HEURISTICS FOR INTEGER PROGRAMMING USING SURROGATE CONSTRAINTS

Fred Glover, *University of Colorado*

ABSTRACT

This paper proposes a class of surrogate constraint heuristics for obtaining approximate, near optimal solutions to integer programming problems. These heuristics are based on a simple framework that illuminates the character of several earlier heuristic proposals and provides a variety of new alternatives. The paper also proposes additional heuristics that can be used either to supplement the surrogate constraint procedures or to provide independent solution strategies. Preliminary computational results are reported for applying one of these alternatives to a class of nonlinear generalized set covering problems involving approximately 100 constraints and 300-500 integer variables. The solutions obtained by the tested procedure had objective function values twice as good as values obtained by standard approaches (*e.g.*, reducing the best objective function values of other methods from 85 to 40 on the average. Total solution time for the tested procedure ranged from ten to twenty seconds on the CDC 6600.

INTRODUCTION

Heuristic solution methods for integer programming have maintained a noticeably separate existence from algorithms. Algorithms have long constituted the more respectable side of the family, assuring an optimal solution in a finite number of steps. Methods that merely claim to be clever, and do not boast an entourage of supporting theorems and proofs, are accorded a lower status. Algorithms are conceived in analytic purity in the high citadels of academic research, heuristics are midwifed by expediency in the dark corners of the practitioner's lair.

Recently, however, there has been a growing recognition that the algorithms are not always successful, and that their heuristic cousins deserve a chance to prove their mettle. Partly this comes from an emerging awareness that algorithms and heuristics are not as different as once supposed – algorithms, after all, are merely fastidious heuristics in which epsilons and deltas abide by the dictates of mathematical etiquette. It may even be said that algorithms exhibit a somewhat compulsive aspect, being denied the freedom that would allow an occasional inconsistency or an exception to ultimate convergence. (Unfortunately, ultimate convergence sometimes acquires a religious significance; it seems not to happen in this world.)

The heuristic approach, robust and boisterous, may have special advantages in terrain too rugged or varied for algorithms. In fact, those who are fond of blurring distinctions suggest that an algorithm worth its salt is one with "heuristic power."

If there is any merit to this view, it would be useful to single out specific heuristic principles that appear to show promise. The distillation of such principles can lead to the discovery of new possibilities and to more effective solutions.

One such principle comes from the surrogate constraint approach. This approach is based on the idea that it may be possible to capture useful information not directly available from the original problem constraints, taken one at a time, by forming a non-negative linear combination of these constraints. The combination serves as a proxy (or surrogate) for the others.

The surrogate constraint framework provides a convenient organizing principle around which a class of heuristics can be formed. For example, as will be shown, the Senju-Toyoda method [10], for 0-1 capital budgeting problems, and the Kochenberger-McCarl-Wyman method [9], for more general integer programming problems, constitute two attractively simple members of this class.

The general surrogate constraint framework substantially increases the available solution strategies. At the same time, the framework enlarges the classes of problems that are potentially accessible to solution. Several important auxiliary heuristics amplify these considerations.

Framework for Surrogate Constraint Heuristics

The general procedural composition of the surrogate constraint framework is quite simple and may be stated as a sequence of four principal steps.

1. Generate one or more surrogate constraints.
2. Determine one or more starting solutions.
3. By reference to the surrogate constraint(s), periodically or regularly updated, establish measures of the goodness of increasing and decreasing each variable.
4. Sequentially change—increase or decrease—the values of the variables, singly or in blocks, in accordance with their goodness measures, and keep track of the best solution(s) found in the process.

In spite of the simplicity of this framework, a variety of specific heuristic methods, old and new, are encompassed within it. It is especially important to note, for example, that no restrictions are imposed on the problem coefficients (such as nonnegativity) or on the character of the starting solution (such as feasibility). This feature by itself immediately distinguishes this framework from less general methods such as [9, 10]. Other features provide other significant departures from less general methods. The departure is caused by the general character of the surrogate constraints themselves and by the methods that have been developed for exploiting them.

The use of the framework for surrogate constraints, beginning with immediate and natural uses and progressing to higher level considerations, will be discussed.

First-level aspects of implementation can be summarized for each of the four steps. These aspects draw upon ideas relating to surrogate constraints in other contexts, ideas which are equally appropriate in the heuristic setting. We mention these ideas only in brief outline, not because they lack importance, but because our purpose here chiefly is to sketch broad considerations and to discuss possible variants before proceeding to more refined strategies.

Step 1. The generation of surrogate constraints can be carried out by the "adaptive weighting" proposals of [3] or the linear programming proposals of [1, 2, 4]. For

example, one can simply normalize and sum a current set of critical constraints, or one can weigh the constraints with values of dual multipliers from the solution to an associated linear program, using auxiliary considerations to reflect desired representations of sign conditions. Information for a trial solution may be incorporated into the process, as in Step 2 or Step 4. Surrogate constraints also can be used as a guide to getting trial solutions in the first place, *e.g.*, by solving a relaxed problem in which surrogate constraints replace certain subsets of the original constraints.

Step 2. To obtain starting solutions, which need not be feasible, as already noted, one may begin with all variables a) at values equal to their upper or lower bounds, b) at values given by reference to an LP solution, c) at values perturbed from another starting point, and d) at values set by prior search.

Step 3. The goodness measure may be stated in terms of ratios of objective function coefficients to surrogate constraint coefficients, as suggested in the original surrogate constraint proposals, and as implicitly used in the heuristic methods of [9] and [10]. Weighted sums and products, and especially conditional measures which make use of thresholds, may be used as alternatives or supplements to ratios.

Step 4. To guide the process of changing the values of the variables, surrogate constraint ratios coupled with sign conditions give a natural indication of whether a value should be raised or lowered. Indeed, an appropriate way to exploit signed ratios has not yet emerged in the heuristic literature, and some confusion has reigned over this area. The way to take advantage of these signed ratios has been developed in the selection rules attending the introduction of surrogate constraints in [3].

For changes more complex than one at a time changes, relative goodness measures can also be used. For example, if x' is the current trial solution and x'' is a candidate for the next trial solution (a trial solution is one generated at any stage of the procedure, and is not the same as the best solution found to date), then the relative goodness measure may be $\alpha(cx'-cx'') + \beta(s(x') - s(x''))$, where cx is the objective function to be minimized, $s(x) \leqslant 0$ is the surrogate constraint, and α and β are decision parameters. (Note that the determination of these parameters can be treated as identifying an appropriate normalization for the surrogate constraint.) One can also use $s'(x') - s''(x'')$ in place of $s(x') - s(x'')$, where $s'(x) \leqslant 0$ is a surrogate constraint determined relative to x', and $s''(x) \leqslant 0$ is a surrogate constraint determined relative to x''.

It is important to note that the foregoing discussion does not impose any necessary prohibition against moving across boundaries of feasibility (in either direction), though the parameters used in decision rules do change under different conditions.

From a purely descriptive point of view, the methods of Senju-Toyoda and Kochenberger-McCarl-Wyman are readily identified as examples of the preceding framework. Specifically, the weights giving rise to the surrogate constraints implicitly relied on in these procedures are the reciprocal of the constant terms of the original constraints (one of the forms of normalization suggested in [4]), where these constant terms are kept updated to reflect the current assignment of values to the problem

variables. The effective gradients employed in these methods then turn out to be the surrogate constraint ratios already discussed.

Advanced Implementations

We now turn to augmenting heuristic considerations, each of which involves new ideas that extend the strategic possibilities available to surrogate constraint frameworks and to other heuristic frameworks as well. We will discuss these under the headings of "scatter search," "oscillating assignment," and "strongly determined and consistent variables."

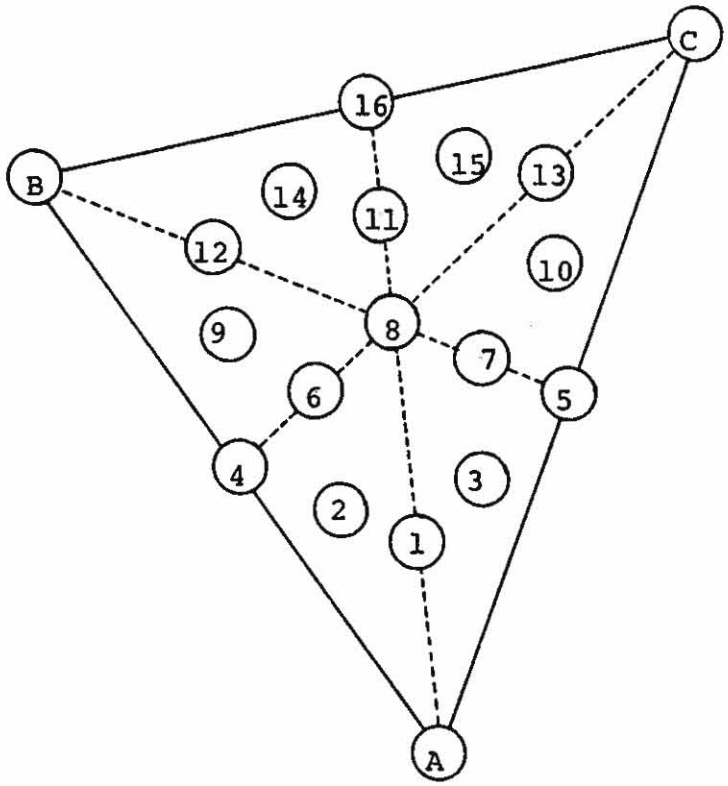
Scatter Search. This method may be used to generate starting solutions and trial solutions. Instead of being limited to a single preliminary effort, as are most procedures for generating such solutions, this approach uses a succession of coordinated initializations. These are purposely (*i.e.*, nonrandomly) generated to take account of characteristics in various parts of the solution space. In particular, scatter search builds upon a kindred strategy suggested a number of years ago in [5], by orienting its explorations systematically relative to a set of reference points. Reference points may consist, for example, of the extreme points of a simplex obtained by truncating the LP basis cone, or of good solutions obtained by prior problem-solving efforts.

The approach begins by identifying a convex combination, or weighted center of gravity, of the reference points. This central point, together with subsets of the initial reference points, is then used to define new subregions. Thereupon, analogous central points of the subregions are examined in a logical sequence (*e.g.*, generally sweeping from smaller objective function values to larger ones). Finally, these latter points are rounded to obtain the desired starting solutions. (Rounding, for any problems other than those with the simplest structures, should be either an iterative or a generalized adjacency procedure, following [5], to accomodate interdependencies in the problem variables.)

A variety of forms of scatter search is possible. The flavor of the approach can be conveyed by one of its simpler versions, illustrated by Figure 1. Each of the points numbered 1 through 16 is the central point of an apparent subregion of the simplex A, B, C. Node 8 is the center of A, B, C itself. Here A, B, C may or may not constitute the orginal reference points (which could, for example, have been 6, 7, 11 or 4, 5, 12, 13). The choice depends on the distribution of the original points relative to each other and the feasible region generally. Thus, for example, it can be desirable to use envelopes containing derived reference points, and to use weightings that bias central points away from centroids. When scatter search is carried out relative to reference points that lie on a single line, then it is reduced to a form of linear search, such as that employed by Hillier [7] and Jeroslow and Smith [8].

In general, to keep the effort within desirable limits for larger dimensions, the points generated according to the example pattern may be examined in their indicated numerical sequence (or in an alternative sequence dictated by the slope of the objective function contour) until a convenient cutoff point is reached. Or one may examine a less refined collection of points.

FIGURE 1



Because scatter search may be applied to reference points obtained from a historical progression of solution attempts and may also be used to influence this progression, the approach is conveniently suited to application with learning strategies.

Oscillating Assignment. This procedure may be used to determine values to be given to variables, as in Step 4 of the surrogate constraint framework. The procedure may be viewed as a unification and extension of outside-in and inside-out procedures. An outside-in approach, exemplified by the Senju-Toyoda method, begins at a distance from the feasible region and changes the values of the variables (as a rule, unidirectionally) until feasibility is attained. By contrast, an inside-out procedure, exemplified by the Kochenberger-McCarl-Wyman approach, starts from inside the feasible region (after an initialization step, if required) and changes values of the variables until no improving moves remain except moves that would force an exit from this region.

The oscillating assignment procedure modifies the form of these procedures and integrates the results into a single method. The first modification is to discard the stopping rule of the outside-in approach upon entering the feasible region, in favor of continuing to go deeper along a path dictated by the surrogate constraints. Thereupon,

the procedure reverses, working back toward the periphery, taking a trajectory designed to obtain improving solutions.

The second modification complements the first. When proceeding from inside the feasible region to a point where no improving move exists except one that violates feasibility, the method is allowed to follow a path that goes outside the feasible region. Then a return to the feasible region is initiated, again guided by the surrogate constraints. The oscillating assignment procedure, which integrates these two modifications by simple alternation, takes the depth of penetration beyond the feasibility boundaries as a decision rule of the process. The process can also be applied relative to implicit perturbations of the feasibility boundaries, thereby creating secondary oscillations in selected regions of the solution space.[1]

Strongly Determined and Consistent Variables. The notions of strongly determined and consistent variables provide a useful adjunct to the foregoing heuristic ideas. A variable may be called strongly determined if its assigned value cannot be changed (by the criteria available on a particular solution attempt), except by inducing a disruptive effect on the objective function value or on the values of the other variables.

To illustrate, consider the process of solving a 0-1 knapsack problem created by the generation of a surrogate constraint. Some of the problem variables will typically have highly profitable ratios and others will have highly unprofitable ratios. The variables with ratios at the extremes qualify as being strongly determined, since they are nearly compelled to assume particular values. The notion of strongly determined variables is a continuum, not an either-or classification, *i.e.*, some variables are more strongly determined than others.)

A consistent variable may be defined as one that frequently is strongly determined at a particular value or within a narrow range relative to the fluctuations of other variables. Its consistency may be illustrated by reference to the approach first proposed for creating a good surrogate constraint [3], which is one of the simplest and most convenient approaches. In this approach, a succession of surrogate constraints are generated which are candidates for selection. The procedure heuristically or algorithmically solves the knapsack problem for each candidate constraint. Weights of component constraints violated by the knapsack solution are then increased, and weights of component constraints satisfied by this solution are decreased, either strictly or in a relative sense. The amount of increase or decrease for a given constraint depends on the extent to which the constraint is violated or satisfied.[2]

---

[1] The inside-out and outside-in moves of the oscillating assignment approach can be replaced by other pairs of complementary moves, as for example, constructive and destructive moves in the context of scheduling and routing problems.

[2] A variation involving multiple surrogate constraints solves a succession of surrogate problems, each composed of a collection of candidate constraints. In this case, the amount of change in the weights that yield a particular surrogate constraint also depends on the degree to which the constraint is satisfied by the surrogate problem solution. An alternative is to divide the problem constraints among the surrogate constraints, either adaptively or based on *a priori* identification of structure.

Good candidate constraints give rise to the most restrictive knapsack solutions; that is, good constraints are those whose optimal knapsack solutions; that is, good constraints are those whose optimal knapsack solutions yield the tightest objective function bounds and come closest to satisfying the overall feasibility conditions for the original problem. Consistent variables may be defined as those most frequently strongly determined, relative to the good solutions and constraints. Consistent variables can also be defined in terms of heuristic solutions to the original problem itself. Like the strongly determined concept, the consistent concept involves a continuum; some variables may be more consistent than others.

There is clearly a healthy latitude in the possible operational specification of what makes up a consistent variable, as there is in most heuristic notions (*e.g.*, in the specification of a strong surrogate constraint itself). Experimentation and context determine whether a variable should be regarded as more consistent if it strongly receives a particular value in 3 out of 5 cases, or if it less strongly receives a value in 4 out of 5 cases. Segregation of various types of solutions may accommodate the possibility that a variable may be strongly determined at one value in some solutions, and strongly determined at another value in other solutions. Consideration of clusterings and interactions among subsets of variables provides logical ways of extending the concept.

However, it is worth stressing that it is the use of strongly determined and consistent variables which provides their *raison d'etre.* Use is based on three conjectures: first, that a variable which is highly consistent over a subset of good solutions is very likely to receive its preferred value—or lie within its preferred range—in optimal and near-optimal solutions; second, that once some variables are assigned specific values or constrained to narrow ranges, other variables that previously seemed not particularly consistent will now become a good deal more so; and, third, that the operation of imposing narrow restrictions on selected variables will yield increasingly reliable measures of the relative consistency of remaining variables, given the imposed restrictions.

These conjectures, of course, refer to general tendencies and not to invariant occurrences. They are inapplicable to single constraint problems, except by allowing an augmentation of the constraint set with cuts. However, the underlying rationale suggests the following heuristic procedure, which may be imbedded in the general surrogate constraint framework:

1. Select one or more variables with the greatest relative consistencies, and constrain these to their preferred values or ranges. Restrict attention to variables that have not previously been given specific values or to variables whose ranges have not been significantly restricted on a recent iteration.
2. Determine new relative consistencies for the variables on the basis of the restriction of Step 1.
3. Repeat the process until all variables have been constrained to specific values (*e.g.*, by progressive narrowing).

This procedure and the others previously described may be applied singly or in a selected combination, within the surrogate constraint framework or outside it.

An Illustration

We now illustrate one of the several surrogate constraint heuristics proposed above—the oscillating assignment heuristic—by applying it to the knapsack problem. In this setting, the knapsack constraint may be viewed as a surrogate constraint. The oscillating assignment heuristic forms the basis of the preliminary computational testing cited in the final section.

The knapsack problem is written as:

$$\text{Maximize} \quad \sum_{j=1}^{n} c_j x_j$$

$$\sum_{j=1}^{n} a_j x_j \leqslant a_0 \tag{1}$$

$$x_j = 0 \text{ or } 1$$

and where $a_j, c_j > 0$ for all j. The variables are indexed so that $c_j/a_j \geqslant c_K/a_K$ for $j \leqslant K$.

The version of oscillating assignment heuristic to be illustrated involves only a simple first order oscillation.

To describe the procedure, the following definitions are used:

1 Set: Indexes of variables that currently = 1

0 Set: Indexes of variables that currently = 0

1-M Set: The 1 Set except for the element most recently transferred to this set.

0-M Set: The 0 Set except for the element most recently transferred to this set.

The 1 Set and the 0 Set implicitly identify the current solution. We provide rules for creating the next solution as follows:

Generating the Next Solution:

Current Solution Is Feasible: Transfer the smallest indexed member of the 0-M Set that satisfies $L_0 \leqslant a_j \leqslant U_0$ to the 1 Set.

Current Solution Is Infeasible: Transfer the largest indexed member of the 1-M Set that satisfies $L_1 \leqslant a_j \leqslant U_1$ to the 0 Set.

In our particular illustration, we specify values for L_0, U_0 and L_1, U_1.

L_0, U_0: Set $L_0 = 0$, $U_0 = \infty$ (*i.e.*, the parameters are nonrestrictive) unless the transfer of the smallest indexed element of the 0-M set to the 1 Set would create an infeasible solution. Then replace U_0 with the average of the a_j for members of the 0-M Set.

L_1, U_1: Set $L_1 = 0$, $U_1 = \infty$ (*i.e.*, the parameters are nonrestrictive) unless the transfer of the largest indexed element of the 1-M set to the 0 set would create a feasible solution. Then replace L_1 with the average of the a_j for members of the 1-M Set.

Finally, we specify the rule for generating the trial solution.

Trial Solution:

Current Solution Is Feasible: Transfer, temporarily, for the purpose of the trial solution only, the member of the 0-M Set to the 1 Set that has the largest c_j value of those that yield a feasible transfer. If ties occur for the largest c_j, pick the one with the smallest index. Repeat this process until the transfer of any further elements of the 0-M set would destroy feasibility.

Current Solution Is Infeasible: Transfer, temporarily, the member of the 1-M set to the 0 set that has the smallest c_j value of those that yield a feasible transfer. If a single transfer doesn't yield a feasible solution, select the smallest c_j and repeat. If ties occur for the smallest c_j, pick the one with the largest index. Once feasibility is attained by this rule, complete the trial solution by the rule for the case where the solution is currently feasible.

This oscillating assignment heuristic will be illustrated for the following problem:

$$\begin{aligned} \text{Maximize } & 18x_1 + 21x_2 + 16x_3 + 18x_4 + 14x_5 + 12x_6 + 8x_7 + 9x_8 \\ & 17x_1 + 20x_2 + 16x_3 + 19x_4 + 15x_5 + 13x_6 + 9x_7 + 11x_8 \leqslant 61 \end{aligned} \tag{2}$$

Table 1 shows the stages of the heuristic. The element most recently transferred into the 1 Set from the 0-M Set is marked with an asterisk to indicate that it is excluded from the 1-M Set. Since the 0 Set is the complement of the 1 Set, only the exceptional member of the 0 set that does not also belong to the 0-M set is indicated. For simplicity, the process begins at the first point where L_1 and U_1 become restrictive (*i.e.*, after transferring indexes 1, 2, 3 from the 0 set to the 1 set), and continues for 5 iterations. The trial solution is represented by its associated 1 Set.

TABLE 1

Current Solution Status	1-Set	Exceptional Member 0-Set	Trial Sol	Objective Value
Feasible	1,2,3*	–	1,2,3	55
Infeasible	1,2,3,6*	–	1,2,6,7	59
Feasible	1,3,6*	2	1,3,5,6	60
Feasible	1,3,5*,6	2	1,3,5,6	60
Infeasible	1,3,5,6,8*	2	1,3,5,8	57

The solution whose 1 Set is 1, 3, 5, 6, obtained on the 3rd and 4th iterations, is optimal for this problem. The procedure can be carried on for a larger number of iterations to an arbitrary or adaptively determined cutoff. The oscillating assignment procedure has the ability to identify good solutions that differ considerably from the solution obtained by the ordinary best ratio determinations. In applications where the knapsack constraint truly represents a surrogate constraint, the form of the constraint may vary from iteration to iteration, and trial solutions may, of course, be based on a larger set of feasibility considerations.

Preliminary Computational Experience

A version of the oscillating assignment approach similar to the one illustrated was preliminarily tested on a class of nonlinear generalized covering problems.

These covering problems occurred in a real world scheduling situation involving 300 to 500 integer variables and approximately 100 constraints with positive right-hand sides. The objective is to minimize a sum of squared deviations of the selected covering from specified target values.

Although the heuristics of [9, 10] do not apply to this problem directly, they can be generalized via the surrogate constraint framework previously outlined to provide augmented methods that are capable of dealing with the present structure. These augmented methods provide a basis for comparing an oscillating assignment version of the general surrogate approach with procedures that are more nearly standard.

The first step of the testing was to produce augmented methods that utilized the surrogate constraint weights implicitly proposed in [9, 10] (*i.e.*, as incorporated into the effective gradients of these references). However, these natural extensions of [9, 10] gave rise to relatively poor solutions. Consequently, the augmented methods were modified to utilize alternative surrogate constraints. The criteria for generating these constraints were the same as those incorporated into the oscillating assignment approach, taking account of which component constraints were locally more restrictive in terms of the nonlinear objective. This change succeeded in reducing the best objective function values obtained by these augmented methods by at least 30 percent, on the average from around 130 to 85. By contrast, the oscillating assignment procedure produced solutions whose objective function values were only half of these new values, ranging on the average from 30-45. The solution times for this superior method were all between 10 and 20 seconds on the CDC 6600, for a total of 50 trial runs with data taken from the setting in which the problems arose.

Details of this computational study, which is presently being extended to additional classes of integer programming problems with accompanying refinement of the internal decision rules, will be reported elsewhere. However, these preliminary results strongly suggest the usefulness of appropriately designed surrogate constraint heuristics in practical settings.

REFERENCES

[1] Balas, E. "Discrete Programming by the Filter Method." *Operations Research*, Vol. 15, No. 5 (September, 1967), pp. 915–957.

[2] Geoffrion, A. "An Improved Implicit Enumeration Approach for Integer Programming." *Operations Research*, Vol. 17, No. 3 (May, 1969), pp. 437–454.

[3] Glover, F. "A Multiphase-Dual Algorithm for the Zero-One Integer Programming Problem." *Operations Research*, Vol. 13, No. 6 (November, 1965), pp. 879–919.

[4] Glover, F. "Surrogate Constraints." *Operations Research*, Vol. 16, No. 4 (July, 1968), pp. 741–749.

[5] Glover, F. "Heuristics in Integer Programming." paper presented at Management Science and Engineering Management Conference, Austin, Texas, September, 1967.

[6] Gulley, D. A., H. S. Swanson, and R. E. D. Woolsey. "On Not Searching for the Multipliers in Knapsack Problems." can be obtained from R. E. D. Woolsey, Colorado School of Mines, Golden, Colorado.

[7] Hillier, F. S. "Efficient Heuristic Procedures for Integer Linear Programming with an Interior." *Operations Research*, Vol. 17 (1969), pp. 600–637.

[8] Jeroslow, R. G. and T. H. C. Smith. "Experimental Results on Hillier's Search Imbedded in a Branch-and-Bound Algorithm." Management Sciences Research Group, Carnegie-Mellon University, Pittsburgh, Pennsylvania, November, 1973.

[9] Kochenberger, G. A., B. A. McCarl, and F. P. Wyman. "A Heuristic for General Integer Programming." *Decision Sciences*, Vol. 5, No. 1 (January, 1974), p. 36.

[10] Senju, S. and Y. Toyoda. "An Approach to Linear Programming with 0-1 Variables." *Management Science*, Vol. 15 (1968), pp. B-196–207.

[11] Trauth, C. A. and R. E. Woolsey. "Practical Aspects of Integer Linear Programming." Sandia Corporation Monograph, August, 1966, SC-R-66-925.

[12] Wyman, F. P. "Binary Programming: A Decision Rule for Selecting Optimal vs. Heuristic Techniques." *The Computer Journal*, Vol. 16 (1973), pp. 135–140.