Management Science and Operations Research

Harvey J. Greenberg Editor

Experimental Investigation of an Interior Search Method Within a Simplex Framework

GAUTAM MITRA, MEHRDAD TAMIZ and JOSEPH YADEGAR

ABSTRACT: A feasible direction method for solving Linear Programming (LP) problems, followed by a procedure for purifying a non-basic solution to an improved extreme point solution have been embedded within an otherwise simplex based optimizer. The algorithm is designed to be hybrid in nature and exploits many aspects of sparse matrix and revised simplex technology. The interior search step terminates at a boundary point which is usually non-basic. This is followed by a series of minor pivotal steps which lead to a basic feasible solution with a superior objective function value. It is concluded that the procedures discussed in this article are likely to have three possible applications, which are

- *(i) improving a non-basic feasible solution to a superior extreme point solution,*
- *(ii) an improved starting point for the revised simplex method, and*
- *(iii) an efficient implementation of the multiple price strategy of the revised simplex method.*

1. INTRODUCTION

In recent years there has been growing interest in developing alternative (polynomially bounded) algorithms for the Linear Programming (LP) problem. The long standing open question, "whether there can be *any* polynomial-time algorithm for LP" was resolved when Khachian [14] developed the ellipsoid algorithm. However, this algorithm is unsatisfactory for practical problems (of even small size) and its average behavior is inferior to the modern simplex based LP-codes. The recent work, the polynomial-time projection algorithm, of Karmarkar [13] has sparked off enormous interest in the operations research community.

Our motivation in this research has been to develop a feasible direction method for LP which exploits many aspects of sparse matrix and revised simplex technology. The reason for working within a simplex framework is to exploit its descriptive properties. For instance, shadow price, post optimal information, uniqueness or otherwise of the optimal solution are easily computed.

The outcome of this work may lead to three possible applications:

- (i) A part of our method may be used as a 'purification' step to terminate an interior search procedure. In this context, we define the purification step as the algorithmic procedure by which we turn a non-basic feasible solution to a 'nearby' as well as improved extreme point (basic feasible) solution. A number of algorithms [13, 17] which use interior search method (see Section 2.1) are able to process large LP problems with special structure, in a computationally efficient manner. In these procedures the purification step may be used as the most apt termination step which can also provide all the simplex information. In the computational results reported by Nickels et al. [24], it is interesting to note that in about 25–30 percent of the total number of "Karmarkar iterations," one reaches around 80 percent of the optimum value of the objective function. We believe that introduction of a purification step would be most appropriate at this point.
- (ii) It is well-known that in many contexts an advanced starting basis, as obtained by the 'crashing' method [3], reduces the number of iterations in the simplex algorithm. Depending on the context of the problem our method may be applied initially to provide such an advanced basis.

(iii) Our experimental investigations show that when the search directions are restricted to small numbers (around 10 in most models) this approach performs efficiently. We later outline that in this situation the method is nearly equivalent to the well-established method of multiple pricing [1, 19, 21, pp. 50–55]. However, our method uses only one working area and amounts to an efficient implementation of the multiple price strategy.

2. BACKGROUND AND OVERVIEW

The simplex algorithm is still established as the most efficient and preferred method to solve general linear programming problems. Borgwardt [4, 5] proved that the expected number of iterations in the solution of an LP problem by a simplex based algorithm is polynomial, thereby explaining the efficiency obtained when simplex algorithms are used for practical problem solving. However, its worst case behavior is not polynomial [15].

Since the first publication of the simplex method by Dantzig [8], there have been many attempts to find better ways to solve LP problems. These experiments may be classified as either improvements of the simplex algorithm or non-simplex methods. Examples of some improvements are: LU factorization and sparse update procedures [10], price strategies—the Devex pricing method of Harris [12], strategies for (multiple) pivot columns selection during the price pass [21, pp. 50–55], procedures for obtaining advanced starting basis known as 'Crash' procedure [1, 19, 27]. For a discussion of 'Crash' procedure we refer the readers to [11, 25]. However, as the term 'improvements' suggests, the basic idea of the simplex algorithm to move from an extreme point (basic solution) to an adjacent extreme point of the polytope has been maintained.

To reduce computational effort, various methods of the non-simplex type have been proposed which avoid the 'crawling along the edges' of the polytope in the simplex algorithm. These methods are loosely referred to as Interior Search methods. Of these methods, we can mention the feasible direction method of Murty and Fathi [22], the block pivoting approach of Sherali et al. [28], and more recently an augmented Lagrangean method of Beale et al. [3]. All these methods, however, have so far not substituted the simplex algorithm.

More recently Karmarkar [13] proposed a polynomial-time algorithm for solving LP problems. This algorithm is shown to have the complexity of $O(n^{3.5}L^2)$, where n is the number of variables and L is the number of bits in the input. This is superior to Khachian's algorithm [14] which has the $O(n^6L^2)$ complexity. In addition to the complexity result Karmarkar's algorithm has also been shown to be an efficient computational method in some contexts. However, the wide ranging claims that it is superior to the simplex algorithm in all instances has not been established and continues to be disputed. We refer the reader to the November 1985 issue of *SIAM Newsletter* [30] which contains three articles on Karmarkar's algorithm and its reception.

As stated earlier a number of interior search methods are of relevance to our approach. In the following section we review them very briefly and classify them in two groups.

2.1 Approaches Within Simplex Framework

a) Zoutendijk (1960)
Various methods of feasible directions have been studied by Zoutendijk [32, 33]. He shows in detail how the simplex method can be considered as a method of feasible directions, and he also shows how the latter can be applied to solve linearly constrained non-linear programming problems.

b) Dantzig and Wolfe: The Decomposition Scheme (1961)
This scheme [9] is in effect an interior search method in relation to the 'Full Problem.' This is because basic feasible solutions to the 'Subproblems' are linearly combined to create a non-basic feasible trial solution to the full problem.

c) 'BASIC' Procedure of MPSX (1965)
The BASIC procedure of MPSX [19] is a system macro that provides a basic solution to a problem. The basic solution is obtained from the variables of a number of sub-problems which are amalgamated by this macro. This procedure assumes that basic solutions of each sub-problem have been supplied. Thus, it applies the simplex method to a restricted problem made up of the columns of the indicated basic variables. The procedure terminates when a basic (possibly feasible) solution is obtained.

d) Cooper and Kennington (1979)
In their paper [7], they describe a block pivoting approach for linear programs in which at most two non-basic variables are exchanged at any iterative step. They also give a feasible direction method which is essentially Wolfe's [31] reduced gradient procedure for convex non-linear programs over polyhedral feasible region. No computational result is given.

e) Sherali, Soyster and Baines (1983)
The paper [28] describes an advanced basis creation method (or block-pivoting) within the simplex approach. This involves exchanging several non-basic variables at each iterative step. They also implement a variation of the feasible directions method of [7] and, in addition, attempt to prevent near binding constraints from quickly restricting motion along an improving feasible direction. Computational results are presented for randomly generated problems with a maximum of 50 constraints and 100 variables. They conclude that creating an advanced starting point (basis) may computationally be an attractive approach for solving LP problems, whereas the feasible directions method is not.

f) Murty and Fathi (1984)
Each major iterative cycle of the method [22] starts with a Basic Feasible Solution (BFS), and then one moves in a profitable direction to a non-basic solution $\bar{x}$

while retaining feasibility. The direction to move is obtained by using the updated columns of the non-basic variables eligible to enter the basis (by the negative reduced cost criterion). The point $\bar{x}$ is not, in general, a basic solution. Subsequently, the algorithm goes through several reduction steps until a new BFS is obtained at which the objective value is better than or the same as that at $\bar{x}$. The major iterative cycle is repeated with the new computed BFS. Under non-degeneracy assumption, this method terminates after a finite number of major iterative cycles.

It is shown that the procedure (of moving from a non-basic feasible solution to a BFS with the same or better objective value) can be carried out using pivot steps and maintaining a basis inverse as in the usual simplex algorithm.

No computational results are given, but they state that the initial results on randomly generated problems are encouraging. This method is close to our approach. However, we have investigated an efficient revised simplex implementation, and a full description of our method is postponed until Section 3.

g) Beale, Hattersley and James (1985)
The main motivation of their approach [3] is to generate an advanced starting basis. This is accomplished in the following manner.

Given the standard LP problem

$$\text{P1: minimize} \quad \sum_{j=1}^{n} c_j x_j$$

subject to

$$\sum_{j=1}^{n} a_{ij} x_j = b_i \quad i = 1, \ldots, m,$$

$$l_j \leq x_j \leq u_j \quad j = 1, \ldots, n,$$

they transform it to a related (relaxed) quadratic problem

$$\text{P2: minimize} \quad \sum_{j=1}^{n} c_j x_j + M \sum_{i=1}^{m} r_i^2$$

subject to

$$l_j \leq x_j \leq u_j \quad j = 1, \ldots, n,$$

and

$$r_i = b_i - \sum_{j=1}^{n} a_{ij} x_j \quad i = 1, \ldots, m.$$

This formulation becomes equivalent to P1 as $M \rightarrow \infty$.

Having obtained an approximate solution to P2, which is a non-basic solution to P1, they apply the 'BASIC' algorithm to achieve an advanced starting basis for P1. They have reported encouraging results for representative LP problems.

2.2 Approaches Outside Simplex Framework

a) Mangasarian (1981/1983)
Given an LP problem, he considers [17] a Convex Quadratic Programming (QP) problem which is a perturbation of the original LP problem. He then applies the well-known iterative technique of successive over relaxation to the dual of the QP problem. This in turn leads to an optimum solution of the original LP. In [18], Mangasarian reports solution of randomly generated LP problems of substantial dimensions, ranging from 500 × 1000 to 5000 × 20000.

b) Karmarkar (1984)
The projection method of Karmarkar [13] first transforms the original LP to an equivalent canonical form. Subsequently at each iteration the current feasible solution is projected to the center of a simplex, and this is essentially a scaling operation. The algorithm then follows a direction of descent with a prescribed step size to ensure feasibility and to guarantee reasonable progress. A cleverly formulated potential function is employed to monitor the progress of the algorithm. This is essential to the proof of polynomial time complexity of the algorithm.
The projection method approaches the optimal solution from an interior feasible point and never visits any extreme point solution until an optimal solution is reached. In [6] Chiu and Ye describe how the Simplex and Karmarkar algorithms can come under a unified framework. This is achieved by varying the weights in a weighted gradient projection method.

c) Murty (1985)
The algorithm as described in [23] is a variant of the gradient projection method for LP and starts with an interior point of the set of feasible solutions. The algorithm terminates after a finite number of (major iterative) cycles, each of which consists of at most n steps (minor iterative) cycles where n is the number of variables in the LP. At each minor cycle (within a major iterative cycle) a tentative steepest descent direction is computed. Subsequently, one tests to establish if a move of sufficient length can be made in that direction without reaching a boundary point. If this is not possible, it then moves to the next step until the n steps are exhausted. If a direction for the move is not selected in a (major) cycle after n steps, this indicates that the current (feasible interior) point is close to an extreme point optimum solution of the LP. Murty mentions that a well-known subroutine (similar to our purification step, described in Section 3) can be used to move from an interior point to an extreme point.
Other authors such as Lemke [16] and Rosen [26] have also described gradient projection methods for solving LP problems. However, these algorithms were not restricted to interior feasible points, and no positive computational results were reported.

3. DESCRIPTION OF THE METHOD

Consider the LP problem:

$$\text{maximize} \quad x_0 = \sum_{j=1}^{n} c_j x_j \tag{3.1}$$

subject to

$$\sum_{j=1}^{n} a_{ij} x_j = b_i \quad i = 1, \ldots, m \tag{3.2}$$

$$x_j \geq 0.$$

For the convenience of exposition, the above statement is also set out in vector notation as:

$$\text{maximize} \quad x_0$$

$$\text{subject to} \quad \sum_{j=0}^{n} a_j x_j = b \tag{3.3}$$

where a_j and b are $m + 1$ vectors as defined below:

$$a_0 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad a_j = \begin{bmatrix} -c_j \\ a_{1j} \\ \vdots \\ a_{mj} \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} 0 \\ b_1 \\ \vdots \\ b_m \end{bmatrix}$$

$$j = 1, \ldots n.$$

An equivalent but transformed system of equations, as obtained after a number of pivotal operations may be expressed as:

$$\sum_{j=0}^{n} \bar{a}_j x_j = \bar{b} \tag{3.4}$$

Traditionally the first entry of the column $\bar{a}_j$ is denoted by d_j ($j = 1, \ldots, n$) which is the reduced cost coefficient for the column j.

Let S_B and S_N denote the sets defining the indices of the Basic and Non-basic variables respectively,

$$\begin{aligned} S_B &= \{0, i_1, \ldots, i_m\} \\ S_N &= \{i_{m+1}, \ldots, i_n\} \end{aligned} \tag{3.5}$$

whereby $S_B \cup S_N = \{0, 1, \ldots, n\}$.

3.1 Computing a Gradient Direction

In simplex algorithm by increasing the value of a single non-basic variable x_j whose corresponding d_j is negative the objective function value is increased. Since all the remaining non-basic variables are held at zero level, this is an edge following direction.

By considering a number of directions for which the d_j's are negative and taking their linear combination, we obtain a new profitable direction to move which may point to the interior of the polytope.

Consider a subset Q of S_N such that,

$$Q = \{j \mid j \in S_N \text{ and } d_j < 0\} \tag{3.6}$$

define α, a trial direction of search which is expressed as:

$$\alpha = \sum_{j \in Q} -d_j \bar{a}_j = \sum_{j \in Q} -d_j B^{-1} a_j = B^{-1} \sum_{j \in Q} -d_j a_j \tag{3.7}$$

where B^{-1} is the inverse of the basis matrix for the system of transformed equations set out in (3.4). At any step the set S_B (3.5) defines the basis matrix B.

From a computational point of view it should be noted that the direction α, defined in (3.7), is obtained by first creating a column using the linear combination $\sum_{j \in Q} -d_j a_j$ and then performing the standard FTRAN (forward transformation) operation [25].

3.2 Computing an Improved Non-Basic Feasible Solution

Define a scalar t as

$$t = \frac{\bar{b}_p}{\alpha_p} = \min_{i=1,\ldots,m} \left\{ \frac{\bar{b}_i}{\alpha_i} \;\middle|\; \frac{\bar{b}_i}{\alpha_i} \geq 0 \text{ and } \alpha_i > 0 \right\} \tag{3.8}$$

then t is the maximum value of a feasible step length in the newly computed gradient direction α. Naturally, the standard ratio test, that is the choose row operation of the simplex method is applied to compute t. In this step at least one basic variable drops to zero and the number of variables taking positive values (assuming non-degeneracy) are given by $|S_B \cup Q| - 1$. The corresponding non-basic solution may be represented using the current basis and the updated solution values of the basic variables and the non-basic variables held at level θ_j, where

$$\theta_j = -t d_j \qquad \text{for} \quad j \in Q. \tag{3.9}$$

thus

$$\sum_{j \in Q} \theta_j \bar{a}_j = t \sum_{j \in Q} -d_j \bar{a}_j = t\alpha \tag{3.10}$$

It is easily deduced from (3.7), (3.9) and (3.10) that

(i) the improvement in solution value is by the amount

$$t \sum_{j \in Q} d_j^2,$$

and

(ii) the basic variable values are updated by the relation

$$\bar{b}' = \bar{b} - t\alpha = \bar{b} - \left(\frac{\bar{b}_p}{\alpha_p}\right)\alpha$$

or in vector notation

$$p\text{th row} \rightarrow \begin{bmatrix} \bar{b}'_0 \\ \vdots \\ 0 \\ \vdots \\ \bar{b}'_m \end{bmatrix} = \begin{bmatrix} \bar{b}_0 \\ \vdots \\ \bar{b}_p \\ \vdots \\ \bar{b}_m \end{bmatrix} - \frac{\bar{b}_p}{\alpha_p} \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_p \\ \vdots \\ \alpha_m \end{bmatrix}$$

It is easy to see that the variable x_{i_p} in the pth row after update takes the value

$$x_{i_p} = \bar{b}'_p = \bar{b}_p - \frac{\bar{b}_p}{\alpha_p}\alpha_p = 0. \qquad (3.11)$$

3.3 Purification Procedure

Starting from a non-basic feasible solution, as obtained above, a basic feasible solution with a superior objective function value is obtained by following a sequence of pivotal steps which we call a purification procedure. In these steps starting with $|S_B \cup Q| - 1$ variables taking positive values, we obtain a basic solution by reducing one variable to zero level at each pivotal step and increasing the objective function value.

Non-basic Variables with Negative d_j

These variables are increased from their current level of θ_j to a higher level by the standard ratio test of the simplex method.

Let

$$\frac{\bar{b}_p}{\bar{a}_{pj}} = \min_i\left\{\frac{\bar{b}_i}{\bar{a}_{ij}} \;\middle|\; \bar{a}_{ij} > 0,\ \bar{b}_i \geq 0\right\} \qquad (3.12)$$

then a pivotal operation is carried out on $\bar{a}_{pj}$ whereby a new ETA-vector is created and the solution values are updated. The solution value for the variable x_j which pivots into the pth row is now updated by the simple upper bound algorithm. Thus x_j takes the value

$$x_j = \theta_j + \frac{\bar{b}_p}{\bar{a}_{pj}}$$

If no positive pivot is found then the problem is unbounded.

Non-basic Variables with Positive d_j

These variables are decreased from the current level of θ_j to a lower level which is

$$x_j = \max\left\{0,\ \theta_j + \frac{\bar{b}_p}{\bar{a}_{pj}}\right\} \qquad (3.13)$$

where

$$\frac{\bar{b}_p}{\bar{a}_{pj}} = \min_i\left\{\frac{\bar{b}_i}{-\bar{a}_{ij}} \;\middle|\; \bar{a}_{ij} < 0,\ \bar{b}_i \geq 0\right\}$$

If in expression (3.13) x_j is set to zero, then the solution values of the current basic variables are updated by the standard procedure of the simple upper bound algorithm, whereby

$$\bar{b}' = \bar{b} + \theta_j\bar{a}_j$$

Otherwise a pivotal operation is carried out on $\bar{a}_{pj}$ and a new ETA-vector is created.

3.4 Statement of the Algorithm

This algorithm can now be stated as a finite number of major iterative cycles. Each major cycle comprises two minor procedures which are:

minor procedure 1
computing a direction of gradient—Section 3.1—followed by computing an improved non-basic feasible solution—Section 3.2.
minor procedure 2
purification procedure—Section 3.3.

Since in each major cycle, we move from one basic feasible solution to a superior basic feasible solution, two basic feasible solutions separated by one major cycle are most unlikely to be adjacent. The finiteness of the algorithm follows naturally from the finiteness proof of the simplex method.

The computational effort required in each step of the minor procedure 2 (the purification procedure) is essentially equivalent to an iterative step of the simplex method. It follows from the description given in Section 3.3 that in $|Q|$ such steps this purification procedure is completed. At the beginning of each major cycle when the first non-basic solution is derived, a basic variable is driven to zero value—see (3.11). Hence at least one move of the minor cycle is always equivalent to a zero move.

It is well-known that a large number of practical LP problems are presented with simple upper bound specifications. Within simplex, these are dealt with implicitly [25]. Similarly the minor procedure 1 and minor procedure 2 require to be modified and extended to deal with these simple upper bound problems implicitly. The corresponding extended procedures are easy to derive and are explained in [29].

3.5 An Example

Consider the problem

maximize $x_1 + x_2 + x_3$

subject to

$$\begin{aligned}
3x_1 + 2x_2 - x_3 + x_4 &= 6\\
3x_1 + 2x_2 + 4x_3 + x_5 &= 16\\
3x_1 - 4x_3 + x_6 &= 3\\
\tfrac{9}{4}x_1 + 4x_2 + 3x_3 + x_7 &= 17\\
x_1 + 2x_2 + x_3 + x_8 &= 10
\end{aligned}$$

$x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8 \geq 0$

where x_4 to x_8 are slack variables.

The bounded polytope representing the constraints is illustrated in Figure 1.

Minor Procedure 1

Starting from the point $P^0 \equiv X^0 = (0, 0, 0)$, we compute the steepest direction α by the relation (3.7), whereby

$$\alpha = \begin{bmatrix} -3 \\ 4 \\ 9 \\ -1 \\ 37/4 \\ 4 \end{bmatrix} \text{ and initially } \bar{b} = \begin{bmatrix} 0 \\ 6 \\ 16 \\ 3 \\ 17 \\ 10 \end{bmatrix}$$

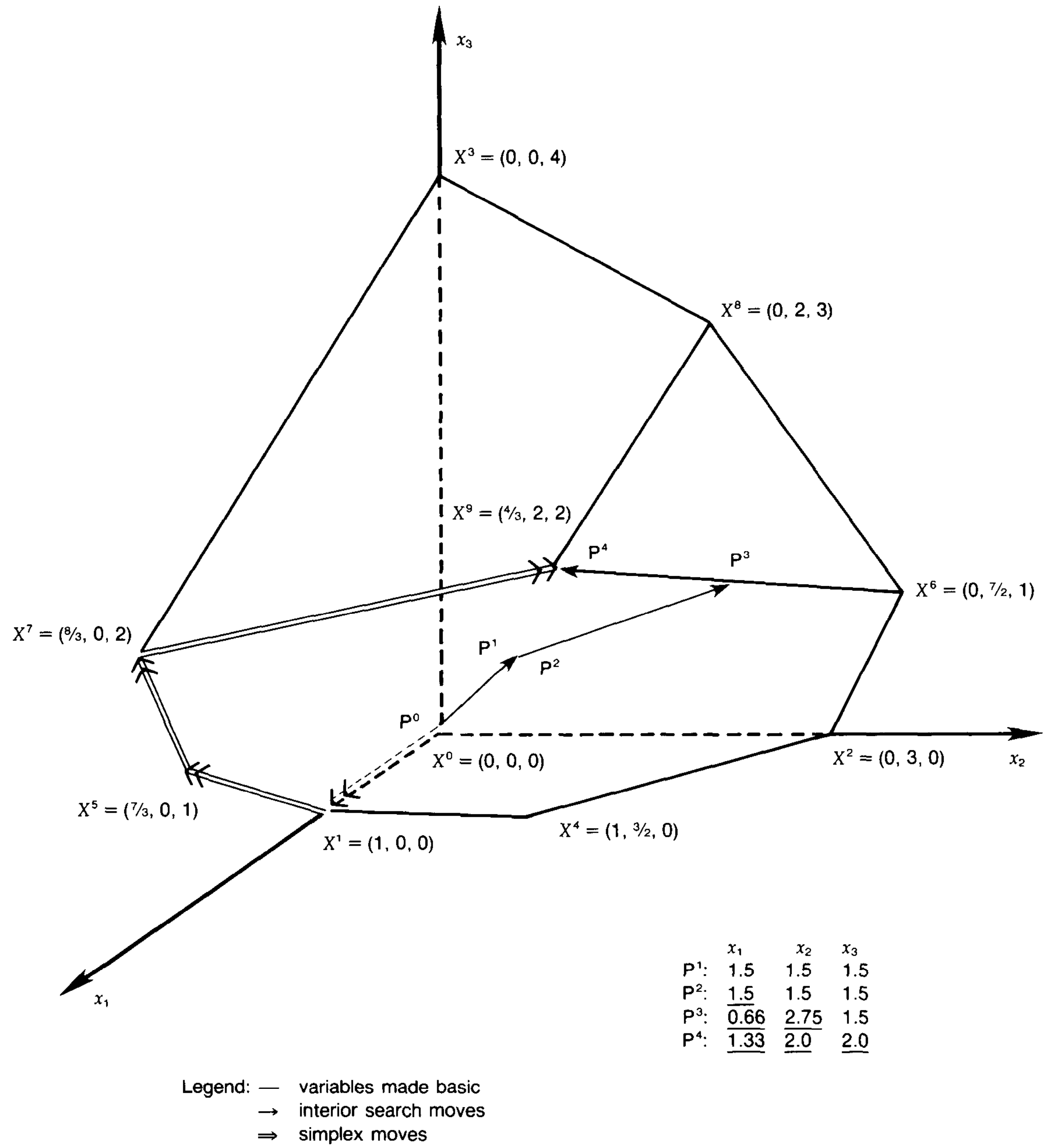

FIGURE 1. Representation of the Problem and the Algorithmic Steps

Then by the ratio rule of (3.8), we have $t = 6/4$, which gives the maximum feasible step length along direction α terminating on the face where the slack variable $x_4 = 0$. Thus the values of the updated basic variables are given as

$$\bar{b} = \begin{bmatrix} 0 \\ 6 \\ 16 \\ 3 \\ 17 \\ 10 \end{bmatrix} - 6/4 \begin{bmatrix} -3 \\ 4 \\ 9 \\ -1 \\ 37/4 \\ 4 \end{bmatrix} \quad \text{then} \quad \begin{aligned} x_0 &= 9/2 \\ x_4 &= 0 \\ x_5 &= 5/2 \\ x_6 &= 9/2 \\ x_7 &= 25/8 \\ x_8 &= 4 \end{aligned}$$

and the non-basic variables are held at $x_1 = x_2 = x_3 = 1.5$.

Minor Procedure 2

Choose x_1 as the non-basic variable and by the ratio rule (3.12) we choose the element α_1 (=4) in row 1 as pivot. This leads to a zero move whereby $P^1 \equiv P^2$ but x_4 becomes nonbasic at zero level. The variable x_1 becomes basic and after bound update takes the value 1.5. After two further minor iterations whereby x_2 and x_3 are made basic variables, the optimum solution is reached at P^4.

4. EXPERIMENTAL RESULTS

The summary information covering five test problems used in our investigation is set out in Table I. These problems are representative industrial test problems and are taken from the lower end of the collection of benchmark problems which have been compiled to validate our FORTLP system [20], [29].

For the purpose of comparison a number of alternative strategies were used to solve these test problems. These strategies are described in Section 4.1 and the results are discussed in Section 4.2.

TABLE I. Test Problems

No	Name	Source	No of rows	No of columns	Density in %	No of non-zeros	No of distinct non-zeros
1	ATLAS464	BP	315	458	2.1	2965	413
2	BLMODEL2	HAVERLY	255	550	1.5	2100	176
3	DOAE	SIA	339	1066	2.3	8142	538
4	AIRCRAFT	BA	162	202	1.5	505	301
5	MULTITIM	BRUNEL	28	48	9.7	130	11

4.1 Alternative Solution Strategies

The following four strategies were used in our investigation:

(i) Simplex
The primal simplex algorithm with full price of the A-matrix in each pass was used. A single price strategy was followed whereby the variable with most negative d_j was chosen.

(ii) Full Interior Search: All Dir
The interior search method as described earlier was used, whereby the direction of steepest descent was computed by first choosing the variables (directions) with negative d_j and then weighting them by the d_j values themselves.

(iii) Interior Search with 10 Best Directions: 10 Best Dir
In this strategy we considered up to a maximum of 10 variables chosen in the order of the most negative d_j. The search direction was computed as in (ii) above.

(iv) As in (iii) with Modifications: 5/10 Best Dir
In this strategy, initially 10 variables were chosen as in (iii) above. In each major cycle, however, 5 minor iterations were carried out. The five variables made basic in this way were chosen in the order of the solution values at which they were held. In all major cycles up to 5 variables were chosen. Also, if any of the residual variables were chosen again, i.e., variables not pivoted into the basis, then the corresponding solution values were updated.

4.2 Presentation and Discussion of Results

The experimental results are presented in Tables II and III. Comparing the solution times in Table II, taken by the alternative strategies to solve the five test problems, the strategy (iv) seems to perform uniformly well and generally superior to the simplex method. Performance of strategy (iii) is less uniform and is only comparable to the simplex method. Strategy (ii) involving the full interior search is computationally inefficient. In analyzing the results set out in Table II and Table III it is worth noting that a full iteration comprises FTRAN, BTRAN, PRICE and CHUZRO, whereas a minor iteration involves only FTRAN and CHUZRO. For a discussion of these main computational subroutines of the simplex method the reader is referred to [25].

In Table III, the times spent in each of the four major processing subroutines are set out. Comparing simplex and the strategy (iv), we see considerable improvement in times spent in BTRAN and PRICE. This is of course at the expense of mainly FTRAN and also CHUZRO.

4.3 Other Computational Considerations

Within the structure of the simplex, the minor procedure 2 has a number of computational relations and implications.

Flagging of Columns

During the application of the minor procedure 2, it is possible to identify and flag out columns. Consider a (transformed) column $\bar{a}_j$ such that the $d_j \geq 0$. The corresponding variable x_j is reduced from its current solution value using relations as in (3.13).

If it is established that $\bar{a}_{ij} \geq 0$ for all $i = 1, \ldots, m$, then the variable x_j can be flagged to zero. This is because the column represents a redundant relation in the dual problem which is always satisfied. Such a step is not worthwhile within the revised simplex method as it requires additional work involving columns with $d_j \geq 0$.

Equivalence with Multiple Price

When a restricted set of directions are chosen during minor procedure 1, the subsequent minor procedure 2 can be interpreted as the multiple price strategy within the simplex method. In the traditional multiple pricing method, if ten variables with negative d_j are chosen, then ten work areas are used. After a number of minor iterative steps a superior basic feasible solution is obtained out of the m original and the ten chosen variables. In our minor procedure 2, exactly the same result is achieved, but, it has the advantage of requiring only one work area.

TABLE II. Experimental Results

[illegible]	Alternative strategies	No of full iterations	No of minor iterations: With eta created	No of minor iterations: Set to lower bound	Total time (secs)
1	Simplex	2328	—	—	2045
	All Dir	90	2225	5605	3275
	10 Best Dir	411	1952	2141	214
	5/10 Best Dir	518	1609	1093	1440
2	Simplex	399	—	—	68
	All Dir	36	342	1602	182
	10 Best Dir	69	267	418	79
	5/10 Best Dir	77	248	144	55
3	Simplex	444	—	—	129
	All Dir	15	690	2463	236
	10 Best Dir	90	425	474	93
	5/10 Best Dir	146	458	380	118
4	Simplex	161	—	—	4.4
	All Dir	3	67	76	4.7
	10 Best Dir	7	60	6	3.5
	5/10 Best Dir	12	60	6	3.9
5	Simplex	44	—	—	0.39
	All Dir	13	51	82	0.60
	10 Best Dir	14	39	68	0.54
	5/10 Best Dir	19	44	54	0.73

All times are in seconds of Honeywell Multics DP68 cpu processing.

The 'BASIC' *Procedure*

The MPSX BASIC procedure as described in Section 2.1 is often used to obtain a 'basic feasible solution from a nonbasic feasible solution to a constraint set'. In this approach only the chosen variables are admitted in a restricted problem and subsequently the simplex method is applied to obtain a basic solution. This compares with the multiple price strategy described above. Again our purification procedure achieves exactly the same result through a series of pivotal operations. We understand that SCICONIC [2] also uses an implementation of BASIC similar to ours.

5. CONCLUSIONS

A number of methods for solving linear programming problems have been reviewed in this article. Some fall

TABLE III. Experimental Results

[illegible]	Subroutine	Time in Seconds for the Strategies: Simplex	All Dir	10 Best Dir	5/10 Best Dir
1	FTRAN	644	2252	1278	813
	BTRAN	570	33	123	133
	PRICE	101	0.4	26	32
	CHUZRO	45	165	95	67
2	FTRAN	8	132	47	27
	BTRAN	14	7	9	9
	PRICE	5	0.03	3	4
	CHUZRO	2	27	10	6
3	FTRAN	17	162	42	43
	BTRAN	19	1	4	8
	PRICE	79	2	22	42
	CHUZRO	7	60	18	18
4	FTRAN	0.9	1.8	1.2	1.3
	BTRAN	1.5	0.6	0.7	0.8
	PRICE	0.5	0.06	0.1	0.1
	CHUZRO	0.5	1.2	0.6	0.7
5	FTRAN	0.06	0.27	0.17	0.23
	BTRAN	0.11	0.05	0.04	0.07
	PRICE	0.14	0.02	0.08	0.13
	CHUZRO	0.07	0.27	0.24	0.22

All times are in seconds of Honeywell Multics DP68 cpu processing.

within the simplex structure and others are outside it. We make a special case for integrating interior search methods within the simplex structure. As a result such methods can be more applicable to real LP problems and implemented within long standing and established LP systems. We also observe that although a number of the methods reviewed have attractive theoretical results, the experimental results given by most except for [3], are not extensive.

The method as described in this article and our limited experimental results indicate that some of the procedures are worthwhile in their own right and fit naturally within the revised simplex structure. In view of the upsurge of interest in interior search methods, we believe it is necessary to provide a procedure which given an interior point, generates a nearby extreme point (optimal or nonoptimal) with a superior function value. Our purification procedure naturally fulfills this role.

Acknowledgments. The experimental investigations reported here were carried out by Mr. M. Tamiz as part of his Ph.D. research. We are grateful to the SERC who supported Dr. J. Yadegar as a research fellow under grant number GR/C/80134.

REFERENCES

1. APEX III. *Reference Manual, version 1.1.* Control Data Corporation, Minneapolis, Minn., 1977.
2. Beale, E.M.L. Private Communication, August 1985.
3. Beale, E.M.L., Hattersley, B., and James, L. An augmented Lagrangian method for linear programming. Represented at the 12th International Symposium on Mathematical Programming, MIT, Boston, Mass., 1985.
4. Borgwardt, K.H. Some distribution-independent results about the asymptotic order of the average number of pivot steps of the simplex-method. *Mathematics of Operations Research 7*, 3 (1982), 441–462.
5. Borgwardt, K.H. The average number of pivot steps required by the simplex method is polynomial. *Zeitschrift fur Operational Research, Series A: Theory 26*, 5 (1982), 157–177.
6. Chiu, S.S., and Ye, Y. *Simplex method and Karmarkar's Algorithm: A Unifying Structure.* Tech. Rep., Engineering Economic Systems Department, Stanford University, Stanford, Calif. 1985.
7. Cooper, L., and Kennington, J., Non-extreme point solution strategies for linear programs. *Naval Research Logistics Quarterly 26*, 3 (1979), 447–461.
8. Dantzig, G.B. Minimisation of a Linear Function of Variables Subject to Linear Inequalities, in *Activity Analysis of Production and Allocation*, Koopmans, T.C., Ed. Wiley, New York, 1951.
9. Dantzig, G.B. *Linear Programming and Extensions.* Princeton University Press, Princeton, N.J., 1963.
10. Forrest, J.J.H., and Tomlin, J.A. Updating triangular factors of the basis to maintain sparsity in the product form simplex method. *Mathematical Programming 2*, 3 (1972), 263–278.
11. Greenberg, H.J. Advanced basis construction in linear programming. Mathematics Department, Univ. Colorado, 1985. To appear in *Annals of Operations Research.*
12. Harris, P.M.J. Pivot selection methods of the Devex LP code. *Mathematical Programming 5*, 1 (1973), 1–28.
13. Karmarkar, N. A new polynomial-time algorithm for linear programming. *Combinatorica 4*, 4 (1984), 373–395.
14. Khachian, L.G. A Polynomial Algorithm in Linear Programming. *Doklady Akademii Nauk*, USSR, Nova Seria, 244, 5, 1093–1096, (1979), translated in *Soviet Mathematics Doklady, 20*, 191–194, (1979).
15. Klee, V., and Minty, G. How Good is the Simplex Algorithm? In *Inequalities III*, Shisha, O., Ed., Academic Press, New York, 1972, 159–179.
16. Lemke, C.E. The Constrained Gradient Method of Linear Programming, *J. Society for Industrial and Applied Mathematics 9*, 1 (1961), 1–17.
17. Mangasarian, O.L., Iterative solution of linear programs. *SIAM J. Numerical Analysis 18*, 4 (Aug. 1981), 606–614.
18. Mangasarian, O.L. Normal Solutions of Linear Programs. Computer Sciences Tech. Rep. #498, Univ. of Wisconsin-Madison, 1983.
19. "Mathematical Programming System Extended". Program number 5734 XM4, IBM Trade Corporation, New York, (1971).
20. Mitra, G., Tamiz, M., and Yadegar, J. *FORTLP: A Linear, Integer and Nonlinear Programming System, Preliminary User Manual.* Brunel University, England, July 1985.
21. Murtagh, B.A. *Advanced Linear Programming: Computation and Practice.* McGraw-Hill, New York, 1981.
22. Murty, K.G., and Fathi, Y. A feasible direction method for linear programming. *Operations Research Letters 3*, 3 (Aug. 1984), 121–127.
23. Murty, K.G. *A New Interior Variant of the Gradient Projection Method for Linear Programming.* Tech. Paper 85-18, Department of Industrial and Operations Engineering, Univ. Michigan, Ann Arbor, Mich., May 1985.
24. Nickels, W.W., Rodder, L.X.U., and Zimmermann, H.J. Intelligent gradient search in linear programming. *European Journal of Operational Research 22* (Dec. 1985), 293–303.
25. Orchard-Hays, W. *Advanced Computing Techniques in Linear Programming.* McGraw-Hill, New York, 1968.
26. Rosen, J.B. The gradient projection method for nonlinear programming: Part I, Linear Constraints. *J. Society for Industrial and Applied Mathematics 8* (Mar. 1960), 181–217.
27. *SCICONIC User Manual.* SCICON, Milton Keynes, U.K., (1978).
28. Sherali, H.D., Soyster, A.L., and Baines, S.G. Nonadjacent Extreme Point Methods for Solving Linear Programs. *Naval Research Logistics Quarterly 30* (1983), 145–161.
29. Tamiz, M. *Design, implementation and testing of a general linear programming system exploiting sparsity.* Ph.D. Thesis, Department of Mathematics and Statistics, Brunel University, London, 1986.
30. The Karmarkar algorithm: Is it for real? *SIAM-NEWS 18*, 6, November 1985.
31. Wolfe, P. Methods of Nonlinear Programming. In *Recent Advances in Mathematical Programming.* Graves, R.L., and Wolfe, P., Eds. McGraw-Hill, New York, 1963.
32. Zoutendijk, G. *Methods of Feasible Directions.* Elsevier Publishing Company, Amsterdam, 1960.
33. Zoutendijk, G. *Mathematical Programming Methods.* North Holland Publishing Company, Amsterdam, 1976.

CR Categories and Subject Descriptors: G.1.0 [**General**]: Numerical Algorithm; G.1.3 [**Numerical Linear Algebra**]: Linear Systems, Sparse and Very Large Systems; G.1.6 [**Optimization**]: Gradient Methods, Linear Programming; G.4 [**Mathematical Software**]: Efficiency

Additional Key Words and Phrases: Interior search method, multiple price, sparse simplex method

ABOUT THE AUTHORS:

GAUTAM MITRA heads the Mathematical Programming Group at Brunel University. The group is involved in the development of efficient algorithms for linear and integer programming problems and computer-based modeling tools for constructing and analyzing optimization and planning problems. His current research interests include many aspects of mathematical models for decision support.

MEHRDAD TAMIZ is a research fellow in the Department of Mathematics and Statistics at Brunel University and is currently working on linear and integer programming algorithms for large sparse problems. He has worked on many aspects of the FORTLP software system which is distributed by Numerical Algorithms Group Limited.

JOSEPH YADEGAR is working at the Distributed Array Processor Support Unit at Queen Mary College, London. His current research interests include design and implementation of parallel algorithms for optimization and pattern recognition problems. Authors' present addresses: Gautam Mitra and Mehrdad Tamiz, Brunel University, Uxbridge, Middlesex, England, UB8 3PH; Joesph Yadegar, DAPSU, Queen Mary College, University of London, Mile End Road, London, E1.