lower bound on an optimal continuation of x^k obtained from the group problems. Thus, if bounds have been obtained from relevant group problems 1,...,T with corresponding right hand sides β_k^t, then the plausibility value of x^k is

$$J^k = \bar{c}x^k + \max_{t=1,\dots,T} \{\bar{c}h^t(\beta_k^t;F_k^t)\} \tag{3.4}$$

and

$$h^t \text{ is either } u^t \text{ or } v^t \text{ (see (2.26), (2.27)).} \tag{3.5}$$

Clearly if at any time $J^k \geq \hat{z}(b^0)$, then x^k is fathomed.

As indicated in Section 2, a number of ways are available for bounding a given correction x^k. In the following discussion of subproblem analysis, we will investigate the manner in which bounding procedures are chosen

for a particular subproblem.

Several comments about the effect of plausibility analysis are relevant here. Notice that the subproblem selection procedure employed means that the search is in a sense intermediate between a depth-first and a breadth-first search. A very promising path (as indicated by the plausibility analysis) may be explored in depth immediately. If the current path appears less promising than some other path in the tree, attention is switched to the new path. This helps avoid the "single-mindedness" of the strictly depth-first search. Also note, that sub-problems which are not solved at one stage in the search may be pruned without further analysis later if a new incumbent is found.

Plausibility analysis is really of use only when it is believed that the incumbent can be improved. If it is believed that the incumbent is optimal, then plausibility analysis should be abandoned in favor of the linear search described in Section 12 . Therefore, the supervisor should monitor the progress of the search and attempt to estimate the potential improvement in the incumbent. At some point, when it is determined that little improvement in the incumbent can be expected, plausibility analysis should be suppressed. A more extensive discussion of this point is presented in Section 10.

One final point should be made about the plausibility value used in our search procedure. It uses only a bound from the group problem as a measure of the promise of a given subproblem. A better measure might be one which incorporated the number of free variables for the correction

as well as the bound. Whether any additional benefit can be derived from such a modification is a matter for further investigation.

3.3 Subproblem Analysis

We now turn to a discussion of the manner in which a selected problem is tested. The collection of methods employed by the supervisor in an attempt to fathom a given subproblem is called subproblem analysis. At the heart of subproblem analysis is the processes which marshall information gleaned from the analysis of past subproblems and integrate it into the analysis of the current subproblem. Also incorporated in this part of the supervisor is information obtained from the analysis of other IP problems. In this section we will discuss the information-gathering or diagnostic function of the supervisor. The foundation for the particular diagnostic function described here is the theoretical analysis presentedin Section 2.3.

The correction x^k is the last in a series

$$P(x^k) = \{x^{k_P}\}_{P=0}^{K(x^k)} \tag{3.6}$$

Notice that $|P(x^k)| = K(x^k)+1$. Much of the information which is relevant to the analysis of x^k is associated with $P(x^k)$. Of major importance is the sequence of cones penetrated and repenetrated by $P(x^k)$. Recall that the cones in this sequences are of the form

$$K_t = \{b | B_t^{-1} b \geq 0\} \tag{3.7}$$

where the condition $1 - \rho_i^t b \geq 0$ for $i \varepsilon S$ is understood to hold implicitly. (See Section 2.3.) Therefore, the determination of a dual feasible

basis B_t such that $B_t^{-1}(b^0-Rx^k)$ identifies the cone of primary interest in fathoming x^k. The AGT Algorithm does not necessarily obtain a basis B_t for each correction x^k. As a result the identity of certain cones corresponding to corrections in $P(x^k)$ may not be known. If, however, a correction x^k is such that $b^0 - Rx^k \in B_t$ for some previously determined B_t, this fact will be noted by the algorithm. The supervisor may or may not decide to determine an optimal LP basis for a right hand side $b^k = b^0 - Rx^k$. The considerations involved in this decision will be discussed below.

Therefore we define

$$C(x^k) = \{t_p\}_{P=0}^{K(x^k)} \quad . \tag{3.8}$$

as the index set for the cones penetrated and repenetrated by the $P(x^k)$ with the provision that $t_p = 0$ whenever the cone (basis) corresponding to x^P at level P in $P(x^k)$ either has not been identified or has been erased by the supervisor as will be discussed below.

The set $C(x^k)$ is an important factor in determining fathoming strategies at x^k. If, for example, $C(x^k) = \{\ldots, 1,\ldots 1, 2,\ldots,2, 3,\ldots,3\}$, then it is clear that the group solutions preferred for bounding x^k are those for group 3 followed by those for group 2 and so forth. If, on the other hand, $C(x^k)$ is a more random sequence, the preference ordering is less clear.

Another consideration is the number of cones in $C(x^k)$. If this number is large in some sense and the sequence $C(x^k)$ is somewhat random,

then solving the group problems for the subproblem x^k may be relatively ineffective. If the number of cones penetrated is small, the group problems may be more useful. Our reasoning here is the following. If $b^0 - Rx^k$ represents the end of a long path in m-space starting at b^0, which has penetrated just a few cones of the form (3.7), then the majority of the activities a_j are such that $B^{-1}a_j$ is an order of magnitude less than $B^{-1}b$ for bases B and right hand sides b encountered in the path from b^0 to $b^0 - Rx^k$. Thus, there is a greater likelihood that group corrections fathom enumerated corrections by providing feasible, and hence optimal continuations.

Another good source of information derives from the dominance relations developed in Section 2.3. In general, we observe that the solution of a group problem t is relevant to the analysis of x^k provided that $(F_k \; \sigma_1) \quad L_t = \phi$ where $L_t = \{a_j | c_j - \pi_t a_j < 0\}$. In addition, we note that if this condition holds <u>and</u> $(F_k \cup \sigma_1) \cap \sigma_t^c \subseteq U^t$, the solution to group problem t can be simply retained for use in the analysis of x^k. This provides us with a very simple fathoming test. In the event that the retrieved group solutions fail to fathom x^k, we can dynamically resolve any or all of these group problems for the group structures G_t, t=1,...,T.

If we cannot fathom x^k using this approach, we still have the option of obtaining a new basis for the cone containing $b^0 - Rx^k$, and solving the corresponding Unconstrained Group Problem. Below we will incorporate these alternatives into our procedure for subproblem analysis.

With these considerations in mind, we can turn to the development

of a diagnostic procedure which the supervisor can use as in subproblem analysis. There are a number of options available to the supervisor in this regard. First, recall that any dual feasible basis for (1.1) can be used to obtain a group structure over which a group problem can be solved in attempting to fathom subproblem x^k. If more than one such group structure is available, the supervisor must have some means for ranking them in terms of their potential value for fathoming x^k. Once a basis with its corresponding group structure has been selected, the choice of solving group problems (2.26) or (2.27) must be made.[1] Both the choice of a group structure as well as the choice of a group optimization problem can be based on information from two sources: 1) diagnosis of the status of the IP problem at hand, and 2) historical experience with other IP problems. Here we will present one relatively simple scheme for incorporating information of both kinds into subproblem analysis. This scheme is undoubtedly incomplete. Hopefully it will be improved as a result of computational experience on the one hand and deeper insights into the structure of IP problems on the other.

First, we focus on the choice of a group optimization problem ((2.26) or (2.27)) given a group structure. Our initial computational experience with our algorithms for solving the Unconstrained and the Zero-one Group Optimization Problems respectively indicates that the

---

[1]Note that both might be solved. If solving (2.26) fails to fathom x^k, there still remains the possibility that solving (2.27) may succeed.

computational cost of the latter is about 5 times that of the former for the same group structure. In addition, the dynamic version of the algorithm for the Unconstrained Problem permits this problem to be solved at each x^k (i.e. for right hand side $b^k = b^0 - Rx^k$) at essentially zero set-up cost. In order to see this, notice that each new correction x^k is a continuation $x^\ell + e_j$ of some previously enumerated correction x^ℓ. Thus, the optimal group table for the Unconstrained Problem can be updated by deleting only one arc and reoptimizing (see Section 8). Hence with regard to computation alone, the Unconstrained Problem is far more easily solved than the Zero-one Problem.

If, however, there are zero-one constraints in (1.3), the corrections and bounds obtained from the Zero-one Problem may be better than those from the unconstrained problem. This difference may be significant in pruning subproblems from the tree. Our current feeling, however, is that on the whole the Unconstrained Problem is to be favored. The Zero-one Problem should be used only when the performance of the unconstrained algorithm falls below some prescribed level. In this way, the <u>mesa phenomenon</u> discussed by Minksy [41] and widely observed in IP [42] hopefully can be avoided. This term refers to the situation in which a search procedure encounters a relatively wide region (mesa) in the solution space for which no significant improvement in the objective function is obtained. Our version of the mesa phenomenon is the failure of the Unconstrained Group Problem to provide significant improvements in bounds or the completion of a path in the tree as the path is developed.

In such a case, we may turn to the Zero-one Group Problem (only if it is relevant, of course) in the hope that solving this problem (at greater computational cost) will provide us with a marked improvement in the bound which will fathom the current subproblem.

In what follows we will assume the above strategy for employing the Zero-one Group Problem. It is important to note at this point that we will not solve dynamically the Zero-one Group Problem. Instead we employ the dominance relations between group problems discussed in Section 2.3. The Zero-one Group Problem will always be solved at $x^0=0$. It will be solved at other corrections only if the supervisor deems the value of the Unconstrained Group Problems to be sufficiently small as to merit it. Whenever a new group structure is found, the Unconstrained Group Problem corresponding to it will always be solved.

Let the supervisor maintain a <u>diagnostic table</u> with an entry for each <u>active</u> basis. An active basis is a dual feasible basis for (1.1) which has been generated and retained during the search to date. An entry in the diagnostic table consists of the basis (and σ and L) the corresponding group structure and unconstrained problem solution, the set U and a counter δ.

Because of storage limitations, the number of entries in the diagnostic table is limited. As a result, after the table has been filled, entries are purged whenever the supervisor considers them to be of little use. This permits new (hopefully more useful) entries to be created.

A counter δ_0 is maintained for the diagnostic table and it is incremented by one every time any entry in the table is used by the fathoming procedures. Each entry t in the table contains a counter δ_t which is the value of δ_0 at the time this entry was last successfully used in fathoming a correction, or if the entry has never been so used, δ_t is the value of δ_0 at the time the entry was created. A performance measure, ε_t, is defined as follows

$$\varepsilon_t = \frac{1}{1+\delta_0-\delta_t} \tag{3.9}$$

The longer an entry remains without a successful use in fathoming, the lowerthe value of ε_t will be. As will be indicated below, ε_t is used both in preference ordering the table entries for a given correction x^k as well as in controlling the use of the Zero-one Group Problem and the purging of entries from the diagnostic table.

Whenever the supervisor through plausibility analysis selects a subproblem for testing, it invokes subproblem analysis. The purpose of subproblem analysis is to use its resources to the extent indicated by the supervisor in an attempt to fathom the subproblem. The supervisor controls the expenditure of effort in subproblem analysis, because at certain points in the search, some methods available for subproblem analysis may be deemed to be of relatively little value.

As noted, the value of alternative strategies for subproblem analysis

may vary during the search. Consider, for example, obtaining a new basis for a correction x^k. If the basis will appear in many $C(x^p)$, then, determining it may have both a long range as well as a short range value. Whether a particular basis will appear in other $C(x^p)$, then, should influence the decision to create and save it. One simple measure of the potential usefulness of a given basis is the number of free variables in F_k for the x^k in question. If this number is relatively large, the basis is of potential value in many fathoming tests (all the tests of descendants of x^k).

On the other hand, if a path $P(x^k)$ is very long (i.e. F_k contains few elements), then the value of obtaining a basis for x^k probably is much less.

Similarly, if we consider the continuation of x^k of the form x^k+e_j for $j=j(x),\ldots,n$, we see that the number of descendants of $x^k+e_{j_1}$ is greater than that of $x^k+e_{j_2}$ if $j_1 < j_2$. (Recall the Form of the Optimal Correction Lemma.) Hence we are less inclined to permit the creation of new diagnostic table entries for $x^k+e_{j_2}$ than we are for $x^k+e_{j_1}$.

In keeping with these considerations, we propose the following simple scheme for controlling the computational effort expended in subproblem analysis as a function of depth and lateral position in the tree. Define the <u>threshold vector</u>, $\omega = \{\omega_i\}_{i=1}^n$ where $\omega_i \geq \omega_{i+1}$ to be a vector of constants. Let the threshold ξ of a subproblem be defined recursively as follows: 1) $\xi(x^0) = \xi^0 = 1$ and 2) Assume that we continue the correction x^k with threshold ξ^k and that the set of

continuations $\{x^k+e_{j_i}\}$ has been ordered (by plausibility analysis for example). Then the threshold of the ith correction in the list of continuations is given by $\xi(x^k+e_{j_i}) = \xi^k \cdot \omega_i$.

As will be seen below, the threshold of a node is used to control the extent of subproblem analysis devoted to that node. As more computational experience with the algorithm is obtained, we will be able to improve on this simple control mechanism.

The first tests applied by subproblem analysis to a correction x^k are real space feasibility tests. The first tests $y^k = b^0 - Rx^k$ for feasibility. If y^k is feasible, x^k is fathomed. The second test is directed at proving that x^k has no feasible completion better than the incumbent (See Section 2.3.). If these two tests fail to fathom x^k, the entries in the diagnostic table are employed.

First, the set τ_1^u is determined by

$$\tau_1^u = \{t \mid F_k \cup \sigma_1) \cap L_t = \phi\}. \tag{3.10}$$

The set τ_1^u is the set of entries relevant to the correction x^k. The scan of the table also yields a second set $\tau_2^u (\subseteq \tau_1^u)$ defined by

$$\tau_2^u = \{t \mid t \in \tau_1^u \text{ and } (F_k \cup \sigma_1) \cap \sigma_t^c \leq U^t$$

The set τ_2^u contains the indices of all the table entries from which the group solution can simply be retrieved for use in testing x^k. If continuations (x^k+e_j) was used [illegible] (by [illegible]) analysis for example). Then the [illegible] of [illegible] in the list of continuations be given [illegible].

As will be [illegible], [illegible] is used to control the extent of [illegible] analysis [illegible]. [illegible] more computational experience with [illegible], we will be able to improve on [illegible].

The [illegible] analysis [illegible] correction z^k are real [illegible] $- Rz^k$ for feasibility [illegible] second test is [illegible] than the incumbent [illegible] tation z^k, the [illegible]

[illegible]

(3.10)

[illegible]

no entry $t \varepsilon \tau_2^u$ provides a bound which fathoms x^k, the dynamic version of the algorithm for the Unconstrained Group Problem mav be used. Its use is determined by a threshold test $\xi^k > \xi_1$ for the threshold ξ_1. That is, if the condition $\xi^k > \xi_1$ holds, then subproblem analysis is permitted to update certain group tables and to attempt to fathom x^k with the resulting bounds. If the condition does not hold, however, then attempts to fathom x^k directly are abandoned. We do this because the portion of the tree below the node x^k is judged to be too small to warrant the use of these methods.

The value of ξ_1 is determined from computational experience.

If $\xi^k > \xi_1$, the set τ_2^u is ordered in accordance with $C(x^k)$ as follows. The last (most recent) ℓ_0 elements of $C(x^k)$ are used to generate a ranking for each entry $t \varepsilon \tau_2^u$. The rank of entry t is given by

$$E_t = \sum_{p \varepsilon Q(t)} \xi_t^{P-\ell_0} \qquad (3.12a)$$

where

$$Q(t) = \{p | p \geq \ell_0 \text{ and } t_p = t \text{ for some } t_p \varepsilon C(x^k)\} \qquad (3.12b)$$

For example if $C(x^k) = \{\ldots,1,2,1,1\}$ and $\ell_0 = 4$, we have $E_1 = 1 + \xi_1 + \xi_1^3$ and $E_2 = \xi_2^2$.

The entries $t \varepsilon \tau_2^u$ are ranked in terms of decreasing E_t. Then the dynamic version of the algorithm for the Unconstrained Group Problem is

used to update the group solution indicated by the first $t \in \tau_2^u$. If the solution fails to fathom x^k, the second $t \in {}_2$ is used. This continues until $E_t \leq E_{min}$ for some supervisor-controlled threshold $E_{min} \leq 1$. Therefore at least one (but not necessarily all) the entries in τ_2 will be used. If none of the selected elements in τ_2 lead to the fathoming of x^k, the next stage of subproblem analysis is entered.

This stage is concerned with the identification of the relevant cone for x^k and the possible inclusion in the diagnostic table of a corresponding entry. First the indicator Δ^k is checked. (See Section 2.3.) If $\Delta^k \neq 0$, then Δ^k is an index t in the diagnostic table. This index was set when x^k was generated as a correction in the path $P(x^k)$, and indicates the relevant cone for x^k. Therefore, when $\Delta^k \neq 0$, no further identification of the cone for x^k is required.

If $\Delta^k = 0$, the cone for the immediate predecessor of x^k in $P(x^k)$ was not determined when x^k was generated. Subproblem analysis seeks to establish an entry for the new cone. If, however, $\xi^k \leq \xi_2$, the supervisor will not permit a new entry in the diagnostic table, and hence the analysis of x^k terminates with a failure to fathom the correction. In the case that $\xi^k > \xi_2$, the dual simplex method is used to obtain a primal feasible basis for (1.1) with a right hand side b^k. The resulting basic variables σ_{T+1} are tested against σ_t for each entry in the diagnostic table. If $\sigma_{T+1} = \sigma_t$ for some entry in the table, no new entry need be created. If not match is found, a new table entry is created, replacing the current

entry with the poorest performance measure.[1] An attempt is made to fathom x^k with the new group solution. This is the last step in subproblem analysis of x^k.

In the above discussion, we have deliberately simplified matters by ignoring the Zero-one Group Problem. We can remedy this somewhat with the following procedure for use when the Zero-one Problem is relevant. Recall that the Zero-one Problem is always solved at x=0. It is not solved again unless the performance of the Unconstrained Problem falls below some prescribed level. For example, after the entries in the diagnostic table have been ranked prior to the use of the dynamic version of GTIP1, the value of $E_{max} = \max \{E_t\}$ is compared with a threshold. If it is greater than the threshold, the Unconstrained Problems are used. If not, the Zero-one Problem is solved for the group structure associated with E_{max}. This is used in attempting to fathom x^k. No further fathoming tests are employed for this subproblem. Again the use of the Zero-one Problem can be controlled for correction x^k by making it conditional on $\xi^k > \xi_3$ for some ξ_3.

Solutions to Zero-one Problems can be saved for use at other nodes in the tree. Thus in the phase of subproblem analysis in which retrievals are tested, retrievals from Zero-one Solutions can be used as well. The schemes for managing the storage of diagnostic table entries outlined

---

[1]If this entry is t, then for all $G(x^p)$ which contain a $t_p = t$ for some p, we must set $t_p = 0$. Similarly, all $L^p = t$ must be reset to zero.

above can easily be adapted for use here.

While this discussion is incomplete, it hopefully provides some insights into the diagnostic procedure designed for the AGT Algorithm. This procedure undoubtedly will be refined as computational experience is obtained.

3.4 Pre-Search Analysis

In this Section, we consider procedures to be used before the implicit search through the tree of feasible corrections is performed. Specifically, we will consider the following topics:

1) Ordering or ranking schemes for the columns and rows of (1.3);

2) Procedures to handle problems when $D = |\det B|$ is too large for any optimal LP basis B for (1.1);

3) Procedures for finding an initial feasible solution.

It has been observed that the performance of existing IP codes can depend significantly on the ordering of the columns and rows. To a certain extent, the supervisory procedures discussed above tend to reduce this sensitivity to the columns and rows. However, it appears to us that the ordering can still be important. At this time, we will not develop a rigorous algorithm for re-ordering, but rather, we will discuss qualitatively some of the relevant factors we have perceived.

The reader may have noticed that the continuation rule implied by the Form of an Optimal Path Lemma is a basic tool of all our algorithms. With this rule, variables of higher index tend to be used more often than variables of lower index, and the ordering choice should be influenced by this consideration. Thus, when solving unconstrained group problems, reordering the non-basics x_j, $j \varepsilon (F_k \cup \sigma_1) \cap \sigma_t^c$ by decreasing relative cost factors $\bar{c}_j^t$ appears to promote efficiency. Moreover, we assume that the non-basics $x_1, \ldots, x_n$ relative to B_1 are ordered so that $\bar{c}_1^1 \geq \bar{c}_2^1 \geq \ldots \geq \bar{c}_n^1$.

Computational experience with the algorithm for the zero-one group problem is as yet fairly limited, but the indications are that a favorable ordering is to place the non-zero-one non-basics first, followed by the zero-one basics.

For a brief discussion of the problem of choosing an optimal ordering for the knapsack problem, see [36; p. 331]. Balas also discusses the problems of an optimal ordering in [2]. As we shall see, the ordering of the rows is important only if the groups induced by optimal LP bases are too large. We turn our attention to this problem.

One of the initial drawbacks of the group theoretic analysis of IP problems is the possibility that the groups encountered will be too large, or equivalently, that the basis determinants will be too large. Computational experience indicates that group problems of order 5000 or less can be easily handled by the existing group algorithms. Nevertheless, the algorithms are coded to handle groups with orders as high as 50,000. For a further discussion of computational experience, along these lines, see Section 10.

Our purpose in the paragraphs below is to describe a method for handling problems for which the optimal LP basis determinant D is too large. Of course, it is possible for D to be too small for meaningful combinatorial resolution from the group theoretic methods. However, we ignore in this Section the possibility that D may be too small. Suppose that we set an upper bound D_0 on the size of groups with which we would like to work.

For simplicity, let us consider problem (1.1) with an empty set S of zero-one variables. The argument can be easily adapted when $S \neq \phi$. Without loss of generality, we can assume that $c_j \geq 0$, $j=1,\ldots,m+n$ in (1.1). To see that this is so, let u_j be an upper bound on x_j with $c_j < 0$ and make the substitution $x_j' = u_j - x_j$ in (1.1). Henceforth, we assume (1.1) has the desired form. Reorder the rows of (1.1) so that the constraint judged most important is first, the second most important constraint is second, etc. In this context we mean by an important constraint that the constraint can be used to yield significant combinatorial resolution. As an example of this, consider the IP formulation of the traveling salesman problem stated in [7;p. 547]. In addition to an imbedded assignment problem, this formulation contains sub-tour breaking constraints. If the problem was solved first as an assignment problem, and then the sub-tour breaking constraints afterwards, the important constraints would be those that break up the sub-tours present in the assignment problem solution. A synthesis of group theoretic and branch and bound method for the travelling salesman problem is being considered in [5].

We proceed to solve (1.1) as an LP problem using the Dual Simplex algorithm. Since $c_j \geq 0$, we can take the surplus solution with basis $-I$ as the initial dual feasible basis. Suppose now that we are at iteration k of the following modified dual simplex procedure. Let $y_1,\ldots,y_m$ denote the basic variables and $x_1,\ldots,x_n$ denote the non-basics. We have min z subject to

$$z = z_0 + \sum_{j=1}^{n} \bar{c}_j x_j$$

$$y_i = \bar{b}_i - \sum_{j=1}^{n} \bar{a}_{ij} x_j, \qquad i=1,\ldots,m$$

x_j, y_i non-negative integer.

Define i_k by

$$i_k = \min \{i | \bar{b}_i < 0\} .$$

We are interested in eliminating the infeasibility on row i_k. Let x_k be the non-basic variable chosen to enter the basis. It can easily be shown that the determinant of the new basis will be $D\rho_{i_k} a_r$ where ρ_{i_k} is row i_k of the current basis (with $|\det| = D$). Thus we make the change of basis only if $D\rho_{i_k} a_r \leq D_0$. If the change is made, the above procedure is repeated.

On the other hand, if $D\rho_{i_k} a_r > D_0$, we do not make the indicated change of basis. We can either choose to eliminate the infeasibility on a different row, or convert (1.1) to group optimization problems with respect to the current basis. If these problems yield a feasible solution to (1.1), then it is optimal. If the group optimization problems fail, then we begin again with the dual feasible basis-I and repeat the above procedure with the following exception. Reorder the rows so that the first k_1 rows were dual feasible for the basis just

obtained. Successively eliminate the infeasibilities on the rows $k_1+1,\ldots,m$ until either all infeasibilities have been eliminated or the determinant becomes too large. In the latter event, extract a new basis and repeat the group analysis.

The solution of dual feasible bases should continue until allocated storage is exceeded or all the (important) constraints are covered in the sense that there is at least one extracted dual feasible basis such that any given constraint is feasible with respect to that basis.

We remark in passing that it may be worthwhile to solve the group problems derived above as one multi-dimensional group problem. For example, if two groups G_1 and G_2 are found, then we form the two dimensional group $\{(\lambda_s^1, \lambda_t^2): s=0,1,\ldots,D_1-1; t=0,1,\ldots,D_2-1\}$ and look for an unconstrained shortest route path connecting (θ^1, θ^2) to (β_0^1, β_0^2) where the arcs are of the form (α_j^1, α_j^2) with arc costs $\bar{c}_j^1$.

As for the procedures for finding an initial feasible solution, we suggest two. The first is the backtracking algorithm of [39]. An adaptation of this algorithm may be useful to the implied procedures of Theorem 1 of [19] for generating good cuts. A second method for finding an initial feasible solution is Balas' algorithm in [1] or [11].

4. THE ADAPTIVE GROUP THEORETIC ALGORITHM

In this section, we present an outline of the AGT Algorithm. The description of the algorithm is at a relatively high level and a number of details are omitted. We have chosen to present the algorithm in this way not because we consider the detail to be unimportant, but because the inclusion of all the detail would obscure the overall structure of the procedure. Also, parts of the algorithm will undoubtedly be modified in the light of computational experience. Thus, the version presented here is really a prototype of the AGT Algorithm. All the "bookkeeping" details have been eliminated as well as the portions of the algorithm concerned with threshold settings. With these exceptions, the analytical realizations of steps in the algorithm have been discussed in Sections 2 and 3.

1) Solve the integer programming problem as a linear programming problem. Use the resulting optimal LP basis to transform to the corresponding group problem. Solve the resulting Unconstrained Group Problem for $x^0 = 0$. If x^0 is fathomed, terminate. Otherwise solve the Zero-one Problem for $x^0 = 0$. If x^0 is fathomed, terminate. Otherwise go to Step 23 with $x^k = 0$. (Section 3.4).

2) If a new incumbent has been found, prune from the subproblem list all subproblems with plausibility values greater than $z(b^0)$.

3) If the subproblem list is empty, terminate.

4) If plausibility analysis is active, use it to remove a new subproblem from the subproblem list. Go to Step 6 , with this subproblem x^k.

5) Remove the first subproblem from the subproblem list. Use linear search subalgorithm to solve this subproblem. Go to Step 2. (Section 7)

6) Determine the set τ_2^v of the Zero-one Group Problems from which retrievals can be used in attempting to fathom x^k. (Section 3.3)

7) If τ_2^v is empty, go to Step 9.

8) For each $t\epsilon\tau_2^v$, attempt to fathom x^k with a retrieval. If x^k is fathomed for some t, go to Step 2. (Section 2.3)

9) Determine the set τ_2^u of the Unconstrained Group Problems from which retrievals can be used in attempting to fathom x^k. (Section 3.3)

10) If τ_2^u is empty, go to Step 13.

11) For each $t\epsilon\tau_2^u$, attempt to fathom x^k with a retrieval. If x^k is fathomed for some t, go to Step 2. (Section 2.3)

12) Rank the problems in τ_2^u in accordance with the measure E. Let τ_3^u be the set $\{t | t\varepsilon\tau_2^u$ and $E_t > E_{min}^u\}$. If τ_3^u is empty, go to Step 14. (Section 3.3)

13) For each $t\varepsilon\tau_3^u$, update the indicated Unconstrained Group Problem solution and attempt to fathom x^k. If x^k is fathomed, go to Step 2. (Section 8)

14) If the updating of Zero-one Group Problems is not permitted, go to Step 17.

15) Rank the problems in τ_2^v in accordance with the measure E. Let τ_3^u be the set $\{t | t\varepsilon\tau_2^v$ and $E_t > E_{min}^v\}$. If τ_3^v is empty, go to Step 17. (Section 3.3)

16) For each $t\varepsilon\tau_3^v$, update the indicated Zero-one Group Problem solution and attempt to fathom x^k. If x^k is fathomed, go to Step 2. (Section 9)

17) If no new entry is permitted in the diagnostic table, go to Step 23.

18) Determine the relevant cone for x^k either from $L^k(L^k \neq 0)$ or from the use of the Dual Simplex method $(L^k=0)$. If $L^k \neq 0$, go to step 23. (Section 2.3)

19) Let σ_s be the set of basic variables for the basis found in Step 18 (for $\Delta^k=0$). If $\sigma_s = \sigma_t$ for some t in the diagnostic table, set $\Delta^k=t$, add the appropriate element to $c(x^k)$, and go to Step 23.

(20) Create a new entry, t*, for the diagnostic table (perhaps deleting an old entry). Set $\Delta^k = t^*$ and add appropriate element to $c(x^k)$. Use the solution to the Unconstrained Group Problem for the new entry in an attempt to fathom x^k. If x^k is fathomed, go to Step 2. (Section 3.3)

21) If a Zero-one Problem is not permitted for the new group structure go to Step 23.

22) Solve the Zero-one Group Problem for the new group structure and attempt to fathom x^k. If x^k is fathomed, go to Step 2.

23) Generate continuations x^k+e_j for $j=j(x),\ldots,n$. Add x^k+e_j to the subproblem list only if the bounds computed from (2.47) are less than $\hat{z}(b^0)$. Associate with each x^k+e_j added to the subproblem list: Δ^k, $c(x^b)$, ξ^k, and a plausibility value. Go to Step 4. (Section 2.3)

8. CONCLUSION

Although we feel that the above discussions in total are a significant step toward the development of an adaptive group theoretic algorithm for the integer programming problem, it is evident that further research is required in certain areas. First, there are a number of theoretical developments which are as yet incomplete. These include first the group representational procedures discussed in Section 11. Very efficient representational procedures can be developed for problems with special structures, if these structures can be recognized without excessive computational effort.

The development of better algorithms for the Unconstrained and Zero-one Group Problems based on supervisory procedures similar to those discussed in Section 3 is also a promising area. Finally, an extension of the group theoretic methods to the mixed integer problem appears possible, and such a development would greatly enhance the value of these methods.

Not only do we wish to extend the group theoretic methods *per se*, but we also seek to characterize the fundamental structure of the problem in order to construct better supervisory procedures. Finally, we need further computational experience in order to appropriately set the various threshold parameters for the supervisor. Moreover, the additional computational experience will lead to improved supervisory procedures and deeper theoretical insights into the problem.

Section 6. Appendix A - Computational Experience

Omitted in preliminary version: see [38], [39] for partial results.

Section 7. Appendix B - Linear Search Sub-algorithm

Omitted in preliminary version: see [39] for a linear search, group theoretic IP algorithm.

Section 9. Appendix D - Algorithm for the Zero-one Group Problem

Omitted in preliminary version: see [39].

Section 11. Appendix F - Group Representational Algorithm

Omitted in preliminary version: see [37].

8. APPENDIX C

STATIC AND DYNAMIC ALGORITHMS FOR THE UNCONSTRAINED GROUP PROBLEM

This appendix begins with a statement of the algorithm from [37] for solving the Unconstrained Group Problem (2.11). We call this algorithm GTIP1 (Static). The algorithm is stated for the generic Group $G = \{\lambda_s\}_{s=0}^{D-1}$ (with the added element λ_D) and the generic set $F = \{f_1,\ldots,f_t\}$. There follows an adaptation of GTIP1 to be used when F has changed to some set F' such that $F \cap F' \neq \phi$. This algorithm is called GTIP1 (Dynamic).

GTIP1 (Static)

STEP 1 (Initialization) Set $Z(\lambda_2) = Z(D+1)$ for s=1,...,D, where

$$Z = \max_{f\varepsilon F} \overline{c}_f;$$

set $Z(0) = 0$. Also, set $j(\lambda_s) = 0$ for $s = 0,1,2,\ldots,D$. For all $f\varepsilon F$ and $\alpha_f \neq \theta$, set $Z(\alpha_f) = \overline{c}_f$, $j(\alpha_f) = f$, and $a_{\alpha_f} = 2$ only if $\overline{c}_f < Z(\alpha_f)$. For $S\varepsilon F$ and $\alpha_f = \theta$, set $Z(\lambda_D) = c_f$, $j(\lambda_D) = f$ only if $\overline{c}_f < Z(\lambda_D)$. For all nodes λ_s for which a_{λ_s} is not specified set $a_{\lambda_s} = 1$. Go to Step 2 with $\lambda_2 = \theta$.

STEP 2 (Stop if $a_{\lambda_s} = 1$ for $s = 0,1,\ldots,D$. Then $z_u(\lambda_s;F) = Z(\lambda_s)$ for s=0,1,...,D, and the optimal group solutions can be found by backtracking.

Otherwise (1) if there is a s' > s such that $a_{\lambda_{s'}} = 2$, index s to s', or (2) if there is no s' > s such that $a_{\lambda_{s'}} = 2$, index s to the smallest s"(<s) with $a_{\lambda_{s''}} = 2$. Go to Step 3.

STEP 3 For $f \varepsilon F$ and $f \geq j(\lambda_s)$, and $\lambda_s + \alpha_f \neq \theta$, set $Z(\lambda_s + \alpha_f) = \bar{c}_f + Z(\lambda_s)$, $j(\lambda_s + \alpha_f) = f$, and $a_{\lambda_s + \alpha_f} = 2$ only if $\bar{c}_f + Z(\lambda_s) < Z(\lambda_s + \alpha_f)$. For $f \varepsilon F$ and $f \geq j(\lambda_s)$, and $\lambda_s + \alpha_f = \theta$, set $Z(\lambda_D) = \bar{c}_f + Z(\lambda_s)$ and $j(\lambda_D) = f$ only if $\bar{c}_f + Z(\lambda_s) < Z(\lambda_D)$. Return to Step 2.

In order to adapt this algorithm to changing F, note that GTIP1 (Static) finds a tree of shortest route paths connecting θ to λ_s, s=1,...,D. These paths and their values are recorded in the optimal group table which has three columns: λ_s, $j(\lambda_s)$, $Z(\lambda_s)$. If F has changed to F' and F' is not substantially different from F, then part or most of the shortest route tree relative to F can be retained and used in finding the tree relative to F'. To this end, let $Q \equiv F \cap F';$, $R \equiv F^c \cap F'$, $W \equiv F \cap (F')^c$.

An intuitive explanation of GTIP1 (Dynamic) is the following. First, go through the optimal group table and for each λ_s such that $j(\lambda_s) = f$ for $f \varepsilon W$, set $j(\lambda_s) = 0$ and $Z(\lambda_s) = (D+1) \min_{f \varepsilon F'} \bar{c}_f$. In other words, remove the discarded arcs. Second, after ordering Q and R as desired, reorder F' so that F' = {Q,R} . Thus, the new activities are placed last. Then set $a_{\lambda_s} = 2$ for s = 1,...,D and use GTIP1 (Static) to reoptimize.

<u>[illegible] (Dynamic)</u>

STEP 1 (Initialization) Assume the elements in F' are ordered so that F' = {1,2}. For s=1,...,S, if [illegible] set [illegible] where

$$[illegible] = \max_{f \in F} [illegible] .$$

For all $f \in F$ and [illegible], set [illegible], and [illegible] only if [illegible]. For all $f \in F$ and [illegible], set [illegible] only if [illegible]. Set [illegible], s=1,...,S-1; [illegible]. Go to Step 2 with [illegible].

STEP 2 Unchanged, except we are now finding [illegible].

STEP 3 Unchanged.

10. APPENDIX E

CONTROL OF PLAUSIBILITY ANALYSIS

In deciding whether plausibility analysis is potentially useful, the supervisor draws on historical information which has been collected during the search to date. Basically the supervisor solves this problem through the prediction of future values of the incumbent. If this prediction indicates that it is unlikely that the current incumbent will be displaced, plausibility analysis is suppressed and the search is conducted in linear mode.

Predictions about the future value of the incumbent can be formulated in numersou ways. Certainly computation experience will yield significant insights into the form such predictions should take. Here we offer a prediction scheme which is simple to use and can be adapted to reflect computational experience.

Two factors enter into this scheme. The first and more obvious of these is the historical behavior of the value of the incumbent. The value of the incumbent as a function of time is easily visualized. It is a constant until a new incumbent is found, at which point it is a new, smaller constant and so on. It can be argued heuristically that the use of plausibility analysis by the supervisor causes the successive differences in incumbent values to decrease on the average as the optimal correction is approached [41]. One factor considered by the supervisor, then, in deciding whether plausibility analysis should be discarded is the historical behavior of the decreases in the incumbent.

The second factor is the number of subproblems fathomed by a given incumbent. The heuristic argument here is that the more subproblems fathomed by an incumbent, the more likely the incumbent is to be optimal. In a sense, this measures the stability of the incumbent. The predictive function of the supervisor assumes that the closer the incumbent value is to the optimal value, the greater the number of fathomed subproblems will be. As above, this factor is developed so as to reflect historical experience. The function employed by the supervisor is as follows:

$\overline{\Delta Z}_i = -(Z_i - Z_{i-1})$ where Z_i is the value of the ith incumbent

f_i = the number of subproblems fathomed by the ith incumbent

$$\overline{\Delta Z}_i = \alpha_z \Delta Z_i + (1-\alpha_z)\overline{\Delta Z}_{i-1}$$

$$\overline{f}_i = \alpha_f f_i + (1-\alpha_f)\overline{f}_{i-1}$$

where $\overline{\Delta Z}i$ and $\overline{f}_i$ are predicted values and α_f, and α_Z are constants.

Let

$$V_{Z,i} = \begin{cases} 0 \text{ if } \overline{Z}_i \geq \Theta_Z \\ 1 \text{ if } \overline{Z}_i < \Theta_Z \end{cases}$$

$$F_{f,i} = \begin{cases} 0 \text{ if } \overline{f}_i \leq \Theta_f \\ 1 \text{ if } \overline{f}_i > \Theta_f \end{cases}$$

where Θ_Z and Θ_f are constants determined by the problem in question.

Finally $\overline{T}_i = c_f T_{i,f} - c_s T_{i,s}$ for constants c_f, c_s and if $T_i \leq 1$ plausibility analysis is abandoned when the i-th incumbent is found. Thus plausibility analysis can be abandoned either because:

1) The value of the current incumbent is believed to approximate that of the optimal correction, or

2) The increasing stability of incumbents indicates that many subproblems are being fathomed per incumbent and plausibility analysis is not required.

One additional control should be added. Periodically, the supervisor should check the value of f_i. If $f_i > k \cdot \overline{f}_i$ for some $k > 1$, then plausibility analysis should be suppressed. This situation can arise if the optimal incumbent is found very early in the search before sufficient historical data has been acquired for use by the control function.

REFERENCES

1) Balas, E., "An Additive Algorithm for Solving Linear Programs with Zero-one Variables," Operations Research, 13, pp. 517-546 (1956).

2) Balas, E., "Discrete Programming by the Filter Method," Operations Research, 15, pp. 915-957 (1967).

3) Balas, E., "Duality in Discrete Programming," Technical Report 67-5, Graduate School of Industrial Administration, Carnegie-Mellon University, Revised December, 1967.

4) Balinski, M. L., "Integer Programming: Methods, Uses, Computation," Management Science, 12, pp. 253-313 (1965).

5) Baxter, W. W., "A Group Theoretic Branch and Bound Algorithm for the Traveling Salesman Problem," unpublished Master's thesis, M.I.T. (1968).

6) Bellman, R. E., "Notes on the Theory of Dynamic Programming -- IV -- Maximization Over Discrete Sets," Naval Research Logistics Quarterly, 3, 67-70 (1956).

7) Benders, J. F., "Partitioning Procedures for Solving Mixed-Variable Programming Problems," Numerische Mathematik, 4, pp. 238-252 (1962).

8) Crowston, W. B. S., "Network Planning Models," unpublished Ph.D. thesis, Carnegie-Mellon University (1968).

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

10) Fuchs, L., Abelian Groups, Pergamon Press (1960).

11) Geoffrion, A. M., "Integer Programming by Implicit Enumeration and Balas' Method," SIAM Review, 9, No. 2, pp. 178-190 (1967).

12) Geoffrion, A. M., "Implicit Enumeration Using an Imbedded Linear Program," The RAND Corporation, P-5406-PR, September (1967).

13) Geoffrion, A. M., "Recent Computational Experience with Three Classes of Integer Linear Programs," The RAND Corporation, P-3699, October (1967).

14) Glover, F., "A Multiphase-Dual Algorithm for the Zero-One Integer Programming Problem," Operations Research, 13, pp. 879-919 (1965).

15) Glover, F., "An Algorithm for Solving the Linear Integer Programming Problem Over a Finite Additive Group, with Extensions to Solving General Linear and Certain Nonlinear Integer Problems," Operations Research Center Report 66-29, University of California, Berkeley (1966).

16) Glover, F., "Integer Programming Over a Finite Additive Group," Department of Statistics and Operations Research, University of Texas, Austin, March, 1968.

17) Gomory, R. E., "An Algorithm for Integer Solutions to Linear Programs," Recent Advances in Mathematical Programming, McGraw-Hill Book Company, New York, 1963, p. 269.

18) Gomory, R. E., "On the Relation Between Integer and Non-Integer Solutions to Linear Programs," Proceedings of the National Academy of Sciences, Vol. 53, pp. 250-265 (1965).

19) Gomory, R. E., "Faces of an Integer Polyhedron", Proceedings of the National Academy of Sciences, Vol. 57, pp. 16-18 (1967).

20) Gorry, G. A., "Strategies for Computer-Aided Diagnosis" to appear in Mathematical Biosciences.

21) Gorry, G. A., "A System for Computer-Aided Diagnosis," Project MAC Technical Report MAC-44 (Thesis), M.I.T., Cambridge, Massachusetts (1967).

22) Gorry, G. A., and Barnett, G. O., "Experience with a Model of Sequential Diagnosis," to appear in Computers in Biomedical Research, Vol. 1, No. 5.

23) Haldi, J., "25 Integer Programming Test Problems," Working Paper No. 43, Graduate School of Business, Stanford University (1964).

24) Hu, T. C., "A Decomposition Algorithm for Paths in a Network," Operations Research, 16, pp. 91-102 (1968).

25) Ignall, E., and Schrage, L., "Application of the Branch-and-Bound Technique to Some Flow-Shop Scheduling Problems," Operations Research, 11, pp. 972-989 (1963).

26) Karp, R. M., and Held, M., "Finite-State Processes and Dynamic Programming," Journal of The Society of Industrial and Applied Mathematics, 15, 3, pp. 693-717 (1965).

27) Lawler, E. L., and Ward, D. E., "Branch-and-Bound Methods: A Survey," Operations Research, 14, pp. 699-719 (1966).

28) Lemke, C. E., and Spielberg, K., "Direct Search Algorithms for Zero-One and Mixed-Integer Programming," Operations Research, 15, pp. 892-914 (1967).

29) Little, J. D. C., Murty, K. G., Sweeney, D. W., and Karel, C., "An Algorithm for the Travelling Salesman Problem," Operations Research, 11, pp. 972-989 (1963).

30) Lomnicki, Z. A., "A Branch and Bound Algorithm for the Exact Solution of the Three-Machine Scheduling Problem," Operational Research Quarterly, 16, pp. 89-100 (1965).

31) Mostow, G. D., Sampson, J. D., and Meyer, J. P., Fundamental Structures of Algebra, McGraw-Hill (1963).

32) Pierce, J. F., "Application of Combinatorial Programming to a Class of All-Zero-One Integer Programming Problems," IBM Cambridge Scientific Center Report, Cambridge, Massachusetts, July, 1966.

33) Reiter, S., and Sherman, G., "Discrete Optimizing," Journal of the Society of Industrial and Applied Mathematics, 13, 3, pp. 865-889 (1965).

34) Samuel, A., "Some Studies in Machine Learning Using the Game of Checkers II -- Recent Progress," IBM Journal, November 1967, pp. 601-617.

35) Shapiro, J. F., and Wagner, H. M., "A Finite Renewal Algorithm for the Knapsack and Turnpike Models," Operations Research, 15, No. 2, pp. 319-341 (1967).

36) Shapiro, J. F., "Finite Renewal Algorithms for Discrete Dynamic Programming Models," Technical Report No. 20, Graduate School of Business, Stanford University (1967).

37) 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, pp. 91-102 (1968).

38) Shapiro, J. F., "Group Theoretic Algorithms for the Integer Programming Problem - II: Extension to a General Algorithm," Technical Report, Operations Research Center, M.I.T., March, 1968.

39) Shapiro, J. F., "A Group Theoretic Branch and Bound Algorithm for the Zero-One Integer Programming Problem," Sloan School of Management, Massachusetts Institute of Technology, Working Paper 302-67, December, 1967.

40) White, W. W., "On a Group Theoretic Approach to Linear Integer Programming," Operations Research Center Report 66-27, University of California, Berkeley (1966).

41) Minsky, M., "Steps Toward Artificial Intelligence", Proceedings of the Institute of Radio Engineers, January, 1961.

42) Pierce, J.F., Operations Research and the Design of Management Information Systems, Special Technical Association Publication, TAPPI No. 4, 1967.