A NEW FOUNDATION FOR A SIMPLIFIED PRIMAL INTEGER PROGRAMMING ALGORITHM†

Fred Glover

University of Texas, Austin, Texas

(Received June 23, 1967)

Following the approach underlying the Pseudo Primal-Dual Integer Programming Algorithm, a new foundation for a simplified primal integer programming algorithm is given. In addition, new choice rules are prescribed which guarantee finite convergence, and a criterion of optimality is introduced that permits the algorithm to terminate before dual feasibility is achieved.

FOR SEVERAL years a variety of researchers have stressed the importance of a primal integer programming algorithm (*see*, e.g., references 1 and 5), but it was not until RICHARD D. YOUNG's‡ pioneering paper[6] that a finite primal algorithm became available. Young's algorithm was an outstanding contribution to integer programming, but difficult to implement. Subsequently, ideas underlying the author's pseudo primal-dual integer programming algorithm[3] suggested the possibility of a primal integer algorithm that does not require the multiple pivot prescriptions of the algorithm of reference 6. However, the simpler algorithm, as initially made explicit in correspondence with Young and T. C. Hu§, had an essential gap in its convergence proof. This gap, too, was later found to be mendable by results of the pseudo primal-dual algorithm, but it was Young who first discovered that the procedure could indeed be made finite, by noting strong correspondences between some of its ideas and those of his original algorithm.

Young's paper,[7] which is a companion to this, contains the results that support this discovery and tie the simplified algorithm to theory underlying his earlier work. Here, drawing more fully on reference 3, we give other

† This research was supported in part by the National Science Foundation, Grant GP-4593, and the Office of Naval Research under Contract Nonr-222(83), with the University of California.

‡ The author is deeply indebted to RICHARD D. YOUNG for many stimulating discussions that have contributed to the development of this paper.

§ T. C. Hu was one of the first to appreciate the significance of Young's primal method, and was responsible for organizing the 1965 NATO Conference on Integer Programming and Network Flows at Lake Tahoe, at which Young and I presented the papers[6, 3] that provide the foundation for the simplified primal method.

rules that make the simplified procedure finite, and tie the development to results underlying the pseudo primal-dual method. As already intimated, several of these results and those of Young,[7] while evolved from different formulations and perspectives, significantly overlap in content. We have attempted in these companion papers to interrelate our presentations to emphasize the complementarities of our differing analytical points of view, as well as divergences in theory and their consequences for computation.

DESCRIPTION OF THE PROBLEM

WE REPRESENT the ordinary linear programming problem P1 as that of maximizing in nonnegative variables

$$x_o = a_{oo} + \sum_{j=1}^{j=n} a_{oj}(-t_j),$$

subject to

$$\begin{aligned} x_i &= a_{io} + \sum_{j=1}^{j=n} a_{ij}(-t_j), && (i=1,\cdots,m) \quad (1)\\ x_{m+j} &= -(-t_j), && (j=1,\cdots,n) \end{aligned}$$

or in matrix form to maximize x_0 subject to

$$X = A^0 T^0,\ A^0 = (A_0, A_1, \cdots, A_n)$$

$$X = \left\| \begin{matrix} x_0 \\ x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_{m+n} \end{matrix} \right\|, \qquad T = \left\| \begin{matrix} 1 \\ -t_1 \\ -t_2 \\ \cdot \\ \cdot \\ \cdot \\ -t_n \end{matrix} \right\|.$$

The matrix A^0 is *dual feasible* if $a_{0j} \geqq 0$ for $j=1, \cdots, n$, and *primal feasible* if $a_{i0} \geqq 0$ for $i=1, \cdots, m+n$. As is well known, an optimal solution to $P1$ is immediately given by $X = A_0$ when both primal and dual feasibility hold.

The pure integer programming problem $P2$, which provides the chief focus of this paper, is the same as $P1$ except that the components of X are additionally required to be integers. Following the lead of Young,[6] we will specify a method for solving $P2$ that yields a nonnegative (primal feasible) integer X and a nondecreasing value of x_0 at each stage of the solution process.† To provide a foundation for this method, we review the ver-

† The value of such an approach is at least threefold. First, it is possible to begin with a known feasible integer solution and obtain progressively better ones. Second, one may discontinue the process of solving $P2$ at any stage and still have a workable, if not optimal, solution. Third, the method typically provides a range of feasible integer solutions instead of single best one, which may be useful in certain

sion of the simplex algorithm that exhibits the same characteristics in solving $P1$ except that the successive X vectors may not be integer.

THE PRIMAL SIMPLEX ALGORITHM (PSA)

BEGINNING WITH A^0 primal feasible, the primal simplex method for solving $P1$ determines a sequence of representations for X:

$$X = A^0T^0 = A^1T^1 = A^2T^2 = \cdots = A^kT^k,$$

$$T^h = (1, -t_1^h, -t_2^h, \cdots, -t_n^h), \qquad (h=0, 1, \cdots, k)$$

where $t_j^h\,(j=1, \cdots, n)$ is nonnegative, $a_{00}^h \leqq a_{00}^{h+1}$, and A^h is primal feasible for each h. When $P1$ is bounded for optimality, the matrix A^k (for finite k) is also dual feasible.

For simplicity, we will let A and T denote any matrix A^h and vector T^h, and let $\bar{A}$ and $\bar{T}$ denote the corresponding A^{h+1} and T^{h+1}. Then the precise rules of the PSA are as follows:

1. If $a_{0j} \geqq 0$ for all $j \geqq 1$, then $X = A_0$ is an optimal solution. Otherwise, select $s \geqq 1$ such that $a_{0s} < 0$.

2. If $a_{is} \leqq 0$ for all $i \geqq 1$, $P1$ has an unbounded optimum. Otherwise, compute

$$\theta_s = \min_{a_{is}>0}\,(a_{i0}/a_{is}).$$

3. Select v such that $a_{vs} > 0$ and $a_{v0}/a_{vs} = \theta_s$.‡

4. Determine $\bar{A}$ by the rules:

$$\bar{A}_s = -A_s/a_{vs},$$

$$\bar{A}_j = A_j - A_s(a_{vj}/a_{vs}) \qquad \text{if} \qquad j \neq s.$$

5. Let $\bar{t}_s \equiv x_v$ and $\bar{t}_j \equiv t_j$ for $j \neq s$. Designate $\bar{A}$ and $\bar{T}$ to be the current A matrix and T vector, and return to instruction 1.

Because the PSA is quite effective for solving $P1$, it is natural to seek an adaptation of this algorithm for solving P2 that maintains A primal feasible *and integer* at each stage. The first step toward such an adaptation (the Rudimentary Primal Algorithm) is straightforward, and has apparently been rediscovered on several occasions (*see*, e.g., references 1, 5, and 6).

THE RUDIMENTARY PRIMAL ALGORITHM (RPA)

ASSUME THAT the constant components of the initial A matrix are integers and the t_j are nonnegative integer variables.

---

applications. (These features may of course be of only slight advantage in solving problems that are extremely resistant to the method, unless such problems are correspondingly difficult for other algorithms.)

‡ We do not concern ourselves with tiebreaking rules in the choice of v, since those that assure finiteness for the PSA have little bearing on finiteness for a primal integer method.

As GOMORY[4] has shown, equation (1) then implies the cut†

$$s=[a_{i0}/\lambda]+\textstyle\sum_{j=1}^{j=n}[a_{ij}/\lambda](-t_j), \tag{2}$$

where $\lambda>0$ and s is a nonnegative integer variable. Thus $X=AT$ may be augmented to include (2) without altering the set of feasible integer solutions to $P2$. Based on this, the RPA occurs by replacing instruction **3** of the PSA with the instruction **3A** below.

3A. Select as the source equation for (2) any equation i such that $a_{is}>0$ and $0\leqq[a_{i0}/a_{is}]\leqq\theta_s$ (e.g., the equation v of instruction 3). Let $\lambda=a_{is}$, and designate (2) to be equation v of $X=AT$.

Designating (2) to be equation v is simply a symbolic device to specify the transformation of A into $\bar{A}$ by instruction **4** of the PSA (hence of the RPA). It is unnecessary to augment $X=AT$ by (2) if one is interested only in the values of the original variables x_i.

On the other hand, it is obvious that if equation (2) as determined by **3A** were adjoined to $X=AT$, then it would qualify to be selected as equation v in instruction **3** of the PSA. Also, the coefficient a_{vs} bequeathed by instruction **3A** is always 1 ($[a_{is}/a_{is}]$). These facts clearly assure that the successive A matrices determined by the RPA will be all integer and primal feasible. Unfortunately, however, there is no assurance that the RPA will converge to an optimal integer solution.

In a highly original paper,[6] Young showed how the RPA could be extended by the addition of a complex set of rules to produce a finite primal integer algorithm. Subsequently, certain ideas and procedure from reference 3 gave rise to a markedly simpler set of rules which Young was the first to justify by a slight modification of his earlier results. This justification, however, remained complicated. Relying still more heavily on reference 3, we now introduce an alternate framework that provides the simpler rules from a few elementary theorems, and in addition provides other rules that lead to a convergent algorithm.

THE SIMPLIFIED PRIMAL ALGORITHM

TO PRODUCE a convergent primal integer algorithm, it suffices to‡

(i) adjoin to $X=AT$ an additional primal feasible and all integer equation (call it r) whose coefficients a_{rj} satisfy certain properties in relation to the A_j,

† $[u]$ denotes the greatest integer $\leqq u$.

‡ One may stop and restart this procedure at finite intervals in the execution of the RPA if the number of digressions is finite. This can be assured for example by reliance on an intervening criterion of progress, such as an uninterrupted increase in the sum of the negative a_{0j} for $j\geqq1$, or in the value of a_{00} (Young's 'transition cycles').

(ii) select s in instruction 1 of the RPA by reference to the properties of equation r,

(iii) periodically select the source equation for (2) in instruction **3A** of the RPA by reference to the size of the coefficients a_{is} (or other criteria to be introduced subsequently).

We develop the properties that we wish equation r to satisfy (in addition to being consistent with the other equations of $X=AT$), while simultaneously motivating and justifying the choice of s according to (ii). To this end, we introduce the following result:

LEMMA.† *Let $\bar{A}_j=A_j-kA_s$ for some scalar k. Then for each pair of indices i, r:*

$$a_{rj}a_{is}<a_{rs}a_{ij}\,(>a_{rs}a_{ij}\ =a_{rs}a_{ij})$$

if and only if

$$\bar{a}_{rj}a_{is}<a_{rs}\bar{a}_{ij}\,(>a_{rs}\bar{a}_{ij}\ =a_{rs}\bar{a}_{ij}).$$

Proof. By definition,

$$\bar{a}_{rj}a_{is}=(a_{rj}-ka_{rs})a_{is}=a_{rj}a_{is}-ka_{rs}a_{is}.$$

Also,

$$a_{rs}\bar{a}_{ij}=a_{rs}(a_{ij}-ka_{is})=a_{rs}a_{ij}-ka_{rs}a_{is}.$$

Thus

$$\bar{a}_{rj}a_{is}-a_{rs}\bar{a}_{ij}=a_{rj}a_{is}-a_{rs}a_{ij}.$$

This proves the lemma.

Note that the definition of $\bar{A}_j$ in the foregoing lemma accords with the definition of $\bar{A}_j$ for $j\neq s$ in instruction 4 of the RPA. Using this same definition, we now extend the result of the lemma to a lexicographic relation‡ between vectors.

THEOREM 1.

$$(H1)\,a_{rj}A_s\overset{l}{<}a_{rs}A_j\,(\overset{l}{>}a_{rs}A_j,\ =a_{rs}A_j),$$

if and only if

$$(H2)\,\bar{a}_{rj}A_s<a_{rs}\bar{A}_j\,(\overset{l}{>}a_{rs}\bar{A}_j,\ =a_{rs}\bar{A}_j).$$

Proof. Let $p=\min(i:\bar{a}_{rj}a_{is}\neq a_{rs}\bar{a}_{ij})$ and $q=\min(i:a_{rj}a_{is}\neq a_{rs}a_{ij})$. By the lemma, $p=q$ and Theorem 1 follows immediately.

Theorem 1 gives direct access to the properties we desire the reference equation r to possess. Indeed, we will want equation r to be created so that s may be selected to satisfy $(H1)$ of Theorem 1. The power of Theorem 1 is that it implies, for s so selected, that equation r will still satisfy the

† This lemma is essentially Lemma 1 of reference 3.

‡ A vector A_h is defined to be lexicographically smaller than a vector A_k (symbolized $A_h\overset{l}{<}A_k$ or $A_k\overset{l}{>}A_h$) if and only if the first nonzero component of A_k-A_h is positive.

same properties relative to the new matrix $\bar{A}$. In addition, when the appropriate choice rules are implemented, the theorem assures that a form of lexicographic progress will occur in passing from A to $\bar{A}$.

To make the foregoing precise, define

$$A_j^* = A_j / a_{rj}$$

for those $j \geqq 1$ such that $a_{rj} \neq 0$. Then we specify the choice of s in instruction 1 so that:

$$a_{rs} > 0 \quad \text{and} \quad A_s^* \overset{l}{<} A_j^* \quad \text{for all} \quad j \neq s\,(s,\, j \geqq 1) \text{ such that } a_{rj} > 0.$$

Note that if there exists a j such that $A_j \overset{l}{<} 0$ and $a_{rj} > 0$, then A_s exists and is lexicographically negative. (There is clearly no need to consider the possibility $A_j^* = A_k^*$ for $j \neq k$ since the initial A_j for $j \geqq 1$ include the $-I$ matrix, and hence begin and remain linearly independent.)

The properties that we require equation r to satisfy are then as follows:†

$$(J1)\, A_j \overset{l}{<} 0 \Rightarrow a_{rj} > 0,$$

$$(J2)\, a_{rj} < 0 \Rightarrow A_j^* \overset{l}{>} A_s^*.$$

It may be observed that any equation with all positive coefficients will automatically satisfy both $(J1)$ and $(J2)$. In particular it suffices to create equation r initially with $a_{rj} = 1$ for all $j \geqq 1$ and a_{r0} equal to an upper bound for $\sum t_j$.‡

An interesting consequence of $(J1)$ is that if $a_{0s} \geqq 0$ (equivalently, $a_{0s}^* \geqq 0$), then the relation $A_s^* \overset{l}{<} A_j^*$ for $a_{rj} > 0$ implies $a_{0j} \geqq 0$ for all $j \geqq 1$, and hence A is dual feasible. But $a_{0s} < 0$ implies that any change in A_0 must produce an (integer) increase in a_{00}. Thus, in a bounded problem, the number of changes in A_0 must be finite whether the algorithm itself is finite or not. Our next theorem summarizes the joint implications of Theorem 1 and the properties $(J1)$ and $(J2)$ we have required of equation r.

THEOREM 2.§ *If $(J1)$ and $(J2)$ are satisfied for the matrix A, then $(H1)$ of*

† These properties, as developed in connection with the Pseudo Primal-Dual Integer Programming Algorithm, are approximately equivalent to Young's subsequent definition of an 'arranged tableau' in reference 7. Note that $(J2) \Rightarrow (J1)$ if there exists a j such that $a_{rj} > 0$ and $A_j \overset{l}{<} 0$.

‡ Under the assumption that $P2$ is bounded, any all-integer linear form $\sum a_{rj} t_j$ can be maximized by the simplex method subject to $X = AT$ to determine an upper bound a_{r0}. It is implied by our results to follow that if the a_{rj} satisfy $(J1)$ and $(J2)$, and if $a_{r0} < 1$, then $X = A_0$ already provides an optimal solution to $P2$.

§ Theorem 2, from results of reference 3, has a close resemblance to some of the results of Young's original primal algorithm,[6] and very probably is collectively implied by them under appropriate reformulation.

Theorem 1 is true, and (J1) and (J2) are also satisfied for $\bar{A}$. Moreover, $\bar{A}_j^* \overset{l}{>} A_s^*$ for all j such that $\bar{a}_{rj} > 0$ (in particular, $\bar{A}_{\bar{s}}^* \overset{l}{>} A_s^*$, where $\bar{s}$ is defined relative to $\bar{A}$ as s is to A).

Proof. The definition of A_s^* directly implies $(H1)$ for those $j \neq s$ such that $a_{rj} > 0$, and $(J2)$ implies $(H1)$ for those j such that $a_{rj} < 0$. If $a_{rj} = 0$, $(H1)$ follows from $(J1)$ and the fact that $A_j \neq 0$ for all j. Thus, $(H2)$ is true by Theorem 1 and $\bar{A}_j \overset{l}{<} 0 \Rightarrow \bar{a}_{rj} > 0$. Hence $(J1)$ is satisfied for $\bar{A}$. Also, dividing $(H2)$ through by $\bar{a}_{rj} a_{rs}$ gives $\bar{A}_j^* \overset{l}{>} A_s^*$ when $\bar{a}_{rj} > 0$ and $A_s^* \overset{l}{>} \bar{A}_j^*$ when $\bar{a}_{rj} < 0$, $j \neq s$. The former proves the last assertion of the theorem, and the latter in conjunction with $\bar{A}_{\bar{s}}^* \overset{l}{>} A_s^*$ proves $(J2)$ holds in $\bar{A}$ if $j \neq s$. But $(J2)$ also holds if $j = s$ since $\bar{A}_s^* = A_s^*$. This completes the proof.

Having now established the form of equation r and the definition of the index s, it remains to specify the choice of the source equation for equation (2) in instruction **3A**. The following result, in conjunction with Theorem 2, lays the foundation for this choice.

THEOREM 3.† *If $\lambda = a_{is} > 0$ and equation (2) is designated to be equation v in instruction 4 of the RPA, then*

$$\bar{a}_{is} = -a_{is} \qquad \textit{and}$$

$$a_{is} > \bar{a}_{ij} \geqq 0 \qquad \textit{for} \qquad j \neq s.$$

Proof. Since $a_{vs} = 1$, $\bar{a}_{is} = -a_{is}$. Also, for $j \neq s$, $\bar{a}_{ij} = a_{ij} - [a_{ij}/a_{is}] a_{is}$. The theorem follows at once from the fact that $u \geqq [u] > u - 1$ for all u.

Note that if $a_{is} > a_{i0}$,‡ then Theorem 3 implies that one may repeatedly select equation i as source equation for (2) in instruction **3A** and thereby eventually assure $a_{is} \leqq a_{i0}$, unless, of course, A becomes dual feasible in the interim.

By reference to this fact, Young then gives the following prescription for the selection of the source equation: use any rule that assures, for each $i \geqq 1$ (including $i = r$), $a_{is} \leqq a_{i0}$ will occur at finite intervals. Theorem 3 provides a ready mechanism for implementing this prescription, as indicated by our foregoing remarks.

We will here give some alternate choice rules that also produce a convergent primal algorithm and are easily justified within the framework of our present development. The first rule is slightly more flexible than the one given above.§

† Theorem 3 was first introduced in the context of a dual algorithm by the author in reference 2, and in the context of a primal algorithm at about the same time by Young in reference 6.

‡ Since $a_{i0}/a_{is} = 0$, equation i is a permissible source equation in instruction **3A**.

§ It is, however, close in spirit to Young's justification of his rule.

Rule 1: Make any choice that assures $a_{rs} \leqq a_{r0}$ at finite intervals, and periodically reduces a^*_{is} for the least $i \geqq 1$ such that $a^*_{is} > a_{i0}$.

We note that, since $a_{rs} \geqq 1$ [by $(J1)$], it follows that $a_{is} \geqq a^*_{is}$ for $a_{is} \geqq 0$, and hence $(a^*_{is} > a_i) \Rightarrow (a_{is} > a_{i0})$ [equivalently, $(a_{is} \leqq a_{i0}) \Rightarrow (a^*_{is} \leqq a_{i0})$]. Consequently, Rule 1 is meaningful and can be implemented by repeatedly selecting equation i as the source equation until a^*_{is} is decreased, unless A becomes dual feasible and the algorithm terminates first. We prove that this rule provides a finite algorithm as follows.

Justification of Rule 1: As observed earlier, A_0 can be changed only a finite number of times. Hence, for each i, there exists a finite constant U_i such that $a_{i0} \leqq U_i$ for all values assumed by a_{i0}. Assume that Rule 1 is not finite. Then there is an infinite set T of A matrices in which $a_{rs} \leqq a_{r0} \leqq U_r$. Since $A_s{}^*$ is lexicographically strictly increasing and $a_{0s} < 0$ it follows that a^*_{0s} can only assume a finite number of values in T and hence must eventually become constant (both in T and outside of T). Applying this argument to successive components of $A_s{}^*$, at least one of which must be unbounded, there exists an index $q \geqq 1$ such that for all A matrices after an initial finite number (call the infinite remaining set of matrices S), $a^*_{qs} > U_q \geqq a_{q0}$ and a^*_{is} is nondecreasing for all $i \leqq q$. But at some point in S, Rule 1 will reduce a^*_{is} for some $i \leqq q$ such that $a^*_{is} > a_{i0}$, which is impossible. This completes the justification.†

The second rule we give has a somewhat different character than the first, drawing on additional results underlying the Pseudo Primal-Dual Algorithm. The Pseudo Primal-Dual Algorithm establishes dual feasibility‡ by a strict increase in the first nonzero component of A_s at each step, thus providing a more evident push toward convergence than the successive lexicographic increases in $A_s{}^*$. In the interest of achieving comparable advances toward convergence with a primal algorithm, one is tempted to select a source equation whenever possible that will result in $\bar{a}_{0\bar{s}} > a_{0s}$. The somewhat surprising fact is that such a choice will indeed produce a finite algorithm, as we now show.

Rule 2: At finite intervals: select as source equation (if possible) one that will result in $\bar{a}_{0\bar{s}} > a_{0s}$, and continue the selection of source equations by this criterion until there are none that satisfy it.

As a basis for establishing the validity of this rule, we introduce§

† It is clear from this proof that a_{r0} and a_{i0} may alternately be replaced by U_r and U_i in the specification of Rule 1. Also, it is unnecessary to require a_{is} to be decreased unless it has been nondecreasing for some finite duration.

‡ A dual feasible matrix is driven dual infeasible in such a way that equation r occurs naturally in $X = AT$. Then dual feasibility is restored by continued selection of equation r as the source equation, and the net progress in A_0 between two consecutive instances of dual feasibility is at least as great as that produced by pivoting with the dual simplex method.

§ This result abstracts a portion of Theorem 2 of reference 3. Part of the remainder of the theorem is developed in the justification for Rule 2.

THEOREM 4. *For A and $\bar{A}$ satisfying conditions* $(J1)$ *and* $(J2)$, *and* $a_{0s}<0$:

$$(\bar{a}_{r\bar{s}}<a_{rs}) \Rightarrow (\bar{a}_{0\bar{s}}>a_{0s}).$$

Proof. By Theorem 2, $(H1)$ is satisfied, hence by Theorem 1 $a_{0s}\bar{a}_{r\bar{s}} \leqq a_{rs}\bar{a}_{0\bar{s}}$. Since $a_{0s}<0$ and $a_{rs}>0$, we have $\bar{a}_{0\bar{s}}/a_{0s} \leqq \bar{a}_{r\bar{s}}/a_{rs}<1$, hence $\bar{a}_{0\bar{s}}>a_{0s}$.

The completed proof that Rule 2 yields a finite primal algorithm is as follows:

Justification of Rule 2: Assume that the rule is not finite. By Theorem 3, if equation r is selected as the source equation, then $\bar{a}_{r\bar{s}}<a_{rs}$ and hence by Theorem 4 $\bar{a}_{0\bar{s}}>a_{0s}$. Thus, Rule 2 assures $a_{rs} \leqq a_{r0}$ for an infinite number of A matrices. Identify q and S as in the justification for Rule 1. If q were selected as source equation in S (it is always eligible), we have $\bar{a}_{q\bar{s}}<a_{qs}$ (by Theorem 3), and this in conjunction with $\bar{a}^*_{q\bar{s}} \geqq a^*_{qs}$ implies $a_{rs}>\bar{a}_{r\bar{s}}$. Then by Theorem 4, for every A matrix in S there is a source equation available (namely q) that will yield $\bar{a}_{0\bar{s}}>a_{0s}$. Rule 2 will thus produce an infinite number of consecutive integer increases in a_{0s}, which is impossible.†

Until now we have assumed that the algorithm terminates only when A becomes dual feasible. A basis for eliminating this assumption, and hence for increasing the effectiveness of the choice rules, is embodied in the following theorem.‡

THEOREM 5. *Assume that A is dual infeasible and satisfies* $(J1)$ *and* $(J2)$. *Then the optimal value of x_0 must satisfy*

$$x_0 \leqq a_{00}+[-a_{r0}a^*_{0s}].$$

Proof. Let $P2'$ be the problem obtained from $P2$ by introducing a new variable t_w, and replacing each equation of $P2$ by

$$x_i=a_{i0}+\sum_{j=1}^{j=n} a_{ij}(-t_j)+a_{iw}(-t_w), \qquad i=0, 1, \cdots, m+n+1,$$

(where, e.g., $r=m+n+1$), and adjoining the two additional equations

$$x_{m+n+2}=-(-t_w), \qquad x_{m+n+3}=-t_w.$$

These last equations assure that every optimal solution $P2'$ is also optimal

† It may be noted that Rule 2 remains valid by this proof when the condition $\bar{a}_{r\bar{s}}<a_{rs}$ replaces $\bar{a}_{0\bar{s}}>a_{0s}$. An argument mirroring the proof of Theorem 4 also shows that selecting equation i as source equation will result in $\bar{a}_{r\bar{s}}<a_{rs}$ whenever $\bar{a}^*_{i\bar{s}} \geqq a^*_{is}$. Other rules closely related (but not equivalent) to the foregoing ones can immediately be inferred from this: e.g., enforce $a_{rs} \leqq a_{r0}$ at finite intervals and periodically select any source equation i such that $a^*_{is}>a_{i0}$ and $\bar{a}^*_{i\bar{s}} \geqq a^*_{is}$, continuing the selection of i by this criterion as long as possible.

‡ This theorem can alternately be justified by the observation by Young that an appropriate multiple of row r provides a feasible solution to the dual of $P1$ (defined for the current A matrix). In this connection, BEN-ISRAEL AND CHARNES[1] anticipate the present use of this result by advocating the solution of an auxiliary problem to prescribe termination before A is dual feasible.

for $P2$, disregarding t_w, x_{m+n+2} and x_{m+n+3} (which must all equal 0). Let $a_{iw}=-a_{is}/a_{0s}$† for $i=0, 1, \cdots, m+n+1$. Then clearly $\bar{w}$ satisfies the requirements for s in $P2'$. Suppose equation r is selected as the source equation in $P2'$, with a_{rw} taking the place of a_{rs} (whether or not $a_{r0}/a_{rw} \leqq \theta_w$).

It follows from Theorems 3 and 4 that $\bar{a}_{0\bar{w}} > a_{0w}$, where $\bar{w}$ denotes the index that corresponds to $\bar{s}$ in $P2'$. But $a_{0w}=-1$, hence $\bar{a}_{0\bar{w}}=0$ (it must be an integer) and $\bar{A}$ is dual feasible in $P2'$ (though possibly not primal feasible). Consequently, $\bar{a}_{00}$ (in $P2'$) provides an upper bound for x_0. But from instruction 4 of the RPA, with w replacing s, we have $\bar{a}_{00}=a_{00}+[-a_{r0}\overset{*}{a}_{0s}]$, since $a_{0w}=-1$ and

$$a_{rw}=-(a_{rs}/a_{0s})=-(1/\overset{*}{a}_{0s}).†$$

It is immediate from Theorem 5 that $X=A_0$ provides optimal solution to $P2$ whenever $\overset{*}{a}_{0s} > -(1/a_{r0})$. This fact not only permits earlier termination of the algorithm, but also gives rise to an additional choice rule.

Suppose that $X=AT$ is augmented by the equation

$$x_u=a_{u0}+\sum_{j=1}^{j=n} a_{uj}(-t_j),$$

where $a_{uj}=-a_{0j}$ for $j \geqq 1$ and $-a_{u0}$ is a nonpositive lower bound for $\sum a_{0j}t_j$. Then we have

Rule 3: Replace $a_{rs} \leqq a_{r0}$ by $a_{us} \leqq a_{u0}$ in Rule 1, and terminate when $\overset{*}{a}_{0s} > 1/a_{r0}$ (if A is not dual feasible).‡

Justification for Rule 3: The justification is the same as for Rule 1 if $a_{rs} \leqq C$ occurs periodically for some finite constant C. Otherwise, if the algorithm does not converge, $a_{rs} > U_u \cdot U_r \geqq a_{u0}a_{r0}$ for all A matrices except an initial finite number. In some of the subsequent matrices $a_u \geqq a_{us}=-a_{0s}$ and hence $\overset{*}{a}_{0s} > -(1/a_{r0})$, contrary to the nonconvergence assumption.

DETERMINING A REFERENCE EQUATION

THE CHOICE of a reference equation to guide the progress of the algorithm can have a marked influence on the speed of convergence. As noted in the preceding section, one permissible reference equation is given by $x_r=a_{ro}+\sum(-t_j)$, where a_{ro} is an upper bound on $\sum t_j$. In this section we propose a different reference equation that can sometimes reduce the number of steps required to obtain an optimal solution.

† More generally in defining $P2'$ let $a_{iw}=-ka_{is}/a_{0s}$ for $k>0$. Then selecting r as a source equation yields $\bar{a}_{0j}=a_{0j}+k[\rho_j/k]$ for $j=0, 1, \cdots, n$ where $\rho_j=-a_{rj}\overset{*}{a}_{0s}$. Thus a more restrictive upper bound on x_0 is obtained by selecting $k>0$ to minimize $a_{00}+k[\rho_0/k]$ subject to $k[\rho_j/k] \geqq a_{0j}$ (to assure $\bar{a}_{0j} \geqq 0$), $j=0, 1, \cdots, n$. Note that a_{0s}/k must be an integer to satisfy this latter inequality for $j=s$.

‡ Since $\overset{*}{a}_{0s}$ is nondecreasing, $a_{rs} \leqq a_{r0}$ implies $a_{rs} \leqq a_{u0}$ for an appropriately large value of a_{u0}, and hence Rule 3 includes Rule 1 as a special instance.

For convenience, we rewrite our original problem (maximize x_o subject to $X=AT$, etc.) in the form†

$$\text{maximize } cy,$$

subject to $$Qy \leqq b,\ y \geqq 0 \text{ and integer},$$

where $c=-(a_{o1}, a_{o2}, \cdots, a_{on})$, $Q=(a_{ij})$ for $i=1, \cdots, m$ and $j=1, \cdots, n$, $y=(t_1, t_2, \cdots, t_n)^T$, and $b=(a_{1o}, a_{2o}, \cdots, a_{mo})^T$. Dropping the integer restriction on the variables, the dual of this problem is

$$\text{minimize } wb,$$

subject to $$wQ \geqq c,\ w \geqq 0,$$

where $w=(w_1, w_2, \cdots, w_m)$ is a row vector.

If $a_{oj} \geqq 0$ for some $j \geqq 1$, we shall assume that the coefficients of equation (1) of $X=AT$ satisfy $a_{1j}=1$ if $a_{oj} \geqq 0$, and $a_{1j}=0$ if $a_{oj}<0$. Then we state

THEOREM 6.‡ *Let w^* denote a feasible solution to the dual linear program. Then the equation r whose coefficients are defined by $a_{ro}=w^*b$ and $a_{rj}=w^*Q_j$ for $j \geqq 1$ satisfies $(J1)$ and $(J2)$ and qualifies to be a reference equation.*

Proof. By the feasibility of w^* for the dual, $a_{rj} \geqq -a_{oj}$ for $j \geqq 1$. Because of the assumed form of equation (1), we have $A_j \overset{l}{<} 0$ implies $a_{oj}<0$ and hence $a_{rj}>0$, verifying $(J1)$. Also, if $a_{rj}<0$, then $a_{oj}/a_{rj} \leqq -1$, while if $a_{rj}>0$ then $a_{oj}/a_{rj} \geqq -1$. Thus, $a_{rj}<0$ implies $\overset{*}{a}_{oj} \leqq \overset{*}{a}_{os}$, and hence either $A_j^* \overset{l}{<} A_s^*$ or $\overset{*}{a}_{oj}=\overset{*}{a}_{os}$. But $a_{rj}<0$ also implies $A_j \overset{l}{>} 0$. Thus, if $\overset{*}{a}_{oj}=\overset{*}{a}_{os}$, the form of the first equation yields $\overset{*}{a}_{1j}<0$ and $\overset{*}{a}_{1s}=0$, and again $A_j^* \overset{l}{<} A_s^*$. This verifies $(J2)$. Finally, since $w^* \geqq 0$, $a_{ro} \geqq 0$ and $x_r \geqq 0$, completing the proof. (It is not strictly necessary for x_r to be an integer variable since the a_{rj} will all be rational, and the arguments requiring integer a_{rj} easily extend to rational a_{rj}.)

Our proposed use of Theorem 6 is to let w^* be an optimal solution to the dual.§ In the next section we solve an example problem using such a reference equation.

† The constant term a_{oo} can be disregarded.

‡ This result is also implicit in comments of reference 7.

§ We do not mean to suggest that this is the most effective reference equation available. One that may be better would be to select the a_{rj} so that, if t_j takes a larger value in the linear programming solution, A_j^* tends to be smaller for $a_{oj}<0$ and larger for $a_{oj}>0$. There may also be strategic points at which to append such a reference equation; for example, when $a_{oj}<0$ holds for a large number of the j for which t_j is positive in the continuous solution. Periodically introducing new reference equations of this type (or the type proposed above) would seem reasonable.

A NUMERICAL EXAMPLE

We shall solve the problem summarized by the following A matrix:

$$
\begin{array}{rr|rrr}
 & & & & s \\
 & 0 & -4 & -6 & -3 \\
\hline
 & 5 & 1 & 2 & 0 \\
 & 8 & 9 & 2 & -4 \\
* & 1 & -3 & -2 & 2 \\
 & 16 & -5 & 4 & 6 \\
r: & 500 & 68 & 125 & 51 \\
 & 0 & -1 & 0 & 0 \\
 & 0 & 0 & -1 & 0 \\
 & 0 & 0 & 0 & -1
\end{array}
$$

The horizontal line separates the top (objective function) row and the vertical line separates the A_o column from the rest of the matrix. Equation r is indicated by the r to its left and is obtained as indicated in the preceding section. The optimal solution w^* to the dual linear program assigns a weight of 39/34 to equation (2) and 43/34 to equation (4). We have multiplied the resulting equation r by 17 to make its coefficients integers. Column s is similarly indicated by the s above it. In this instance, $s=3$ since A_3^* is the lexicographically smallest A_j^* such that $a_{rj}>0$.

Equations that are eligible to be the source equation† are indicated by an asterisk [as equation (3) above]. When more than one equation is eligible, the one that is actually selected is indicated by a double asterisk. The selection is made by picking the candidate that maximizes $\bar{a}_{os}$ (a_{os} in the next A matrix). Ties are broken by selecting the equation that also maximizes the sum of the negative $\bar{a}_{oj}$, and then selecting the equation with the smallest index if ties still remain. This rule, while mildly heuristic, is of course a special instance of Rule 2 and therefore ensures finite convergence.

We have not bothered to write the cut equation (2) associated with the source equation above. The reason for this is that the transformation that gives $\bar{A}$ from A is immediately determined by reference to Theorem 3. Thus, if $a_{ij}<0$ (where i is the source equation) one adds the smallest (positive) integer multiple of A_s to A_j that yields $\bar{a}_{ij}\geqq 0$, while if $a_{ij}\geqq 0$, then one subtracts the largest (nonnegative) integer multiple of A_s from A_j that yields $\bar{a}_{ij}\geqq 0$. Finally, A_s is replaced by $\bar{A}_s=-A_s$. Thus, to obtain $\bar{A}$ from the A matrix above, we see that $2A_s$ is to be added to A_1 and $1A_s$ is

† To identify these equations, determine the largest integer multiple of A_s that can be subtracted from A_o and leave all resulting coefficients ($\bar{a}_{io}$) nonnegative for $i\geqq 1$. Each equation that prevents this multiple from being larger is eligible for the source equation.

to be added to A_2, with A_o unchanged. Thereupon, replacing A_s by its negative, we obtain the new A matrix below.

1.

		s		
	0	−10	−9	3
	5	1	2	0
	8	1	−2	4
*	1	1	0	−2
	16	7	10	−6
r	500	170	176	−51
	0	−1	0	0
	0	0	−1	0
	0	−2	−1	1

The foregoing procedure is reapplied to yield the successive A matrices following:

2.

				s
	10	10	−9	−17
	4	−1	2	2
*	7	−1	−2	6
	0	−1	0	0
**	9	−7	10	8
* r	330	−170	176	289
	1	1	0	−2
	0	0	−1	0
	2	2	−1	−3

3.

		s		
	27	−7	8	17
	2	1	0	−2
*	1	5	−8	−6
	0	−1	0	0
	1	1	2	−8
** r	41	119	−113	−289
	3	−1	2	2
	0	0	−1	0
	5	−1	2	3

The solution $X = A_o$ for A_o given in Matrix 3 is optimal. Nevertheless, it requires three additional steps to verify this fact.

4.

				s
	27	7	1	−4
	2	−1	1	1
*	1	−5	−3	9
	0	1	−1	−3
	1	−1	3	−5
** r	41	−119	6	68
	3	1	1	−1
	0	0	−1	0
	5	1	1	0

5.

		s		
	27	−1	1	4
*	2	1	1	−1
	1	13	−3	−9
	0	−5	−1	3
	1	−11	3	5
r	41	17	6	−68
	3	−1	1	1
	0	0	−1	0
	5	1	1	0

6.

	27	1	0	3
	2	-1	2	0
	1	-13	10	4
	0	5	-6	-2
	1	11	-8	-6
r	41	-17	23	-51
	3	1	0	0
	0	0	-1	0
	5	-1	2	1

Matrix 6 is both primal and dual feasible, thus verifying that the optimal solution has been obtained.†

REFERENCES

1. A. BEN-ISRAEL AND A. CHARNES, "On Some Problems of Diophantine Programming," *Cahiers du Centre L'Etudes de Recherche Operationnelle*, Brussels (1962).
2. FRED GLOVER, "A Bound Escalation Method for the Solution of Integer Linear Programs," *Cahiers du Centre L'Etudes de Recherche Operationnelle*, **6,** Brussels (1964).
3. ———"A Pseudo Primal-Dual Integer Programming Algorithm," Management Sciences Research Report No. 90, Carnegie-Mellon University (December 1966) (to appear in *J. Research* of the National Bureau of Standards).
4. RALPH E.GOMORY, "All-Integer Programming Algorithm," *Industrial Scheduling*, J. F. Muth and G. L. Thompson (eds.), Prentice Hall, New York, 1963.
5. ROMULO H. GONZALEZ-ZUBIETA, "Fundamental Investigations in Methods of Operations Research," Technical Report No. 16, Massachusetts Institute of Technology (June 1965).
6. R. D. YOUNG, "A Primal (All-Integer) Integer Programming Algorithm," *J. Research* of the National Bureau of Standards (July–September, 1965).
7. ———, "A Simplified Primal (All-Integer) Integer Programming Algorithm," *Opns. Res.* **16,** 750–782 (1968).

† If the reference equation in the original A matrix is instead given by $x_r=8+\sum(-t_j)$, then 15 pivot steps are required to obtain the optimal solution (using the same rule for selecting the source equation).