ON FINDING AN INITIAL SOLUTION FOR THE DUAL SIMPLEX ALGORITHM

David I. Steinberg, *Southern Illinois University at Edwardsville*

ABSTRACT

Torrance [4] has proposed a "new" approach to finding an initial solution of a linear programming problem for use in conjunction with the dual simplex algorithm. The purpose of this note is to comment on two aspects of that paper. Firstly, Torrance's method is not new at all, but was proposed in 1958 by Wagner. Secondly, the method can be implemented much more efficiently than Torrance suggests.

Torrance [4] has proposed a "new" approach to finding an initial solution of a linear programming problem for use in conjunction with the dual simplex algorithm. The purpose of this note is to comment on two aspects of that paper. Firstly, Torrance's method is not new at all, but was proposed in 1958 by Wagner [5]. Secondly, the method can be implemented much more efficiently than Torrance suggests.

For clarity of discussion, the initialization procedure proposed by Wagner and Torrance [4] is briefly reviewed. Consider the linear programming problem:

$$\text{Maximize } x_0 = \sum_{j=1}^{n} c_j x_j \tag{1}$$

$$\text{subject to } \sum_{j=1}^{n} a_{ij} x_j \leqslant b_i,\ i = 1,2,\ldots,m \tag{2}$$

$$x_j \geqslant 0, j = 1,2,\ldots,n\ . \tag{3}$$

For each variable x_j whose $c_j > 0$, substitute $x_j = u_j - x_j'$, where u_j is a known upper bound on x_j. Then, an initial basic solution consisting of the m slack variables has the required property for beginning the dual simplex algorithm, since coefficients of all nonbasic variables in the resulting Equation (1) are nonpositive for this initial solution.

The problem can be restated as follows:

$$\text{Maximize } x_0 = \sum_{j\notin S} c_j x_j - \sum_{j\in S} c_j x_j' + \sum_{j\in S} c_j u_j \tag{4}$$

$$\text{subject to: } \sum_{j\notin S} a_{ij} x_j - \sum_{j\in S} a_{ij} x_j' \leqslant b_i - \sum_{j\in S} a_{ij} u_j,\ i = 1,2,\ldots m \tag{5}$$

$$0 \leqslant x_j' \leqslant u_j \quad j\in S \tag{6}$$

$$x_j \geqslant 0 \quad j\notin S, \tag{7}$$

where S is the set of subscripts for which $c_j > 0$.

Torrance [4, p. 423] claims that for each variable which is transformed (i.e., the number of elements in S), an additional constraint must be added (i.e., (6) above). Thus, the size of the basis to be manipulated is increased accordingly. He further states, "Whether or not this results in more computation than a primal algorithm depends upon the relative number of iterations required, which in turn depends upon the characteristics of the particular problem" [4, p. 423].

While these statements may be true as far as they go, it is also true that the number of iterations required for solving a linear programming problem seems to be primarily a function of the number of constraints.[1] Moreover, the amount of computation per iteration increases as the square of the number of constraints. These two facts strongly suggest that Torrance's approach would not be preferred over a primal simplex algorithm if the set S contained more then just a few members, a situation which would be very likely in a problem in which the objective function is one of maximizing profits (rather than minimizing costs).

However, as Wagner [5] has shown, it is not necessary to treat the upper bound constraints (6) explicitly. Thus, the size of the basis need not be increased. Instead, one can introduce an upper bound u_j for every variable and define for every variable x_j a *complementary variable*[2], x_j', by

$$x_j + x_j' = u_j. \tag{8}$$

Then, at any given iteration, only one of the pair x_j, x_j' needs to be in the problem explicitly for each j. The other variable for each pair is eliminated by means of (8). The initial solution is chosen as per Torrance. However, at subsequent iterations the set S is updated to reflect which of the variables x_j, x_j' is in the problem explicitly. Thus, S contains those subscripts for which the corresponding complementary variable x_j' is explicitly in the problem. With this convention, the linear programming problem under consideration consists of the objective function (4), the m constraints (5), and non-negativity restrictions on all variables.

At the end of each iteration, one examines each basic variable to see if it exceeds its upper bound. If so, it is replaced by its complementary variable in the basic solution. The complementary variable will then be negative. When updating the rest of the simplex tableau to account for this change, the basic variable in question will only appear in one of the constraints (5). In this constraint, each coefficient is multiplied by -1, and the new right-hand side becomes the upper bound minus the current right-hand side value.

It must be pointed out here that one of the primary purposes of Torrance's article was to provide a *simple* starting procedure for the dual simplex method, while

[1]Wagner, [6, p. 116] states: "Considerable empirical evidence suggests that most actual applications are solved within the range of 1.5 m to 3 m iterations, where m is the number of constraints and the starting basis is comprised only of slack, surplus, and artificial variables."

[2]The variable x_j is also called the complementary variable of x_j'.

retaining the simplicity of execution, and Wagner's method loses a bit of the simplicity of execution of Torrance's variation. However, there is another quite simple method for finding an initial solution for the dual simplex method which should be mentioned in this context. This approach is usually called the "artificial constraint method," and may be found in Simonnard [3] and Cooper and Steinberg [1]. It is a variation of the bounding row technique mentioned by Torrance and described in [2].

Consider again the original problem, defined by (*1*), (2), and (3). Again, let S be the set of subscripts whose $c_j > 0$. Let also

$$c_k = \max_{j \in S} \{c_j\} . \tag{9}$$

Now, add the following constraint to the problem:

$$\sum_{j \in S} x_j \leqslant M . \tag{10}$$

The constraint can be made redundant by choosing M large enough; e.g., let M exceed the sum of the upper bounds of the variables appearing in (10). Upon adding non-negative slack variables x_{n+i} to the $(m + 1)$ constraints (2) and (10), one has

$$\begin{aligned} x_{n+i} + \sum_{j=1}^{n} a_{ij}x_j &= b_i \qquad i = 1, 2, \ldots, m \\ x_{n+m+1} + \sum_{j \in S} x_j &= M . \end{aligned} \tag{11}$$

To find an initial basic solution of (11), select as the $(m + 1)$ basic variables $x_{n+1}, x_{n+2}, \ldots, x_{n+m}$, and x_k, where x_k is found from (9). If $S' = S - \{k\}$ and if x_k is eliminated from each of the first m equations of (11), one obtains

$$\begin{aligned} x_{n+i} + \sum_{j \notin S} a_{ij}x_j + \sum_{j \in S'} (a_{ij} - a_{ik})x_j - a_{ik}x_{n+m+1} &= b_i - a_{ik}M, \; i = 1, 2, \ldots, m \\ x_k + \sum_{j \in S'} x_j + x_{n+m+1} &= M . \end{aligned} \tag{12}$$

Similarly, eliminating x_k from the objective function (1) yields

$$x_0 = \sum_{j \notin S} c_jx_j + \sum_{j \in S'} (c_j - c_k)\, x_j - c_kx_{n+m+1} + c_kM. \tag{13}$$

Since, by virtue of (9) and the definition of S, the coefficients of all variables in the objective function (13) are nonpositive, the dual simplex method may now be implemented.

The "artificial constraint method" requires the addition of only one extra constraint and is conceptually quite simple, thus making it seem an attractive alternative to Torrance's method.

We note in closing that Torrance has attempted to provide a simple alternative solution procedure to the primal simplex method. Whether or not it or Wagner's version of the procedure is a computationally effective alternative is still open to question. Wagner did in fact propose his method as a serious competitor to the simplex method. The reason these various starting procedures for the dual simplex procedure are not usually described in general operations research texts, as Torrance observes, is simply that the dual simplex method is not widely used today as a general purpose solution procedure for linear programming problems. Rather, it is used in special situations (e.g., in post-optimal analysis, as a subroutine in many integer programming algorithms) in which an initial optimal (but not necessarily feasible) solution is already known.

REFERENCES

[1] Cooper, Leon and David I. Steinberg. *Methods and Applications of Linear Programming.* Philadelphia, Pennsylvania: W. B. Saunders, 1974.

[2] Orchard-Hays, William. *Advanced Linear-Programming Computing Techniques.* New York, N. Y.: McGraw-Hill, 1968.

[3] Simonnard, Michel. *Linear Programming* (Translated by William S. Jewell). Englewood Cliffs, N. J.: Prentice-Hall, 1966.

[4] Torrance, George W. "Initial Solution for the Dual Simplex Algorithm–A Tutorial Note." *Decision Sciences,* Vol. 5, No. 3 (July, 1974), pp. 422-424.

[5] Wagner, H. M. "The Dual Simplex Algorithm for Bounded Variables." *Naval Research Logistics Quarterly,* Vol. 5, No. 3 (1958), pp. 257-261.

[6] Wagner, Harvey M. *Principles of Operations Research.* Englewood Cliffs, N. J.: Prentice-Hall, 1969.