# BOUNDS FOR GENERALIZED INTEGER PROGRAMS

D. E. Bell

November 1973 WP-73-9

## Abstract


## Bounds for Generalized Integer Programs



David E. Bell

International Institute for Applied Systems Analysis
2361 Laxenburg
Austria



Generalized linear programming problems have been well solved by column generation and dual ascent procedures. The same problems with the variables restricted to be integer have only been solved when all the coefficients are known explicitly. This paper finds lower bounds for the optimal value of such programs requiring only the implicit definition of the activities.

# Bounds for Generalized Integer Programs

Some linear programs having a large number of variables may be solved relatively easily because each column of the coefficient matrix is defined implicitly as a feasible solution to some other problem. For example, the columns of the maximal flow problem (Ford and Fulkerson [4]) and multicommodity flow problem (Tomlin [14]) are defined implicitly as all routes between sources and sinks in a network. Other column defining subproblems are the minimum spanning tree calculation for the traveling salesman problem (Held and Karp [10]) and the knapsack problem for the cutting stock problem (Gilmore and Gomory [5]). Dantzig and Wolfe [2] have generalized this approach in connection with the decomposition of general large scale linear programs.

Little progress has been made in adapting these methods to deal with the same problems when the variables are constrained to be integral although Shapiro [13] has given a dual method for cases in which all coefficients may be generated explicitly beforehand.

The aim of this paper is to solve such problems in the manner of the linear program, that is by considering most of the activities implicitly. It will be shown how a lower bound for the optimal value of the program may be obtained which can then be incorporated into a branch and bound procedure if necessary.

The first section sets out the problem in more detail and is followed by two sections giving details of two separate bounding procedures. Section four gives a worked example of a cutting stock problem as a demonstration of the ideas. It might be useful to glance at that example before reading on.

## 1. The Problem

The integer program to be considered is

$$Z^* = \min \sum_{j\varepsilon J} c_j x_j$$

$$\text{s.t.} \quad \sum_{j\varepsilon J} a_j x_j \geq b \tag{1}$$

$$x_j \geq 0 \text{ and integer}$$

where $\{a_j\}_{j\varepsilon J}$ are activities defined as the set of solutions to some subproblem with $|J|$ assumed to be large. All the coefficients in (1) will be taken as integral.

Considering (1) as an ordinary integer program and using Gomory's group reformulation (see [6]), an equivalent problem is

$$\begin{aligned} Z^* = Z_o + \min \quad & \sum_{j\varepsilon N} \bar{c}_j x_j + \bar{c}_s s \\ & \sum_{j\varepsilon N} (B^{-1}a_j)x_j - B^{-1}Ls \leq B^{-1}b \\ & \sum_{j\varepsilon N} (B^{-1}a_j)x_j - B^{-1}Ls \equiv B^{-1}b \\ & x_j,\ Ls \geq 0 \text{ integer} \end{aligned} \tag{2}$$

B is an optimal L.P. basis, N the non-basic activities, $L = (l_{ij})$ is defined by

$$l_{ii} = \begin{cases} 1 \text{ if slack } s_i \text{ is non-basic} \\ 0 \text{ otherwise} \end{cases}$$

and $\bar{c}_j = c_j - c_B B^{-1} a_j$ are the revised cost coefficients. The symbol '$\equiv$' will represent equality modulo 1, that is, $a \equiv b$ if and only if $a - b$ is integral.

A lower bound for Z* may be obtained by forming the unconstrained group problem suggested in [6] which is precisely (2) with the inequality constraints relaxed

$$Z^* \geq \overline{Z} = Z_o + \min \sum_{j\varepsilon N} \overline{c}_j x_j + \overline{c}_s s$$

$$\text{s.t.} \sum_{j\varepsilon N} (B^{-1}a_j)x_j - B^{-1}Ls \equiv B^{-1}b \qquad (3)$$

$$x_j,\ Ls \geq 0 \text{ integer}$$

The group associated with this problem is that generated, with addition modulo 1, by the columns $B^{-1}a_j$ and may be shown to have an order which is a factor of $|\det B|$ (see Wolsey [15]). This group problem may be solved very quickly by considering it as a shortest route problem (see Gorry and Shapiro [8] and Gorry, Northup and Shapiro [7]) and provides a lower bound $\overline{Z}$ for $Z^*$.

For the problems under consideration, $|J| >> |\det B|$ so that many activities will be mapped into the same group element in (3) thus giving a decomposition of J into equivalence classes. Let the group be $G = \{g_0, g_1, \ldots g_{D-1}\}$, say, then define

$$J_i = \{j\varepsilon J |\ B^{-1}a_j \equiv g_i\} .$$

Now consider the following problem

$$\min \sum_{i=1}^{D-1} h_i x_i + \overline{c}_s s$$

$$\text{s.t.} \sum_{i=1}^{D-1} g_i x_i - B^{-1}Ls \equiv B^{-1}b \qquad (4)$$

$$Ls,\ x_i \geq 0 \text{ integer,}$$

where

$$h_i = \min_{j\varepsilon J_i} \overline{c}_j \quad i = 0, 1, \ldots D-1 \qquad (5)$$

and $h_i = +\infty$ if $J_i = \emptyset$. The optimal value of this problem is evidently equal to that of (3) since at most one variable from each equivalence class will be used in the solution to (3), and (4) includes the cheapest from each class. So the lower bound $\bar{Z}$ may now be found if the value of $h = (h_0, h_1, \ldots h_{D-1})$ is known. It could be found by explicit calculation ([13]) but this is prohibitive if $|J|$ is too large.

The next section discusses how, for certain subproblems, h may be found by dynamic programming.

## 2. A Dynamic Programming Approach

If the subproblem which generates the activities of J is a dynamic programming problem, it may be possible to find h by means of a simple extension of the state space.

For a problem having a finite state space $\mathcal{S}$ and a function $C : \mathcal{S} \times \mathcal{S} \to R$ representing the cost of transferring from one state to another, define $\phi : \mathcal{S} \to R$ to give the minimum cost of reaching a given state S from some initial state by a sequence of transferals. The recursion

$$\phi(S) = \min_{S^1 \varepsilon \mathcal{S}} \{\phi(S^1) + C(S^1,S)\}$$

together with initial values, will give an optimal routing to each state. Now if each transferal is assigned a group value from G, we may consider the problem of reaching a given state by a sequence of transferals whose group sum is a given element of G. If $\phi^* : \mathcal{S} \times G \to R$ is defined by the recursion

$$\phi^*[S,g] = \min_{S^1 \varepsilon \mathcal{S}} \{\phi^*[S^1,g-g(S^1,S)] + C(S^1,S)\} \tag{6}$$

it can be seen that $\phi^*(S,g)$ represents the minimum cost of reaching state S with group value g. Comparing (6) with (5) it can be seen that

$$h_i = \min_{S \in \bar{\mathcal{S}}} \phi(S, g_i) \qquad (7)$$

where $\bar{\mathcal{S}} \subseteq \mathcal{S}$ is a subset of final states. This extension of the state space may be applied to any monotone sequential decision process, as described by Karp and Held [11]. Thus a lower bound may be obtained for $Z^*$ by solving the sequence of problems (6), (7), (4). The size of this effort evidently depends upon $|G|$, but relaxation procedures exist for reducing it if necessary (see Gorry, Shapiro and Wolsey [9]). If $|G|$ is too large and cannot be reduced the methods of the next section may be applied.

As an example, consider the problem of finding the shortest route between source and sink in the undirected network of Figure 1.

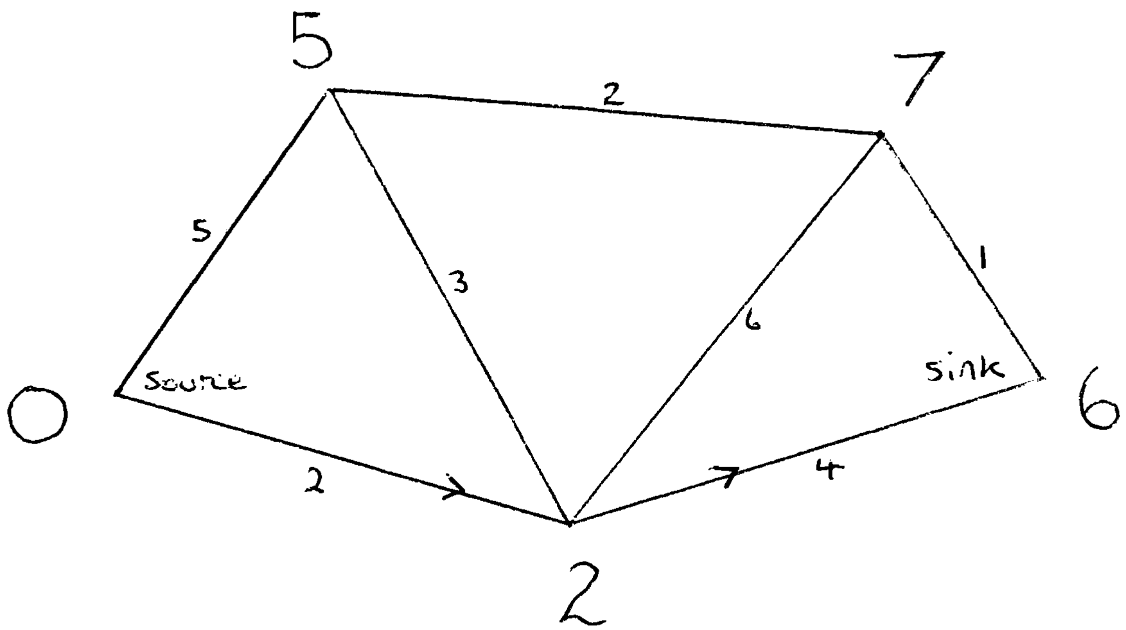


Figure 1

The state space for this problem is the set of nodes, the non infinite state change costs are shown together with the value of ø at each node. The set of final states is $\bar{\mathcal{S}}$ = {sink} so that the shortest route has length 6. Now a group weight from the addition modulo 2 group, G = {0, 1}, is assigned to each arc and as the shortest route has group sum 1, the object is now to find the shortest route with group sum 0. The situation of (6) may be considered diagramatically in Figure 2.

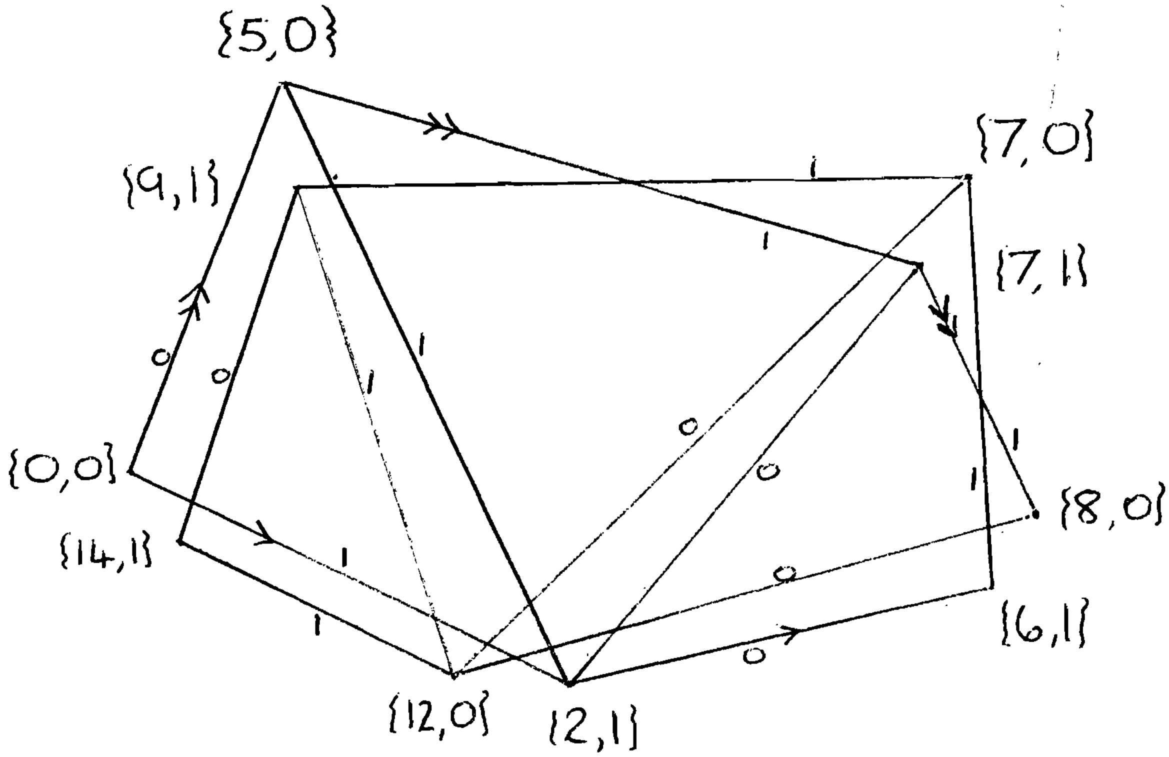


Figure 2

Each node has been replicated (G) times and the shortest route of group sum g$\varepsilon$G is the shortest route from the "0" source node to the "g" sink node. The vector {ø(s,g),g} is marked at each node in Figure 2 showing that the shortest route of group sum 0 has length 8.

## 3. Bounds for the Equivalence Classes

The aim of this section is to find h or a set of lower bounds $\bar{h}$ for h, in cases where dynamic programming is inappropriate for solution of the subproblem. Recall from section 1 that J was divided into equivalence classes where $a^1$, $a^2$ are equivalent if and only if

$$B^{-1}a^1 \equiv B^{-1}a^2 \quad .$$

<u>Theorem 1</u> If $\bar{c}_1$, $\bar{c}_2$ are the modified objective costs of equivalent activities $a^1$, $a^2$ then

$$\bar{c}_1 \equiv \bar{c}_2$$

<u>Proof</u> Since the objective coefficients were assumed to be integral $c_1 \equiv c_2 \equiv 0$

$B^{-1}a^1 \equiv B^{-1}a^2$ implies $c_B B^{-1}a^1 \equiv c_B B^{-1}a^2$

where $c_B$ is the vector of costs associated with the basic variables. Thus

$$\bar{c}_1 = c_1 - c_B B^{-1}a^1 \equiv c_2 - c_B B^{-1}a^2 = \bar{c}_2 \qquad //$$

The important implication of this theorem is that if $j\varepsilon J_i$ then $h_i \equiv \bar{c}_j$.

Define $\bar{h}$ by the relation

$$\bar{h} \equiv h \qquad 0 \leq \bar{h} < 1$$

<u>Theorem 2</u> Let $\bar{\bar{Z}}$ be the optimal value of the unconstrained group problem (4) with the objective coefficients $\bar{h}$. Then $\bar{\bar{Z}}$ is a valid lower bound for $Z^*$.

<u>Proof</u> With B an optimal LP basis

$$\bar{c}_j \geq 0 \quad \text{for all } j\varepsilon J,$$

hence $h \geq 0$.

Thus $\bar{h} \leq h$, and $\bar{\bar{Z}} \leq \bar{Z} \leq Z^*$ //

Note that in cases where J is large, the value of $h_i$ might be expected to be near zero and perhaps less than one in which case $\bar{h}_i = h_i$. Compared with the task of finding h, the problem of finding $\bar{h}$ is trivial. All that is required is any integer vector a for which $B^{-1}a \equiv g_i$ then $\bar{h}_i$ is known from the value of $c_B B^{-1}a$. Indeed if $G = \{g_0, g_1, \ldots g_{D-1}\}$ is known explicitly, then $\bar{h} \equiv (c_B g_0, c_B g_1, \ldots c_B g_{D-1})$. Even if $\overline{h_i} < h_i$ the relative ease with which it may be obtained could more than compensate for any worsening of the resulting bound.

Under certain circumstances it is possible to show that as long as a given equivalence class is non empty then $\bar{h}_i = h_i$. The conditions of Theorem 3 enforce all activities in the same class to have the same revised objective costs but this is overly strict since it only requires one of the activities of each class to be less than one in order to have $\bar{h} = h$.

Since the activities are generated implicitly by a subproblem, so too must the objective coefficient be so generated. Assume that for some integral vector $(r_o, r)$ the cost of an activity $a_j$ is

$$c_j = r_o + ra_j \ .$$

All the examples quoted in the introduction have such a representation.

Partition the optimal L.P. basis B for (1) as

$$B = \begin{pmatrix} B_1 & 0 \\ B_2 & -I \end{pmatrix}$$

Let $\pi = 1.B_1^{-1}$, with $a_j = \binom{a_{1j}}{a_{2j}}$, $r = (r_1, r_2)$ partitioned in accordance with B.

<u>Theorem 3</u> For a non empty equivalence class $J_i$

$$\overline{h_i} = h_i \quad \text{if}$$

(i) $\pi a_{1j} > 1 - 1/r_o$ for all $j \varepsilon J_i$

(ii) $r_2 = 0$

<u>Proof</u>

$$B^{-1} = \begin{pmatrix} B_1^{-1} & 0 \\ B_2 B_1^{-1} & -I \end{pmatrix}$$

Let $r = (r_1, r_2)$ so that

$$\begin{aligned} \bar{c}_j &= c_j - c_B B^{-1} a_j \\ &= r_o + r_1 a_{ij} + r_2 a_{2j} - (r_o 1 + r_1 B_1 + r_2 B_2, 0) B^{-1} a_j \\ &= r_o(1 - \pi a_{1j}) + r_2(a_{2j} - B_2 B_1^{-1} a_{1j}) . \end{aligned}$$

Now $\overline{h}_i = h_i$ if

$$0 \le \bar{c}_j < 1 \quad \text{for all } j \varepsilon J_i$$

which with $r_2 = 0$ is equivalent to

$$0 \le r_o(1 - \pi a_{1j}) < 1 .$$

Since B is optimal $\bar{c}_j \ge 0$ is automatic so that condition (i) enforces $\overline{h_i} = h_i$. //

<u>Definition</u> A set of activities will be called <u>complete</u> if for a given activity a, if $\bar{a}$ is any integer vector for which $0 \le \bar{a} \le a$, then $\bar{a}$ is also an activity.

For example, solutions to the Knapsack problem form a complete set as indeed do the set of solutions to inequalities of the form $Ax \le b$ where A is a non-negative matrix.

The shortest route and minimum spanning tree subproblems do not have complete sets of activities.

<u>Theorem 4</u> For problems having a complete set of activities there exists an optimal basis having no slack variables, hence condition (ii) of Theorem 3 may always be satisfied.

<u>Proof</u> The idea is that if a given optimal basis uses slacks then the optimal activities may be reduced so that the slacks become zero.

Consider any row in the given basis which has a basic slack,

$$\Sigma a_{ij} x_j^* = b_i + s_i^* \quad , \qquad s_i^* > 0 \quad .$$

<u>Case 1</u> For some j $\quad a_{ij} x_j^* \geq s_i^*$ .

Then choose $0 \leq \overline{a_{ij}} \leq a_{ij}$ to maximize $\overline{a_{ij}} x_j^* \leq s_j^*$ .

Replace the activity $a_j$ by

$$a_j = \begin{pmatrix} a_{1j} \\ \cdot \\ a_{ij} - \bar{a}_{ij} \\ \cdot \\ a_{mj} \end{pmatrix}$$

so that now $\quad \Sigma a_{ij} x_j^* = b_i + (s_i^* - \bar{a}_{ij} x_j^*)$.

If $\overline{a_{ij}} = 0$ then $x_j^* > s_i^*$ so in this case let

$$\overline{a_j} = \begin{pmatrix} a_{1j} \\ \cdot \\ a_{ij} - 1 \\ \cdot \\ a_{mj} \end{pmatrix}$$

and replace the slack column $\begin{pmatrix} 0 \\ 0 \\ \cdot \\ -1 \\ \cdot \\ \cdot \\ 0 \end{pmatrix}$ by $\overline{a_j}$ .

Replace $x_j^*$ by $x_j^* - s_i^*$ and let $\bar{x}_j^* = s_i^*$ .

Then $\sum_{k \neq j} a_{ik} x_k^* + a_{ij}(x_j^* - s_i^*) + (a_{ij} - 1)s_i^* = b_i$ .

The new matrix has the correct number of columns and is nonsingular because its determinant is unaffected. (Adding one column to another does not affect the determinant).

Case 2 For all j, $a_{ij}x_j^* < s_i^*$ .

Then replace $a_j$ by $\overline{a_j} = \begin{pmatrix} a_{1j} \\ \cdot \\ \cdot \\ 0 \\ \cdot \\ \cdot \\ a_{mj} \end{pmatrix}$ then

$$\sum_{k \neq j} a_{ik} x_k^* + \overline{a_{ij}} x_j^* = b_i + (s_i^* - a_{ij} x_j^*).$$

If $s_i^* = a_{ij} x_j^*$ replace the slack column by any independent activity.

Since either case 1 or case 2 must apply the above process may be repeated until the required matrix has been obtained. The resulting solution is non-negative by construction and must be optimal if $r_i \geq 0$. Indeed if $r_i > 0$ this row would have no slack anyway for then the above construction improves the basis. //

Theorem 5 If the objective cost of all activities is constant, condition (i) of Theorem 3 is automatically satisfied for almost all equivalence classes.

<u>Proof</u> In this case $(r_o, r) = (1, 0)$, so that it need only be shown that

$$1 - \pi a_{1j} < 1$$

$$\text{or} \quad \pi a_{1j} > 0 \quad \text{for all } j \varepsilon J_i \ .$$

The objective function $\Sigma c_j x_j$ may be rewritten as $r_1(\Sigma x_j) + rAx$ which equals $r_o(\Sigma x_j) + rb + rs$ .

The dual of

$$\begin{aligned} \min \ & \Sigma_J x_j + rs \\ & Ax - Is = b \\ & s, x \geq 0 \end{aligned} \qquad (8)$$

is

$$\begin{aligned} \max \ & yb \\ & yA \leq 1 \\ & -y \leq r \end{aligned} \qquad (9)$$

so that with $r = 0$, $y \geq 0$ .

If B is the optimal basis for (8) then the optimal y in (9) is $y = (1.B_1^{-1}, 0) = (\pi, 0)$ which implies that $\pi \geq 0$.

Hence $0 \leq \overline{c_j} \leq 1$

so that $\overline{h_i} = h_i$ unless there exists a class $J_i$ for which every activity has $\overline{c_j} = 1$. In this case $\overline{h_i} = 0$, $h_i = 1$, and each activity satisfies

$$\pi a_{1j} = 0 \ . \qquad //$$

Lagrange multipliers have been used to improve the bound given by the unconstrained group problem [3], [12]. The methods of this section may be adapted for this case, see chapter 2 in [1].

## 4. An Example - A Cutting Stock Problem

The cutting stock problem was first solved using column generation methods by Gilmore and Gomory [5] and concerns the minimization of material required to fulfil given orders.

Consider a situation in which a supplier has rolls of cloth of a given length L. He has orders for $b_i$ rolls of cloth of a smaller length $w_i$ $i = 1, . . . m$. Each roll may be cut into smaller rolls by using any cutting pattern which produces a non-negative integer number $a_i$ rolls of length $w_i$ subject only to the condition

$$\sum_{i=1}^{m} a_i w_i \leq L . \qquad (10)$$

Hence the set of activities for this problem consists of all non-negative integer solutions to (10). Clearly this set is complete if $a \geq \bar{a} \geq 0$ then (10) implies

$$\sum_{i=1}^{m} \bar{a}_i w_i \leq L \quad ,$$

so that this problem satisfies the conditions of Theorem 4. Let $x_j$ represent the number of rolls on which cutting pattern $a_j$ is used so that if the objective function is the number of rolls of length L used to satisfy the order, the coefficients $c_j = r_o + r.a_j = 1$ have $r_o = 1$ $r = 0$ so that this problem also satisfies Theorem 5.

As a numerical example, suppose $L = 58$, $b_1, b_2, b_3 = 7$ with $w_1 = 7$, $w_2 = 11$, $w_3 = 16$. The object is thus

$$\text{minimize} \quad \sum_{j \varepsilon J} x_j$$

$$\text{s.t.} \quad \sum_{j \varepsilon J} a_j x_j \geq b \qquad (11)$$

$$x_j \geq 0 \quad \text{integer}$$

where the $\{a_j\}_{j \varepsilon J}$ are any non-negative integer solutions to

$$7a_{1j} + 11a_{2j} + 16a_{3j} \leq 58 \qquad (12)$$

The optimal L.P. basis uses activities (2, 1, 2), (2, 4, 0) and (1, 0, 3) each 1.4 times for an objective value of 4.2.

$$B = \begin{pmatrix} 2 & 2 & 1 \\ 1 & 4 & 0 \\ 2 & 0 & 3 \end{pmatrix} \qquad B^{-1} = \frac{1}{10} \begin{pmatrix} 12 & -6 & -4 \\ -3 & 4 & 1 \\ 8 & 4 & 6 \end{pmatrix}$$

$$\Pi = 1.B^{-1} = \frac{1}{10}(1, 2, 3).$$

Since two activities are equivalent if they have the same values of $B^{-1}a$ (mod 1) the equivalence classes are determined by the values $k_1$, $k_2$, $k_3$ given by

a) $2a_1 + 4a_2 + 6a_3 \equiv k_1 \pmod{10}$

b) $7a_1 + 4a_2 + a_3 \equiv k_2 \pmod{10}$

c) $2a_1 + 4a_2 + 6a_3 \equiv k_3 \pmod{10}$

Clearly $k_1 \equiv k_3$ and since $k_1 \equiv 6k_2$ the equivalence classes may be determined solely by the value of

$$7a_1 + 4a_2 + a_3 \pmod{10}$$

$$a_1 + 2a_2 + 3a_3 \pmod{10}$$

$$\min \quad \frac{1}{10}\left(\sum_{k=1}^{9} kx_k + s_1 + 2s_2 + 3s_3\right)$$

$$\text{s.t.} \quad \sum_{k=1}^{9} kx + s_1 + 2s_2 + 3s_3 \equiv 8 \pmod{10} \qquad (13)$$

$$x_k,\ s_i \geq 0 \ \text{integral}$$

which has an optimal value of 0.8. This gives a lower bound of 4.2 + 0.8 = 5.0 for the number of rolls required. There are many optimal solutions to (13) a sample of which are

(i) $x_1 = 8$ (ii) $x_2 = 4$

(iii) $x_4 = 2$ (iv) $x_8 = 1$

(v) $s_1 = 8$ (vi) $s_2 = 1$, $s_3 = 2$

(vii) $x_2 = 2$, $s_1 = 1$, $s_3 = 1$.

Some of these solutions may not be feasible in (11) and since there are so many it would be useful to have criteria for choosing amongst them.

Criterion 1 The optimal solution which minimizes $\sum_{j\epsilon N} x_j$ should be chosen.

Reasoning This criterion is just as applicable to all the problems fitting the model of this chapter. The inequality constraints omitted from (13) are

$$\sum_{j\epsilon N} (B^{-1}a_j)x_j - B^{-1}s \leq B^{-1}b \quad , \qquad (14)$$

which, if summed to give a single surrogate constraint yields

$$\sum_{j\epsilon N} (1.B^{-1}a_j)x_j - 1.B^{-1}s \leq 1.B^{-1}b$$

$$\text{or} \quad \sum_{j\epsilon N} (\Pi a_j)x_j - \Pi s \leq \Pi b \qquad (15)$$

indicating that a good choice of optimal solution is one that minimizes the left hand side of (15). Now the objective function of (13) is

$$\sum_{j\in N} (1 - \Pi a_j)x_j + \Pi s$$

or (16)

$$\sum_{j\in N} x_j - \left( \sum_{j\in N} (\Pi a_j)x_j - \Pi s\right) .$$

For all optimal solutions, the quantity (16) is constant hence the minimizing of the left hand side of (15) is equivalent to minimizing $\sum_{j\in N} x_j$ .

This criterion orders the optimal solutions given with (v) and (vi) as the best and (i) as the worst.

<u>Criterion 2</u> The optimal solution $(x^*, s^*)$ which minimizes the value of $\max_{i,j} \{x_j^*, s_j^*\}$ should be selected.

<u>Reasoning</u> Criterion 1 was developed by an averaging of the constraints but it has been observed from hand computations that if one variable has a high value it is likely to cause infeasibility in (14) even though (15) is satisfied. Hence criterion 2 suggests an averaging out of the values of the variables. This criterion would give (iv) as the best and (i) and (v) as the worst of the optimal solutions to choose.

There is evidently some disagreement between the criteria as solution (v) appears as the best and the worst in two lists. A suggested combination criterion is the minimization amongst the optimal values of

$$\sum_{j\in N} x_j^* + \tfrac{1}{2} \max_{i,j} \{x_j^*, s_i^*\} . \qquad (17)$$

This gives a final ordering of the selected optimal solutions of (vi), (iv), (iii), (vii), (v), (ii), (i).

(vi) $s_2 = 1$, $s_3 = 2$ is a feasible correction giving an optimal value of 5 with solution

$$2 \begin{pmatrix} 2 \\ 4 \\ 0 \end{pmatrix} + 2 \begin{pmatrix} 1 \\ 0 \\ 3 \end{pmatrix}$$

Solutions for which $\sum_{j \varepsilon N} x_j^* = 0$ are particularly simple to check since it is not necessary to calculate any elements of the equivalence classes.

(iv) $x_8 = 1$

The activities of this section, which may be found by dynamic programming as indicated in section 2 are

1, 2, 1
2, 3, 0
6, 1, 0
0, 1, 2
0, 4, 0
8, 0, 0
4, 2, 0
2, 0, 2
5, 0, 1
3, 1, 1

of which only the first two will make the correction $x_8 = 1$ feasible. This fact raises an important procedural point. It has already been noted in Theorem 4 that certain problems may be made easier by including activities in the L.P. basis which are strictly dominated by other activities. Here, neither of the two feasible correction activities are maximal, they are dominated by (2, 2, 1) and (2, 4, 0) respectively.

(iii) $x_4 = 2$

The activities here are

0, 2, 0
4, 0, 0
1, 0, 1
2, 1, 0

and although none of these activities on their own provide a feasible correction the activity (0, 2, 0) used once together with either (1, 0, 1) or (2, 1, 0) is feasible.

In summary, solution (vii) has a feasible solution whereas none of (v), (ii), (i) have. The calculation here would, of course, have stopped with (vi), the other solutions were examined for the purposes of the example only. Note that the bound from (13) was exact, as in fact none of the equivalence classes were empty.

This example has shown the importance of testing all the alternative optimal solutions to the group problem and thus of making good ranking decisions amongst them.

## ACKNOWLEDGEMENT

This research was part of a dissertation supervised by Professor J.F. Shapiro at the Operations Research Center, M.I.T.

REFERENCES

1. D.E. Bell, "The Resolution of Duality Gaps in Discrete Optimization", Technical Report No. 81, Operations Research Center, Massachusetts Institute of Technology, 1973.

2. G.B. Dantzig and P. Wolfe, "Decomposition Principle for Linear Programs", Operations Research 8, 101-111, 1960.

3. M.L. Fisher and J.F. Shapiro, "Constructive Duality in Integer Programming", to appear in the SIAM Journal on Applied Mathematics.

4. L.R. Ford, Jr. and D.R. Fulkerson, "A Suggested Computation for Maximal Multicommodity Network Flows", Management Science 5, 97-101, 1958.

5. P.C. Gilmore and R.E. Gomory, "A Linear Programming Approach to the Cutting Stock Problem", Operations Research 9, 849-859, 1961.

6. R.E. Gomory, "Some Polyhedra Related to Combinatorial Problems", Linear Algebra 2, 451-558, 1969.

7. G.A. Gorry, W.D. Northup and J.F. Shapiro, "Computational Experience with a Group Theoretic Integer Programming Algorithm", Mathematical Programming V4, 171-192.

8. G.A. Gorry and J.F. Shapiro, "An Adaptive Group Theoretic Algorithm for Integer Programming Problems", Management Science 17, 285-306, 1971.

9. G.A. Gorry, J.F. Shapiro and L.A. Wolsey, "Relaxation Methods for Pure and Mixed Integer Programming Problems", Management Science V18, 229-239, January 1972.

10. M. Held and R.M. Karp, "The Traveling Salesman Problem and Minimum Spanning Trees", Operations Research 18, 1138-1162, 1970.

11. R.M. Karp and M. Held, "Finite State Processes and Dynamic Programming", SIAM Journal on Applied Mathematics 15, 693-718, 1967.

12. J.F. Shapiro, "Generalized Lagrange Multipliers in Integer Programming", Operations Research 19, 68-76, 1971.

13. J.F. Shapiro, "A Decomposition Algorithm for Integer Programming Problems with Many Columns", Proceedings of the 25th Annual ACM Conference, Boston, 1972.

14. J.A. Tomlin, "Minimum-Cost Multicommodity Network Flows", Operations Research 14, 45-51, 1966.

15. L.A. Wolsey, "Group Representation Theory in Integer Programming", Technical Report No. 41, Operations Research Center, Massachusetts Institute of Technology, 1969.