Journal of the Korean Operations
Research Society
Vol. 7, No. 2, Oct., 1982

—Research Paper—

# An Algorithm for Multiple Compensatory Objectives Problems

Kwang Min Yang*

**Abstract**

This paper presents an efficient algorithm both in computation speed and storage requirement by exploring the special structure of problems involving multiple objective goals. The algorithm developed here is limited to the problems with multiple, compensatory objectives, however it can be extended to 'traditional' preemptive priority goal programming problems. Computational results are included.

## 1. INTRODUCTION

Multiple objectives optimization, or goal programming, was originally proposed by Charnes, Cooper and Ferguson [4] as an approach to developing a scheme for executive compensation. As noted by Charnes and Cooper [3] in a review of the field, this approach to multiple objectives optimization did not receive significant attention until the mid-1960's. However, during the past twenty years, we have witnessed a flood of professional articles (see Lin [15] for a survey) and books (*e.g.*, Igiri [12], Ignizio [11], and Lee [14]) dealing with applications of this methodology. In spite of the apparent special structure of the problems and flourishing applications, there appears to be no specialized solution procedure in the literature except for more special cases: Charnes and others' explicit solution procedure [5] for problems with separable goal functionals, Dyer's non-linear programming approach [6], and Arthur and Ravindran's algorithm [1] using a partitioning and elimination procedure for conventional preemptive priority goal programming problems. This may partly be due to the easy convertibility of the problem to an ordinary linear programming problem on no consideration for computational efficiency whatsoever.

This paper is limited to the development of a solution algorithm for problems with multiple, compensatory objectives. We eliminate problems that do not allow trade-offs among the objectives. For further discussion on this subject (additivity of goal functionals) see Dyer [7].

The principal advantage of the algorithm developed here comes from *i*) a large reduction of the storage requirement ii) less work per pivot, and iii) use of logical operations in lieu of

* Chung Ang University

usual pivot operations thereby reducing the number of arithmetic operations.

In the following, the model, solution approach and a computational comparison with two widely available revised simplex algorithms will be presented.

## 2. MODEL

A simple goal programming model can be stated as

$$\text{(p)} \qquad \underset{x_j}{\text{Minimize}} \sum_{i\in I} \left| \sum_{j\in J} a_{ij}x_j - g_i \right|,$$

where $x_j$ represents decision variables. $a_{ij}$ and $g_i$ are constants and the $g_i$ can be regarded as "goals" since the functionals.

$$f_i(x) = \left| \sum_{j\in J} a_{ij}x_j - g_i \right|,$$

increase with discrepancies from the goals, $g_i$, for each decision variable, $x_j$. Since the goal functionals are non-linear and also nondiffential at $f_i(x)=0$, we convert the model into an equivalent linear form.

One method of conversion is

$$\text{(p1)} \qquad \underset{x_j,\delta_i}{\text{Minimize}} \sum_{i\in I} \delta_i$$

$$\text{subject to} \quad -\delta_i \le \sum_j a_{ij}x_j - g_i \le \delta_i \quad \forall i$$

$$\delta_i \ge 0 \quad \forall i$$

As proved in Charnes and Cooper [2], (p) can also be reformulated as an ordinary linear programming problem, viz.,

$$\text{(p2)} \qquad \underset{x_i,\delta_i^+,\delta_j^-}{\text{Minimize}} \sum_{i\in I} (\delta_i^+ + \delta_i^-)$$

$$\text{subject to} \quad \sum_{j\in J} a_{ij}x_j - \delta_i^+ + \delta_i^- = g_i \quad \forall i$$

$$\delta_i^+,\ \delta_i^- \ge 0 \quad \forall i$$

At a glance, both formulations are about the the same size. (p1) has twice as many constraints as (p2), while (p2) contains twice as many variables as (p1). The choice between these two models will depend on which algorithm we are going to apply.

We may extend (p2) naturally into a more general model which allows asymmetric weights for deviations, viz.,

$$\text{(GP)} \qquad \underset{x_j,\delta_i^+,\delta_i^-}{\text{Minimize}} \sum_{i\in I} (w_i^+\delta_i^+ + w_i^-\delta_i^-)$$

$$\text{snbject to} \quad \sum_{j\in J} a_{ij}x_j - \delta_i^+ + \delta_i^- = g_i \quad \forall i$$

$$\delta_i^+, \delta_i^- \ge 0 \quad \forall i$$

where the $w_i^+, w_i^-$ are non-negative weights for positive and negative deviations from each goal, $g_i$, $i\varepsilon I$.

Note that (GP) can encompass any type of constraint by assigning appopriate weights without any notational change. This is equivalent to introducing both slack and surplus variables for each additional constraint regardless of whether the constraint is equality or inequality of either direction.

## 3. SOLUTION APPROACH

In order to make the presentation of the algorithm clear, we appeal to Graves' primal algorithm [8]. (Note that row labels designate constraints which are not row-basic.)

The problem (GP) can be presented (after possible row and column permutation) as

| | | $X1$ | $X2$ | $\delta_1^+$ | $\delta_2^+$ | $\delta_3^+$ | $\delta_1^-$ | $\delta_2^-$ | $\delta_3^-$ | RHS |
|---|---|---|---|---|---|---|---|---|---|---|
| | $E1$ | $A_{11}$ | $A_{12}$ | $-I_1$ | | | $I_1$ | | | $g_1$ |
| (TO) | $E2$ | $A_{21}$ | $A_{22}$ | | $-I_2$ | | | $I_2$ | | $g_2$ |
| | $E3$ | $A_{31}$ | $A_{32}$ | | | $-I_3$ | | | $I_3$ | $g_3$ |
| | BR | 0 | 0 | $w_1^+$ | $w_2^+$ | $w_3^+$ | $w_1^-$ | $w_2^-$ | $w_3^-$ | 0 |

where A has been partitioned as

$$A=\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \\ A_{31} & A_{32} \end{bmatrix}$$

This partitioning is generic only and will be determined at each step. At this point it is convenient to force the RHS$\geq$0. This is easy to do because each row represents an equation constraint, and the entire row can be multiplied by -1. This merely interchanges the roles of the corresponding $\delta^+$ and $\delta^-$, so they may simply be relabled. The equations are now pivoted into the basis. Since the minimum same sign ratio is used the RHS remains nonnegative. From this point on, the full primal tableau may be written

| | | $E1$ | $X2$ | $\delta_1^+$ | $E2$ | $\delta_3^+$ | $\delta_1^-$ | $\delta_2^-$ | $E_3$ | RHS |
|---|---|---|---|---|---|---|---|---|---|---|
| | $X1$ | $A_{11}^{-1}$ | $A_{11}^{-1}A_{12}\cdot$ | $-A_{11}^{-1}$ | | | $A_{11}^{-1}$ | | | $A_{11}^{-1}g_1$ |
| (T1) | $\delta_2^+$ | $A_{21}A_{11}^{-1}$ | $-A_{22}+A_{21}\cdot A_{11}^{-1}A_{12}$ | $-A_{21}A_{11}^{-1}$ | $I_2$ | | $A_{21}A_{11}^{-1}$ | $-I_2$ | | $-g_2+A_{21}A_{11}^{-1}g_1$ |
| | $\delta_3^-$ | $-A_{31}A_{11}^{-1}$ | $A_{32}-A_{31}\cdot A_{11}^{-1}A_{12}$ | $A_{31}A_{11}^{-1}$ | | $-I_3$ | $-A_{31}A_{11}^{-1}$ | | $I_3$ | $g_3-A_{31}A_{11}^{-1}g_1$ |
| | BR | $-(w_2^+A_{21}-w_3^-A_{31})A_{11}^{-1}$ | $w_2^+A_{22}-w_3^-A_{32}-(w_2^+A_{21}-w_3^-A_{31})A_{11}^{-1}A_{12}$ | $w_1^+ +(w_2^+A_{21}-w_3^-A_{31})A_{11}^{-1}$ | $w_2^+$ | $w_3^+ + w_3^-$ | $w_1^- -(w_2^+A_{21}-w_3^-A_{31})A_{11}^{-1}$ | $w_2^+ + w_2^-$ | $-w_3^-$ | $-wy^0$ |

The state of (T1) at any iteration determines the partitioning of (T0). The $\delta_1$ columns are those deviational variables which have both $\delta^+$ and $\delta^-$ basic. $\delta_2$ and $\delta_3$ have one side basic, the other side feasible.

In the following $\delta_2^+$ and $\delta_3^-$ behave identically, the same is true of $\delta_3^+$ and $\delta_2^-$. We shall restrict our attention to $\delta_2$. Corresponding relationships will hold for $\delta_3$.

The proposed algorithm gains its strength in two areas, concise representation of the tableau and a modified pivot procedure.

### A) Concise Representation of the Tableau

The following properties can be observed from (T1):

i) Columns denoted by $E1$, $E2$ and $E3$ can be permanently removed. $E1$, $E2$ and $E3$ have become basic and we never again pivot in these columns regardless of the values of the

corresponding bottom row, BR, because doing so would cause primal infeasibility.

ii) BR of columns $(\delta_3^+)$ and $(\delta_2^-)$ are always positive since we assume only non-negative weights. These columns can be removed since they are always dual feasible.

iii) The $(\delta_1^+)$ and $(\delta_1^-)$ columns contain identical entries except for their reversed signs and BR. This enables us to drop either of the columns: the other column can always be generated from the one we are carrying. Note that BR of $(\delta_1^-)$ is the $(w^+ + w^-)_1$'s complement of the $(\delta_1^+)$ column.

iv) The optimality conditions are:

$$0 \leq w_1^+ + (w_2^+ A_{21} - w_3^- A_{31}) A_{11}^{-1} \leq w_1^+ + w_1^- \tag{1}$$

$$w_2^+ A_{22} - w_3^- A_{32} - (w_2^+ A_{21} - w_3^- A_{31}) A_{11}^{-1} A_{12} \geq 0 \tag{2}$$

The feasibility conditions are:

$$-g_2 + A_{21} A_{11}^{-1} g_1 \geq 0 \tag{3}$$

$$g_3 - A_{31} A_{11}^{-1} g_1 \geq 0 \tag{3}'$$

$$\text{and } A_{11}^{-1} g_1 \geq 0 \quad \text{if } x \geq 0 \text{ is required} \tag{4}$$

In summary, to update the primal tableau we carry only columns $X_2$, $(\delta_1^+)$ and RHS, $(w^+ + w^-)_i$ and the index of the current basis. Since all constraints are equalities we can start iterations with the RHS$\geq$0 and by using the minimum same sign ratio for selecting a non-basic constraint we can always maintain the conditions (3), (3)', and (4) above. Target row selection steps can be omitted, the target row is always BR.

B) Pivot procedure

In addition to the reduction of the problem representation described above, pivot operations can also be performed efficiently as will be explained below.

Assume that we have primal feasibility (*i.e.* conditions (3) and (4) are satisfied) but BR of $(\delta_1^+)$ does not meet the optimality condition (1). By pivoting in the $(\delta_1^+)$ column (or$(\delta_1^-)$) the objective function value can be improved. However, unlike ordinary linear programming, the optimality condition(1) is bounded on both sides. A pivot operation, in general, may not fulfill condition (1). In other words, it may require a series of pivots in $(\delta_1^+)$ to satisfy condition (1) with respect to $(\delta_1^+)$.

We seek an alternative pivot strategy presenting the following.

**Theorem 1.** suppose that BR of $(\delta_1^+)$ does not meet optimality condition (1), and $(\delta_2^+)$ is an improving direction for the violation. The pivot $(\delta_2^+) \leftrightarrow (\delta_1^+)$ satisfies condition (1) if, and only if, the pivot $(\delta_2^+) \longleftrightarrow (\delta_2^-)$ changes the sign of BR of $(\delta_1^+)$.

**Proof.** We shall refer to elements of the tableau by attaching subscripts $i$ and $j$. The $j$th column violates (1) and the $i$ th row is an improving direction. There are two cases where condition (1) is not satisfied:

a) *Case I:* $BR_j < 0$

Since $RHS \geq 0$ and $(\delta_2^+)_i$ is an improving direction we have

$$(-A_{21} A_{11}^{-1})_{ij} > 0.$$

After the pivot $(\delta_2^+) \longleftrightarrow (\delta_1^+)$, the new $BR_j$ is

$$BR_j' = -BR_j / (-A_{21} A_{11}^{-1})_{ij}$$

which satisfies $BR_j' \geq 0$,

Now $BR_j' \leq (w_2^+ + w_2^-)_i$

iff $BR_j+(w_2^+ + w_2^-)_i(-A_{21}A_{11}^{-1})_{ij} \geq 0,$ (1)

In instead we pivot $(\delta_2^+) \longleftrightarrow (\delta_2^-)$,

$$BR_j'' = BR_j + (w_2^+ + w_2^-)_i(-A_{21}A_{11}^{-1})_{ij}$$

The sign changes iff $BR_j'' \geq 0$

or $BR_j+(w_2^+ + w_2^-)_i(-A_{21}\ A_{11}A_{11}^{-1})_{ij} \geq 0.$ (2)

(1) and (2) are the same.

b) *Case II:* $BR_j > (w_1^+ + w_1^-)_j > 0$

The BR of the corresponding $(\delta_1^-)$ column will be negative. Because $(\delta_2^+)_i$ is an improving direction and $RHS \geq 0$

$$(A_{21}A_{11}^{-1})_{ij} > 0$$

or $$(-A_{21}A_{11}^{-1})_{ij} < 0.$$

After the pivot $(\delta_2^+) \longleftrightarrow (\delta_1^+)$, the new BR is

$$BR_j' = -BR_j/(-A_{21}A_{11}^{-1})_{ij}$$

which satisfies $BR_j' \geq 0$.

Now $BR_j' \leq (w_2^+ + w_2^-)_i$

iff $BR_j+(w_2^+ + w_2^-)_i(-A_{21}A_{11}^{-1})_{ij} \leq 0.$ (3)

If instead we pivot $(\delta_2^+) \longleftrightarrow (\delta_2^-)$, new BR

$$BR_j'' = BR_j + (w_2^+ + w_2^-)_i(-A_{21}A_{11}^{-1})_{ij}$$

The sign changes iff $BR_j'' \leq 0$

which again is the same as (3). Q.E.D.

The modified pivot strategy is used when the most negative BR violates optimality condition (1). The improving direction ratios between $(\delta_1^+)$ and the RHS are computed and saved. The ratio with the smallest absolute value determines the normal pivot. If this element is in an (X1) row the normal pivot is taken.

If the normal pivotal element occurs in a $\delta_2^+$ row (or equivalently $\delta_3^-$) we check if a $(\delta_2^+) \longleftrightarrow (\delta_2^-)$ pivot would change the sign of the $\delta_1^+$ bottom row. By Theorem 1 a sign change indicates the normal pivot will give complete (1)-optimality in the violated $\delta_1^+$ column, and so the normal pivot is taken.

If the $(\delta_2^+) \longleftrightarrow (\delta_2^-)$ pivot does not yield a $\delta_1^+$ sign change Theorem 1 implies that the normal $(\delta_2^+) \longleftrightarrow (\delta_1^+)$ pivot would cause a $\delta_1^-$ dual violation. In this case we do not perform the normal pivot but perform instead the $(\delta_2^+) \longleftrightarrow (\delta_2^-)$ pivot. This pivot has several properties:

a) The pivotal element is -1 so the pivot is easy to perform.

b) The objective fuuction is improved by $(w_2^+ + w_2^-)_i(g_2 - A_{21}A_{11}^{-1}g_1)$.

c) $(\delta_1^+)$ remains in dual violation of (1), but is improved by $|(w_2^+ + w_2^-)_i(A_{21}A_{11}^{-1})_{ij}|$.

d) Besides the BR, the only row that is affected is the improving $(\delta_2^+)$ row.

e) The sign of $RHS_i$ is changed, causing a primal infeasibility.

This last property would seem to violate the proof of convergence given in Graves [8]. The feasibility will be recovered, however, without causing a loss on the objective function.

The algorithm now repeatedly executes $(\delta_2^+) \longleftrightarrow (\delta_2^-)$ pivots in theorder of the stored ratios, until on of them would cause a sign change as in Theorem 1. A normal $(\delta_2^+) \longleftrightarrow (\delta_1^+)$ pivot is now indicated that will bring the indicated column into dual feasibility. Primal feasibility will also be completely restored, as the control ratio is now larger than the ratios for the

primal violated rows.

pivot operations on the implicit column ($\delta_1^-$) (recall that we carry ($\delta_1^+$)) are defined as follows (for notation see Graves [8]):

pivotal element $(s,k)$

$v_{sk}' = -1/v_{sk}$

general elements $(i,j)$

$v_{ij}' = v_{ij} + v_{sj} \cdot v_{ik} \cdot v_{sk}'$

pivotal column $(k)$

$v_{ik}' = v_{ik} \cdot v_{sk}'$

pivotal row $(s)$

$v_{sj}' = -v_{sj} \cdot v_{sk}'$

bottom row of pivotal column

$w_k' = -(w_s^+ + w_s^- - w_k) \cdot v_{sk}'$

## 4. IMPLEMENTATION AND COMPUTATIONAL RESULTS

As pointed out, since we are carrying only information on one of the variables $\delta_i^+$ or $\delta_i^-$, it is necessary to change the signs of the entries (and complement with $(w^+ + w^-)_i$ in the bottom row) when a pivot must be performed on the variable which we are not carring. Instead we duplicate the logic with signs and relational directions reversed. This approach requires two separate pivot routines, however in this way explicit sign reversals can be avoided, thereby saving computing time. The amount of additional storage needed is minimal.

Unlike ordinary linear programming, the optimality condition is bounded by 0 below and $(w^+ + w^-)_i$ above for $\delta_1^+$ and only bounded below for $X_2$. In order to handle this in a streamlined way without knowing which variable is under consideration, we carry a vector of size $(m+n)$ which contains the $m$ ($m$ is the number of goal functionals) elements of of $(w^+ + w^-)$ and $n$ ($n$ is the number of variables) arbitrary large positive numbers which impose no upper limit for the $X_2$ variables. These "upper bounds" must be greater than the largest weight.

The algorithm was coded in ANSI FORTRAN and was tested on an IBM 390/91. The code requires $mn+5m+3n+800$ words to solve a goal programming problem with $m$ goal functionals and $n$ decision variables (including free variables). To demonstrate the efficiency of the approach relative to the straightforward simplex approach, we also solved the same test problems with two widely available revised simplex LP subroutines. Two commercial codes were LA01B of the Harwell Library [10] and ZX3LP of IMSL[13]. These codes were chosen simply because they were included in these well-known mathematical subroutine libraries. The test problems were generated using the uniform random number generator (GGUBF) in IMSL. $a_{ij}$ values were chosen between $-1.0$ and $+1.0$, $g_i$ were $.0-3.0$ and $w_i^+$ and $w_i^-$ were $.0-1.0$. The solution times reported below were measured by a real-time clock accurate to $\pm 1.04 * 10^{-4}$ sec and do not include time for input and output. The proposed algorithm and Harwell were run using double precision whereas IMSL was run using single precision due to the limitation on core availability. All were compiled with the IBM FORTRAN H Extended

compiler with $OPT=2$ compiler option.

**Table 1. Computational Results**

| # of Goal Functionals | # of Decision Variables (# of Free Var.) | Avg. CPU Time (in Milliseconds) | | | Avg. # of Pivots | | | |
|---|---|---|---|---|---|---|---|---|
| | | Harwell | IMSL | Proposed | Harwell | IMSL | Proposed | |
| | | | | | | | Logical | Normal |
| 10 | 5(5) | 27.7 | 37.3 | 1.8 | 17.9 | 16.6 | 2.2 | 6.4 |
| 10 | 10(5) | 36.9 | 40.3 | 3.4 | 21.6 | 16.7 | 3.8 | 9.4 |
| 10 | 20(5) | 46.1 | 42.2 | 5.7 | 22.9 | 15.1 | 4.0 | 10.4 |
| 20 | 5(5) | 87.9 | 200.0 | 3.0 | 29.1 | 38.3 | 5.3 | 7.2 |
| 20 | 10(5) | 105.9 | 228.2 | 5.8 | 33.1 | 42.0 | 7.4 | 11.2 |
| 20 | 20(5) | 151.8 | 278.2 | 14.0 | 42.3 | 47.2 | 12.2 | 17.9 |
| 50 | 5(5) | 590.2 | 3,215.9 | 6.8 | 63.4 | 142.2 | 11.5 | 8.7 |
| 50 | 10(5) | 689.9 | 3,745.6 | 14.5 | 70.5 | 162.7 | 19.2 | 14.1 |
| 50 | 20(5) | 945.5 | 4,451.5 | 40.7 | 85.6 | 185.9 | 29.3 | 27.0 |
| 100 | 5(5) | 3,080.0 | 36,522.9* | 14.8 | 117.6 | 462* | 18.3 | 10.4 |
| 100 | 10(5) | 3,375.0 | — | 29.6 | 124.2 | — | 30.1 | 15.9 |
| 100 | 20(5) | 4,299.4 | — | 80.2 | 147.4 | — | 51.5 | 29.1 |
| 200 | 5(5) | 17,569.4** | — | 32.8 | 222.0** | — | 32.6 | 11.8 |
| 200 | 10(5) | — | — | 69.2 | — | — | 53.0 | 19.1 |
| 200 | 20(5) | — | — | 175.1 | — | — | 85.2 | 33.0 |

* Run terminatde after 1 replication
** Run terminated after 2 replications.

The computation times were each based on 20 random replications. Some of the computations were not performed (these are marked as "—" in Table 1) because either i) the required storage exceeded the available storage or ii) computational times were becoming too large.

The computational results obviously establish the dominance of the proposed algorithm. The solution times are stable throughout the test problem size. As the problem size increases, the solution time increases but at a remarkably slower rate than the revised simplex methods.

The relationship of the computation time and the number of pivots of the proposed algorithm as a function of the number of goal functionals, $m$, and the number of variables(including free variables), $n$, was studied from the data of Table 1. The results of a log-linear multiple regression ($T=k\cdot m^{\alpha}\cdot n^{\beta}$) fitted to this data is shown in Table 2.

**Table 2. Regression Results**

| | $\alpha$ | $\beta$ | $R^2$ |
|---|---|---|---|
| Time | 1.033(78.6) | 1.133(45.3) | 0.9652 |
| No. of Pivots* | 0.263(18.9) | 0.662(25.0) | 0.7680 |

t-statistics are shown in parentheses.
*Regressed on the number of normal pivots alone.

From the regression results, we find that the computational time for the proposed algorithm increases approximately linearly as the parameter values increase. However this can be improved further by only carrying the basic kernel (see Graves [8]) instead of carrying and updating the partial tableau. This seems particularly true if we note that the regression on

the number of pivots is concave for the proposed algorithm, hence faster pivoting should give even more impressive results.

## 5. CONCLUSION

As the computational results show, the original cumbersome-looking absolute value notation turned out to be the one which can be dealt with easily without explicitly carrying any extraneous variables. The efficiency of the algorithm comes from the reduced tableau size and the use of logical operations in lieu of ordinary pivot operations by exploiting the special structure of the problem, *i. e.*, the presence of an identity matrix.

The proposed algorithm is more favorable if there exist many goal functionals and free variables since the reduction of the problem comes from selecting out some deviational variables and the existence of free variables which reduces the size of the index set of rows to be searched in the ratio test.

This approach can be applied to any "linear" programming problem with absolute values in the objective and can yield reductions in both computation time and storage space.

The FORTRAN programs used for the comparison are available to researchers for a nominal handling charge. For further information write the author, Chung-Ang University, College of Business Administration, 221 Heukseok-dong, Dongjak-ku, Seoul 151.

### References


1. Arthur, Jeffrey L., and Ravindran, A., "An Efficient Goal programming Algorithm Using Constraint Partitioning and Variable Elimination," *Management Science*, Vol. 24, No. 8 (April 1978), 867—868.
2. Charnes, A., and Cooper,W.W., *Management Models and Industrial Applications of Linear Programming*, John Wiley and Sons, New York, 1971.
3. _____ and _____, "Goal Programming and Multiple Objective Optimizations," *European Journal of Operational Research*, Vol. 1, No.1 (Jauary 1977), pp.33—54.
4. _____, _____ and Ferguson, R. O., "Optimal Estmation of Executive Compensation by Linear Programming," *Management Science*, Vol. 1, No. 2 (January 1955), pp. 138—151.
5. _____, _____, Klingman, D., and Niehaus, R. J., "Explicit Solutions in Convex Goal Programming," *Management Science*, Vol. 22, No. 4 (December 1975), pp. 438—448.
6. Dyer, J., "Interactive Goal Programming," *Management Science*, Vol. 19, No. 1 (September 1972), pp. 62—70.
7. _____, "On the Relationship Between Goal Programming and Multiattribute Utility Theory," Discussion Paper No. 69, Management Science Study Center, Graduate School of Management, University of California, October 1977.
8. Graves, G. W., "A Complete Constructive Algorithm for the General Mixed Linear Programming Problem," *Naval Research Logistics Quarterly*, Vol. 12, No. 1 (March 1965), pp. 1—34.
9. _____, *Mathematical Programming*, forthcoming.

10. Hopper, M. J., Harwell Subroutine Library, A Catalogue of Subroutines, Suppl. 2, 1973, AERE-R-7477 Supple. 2, August 1977.
11. Ignizio, J. P., *Goal Progiamming and Extensions*, Lexington Books, Lexington, Massachusetts, 1976.
12. Ijiri, Y., *Management Goals and Accounting for Control*, American Elsevier, New York, 1965.
13. International Mathematical and Statistical Libraries, Inc., *Library* 1 (*Reference Manual*), Vol. Ⅱ, 1977.
14. Lee, Sang M. *Goal Programming for Decision Analysis*, Auerbach Publishers, Inc., 1972.
15. Lin, W. Thomas, "A Survey of Goal Programming Applications," presented at ORSA/TIMS Joint National Meeting, Los Angeles, November 13, 1978.