

CUT SEARCH METHODS IN INTEGER PROGRAMMING

Fred GLOVER

University of Colorado, Boulder, Colorado, U.S.A.

Received 18 January 1971

Revised manuscript received 5 November 1971

Cut search is a new approach for solving integer programs based on extending edges of a cone to probe the solution space for sets of hyperplanes that are “proxies” for solution points in the space. Once all proxy hyperplanes associated with a given point have been intersected by at least one of the extended edges, this point is included in a set of points to be examined for feasibility (algorithmically or by inspection). Thereupon, all edges of the cone are extended an additional distance to create a cut by passing a hyperplane through the endpoints of these extended edges.

The flexibility of the cut search procedure permits a variety of strategies for exploring and cutting into the solution space. One useful version arises by taking the proxy hyperplanes to be members of a “positive” or “semipositive” coordinate system. Relative to such a system the procedure can be organized to reduce the set of vectors to be examined for feasibility and also to generate deeper cuts at the end of the edge probe.

1. Introduction

Cut search is a new approach for solving integer programs that consists of two alternating phases: an edge probe phase and a cut phase. The edge probe phase extends edges of a cone to intersections with sets of “proxy” hyperplanes which are associated with candidate solution points. As these sets of proxy hyperplanes are encountered, their associated candidate solutions are examined for feasibility (algorithmically or by inspection). At the elected termination of the edge probe phase, all finite edges of the cone are extended an additional distance, and a cut is adjoined to the problem by passing a hyperplane through the endpoints of the extended edges. A variety of strategies for applying cut search are indicated, with special emphasis on a version of cut search that provides particularly deep cuts and reduces the number of candidate solutions to be tested for feasibility.

2. Preliminary notation and definitions

We shall write the integer programming problem in the form

$$\begin{aligned} &\text{minimize} && cx, \\ &\text{subject to} && Ax \leq b, \\ &&& x \geq 0 \text{ and integer,} \end{aligned}$$

where x is an n -dimensional column vector and the matrix A and vectors b and c are assumed to be dimensioned conformably.

Introducing a vector v of slack variables, we may represent the foregoing in the column tableau formulation

$$\begin{aligned} &\text{maximize} && x_0 = c(-x), \\ &\text{subject to} && x = 0 - (-x), \\ &&& v = b + A(-x), \\ &&& x \geq 0, v \geq 0 \text{ and } x \text{ integer,} \end{aligned}$$

or, still more compactly, in the *current tableau* formulation

$$\begin{aligned} &\text{maximize} && x_0 = c_0 + c(-t), \\ &\text{subject to} && y = B_0 + B(-t), \\ &&& y \geq 0 \text{ and } x \text{ integer.} \end{aligned}$$

Here $y = \begin{pmatrix} x \\ v \end{pmatrix}$ and t is the vector of current nonbasic variables. The components of t are identified as a subset of the components of y in the current tableau by a set of n equations of the form

$$y_i = -(-t_j), \quad j \in N = \{1, \dots, n\},$$

where i depends on j .

We shall define S to be the set of feasible y vectors for the integer program, i.e., $S = \{y | y = B_0 - Bt, y \geq 0 \text{ and } x \text{ integer}\}$, and define the cone C associated with the current linear programming tableau by $C = \{y | y = B_0 - Bt, t \geq 0\}$ (hence $C \subset S$).

Given numbers $t_j^* > 0$, one may identify the hyperplane through the n points $B_0 - B_j t_j^*$, $j \in N$, on the edges of the cone C , by the equation

$$\sum (1/t_j^*) t_j = 1, \tag{1}$$

as may be verified by observing that (1) is “solved” upon setting $t_j = t_j^*$ for any single j . The associated half space

$$\sum (1/t_j^*) t_j \geq 1 \quad (2)$$

determines a “cut” (relative to B_0) since it eliminates the solution $y = B_0$ from the set of feasible solutions to the linear program. (In particular, $y = B_0$ occurs for $y = B_0 - Bt$ only when $t = 0$, due to the full column rank of B in the linear programming tableau.)

For the cut (2) to be compatible with the constraints of the integer program, it must not eliminate any points of the region S . We express this by reference to the *truncated cone* C^* , which we define to be the intersection of C with the open half space

$$\sum (1/t_j^*) t_j < 1, \quad (3)$$

i.e., $C^* = \{y \mid y = B_0 - Bt \in C \text{ and } t \text{ satisfies (3)}\}$. Then the cut (2) is defined to be *legitimate* (relative to S) if $C^* \cap S = \emptyset$ (this definition also applies to “feasible solution sets” S for problems other than the integer programming problem).

3. Convexity cuts

To provide a backdrop against which the cut search ideas may be put in convenient focus for purposes of comparison and contrast, we sketch the related “convexity” or “intersection” cut ideas that were first developed in the context of integer programming by Young [12] and Balas [1], and in the context of concave programming by Tui [11]. Our description will follow the exposition of Glover [4], which slightly extends the developments of [1] and [12].

Broadly speaking, the convexity cuts apply to any mathematical programming problem whose feasible solutions imply $y \in C \cap S$, where C is the cone indicated in the preceding section and S may be any of a rather large class of sets. The key restriction on S is the stipulation that whenever $B_0 \notin S$, it must be possible to identify a convex region R whose interior ($\text{int}(R)$) contains B_0 but no points of S . This stipulation leads to the following result.

Convexity cut lemma. Assume R is convex, $B_0 \in \text{int}(R)$ and $\text{int}(R) \cap S = \emptyset$. Then for any numbers $t_j^* > 0$ such that the points $B_0 - B_j t_j^*$, $j \in N$, are all contained in R , the cut (2)

$$\sum (1/t_j^*) t_j \geq 1$$

is legitimate relative to S .

Proof. The assumptions imply that the truncated cone C^* is contained in $\text{int}(R)$. Thus $C^* \cap S = \emptyset$ follows immediately from $\text{int}(R) \cap S = \emptyset$.

Balas and Young first introduced this result for the case in which R is a hypersphere (or, in the mixed integer problem, a hypercylinder) whose boundary contains integer points but whose interior does not. (Balas' results apply to the general mixed integer problem and Young's results apply to a class of 0-1 problems.) A variety of cuts based on different choices of R are indicated in Glover [5], exploiting the more general statement of the convexity cut lemma given above. More recently, Balas and Young have also developed interesting generalizations of their original work [1, 2, 13]. An intriguing cut, obtained when R is the "dual of the unit hypercube", is developed by Balas, Bowman, Glover and Sommer [3]. Other applications of related ideas, particularly in the context of Tui's approach to concave programming, are developed by Ragavachari [10] and Glover and Klingman [7]. Special results for a class of "generalized lattice point" problems (which includes the mixed integer programming problem as a special case) are also given in Glover and Klingman [8].

4. Cut search ¹

4.1. Fundamental notions

The cut search approach uses the same cut inequality as the convexity cut approach, but generates different values of the cut coefficients and relies on a different strategy for guaranteeing the legitimacy

¹ Several of the results of this section were initially developed in a slightly different manner in [6]. A conceptually related, but procedurally intriguing departure from these ideas has been recently proposed by C.A. Burdet in [4].

of the cut. In particular, rather than determine the numbers $t_j^* > 0$ so that the points $B_0 - B_j t_j^*$ all lie in a predetermined convex set R , the cut search approach verifies the legitimacy of the cut by constructing a superset S^* of $C^* \cap S$ for which the condition $S^* = \emptyset$ is easily determined. This is accomplished in a manner so that the precise composition of S^* is always completely specified. Consequently, whenever $S^* \neq \emptyset$, it is nevertheless possible to “search” among the elements of S^* (either directly, or using an auxiliary algorithm) to discover whether $C^* \cap S$ is itself empty.

If in fact $C^* \cap S \neq \emptyset$, the ability of the procedure to identify the composition of $C^* \cap S$ (as a subset of S^*) makes it possible to obtain feasible solutions to the integer program. This may, of course, be particularly advantageous for problems in which “near-optimal” solutions are acceptable.

Finally, upon determining the composition of $C^* \cap S$, the procedure extends the edges of the cone C still further (ignoring the composition of the new $C^* \cap S$) to provide a particularly deep cut. Alternatively, if desired, one can simply continue the edge probe, although often at an increasing computational expense.

The key to constructing S^* is to associate a set of proxy hyperplanes with each vector $y = \begin{pmatrix} x \\ v \end{pmatrix}$, and include y in S^* only if all of its proxy hyperplanes are intersected by at least one of the edges

$$C_j^* = \{y \mid y = B_0 - B_j t_j, 0 \leq t_j < t_j^*\}$$

of the truncated cone C^* .

The basis for this construction rule is the following result.

First cut search lemma. Assume $y' \in C^*$ and let H be any hyperplane containing y' . Then there is at least one edge C_j^* of C^* that intersects H .

Proof. Let H^+ be the half space determined by H that contains B_0 . If the lemma is false, then $B_0 - B_j t_j \in H^+$ for all j and all t_j satisfying $0 \leq t_j < t_j^*$. But this implies $C^* \subset H^+ - H$, and hence $C^* \cap H$ must be empty, contrary to the assumption that $y' \in C^* \cap H$.

Note that the foregoing lemma also holds (by essentially the same proof) when C^* and C_j^* are replaced by their closures \bar{C}^* and \bar{C}_j^* .

A convenient use of the lemma is to let the proxy hyperplanes for a given y vector be simply the set of coordinate hyperplanes for that vector. In the context of coordinate hyperplanes, the first cut search lemma implies the following result.

Corollary. *If $y' \in C^*$, then for each component y'_i of y' there must be at least one edge C_j^* of C^* and a point $y^* \in C_j^*$ such that $y'_i = y_i^*$.*

Procedurally, this corollary has the following implications. Since each y is uniquely determined from the initial tableau by its associated x vector, and since only the integer x vectors are of interest, it suffices to restrict attention to the integer coordinate hyperplanes for the components of x . Moreover, as the edges C_j^* are parametrically extended (by increasing the t_j^* values from 0), it is easy to keep track of integer bounds L_i and U_i for each variable x_i such that the integer values of x_i encountered by the edges C_j^* are precisely those satisfying $L_i \leq x_i \leq U_i$. The intervals determined by these bounds provide the set S^* referred to earlier. That is,

$$S^* = \{y \mid L \leq x \leq U \text{ and } x \text{ integer}\} .$$

Clearly, from these observations it is particularly easy to verify whether $S^* = \emptyset$, and, if $S^* \neq \emptyset$, to determine the exact composition of S^* . The importance of this is underscored by the fact that, having examined the elements $y \in S^*$, it is possible to introduce a cut defined relative to a new truncated cone C^+ that is strictly larger than C^* , as we now show.

4.2. The edge probe and the resulting cut

In order to specify the exact procedural details of the cut search approach we require the following additional definitions and notation.

4.2.1. *The cone C^+ .* The truncated cone C^+ and its edges C_j^+ are defined relative to a specified set of values t_j^+ in the same way that C^* and C_j^* are defined relative to the values t_j^* . We let \bar{C}^+ and \bar{C}_j^+ denote the closures of C^+ and C_j^+ (just as \bar{C}^* and \bar{C}_j^* denote the closures of C^* and C_j^*). In the present context t_j^* will always be positive for all j . However, possibly some $t_j^* = 0$, in which case we define $\bar{C}_j^* = \bar{C}^* = \{B_0\}$.

4.2.2. *Admissible hyperplanes.* An *admissible* (coordinate) hyperplane is one of the form $x_i = k$, where k is an integer satisfying $L_i^o \leq k \leq U_i^o$, and where L_i^o and U_i^o are known integer bounds on x_i implied by the problem constraints (hence $L_i^o \geq 0$ and $U_i^o \leq \infty$).

4.2.3. *The set S^* .* The set $S^* = \{y | L \leq x \leq U \text{ and } x \text{ integer}\}$ is defined to consist of those y that lie on the intersection of n admissible hyperplanes each of which is intersected by at least one of the edges \bar{C}_j^* of the truncated cone \bar{C}^* . Thus, for $t^* = 0$, we may determine the L and U vectors for S^* by the rule $L_i = \langle b_{i0} \rangle$ (the least integer $\geq b_{i0}$) and $U_i = [b_{i0}]$ (the greatest integer $\leq b_{i0}$), where b_{i0} denotes the i -th component of B_0 (restricted here to $i \in N$, the index set of the components of x).² Then we note that $S^* \neq \emptyset$ (and hence $L = U$) for $t^* = 0$ only if $y = B_0$ implies x is integer. We will subsequently give rules for identifying L and U for S^* as the components of t^* are increased from 0.

4.2.4. *The t_j^+ values.* The number t_j^+ represents the “next value” to be assumed by t_j^* (for some j) in the cut search procedure, and is determined relative to the cone \bar{C}^* by the definition

$$t_j^+ = \text{Min} \{t_j | t_j \geq 0 \text{ and } \bar{C}_j^+ \text{ intersects at least one admissible hyperplane not intersected by any edge of } \bar{C}^* \text{ (where } t_j^+ = \infty \text{ if no such } t_j \text{ exists)}\}$$

or equivalently

$$t_j^+ = \text{Max} \{t_j | C_j^+ \text{ does not intersect any admissible hyperplanes except those intersected by the edges of } \bar{C}^*\} .$$

Numerically, the values of t_j^+ satisfying the foregoing definition may be calculated by reference to the L and U vectors according to the rule

$$t_j^+ = \text{Min}_{i \in N} \{\theta_i\} ,$$

where, relative to the selected index j , θ_i is defined by³

$$\theta_i = \begin{cases} (b_{i0} - L_i + 1) / b_{ij} & \text{if } b_{ij} > 0 \text{ and } L_i > L_i^o , \\ (b_{i0} - U_i - 1) / b_{ij} & \text{if } b_{ij} < 0 \text{ and } U_i < U_i^o , \\ \infty & \text{otherwise .} \end{cases}$$

² We assume that $U^o \geq x \geq L^o$ is satisfied by the solution $y = B_0$. If not, one may set $L_i = \text{Max} \{L_i^o, \langle b_{i0} \rangle\}$ and $U_i = \text{Min} \{U_i^o, [b_{i0}]\}$.

³ The calculation of θ_i can be simplified when a new value of t_j^+ is to be determined, provided a record is kept of those L_i and U_i that have changed since the previous calculation.

The complete cut search procedure can now be specified as follows.

4.2.5. *The cut search procedure*

1. If $B_0 \in S$ and the current tableau is dual feasible, the problem is solved. Otherwise, set $t^* = 0$.
2. Select one of the following two alternatives:
 - (a) *Cut alternative.* Adjoin the cut

$$\sum (1/t_j^+) t_j \geq 1 .$$

Then, using the simplex method, iterate to a new current tableau and return to instruction 1.

(b) *Edge probe ("search") alternative.* Select a finite edge \bar{C}_p^* of \bar{C}^* and extend it until it coincides with \bar{C}_p^+ , thus determining a new cone \bar{C}' (i.e., the edges of \bar{C}' are given by $\bar{C}_p' = \bar{C}_p^+$ and $\bar{C}_j' = \bar{C}_j^*$ for $j \neq p$). Then continue to instruction 3.

3. Examine the vectors y in $S' - S^*$ for feasibility (if any exist), where S' is defined relative to \bar{C}' in the same way that S^* is defined relative to \bar{C}^* .⁴ Keep track of the currently best feasible solution.

4. Redefine \bar{C}^* to be \bar{C}' and S^* to be S' . Return to instruction 2 unless all edges of \bar{C}^* (or of \bar{C}^+) are now infinite. In this latter case, the best feasible solution found at instruction 3 is optimal (or, if no feasible solution has been found, then the problem does not have a feasible solution).

It is particularly interesting to note that, while the cut search procedure extends only one of the edges of \bar{C}^* at instruction 2(b), the cut of instruction 2(a) arises by extending all edges of \bar{C}^* without requiring the examination of any vectors for feasibility. This "something for nothing" boost at instruction 2(a) is made possible by the fact that the cut need not be legitimate relative to S , but only relative to $S - S^*$, since initially $S^* = \emptyset$ and all increments $S' - S^*$ to S^* are examined at instruction 3.⁵ (It is nevertheless true that the depth of the cut may increase, as edges are extended, even though S^* itself may not change.)

The justification of the foregoing procedure, and in particular of the cut of instruction 2(a), is given by the following result.

⁴ Because of the definition of t_p^+ , the rule for determining S' in terms of S^* is particularly simple. If $t_p^+ = \infty$, none of the L_i or U_i values defining S^* change. Otherwise, for those $i \in N$ such that $t_j^+ = \theta_j$, L_i is replaced by $L_i - 1$ if $b_{ij} > 0$ and U_i is replaced by $U_i + 1$ if $b_{ij} < 0$.

⁵ The procedure can be applied in the situation where S^* is not initially empty, i.e., where $S^* = \{B_0\}$, simply by recording B_0 as the first feasible solution.

Second cut search lemma. For the values $t_j^+ > 0$ defined relative to the values t_j^* as indicated,

$$C^+ \cap S \subset S^*$$

and the cut

$$\sum (1/t_j^+) t_j \geq 1$$

is legitimate relative to $S - S^*$.

Proof. First, $y \in S^*$ if and only if y lies on the intersection of n admissible hyperplanes, each of which is intersected by one of the edges of \bar{C}^* . Also, by construction, the edges of C^+ intersect the same set of admissible coordinate hyperplanes as the edges of \bar{C}^* (although each of the non-infinite edges of \bar{C}^+ intersects at least one admissible hyperplane not intersected by the edges of \bar{C}^*). Finally, $y \in S$ only if y lies on an admissible hyperplane. Thus, by the first cut search lemma, it follows that $C^+ \cap S \subset S^*$. Consequently, $C^+ \cap (S - S^*) = \emptyset$, which verifies the legitimacy of the cut relative to $S - S^*$.

5. Characteristic features of the cut search procedure

A number of the key features of cut search can be glimpsed by reference to the description of the preceding section. First, S^* typically, but not invariably, remains empty during the first step (or in general, the first several steps). Second, the cone \bar{C}^* may be “vacuous” in the sense that one of its edges may not be extended a positive length, but the edges of C^+ are always positive and increase at each iteration. Third, the nature of the cut and the composition of S^* depends on the successive choices of the edges to extend at instruction 2(b). This is a particularly critical feature, for it enables the “inclination” of the cut to be determined largely by design. Likewise, it allows some freedom to specify which portion of the cone to probe for feasible integer solutions. This freedom of choice makes it possible to select edges according to a variety of criteria; as, for example, that of minimizing the number of new vectors added to S^* at each iteration.

A particularly interesting characteristic of the method derives from

the fact that the edge extensions are defined relative only to admissible hyperplanes. This implies that the coordinate hyperplanes “beyond” the region of such admissible hyperplanes are ignored by the method and clearly permits some of the edge extensions to be somewhat greater than would otherwise be the case. Also, the number of vectors in S^* is reduced, so that there are fewer vectors to examine for feasibility. In particular, the ability of the method to exploit bounded variables in this fashion suggests the value of redefining the “original” x variables to include slack variables of constraints that are “almost binding” at the vertex B_0 .

It is also possible to anticipate some intrinsic limitations of the method. As the edge extensions increase, the number of vectors in S^* will tend to grow at an erratic but generally increasing rate. Also, for some structures of the cone C , many of the points in S^* may be infeasible, not only relative to the full set of problem constraints, but relative to C itself.

These disadvantages are partially offset by the fact that the composition of S^* will tend to include points that are indeed likely candidates for “good solutions”, at least relative to C . Moreover, the absolute size of S^* may not be a valid indication of the effort involved in determining whether the members of $S' - S^*$ are feasible, particularly if this determination is made by a more sophisticated means than simple inspection (see section 7). Finally, we shall subsequently specify a modified procedure by which $S' - S^*$ may be replaced by a significantly smaller set of vectors, thereby offsetting the connection between the increasing increments to S^* and the number of vectors currently to be examined for feasibility.

6. Finiteness of the cut search procedure

We shall establish finiteness for a simple version of the procedure of section 4 by showing that the cuts produced by this version coincide with cuts for which finiteness has already been established. This version is a particularly “unadventurous” one, because the cut is adjoined before it becomes necessary to examine any elements of $S' - S^*$ (i.e., before $S' - S^*$ becomes nonempty).

To describe this version, we introduce the notion of a “blocked” set S^* , which may be used to provide an advanced starting point for the cut search procedure.

We will say that S^* is blocked (given $S^* = \emptyset$) if, for each finite \bar{C}_p^* , the replacement of \bar{C}_p^* by \bar{C}_p^+ to yield the cone \bar{C}' (as in instruction 2(b)) results in $S' \neq \emptyset$.

It is clearly possible for S^* to be unblocked and yet for there to exist a finite \bar{C}_p^* whose replacement by \bar{C}_p^+ results in $S' \neq \emptyset$. However, one strategy for applying cut search is to select a \bar{C}_p^* at instruction 2(b) that will result in $S' \neq \emptyset$ only if S^* is blocked, i.e., only if a number of the initial iterations of the method can be condensed into a "single step" by the following observation.

A necessary condition for S^* to be blocked is the existence of a noninteger b_{i_0} such that the endpoint of \bar{C}_j^+ intersects $x_i = [b_{i_0}]$ or $x_i = \langle b_{i_0} \rangle$ (i.e., $b_{i_0} - b_{ij}t_j^+ = [b_{i_0}]$ or $\langle b_{i_0} \rangle$) for every j . Thus, the values t_j^+ , $j \in N$, at which this condition holds (for a given index i) can be directly identified, and the corresponding t_j^* values (and the set S^*) that would ordinarily "give rise" to these t_j^+ values can be determined at a generally smaller computational expense than the process of successively iterating through instruction 2(b).

It may of course happen that the set S^* obtained in the manner just indicated is not blocked after all. However, if S^* is not blocked, there must be a noninteger b_{i_0} (other than the one previously selected) for which $[b_{i_0}] < b_{i_0} - b_{ij}t_j^+ < \langle b_{i_0} \rangle$ for all j , and new (larger) t_j^+ values can be computed relative to this b_{i_0} that will cause one of the two sides of the foregoing inequality to hold as an equality for each j . In this manner a blocked S^* still can be identified, conveniently bypassing a number of iterations of instruction 2(b).

The question now arises: what will be the nature of the cut adjoined at instruction 2(a) if the cut search procedure is stopped as soon as a blocked S^* is determined (thereby avoiding the examination of any vectors y at instruction 3)? The answer is that the resulting cut will be precisely one of the Gomory mixed integer cuts of [9]. More particularly, as pointed out in [5], the cuts of [9] arise from the inequality (2) by extending edges of C to the boundaries of the convex set given by $[b_{i_0}] \leq x_i \leq \langle b_{i_0} \rangle$, and hence it follows that the previously indicated necessary condition for S^* to be blocked gives the full set of Gomory cuts.⁶ Moreover, by the foregoing discussion, the cut corresponding to

⁶ The Gomory cuts and the cut search procedure alike may be applied to variables that are integer linear combinations of the x_j , as elaborated more fully in section 7.

a blocked S^* must be one that is deeper along some edge than any other Gomory cut and thus belongs to the subset of “mutually undominated” Gomory cuts. Finiteness of the cut search procedure is thus assured by periodically determining the edge extensions relative to a blocked set, by starting from a blocked S^* and probing further.⁷

In the next section we give a modified cut search procedure that can produce deeper cuts than discussed here even in the case in which no y vectors are examined for feasibility.

7. A modified cut search procedure

The procedure of section 4, while illustrative of several of the main ideas of cut search, is nevertheless naive in two principal ways: (i) there is no need to restrict attention to coordinate hyperplanes for the original problem variables and (ii) it is not truly necessary, if an appropriate set of variables is considered, to examine all points of $S' - S^*$ for feasibility.

The first point is obvious, but significant. The logic of the previous discussion remains valid if x is replaced by a possibly different vector of integer variables, say w . The vector w may be created in a variety of ways, as, for example, from various integer linear combinations of the original variables. It may have more or fewer than n components, some of which are continuous rather than integer (in which case the definitions of S^* and L and U must be modified appropriately).

The use of w in place of x requires the identification of precise correspondences (one-to-one, one-to-many, etc.) between these two vectors in order to check elements of $S' - S^*$ for feasibility. This process of checking for feasibility need not (and in some instances cannot) consist of simple itemization of elements of $S' - S^*$ to determine whether they lie in S . Instead, an auxiliary algorithm may be used for this process.⁸

An alternative is to stop the procedure while $S' - S^*$ is still empty, as discussed in section 6. While this alternative produces the Gomory cuts in the naive version of the method, it may produce cuts that are

⁷ We of course predicate this statement on the assumptions that x_0 is integer valued and that the appropriate lexicographic considerations are accommodated.

⁸ For example, one may apply a branch and bound or implicit enumeration procedure to the problem of maximizing x_0 , subject to $y \in (S' - S^*) \cap S$.

somewhat stronger than the Gomory cuts in more sophisticated versions (as we will show).

The second limiting characteristic of the naive version allows the procedure to be substantially improved once the proper relationships are taken into account. The basic idea for overcoming this limitation – i.e., for avoiding an exhaustive examination of elements of $S' - S^*$ for feasibility (either directly or algorithmically) – is to create a “positive” coordinate system relative to the cone C . That is, x is to be replaced by a new vector of integer variables w created in such a manner that w is a nondecreasing function of t . More particularly, if $x = B_0^1 - B^1 t$ is the current tableau representation of x (taken from the full representation $y = B_0 - Bt$), then w may be written $w = MB_0^1 - MB^1 t$, where M is an all integer matrix with an all integer inverse and $MB^1 \leq 0$. Enlarging y (if necessary) to include the components of w , and denoting the index set of these components by N_0 , the condition that w be nondecreasing as a function of t may be expressed by $b_{ij} \leq 0$ for all j and for all $i \in N_0$.

Clearly, then, the cut search procedure can be applied by replacing the x vector in the definitions of S^* , etc., with the set of $y_i, i \in N_0$. The significance of such a replacement is compelling, for it allows the set $S' = \{y \mid L'_i \leq y_i \leq U'_i \text{ and } y_i \text{ integer, } i \in N_0\}$ in instruction 3 to be replaced by a smaller set $S'' = \{y \mid L''_i \leq y_i \leq U'_i \text{ and } y_i \text{ integer, } i \in N_0\}$, where $L''_i = \text{Min}_j \{b_{i0} - b_{ij} t_j^+\}$.⁹

The justification of this assertion is given by the following result.

Third cut search lemma. If a vector $y \in S' - S^$ is feasible for the integer programming problem, then $y \in S'' - S^*$; i.e.,*

$$(S' - S^*) \cap S = (S'' - S^*) \cap S.$$

Proof. By the nonnegativity of the $b_{ij}, i \in N_0$, we have $S'' \subset S'$ (see footnote 9), and thus it is necessary only to show $(S' - S^*) \cap S \subset (S'' - S^*) \cap S$. Suppose on the contrary there exists a vector $y' \in (S' - S^*) \cap S$ such that $y'_p < L''_p$ for some $p \in N_0$. The cut determined from C^+ is legitimate relative to $S - S^*$ by the second cut search lemma; i.e., $C^+ \cap (S - S^*) = \emptyset$. Thus, since $(S' - S^*) \cap S = S' \cap (S - S^*)$, it follows that $y' \in C^+$. Now, let t'_j be given so that $y'_p = b_p - b_{pj} t'_j$ for each j

⁹ If $y = B_0$ does not satisfy the lower bound restrictions for the $y_i, i \in N_0$, then L''_i can more generally be defined to be $\text{Max}\{L'_i, \text{Min}_j \{b_{i0} - b_{ij} t_j^+\}\}$.

(letting $t'_j = \infty$ if $b_{pj} = 0$). Because $b_{pj} \leq 0$ for all j , if the hyperplane $y_p = y'_p$ is intersected by an edge of C , then $y'_p > b_{pj}$ and hence $t'_j > 0$ for all j . Consequently, the numbers t'_j are the intersection values of edges of C with the hyperplane $y_p = y'_p$. Letting C' be defined in terms of t' in the same way that C^* is defined in terms of t^* , we have $y' \in \bar{C}'$. Furthermore, from the definition of L''_p and again from $b_{pj} \leq 0$, it follows that $t'_j < t^*_j$ for all j (since by assumption $y'_p < L''_p$). Thus, $\bar{C}' \subset C^+$ and hence $y' \in C^+$, proving the lemma by contradiction.

There are a variety of ways of exploiting the foregoing lemma. First, of course, it is not truly essential that the transformation $w = Mx$ provide a fully positive coordinate system; that is, the lemma may be applied to any subset of the $y_i, i \in N_0$, such that $b_{ij} \leq 0$ for all j (replacing L'_i by L''_i only for these i).¹⁰

Second, the earlier remarks of section 7 concerning the latitude of dealing with variables other than the original x_i apply here as well. Thus, for example, M may be integer but lack an all integer inverse (in which case some of the points of $S'' - S^*$ may not be in S because they yield noninteger x vectors).

Third, strictly within the domain of positive (integer) coordinate systems, there are many different transformations $w = Mx$ that are acceptable. Studies devoted to ways of characterizing and generating "good" transformations would seem especially worthwhile. Appropriate characterizations become more difficult and perhaps also more crucial in the case of x variables that are close to their bounds at the solution $y = B_0$, as in the case of 0-1 problems. For such problems the transformation $w = Mx$ must be determined with care since the w_i may lie farther from their identifiable bounds than the x_i when $y = B_0$. (This means, loosely, that there may be more admissible hyperplanes in the vicinity of the w_i than in the vicinity of the x_i , thereby increasing the number of possibilities for "holding back" the edge extensions.)

Fourth, the lemma justifies a procedure of adjoining the cuts $y_i \geq \langle L''_i \rangle, i \in N_0$, at instruction 2(a). This may eliminate somewhat more of the region $S^* \cap C$ than the cut taken from C^+ by itself. (These cuts also have integer slack variables, which can be used in creating further integer transformations to positive or "semi-positive" coordinate systems.)

¹⁰ There is no need to consider a "negative" coordinate system, since such a system becomes positive simply by replacing y_i with $-y_i, i \in N_0$.

Fifth, $S' - S^*$ may contain several vectors to be examined for feasibility in instruction 3, while the set $S'' - S^*$ (which replaces $S' - S^*$ in the modified method) may be empty. One may take advantage of this by calculations that condense several steps into one when these steps leave $S'' - S^*$ empty. As already intimated, S^* may remain unblocked for the modified method¹¹ when it is blocked for the original method.

References

- [1] Egon Balas, "The intersection cut - A new cutting plane for integer programming" (Management Sciences Research Report No. 187, Carnegie-Mellon University, October, 1969) *Operations Research* 19 (1971) 19-39.
- [2] Egon Balas, "Alternative strategies for using intersection cuts in integer programming", presented at the TIMS XVII International Conference, London, July 1970.
- [3] Egon Balas, V. Joseph Bowman, Fred Glover and David Sommer, "An intersection cut from the dual of the unit hypercube" (WP #90-69-7, Carnegie-Mellon University, April 1970) *Operations Research* 19 (1971) 40-44.
- [4] Claude A. Burdet, "A class of cuts and related algorithms in integer programming", Management Sciences Research Report No. 220, Carnegie-Mellon University, September 1970.
- [5] Fred Glover, "Convexity cuts", working paper, School of Business, University of Texas, Austin, December 1969.
- [6] Fred Glover, "Convexity cuts and cut search" (MSRS 70-2, School of Business, University of Colorado at Boulder, July 1970) *Operations Research*, to appear.
- [8] Fred Glover and D. Klingman, "The generalized lattice point problem" (MSRS 71-3, School of Business, University of Colorado, Boulder, May 1971), presented at the 39th National OSRA Meeting, Dallas, May 1971.
- [9] Ralph Gomory, "An algorithm for the mixed integer problem", Research Memorandum RM-2597, Rand Corporation, Santa Monica, 1960.
- [10] M. Raghavachari, "On the zero-one integer programming problem" (Technical Report No. 20, Department of Statistics, Carnegie-Mellon University, Pittsburgh, Pennsylvania), *Operations Research* 17 (1969) 680-685.
- [11] Hoang Tui, "Concave programming under linear constraints", *Doklady Akademii Nauk SSSR*, 1964 (in Russian), *Soviet Mathematics* (1964) 1437-1440 (English translation).
- [12] Richard D. Young, "New cuts for a special class of 0-1 integer programs", Research Report in Applied Mathematics and Systems Theory, Rice University, Houston, Texas, October 1968.
- [13] Richard D. Young, "Hypercylindrically deduced cuts in zero-one integer programs" (Rice University, Houston, Texas, June 1970), *Operations Research*, to appear.

¹¹ We may define S^* to be blocked for the modified method if $S'' - S^*$ has always previously been empty but cannot currently remain empty for any choice of \bar{C}_p^+ at instruction 2(b).