Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Technical Note—A Polynomial Simplex Method for the Assignment Problem

Ming S. Hung,

In *Proc. of the Fourth International Symposium on Operations Research*, Cambridge, Mass.

PIERSKALLA, W. P. 1967. Optimal Issuing Policies in Inventory Management—I. *Mgmt. Sci.* **13,** 395–412.

PIERSKALLA, W. P., AND ROACH, C. D. 1972. Optimal Issuing Policies for Perishable Inventory. *Mgmt. Sci.* **18,** 603–614.

ZEHNA, P. A. 1962. Inventory Depletion Policies. *Studies in Applied Probability.* Stanford University Press, Stanford, Calif.

A Polynomial Simplex Method for the Assignment Problem

MING S. HUNG
Kent State University, Kent, Ohio

(Received January 1981; revised July 1981; accepted October 1982)

We present a polynomial primal simplex algorithm for the assignment problem. For an $n \times n$ assignment problem with integer cost coefficients, the algorithm generates at most $n^3 \ln \Delta$ bases prior to reach the optimal basis, where Δ is the difference in objective value between an initial extreme point and the optimal extreme point.

WE PRESENT a primal simplex algorithm which solves an $(n \times n)$ assignment problem with integer cost coefficients in at most $n^3 \ln \Delta$ basis exchanges (pivots), where Δ is the difference in the objective value between a starting feasible solution and the optimal solution. By a primal simplex algorithm for the assignment problem, we mean a method that proceeds from one feasible basis to another, each obtained from the previous one by the addition of a nonbasic variable and the removal of a basic variable. Primal simplex algorithms differ from each other in the rules for selecting the entering variables and for selecting the leaving variables. Aside from the issue of cycling, certain rules for selecting entering variables have been shown to lead to exponentially long sequences of pivots for general linear programs (see Klee and Minty [1972] and Goldfarb and Sit [1979]), even for network flow problems (Zadeh [1973] and Cunningham [1979]).

The algorithm presented here represents one of the first evidences of the positive kind. By using two commonly known rules for selecting entering variables, the algorithm requires at most $(n - 1)^2$ degenerate

Subject classification: 484 polynomial simplex algorithm for the assignment problem.

Operations Research 0030-364X/83/3103-0595 $01.25
Vol. 31, No. 3, May–June 1983

pivots at an extreme point and visits at most $n \ln \Delta$ extreme points before finding the optimal solution.

1. THE ASSIGNMENT PROBLEM

Consider the $(n \times n)$ assignment problem

$$\text{Minimize } C(x) = \textstyle\sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}$$

$$\text{s.t. } \textstyle\sum_{j \in J} x_{ij} = 1,\ i \in I;\ \sum_{i \in I} x_{ij} = 1,\ j \in J;\ x_{ij} \geq 0,\ i \in I,\ j \in J.$$

where $I = \{1, \cdots, n\}$ is the set of origin nodes and $J = \{1, \cdots, n\}$ is the set of destination nodes. For the purposes of this paper, we assume that each c_{ij} is an integer and that the network is fully dense in the sense that there is an arc for each pair of (i, j).

An extreme point solution to the assignment problem consists of n variables, each having a value of one and corresponding to an arc from a distinct origin to a distinct destination. A basis consists of the n positive variables and $n - 1$ degenerate variables which together form a spanning tree over the assignment network. On the basis tree, we designate one node as the root. Using the notation in Cunningham [1976], we let $R(T, e)$ be the node set of the subtree containing the root if basic arc e were to be removed from the basis tree T. A basic arc $e = (i, j)$ is said to be "directed toward the root" if its destination node $j \in R(T, e)$ and is said to be "directed away from the root" otherwise. A basis is called an "alternating path basis" (A-P basis) by Barr et al. [1977] or a "strongly feasible basis" by Cunningham [1976] if the root is an origin node and every degenerate basic arc is directed toward the root. A distinct feature of an A-P basis is that there is one degenerate basic arc from every origin node, except the root.

Let u_i for $i \in I$ and v_j for $j \in J$ be the associated dual variables of the assignment problem. In this paper, we will always let origin 1 be the root and set $u_1 = 0$. Then the other dual variables are uniquely determined by the following equation: $u_i(T) + v_j(T) = c_{ij}$ if $(i, j) \in T$ for any basis T. Once the dual variables are determined, the "reduced cost" for arc $e = (i, j)$ is $\bar{c}_e = c_e - u_i - v_j$.

A primal simplex algorithm for the assignment problem was developed by Barr et al. and can be summarized as follows:

Step 1: Let T_0 be the current A-P basis tree. Select a nonbasic arc $e^* = (i^*, j^*)$ with $\bar{c}_{e^*}(T_0) < 0$ to enter the basis. If $\bar{c}_e(T_0) \geq 0$ for every arc e, the current basis is optimal.

Step 2: Add e^* to T_0 to form a unique circuit with arc set $\Phi(T_0, e^*)$. Then there is exactly one arc $\bar{e} = (i^*, j)$, $j \neq j^*$ such that $\bar{e} \in \Phi(T_0, e^*)$. The arc $\bar{e}$ will become nonbasic in the new basis T_1. If $x_{\bar{e}}(T_0) = 0$, then $x_{e^*}(T_1) = 0$. If $x_{\bar{e}}(T_0) = 1$, then change every positive arc in $\Phi(T_0, e^*)$ to a degenerate arc and every degenerate arc into a positive arc, and let $x_{e^*}(T_1) = 1$.

The A-P basis structure is automatically maintained by the unique determination of the leaving arc. It is also easy to anticipate whether the ensuing pivot is degenerate (i.e., $x_{e^*}(T_1) = 0$) or not. Relative to an A-P basis tree T_0, Barr et al. call a nonbasic arc $e = (i, j)$ an "upward" arc if node j is on the path from node i to the root; they refer to the arc as a "downward" arc if node i is on the path from node j to the root. Otherwise arc e is a "cross" arc. The ensuing pivot is nondegenerate if and only if e^* is a downward arc on T_0. If the pivot that forms T_1 from T_0 is degenerate, then $u_i(T_1) = u_i(T_0)$, $v_j(T_1) = v_j(T_0)$ for $(i, j) \in R(T_0, \bar{e})$ and $u_i(T_1) = u_i(T_0) + \bar{c}_{e^*}(T_0)$, $v_j(T_1) = v_j(T_0) - \bar{c}_{e^*}(T_0)$ for $(i, j) \notin R(T_0, \bar{e})$. Hence $u_i(T_1) \le u_i(T_0)$ and the strict inequality holds for at least one $i \in I$.

2. THE POLYNOMIAL SIMPLEX ALGORITHM

The simplex algorithm to be presented below specifies that in selecting the entering arcs, those arcs that are upward or cross are to be chosen first. In other words, one is to perform the degenerate pivots at an extreme point until a basis, called "degenerate-pivot-free," is found. An A-P basis is degenerate-pivot-free if every upward or cross nonbasic arc has a nonnegative reduced cost. Once a degenerate-pivot-free basis is found, one then chooses the nonbasic arc with the most negative reduced cost to enter the basis and moves to a new extreme point.

One of the entering-arc selection rules used in the algorithm is the "Modified Row Most Negative" (MRMN) rule which can be described as follows: Let $\bar{e} = (i_1, j)$ be the entering arc that gives rise to the current A-P basis T. Select $e^* = (i_1 + 1, j^*)$ as the next entering arc if $\bar{c}_{e^*}(T) < 0$, e^* an upward or cross arc on T, and $\bar{c}_{e^*}(T) = \text{Min } \bar{c}_e(T)$ over all arcs $e = (i_1 + 1, j)$, $j \in J$, that are upward or cross on T. If every cross or upward nonbasic arc in row $i_1 + 1$ has a nonnegative reduced cost, then go on to row $i_1 + 2, i_1 + 3, \cdots, n, 2, 3, \cdots, i_1$. Row 1 need not be scanned because origin 1 is the root and every arc on row 1 is a downward arc on any A-P tree.

A Polynomial Simplex Algorithm for the Assignment Problem

Step 1 (entering variable selection):

1a: Use the MRMN rule to determine whether degenerate pivots are possible on the current A-P basis. If so, let the entering arc be e^* and go to Step 2. Otherwise, go to Step 1b.

1b: Choose the entering arc e^* with $\bar{c}_{e^*} = \text{Min } \bar{c}_e$ over all arcs e. If $\bar{c}_{e^*} \ge 0$, terminate the algorithm; the optimal basis has been found. Otherwise go to Step 2.

Step 2 (basis exchange): Same as Step 2 in Section 1. Then return to Step 1a.

Since the choice of the leaving arc is the same as before, this procedure preserves the A-P basis structure. To prove that the algorithm is polynomial, we need to show that the number of bases encountered at an extreme point and the number of extreme points generated in the sequence of pivots are polynomial. The former is proved below.

THEOREM 1. *Let $T_0, T_1, \cdots, T_m$ be a sequence of A-P bases chosen by the simplex method at an extreme point x^0 of an $(n \times n)$ assignment problem, where T_m is a degenerate-pivot-free basis. If the modified row most negative rule is used, then $m \leq (n-1)^2$.*

Proof. This proof is modeled closely after the argument of Cunningham's Theorem 4 [1979].

We first prove the claim that if the path on T_m from origin i^* to the root has k degenerate arcs, and $k(n-1) \leq m$, then $u_{i^*}(T_{k(n-1)}) = u_{i^*}(T_m)$. The claim is clearly true for $k = 0$. Assume it is true for all $k < t$, $t \geq 1$. If it is not true for $k = t$, then there is a path from node i^* to the root such that the path has t degenerate arcs and $u_{i^*}(T_{t(n-1)}) > u_{i^*}(T_m)$. Let $e^* = (i^*, j^*) \in T_m$ be the degenerate arc on row i^*. Then by the induction assumption, $v_{j^*}(T_{(t-1)(n-1)}) = v_{j^*}(T_m)$ since $u_i(T_{(t-1)(n-1)}) = u_i(T_m)$ where $e = (i, j^*)$ is the positive arc on x^0. Therefore, for $(t-1)(n-1) \leq q \leq t(n-1)$, $\bar{c}_{e^*}(T_q) = c_{e^*} - u_{i^*}(T_q) - v_{j^*}(T_q) \leq c_{e^*} - u_{i^*}(T_{t(n-1)}) - v_{j^*}(T_m) < c_{e^*} - u_{i^*}(T_m) - v_{j^*}(T_m) = 0$. So e^* was not a basic variable in this sequence of n bases even though it was either a cross or an upward arc. It follows, then, from the MRMN rule that there was an arc $\hat{e} = (i^*, j)$ with a larger (or equal) negative reduced cost than e^* at the time the algorithm scanned row i^*. Then by selecting $\hat{e}$ to enter, the algorithm would have reduced u_{i^*} to a level smaller than (or equal to) $u_{i^*}(T_m)$, and would not have chosen e^* to enter the subsequent bases. This is a contradiction. Therefore no such node i^* could exist and the claim is proved.

Since every path on T_m has at most $n-1$ degenerate arcs, the validity of the theorem follows.

The following provides a bound on the objective value of the optimal extreme point solution.

THEOREM 2. *Let x^0 and x^* be, respectively, the current and the optimal extreme point solutions to the assignment problem. Let T be a basis (not necessarily A-P) for x^0. Then*

$$C(x^0) - C(x^*) \leq \delta = -\textstyle\sum_{i \in I} (\text{Min}_j \bar{c}_{ij}(T)).$$

Proof. It is well known that subtractions of scalars from the cost coefficients (c_{ij}) do not change the relative objective function values of feasible solutions to the assignment problem. That is, if $\hat{c}_{ij} = c_{ij} - a_i - b_j$, then $\hat{C}(x^1) - \hat{C}(x^2) = C(x^1) - C(x^2)$ for any feasible solutions x^1 and x^2. Therefore, consider the case where $\hat{c}_{ij} = c_{ij} - u_i(T) - v_j(T) - r_i$ and $r_i = \text{Min}_{j\in J}\bar{c}_{ij}(T)$. Then $\hat{c}_{ij} \geq 0$. Since $\hat{C}(x^0) = -\sum_{i\in I} r_i = \delta$ and $\hat{C}(x^*) \geq 0$, we have $C(x^0) - C(x^*) = \hat{C}(x^0) - \hat{C}(x^*) \leq \delta$.

Theorem 3. *Let $T_0(x^0), \cdots, T_k(x^*)$ be the sequence of A-P bases generated by the algorithm, where $T_k(x^*)$ is the optimal basis. Then, $k \leq n^3 \ln \Delta$ where $\Delta = C(x^0) - C(x^*)$.*

Proof. We first claim that $k \leq (n-1)^2(1 + \lceil \log \Delta/\log((n-1)/(n-2))\rceil)$ for $n \geq 3$, where $\lceil a \rceil$ denotes the least integer $\geq a$. (The assignment problem with $n \leq 2$ is trivial.) The term $(n-1)^2$ in this expression corresponds to the maximum number of bases that the algorithm encounters at an extreme point, which has been proved in Theorem 1. The other term accounts for the maximum number of extreme points that the sequence may include.

Let $T_{m_0}(x^0)$ be the degenerate-pivot-free basis at x^0 and $\delta_0 = -\sum_i (\text{Min}_j \bar{c}_{ij}(T_{m_0}))$. Since by Step 1b the entering variable e^* has the most negative reduced costs at T_{m_0} and the last origin node of any path on T_{m_0} has no downward nonbasic arcs, it follows that $\bar{c}_{e^*}(T_{m_0}) \leq -(1/(n-1))\delta_0$.

Let x^1 be the next extreme point. Then $C(x^1) = C(x^0) + \bar{c}_{e^*}(T_{m_0})$ and $C(x^1) - C(x^*) = C(x^0) - C(x^*) + \bar{c}_{e^*}(T_{m_0}) \leq \Delta - (1/(n-1))\delta_0 \leq ((n-2)/(n-1))\Delta$. The last inequality follows from Theorem 2. Similarly for the next extreme point x^2, $C(x^2) - C(x^*) \leq C(x^1) - C(x^*) - (1/(n-1))(C(x^1) - C(x^*)) \leq ((n-2)/(n-1))^2\Delta$. Hence $C(x^t) - C(x^*) \leq ((n-2)/(n-1))^t\Delta$ for $t \geq 0$. Since the cost coefficients are integers, $x^t = x^*$ if $C(x^t) - C(x^*) < 1$, which implies that t is the least integer satisfying $t > \log_{((n-1)/(n-2))}\Delta = \log \Delta/\log((n-1)/(n-2))$ for any base for the logarithm. The truth of the claim thus follows.

Finally, since $e^a > 1 + a$ for $a \neq 0$ and e the base of the natural logarithm, $\ln \Delta/\ln((n-1)/(n-2)) < (n-1)\ln \Delta$. Thus $k < (n-1)^2 (1 + (n-1)\ln \Delta)$. Further, for $\ln \Delta \geq ½$ (implying $\Delta \geq 2$), $(n-1)^2 [1 + (n-1)\ln \Delta] < n^3 \ln \Delta$.

It is clear that $\Delta \leq \sum_{i\in I} (\max_j c_{ij} - \min_j c_{ij})$. Thus the number of bases generated by the algorithm is polynomial in the problem size and the encoding of the problem data. Moreover, the number of arithmetic operations needed for constructing a new basis is at most $O(n^3)$. Checking whether a nonbasic arc $e = (i, j)$ is a downward arc or not can be done by tracing the path from destination node j to the root on the basis tree T. A path on T has at most $2n - 1$ arcs. So the number of comparisons needed is of order $O(n)$. One has to scan at most $n(n-1)$ arcs in Step 1a, hence it requires $O(n^3)$ arithmetic operations. Steps 1b and 2 require at worst $O(n^2)$ operations.

3. CONCLUSION

By combining rules for selecting the entering variables, the simplex algorithm presented here achieves polynomial convergence for solving the assignment problem. The algorithm's worse case bound is worse than some nonprimal algorithms (see Hung and Rom [1980]). Further, the algorithm is probably less efficient than the primal simplex algorithm of Barr et al. Nevertheless, the results here present a constructive first step in the development of theoretically efficient simplex algorithms.

REFERENCES

BARR, R., F. GLOVER AND D. KLINGMAN. 1977. The Alternating Path Basis Algorithms for Assignment Problems. *Math. Program.* **13,** 1-13.

CUNNINGHAM, W. 1976. A Network Simplex Method. *Math. Program.* **13,** 1-13.

CUNNINGHAM, W. 1979. Theoretical Properties of the Network Simplex Method. *Math. Opns. Res.* **4,** 196-208.

GOLDFARB, D., AND W. Y. SIT. 1979. Worst Case Behavior of the Steepest Edge Simplex Method. *Discrete Appl. Math.* **1,** 277-286.

HUNG, M. S., AND W. O. ROM. 1980. Solving the Assignment Problem by Relaxation. *Opns. Res.* **28,** 969-982.

KLEE, V., AND G. J. MINTY. 1972. How Good is the Simplex Algorithm? In *Inequalities III,* O. Shicha (ed.). Academic Press, New York.

ZADEH, N. 1973. A Bad Network Problem for the Simplex Method and Other Minimum Cost Flow Algorithms. *Math. Program.* **5,** 255-266.