TR/06/86   June 1986

EXPERIMENTAL INVESTIGATION OF AN INTERIOR SEARCH METHOD WITHIN A SIMPLEX FRAMEWORK

by

Gautam Mitra, Mehrdad Tamiz

And Joseph Yadegar

Contents

Abstract

A steepest gradient 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 optimiser. 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 then 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 paper are likely to have three possible applications, which are

(i)   improving a non-basic feasible solution to a superior extreme point solution,

(iii) 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 [KHAC 79] developed the ellipsoid algorithm.   However, algorithm is unsatisfactory for practical problem (of even small size) and its average behaviour is inferior to the modern simplex based LP-codes. The   recent work the polynomial-time projection algorithm, of Karmarkar [KARM 84] 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 [MANG 81a], [KARM 84] 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. [NIRO 85], It is interesting to note that in about 25% - 30% of the total number of "Karmarkar iterations", one reaches around 80% 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, 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 (5-10 in most models) this approach performs efficiently. We outline later on that in this situation the method is nearly equivalent to the well established method of multiple pricing [MURT 81, pp50-55],   [MPSX 71], [APEX   77]. However, our method uses only one working area and amounts to an efficient implementation of the multiple price strategy.

The rest of the paper is organised as   follows:

A brief overview of the previous work done in this area is given in Section 2. In Section 3, we describe our method in full and give an illustrative example. Results of our experimental investigation are reported in Section 4, and we conclude the paper Section 5.

2. Background and Overview

The simplex algorithm is still established as the most efficient and favoured method to solve general linear programming problems. Borgwardt [BORG 82a], [BORG 82b] 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 behaviour is not polynomial [KLMI 72]

Since the first publication of the simplex method by Dantzig [DANT 51], there has 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. Improvements are for instance: LU factorization and sparse update procedures [FOTO 72], price strategies – the Devex pricing method of Harris [HARR 73], strategies for (multiple) pivot columns selection during the price pass [MURT 81, pp. 50-55], procedures for obtaining advanced starting basis known as 'Crash' procedure [MPSX 71], [APEX 77], [SCIC 78]. 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.

More recently Karmarkar [KARM 84] 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 [KHAC 79] which has the $0\{n^6L^2\}$ 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 85 issue of SIAM Newsletter [SIAM 85] 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 next 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 [ZOUT 60], [ZOUT 76]. 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 [DANT 63] 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 [MPSX 71] 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 the paper [COKE 79], 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 [WOLF 63] 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 [SHSO 83] 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 by Cooper and Kennington [COKE 79] 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 feasible directions method is not.

f)   Murty and Fathi (1984)

Each major iterative cycle of the method [MUFA 84] starts with a Basic Feasible Solution (BFS), and then one moves in a profitable direction to a non-basic solution 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 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 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 BES 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 [BEHA 85] is to generate an advanced starting basis. This is accomplished in the following manner.

Given the standard LP problem

P 1:   minimise $\sum_{j=1}^{n} c_j x_j$

    subject to $\sum_{j=1}^{n} a_{ij} x_j = b_i$ $i = 1,...., m$,

    $l_j \leq x_j \leq u_j$ $j = 1,...., n,$

they transform it to a related (relaxed) quadratic problem

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

    subject to $l_j \leq x_j \leq u_j$ $j = 1,...., n$,

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

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

Having obtained an approximate solution to P2, which is a non-basic solutiontoP1,they apply the 'BASIC' algorithmto 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 [MANG 81] 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 [MANG 83], Mangasrian reports solution of randomly generated LP problems of substantial dimensions, ranging from 500x1000 to 5000x20000.

(b) Karmarkar (1984)

The projection method of Karmarkar [KARM 84] first transforms the original LP to an equivalent canonical form. Subsequently at each iteration the current feasible solution is projected to the centre 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 [CHYE 85] Chiy 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 [MURT 85] 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 [LEMK 61] and Rosen [ROSEN 60] 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:

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

$$\text{subject to}\quad \sum_{j=1}^{n} a_{ij} x_j = b_i \qquad i = 1,....,m \tag{3.2}$$

$$x_j > 0..$$

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

maximise $\qquad x_0$

subject to $\qquad \displaystyle\sum_{j=0}^{n} a_j x_j = b$ $\qquad\qquad\qquad\qquad\qquad$ (3.3)

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

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

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

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

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

Traditionally the first entry of $\bar{a}_j$ is denoted by d_j (j=1,...,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,

$$\begin{aligned} S_B &= \left\{ 0, i_1, ..., i_m \right\} \\ S_N &= \left\{ i_{m+1}, ..., i_n \right\} \end{aligned} \qquad\qquad (3.5)$$

whereby $\qquad S_B \cup S_N = \left\{ 0, 1, ..., n \right\}.$

3.1   Computing a Direction of Steepest Gradient

In simplex algorithm by increasing the value of a single non-basic variable x_j whose corresponding dj 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,
$$Q = \left\{ j | \ j \in S_N \text{ and } d_j < 0 \right\} \qquad\qquad (3.6)$$

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

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

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 $\propto$, defined in (3.7), is obtained by first creating a column using the linear combination $\sum -d_j a_j$ and then performing the standard FTRAN (forward transformation) operation.

3.2 Computing an Improved Non-Basic Feasible Solution

Define a scalar t as

$$t = \frac{\bar{b}_p}{\propto_p} = \min_{i=1,...,m} \left[\frac{\bar{b}_i}{\propto_i} \quad \frac{\bar{b}_i}{\propto_i} > 0 \text{ and } \propto_i > 0 \right] \qquad (3.8)$$

then t is the maximum value of a feasible step length in the new (steepest gradient) direction $\propto$. 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 $|S_B \cup Q|$-l. The corresponding non-basic solution may be represented using the current basis and the updated solution values of the basic variables held at level e_i, where

$$e_j = -t d_j \text{ for } j \in Q. \qquad (3.9)$$

thus

$$\sum_{j \in Q} e_j \bar{a}_j = t \sum_{j \in Q} -d_j \bar{a}_j = t \propto \qquad (3.10)$$

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

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

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

(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

$$
p^{th}\ \text{row} \rightarrow
\begin{bmatrix} \bar{b}'_{o} \\ \cdot \\ \cdot \\ 0 \\ \cdot \\ \cdot \\ \bar{b}'_{m} \end{bmatrix}
=
\begin{bmatrix} \bar{b}_{o} \\ \cdot \\ \cdot \\ \bar{b}_p \\ \cdot \\ \cdot \\ \bar{b}_{m} \end{bmatrix}
- \frac{\bar{b}_P}{\alpha_P}
\begin{bmatrix} \alpha_{o} \\ \cdot \\ \cdot \\ \alpha_P \\ \cdot \\ \cdot \\ \alpha_{m} \end{bmatrix}
$$

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

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

3.3 Purification Procedure

Starting from a nonbasic feasible solution, as obtained above, a basic feasible solution with a superior objective function value is obtained by following a sequence of pivotal steps. This we call a purification procedure. In these steps starting with $|\ SB\ \ U\ \ Q|\ \text{-}1$ 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 e_j to a higher level by the standard ratio test of simplex.

Let

$$\frac{\bar{b}_p}{\bar{a}_{pj}} = \min_{i}\left[\frac{\bar{b}_i}{\bar{a}_{ij}}\ \Big|\ \bar{a}_{ij} > 0,\ \bar{b}_i > 0\right] \tag{3.12}$$

now 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_i which pivots into the pth row is now updated by the simple upper bound algorithm. Thus x_j takes the value

$$x_j = e_j + \frac{\bar{b}_p}{a_{pj}}$$

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

Non-basic Variables with Positive d_i

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

$$x_j = \max\left[0, e_j + \frac{\overline{b}_p}{\overline{a}_{pj}}\right] \qquad (3.13)$$

Where

$$\frac{\overline{b}_p}{a_{pj}} = \min_i\left\{\frac{\overline{b}_i}{-\overline{a}_{ij}} \mid \overline{a}_{ij} < 0, \ \overline{b}_i > 0\right\}$$

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

$$\overline{b}' = \overline{b} + e_j\overline{a}_j$$

Otherwise a pivotal operation is carried out on $\overline{a}_{pj}$ 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 steepest gradient - Section 3,1 - followed by computing an improved non-basic feasible solution - Section 3.2,

minor procedure 2   purificat ion 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 in effect 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 nonbasic 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.

3.5 An Example

Consider the problem

maximise $\quad x_1 + x_2 + x_3$

subject to

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

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

where x_4 to x_e are slack variables.

The bounded polytope representing the constraints is illustrated in Figure 3.1.

Minor Procedure 1

Starting from the point P° X° $(0,0,0)$, we compute the steepest direction $\propto$ by the relation (3.7), whereby

and initiallly $\quad b =$

$$
\propto \;=\; \begin{bmatrix} -3 \\ 4 \\ 9 \\ -1 \\ 37\,/\,4 \\ 4 \end{bmatrix}
\qquad\qquad
\begin{bmatrix} 0 \\ 6 \\ 16 \\ 3 \\ 17 \\ 10 \end{bmatrix}
$$

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

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

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

Figure 3.1

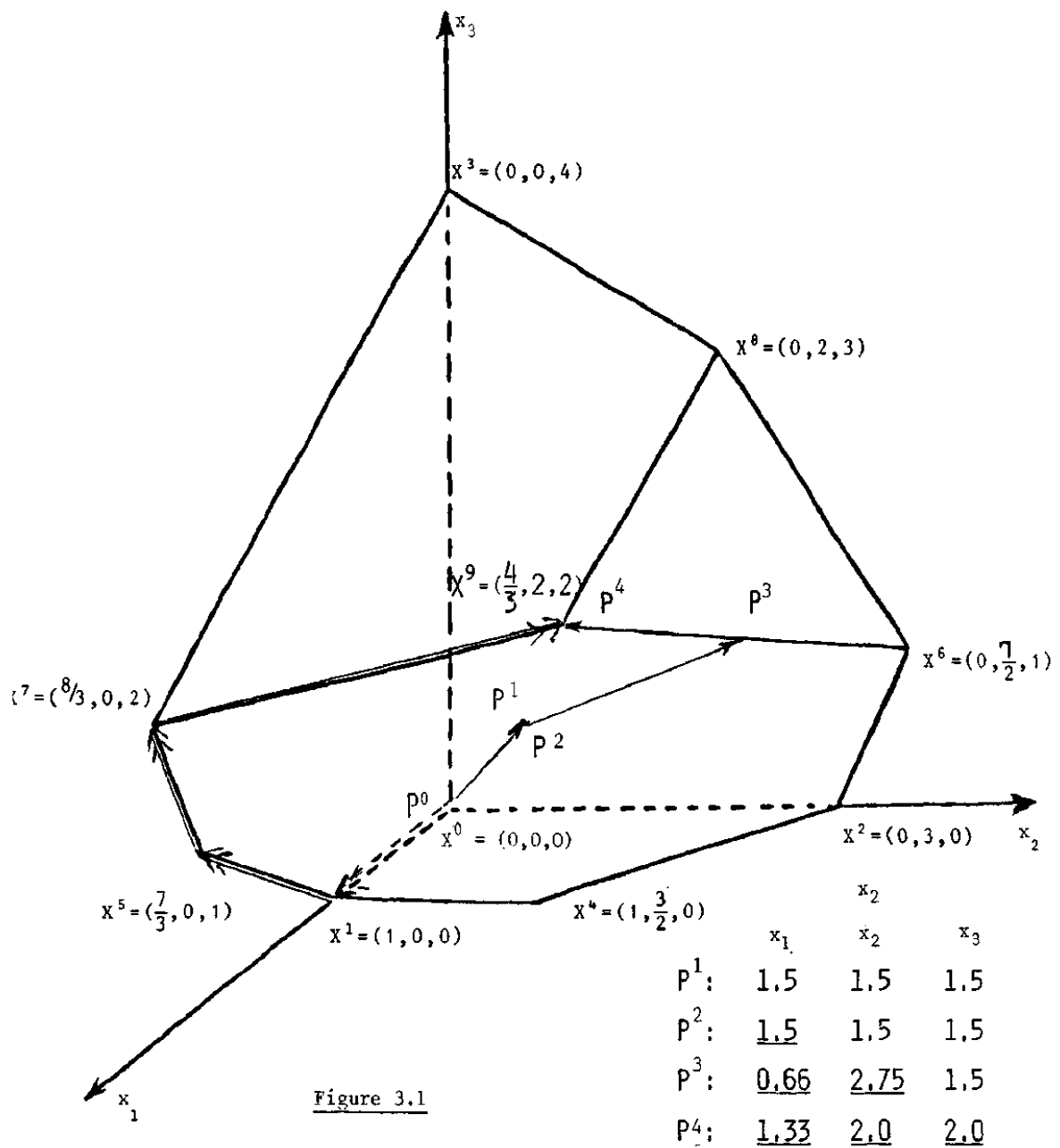|	x_1	x_2	x_3
P^1:	1.5	1.5	1.5
P^2:	1.5	1.5	1.5
P^3:	0.66	2.75	1.5
P^4:	1.33	2.0	2.0

Legend: — variables made basic
→ interior search moves
⇒ simplex moves

Page 11

Minor   Procedure   2

Choose x_1 as   the   non-basic   variable   and   by   the   ratio   rule   (3.12)   we   choose   element 4 in   row 1   as pivot.   This   leads   to   a   zero   move   whereby P^1 m 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 are   set   out   in   Table   4.1. 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   [MITY   85],   [TAMI   86].

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.

4.1 Alternative   Solution Strategies

Altogether   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   dj   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   dj   and   then weighting   them   by   the   dj   values   themselves.

(iii) Interior Search with 10 Best Directions: 10 Best Dir

    In   this   strategy   we   considered   upto   a   maximum   of   10   variables   chosen   in   the   order   of   the   most   negative   dj.   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   upto   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.

TABLE 4.1 TEST PROBLEMS

PROBLEM	NAME	SOURCE	NO OF ROWS	NO OF COLUMNS	NO OF NON-ZEROS	NO OF DISTINCT NON-ZEROS
1	ATLAS464	BP	315	458	2965	413
2	BLMODEL2	HAVERL	255	550	2100	176
3	DOAE	SIA	339	1066	8142	538
4	AIRCRAFT	BA	162	202	505	301
5	MULTITIM	BRUNEL	28	48	130	11

4.2 Presentation and Discussion of Results

The experimental results are presented in Table 4,2 and 4,3. Comparing the solution times in Table 4.2, taken by the alternative strategies to solve the five test problems, the strategy (iv) seems to perform uniformly well and always 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 analysing the results set out in table 4.2 and table 4.3 it is worth noting that a full iteration comprises FTRAN, BTRAN, PRICE and CHUZRO, whereas a minor iteration involves only FTRAN and CHUZRO.

TABLE 4.2 EXPERIMENTAL RESULTS

PROBLEM	ALTERNATIVE STRATEGIES	NO OF FULL ITERATIONS	NO OF MINOR ITERATIONS With eta	set to lower bound	TOTAL TIME Sees
1	Simplex	2328	—	—	2045
	All Dir	90	2225	5605	3275
	10 Best Dir	411	1952	2141	2148
	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.

In Table 4.3, 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 $d_j > 0$. The corresponding variable x_j is reduced from its current solution value using relations as in (3.13).

If it is established that $\bar{a}_{ij} > 0$ for all $i = 1,\dots,m$, 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 revised simplex method as it requires additional work involving columns with $d_j > 0$.

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 dj 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.

The 'BASIC' Procedure

The MPSX BASIC procedure as described in Section 2.1 is often used to obtain a 'basic feasible solution starting 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 [BEAL 85] also uses an implementation of BASIC similar to ours

TABLE 4.3:   Experimental Results

Time in Seconds for the Strategies

PROBLEM	SUBROUTINE	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 PRICE	1.5	0.6	0.7	0.8
	CHUZRO	0.5	0.06	0.1	0.1
		0.5	1.2	0.6	0.7
5	FTRAN	0.06	0.27	0.17	0.23
	BTRAN PRICE	0.11	0.05	0.04	0.07
	CHUZRO	0.14	0.02	0.08	0.13
		0.07	0.27	0.24	0.22

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

5.   Conclusions

The method as described in this paper and our limited experimental results indicate that some of the procedures are worth while 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 near by extreme point (optimal or nonoptimal) with a superior function value.Our purification procedure naturally fulfils this role.

6.   Acknowledgments

7.   References

[APEX 77] APEX III, "Reference Manual, version 1.1", Control Data Corporation, Minneapolis, USA (1977).

[BEAL 85] Beale, E.M.L., Private Communication, August 1985.

[BEHA 85] 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, Massachusetts, USA (1985).

[BORG 82a] 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, vol. 7, No. 3, pp.441-462 (1982).

[BORG 82b] Borgwardt, K.H., "The Average Number of Pivot Steps Required by the Simplex Method is Polynomial", Zeitschrift fur Operational Research, Serie A: Theory, vol. 26, No. 5, pp. 157-177 (1982).

[CHYE 85] Chiu, S.S., and Ye, Y., "Simplex Method and Karmarkar's Algorithm: A Unifying Structure", Technical Report, Engineering Economic Systems Department, Stanford University, Stanford, CA 94305 (1985).

[COKE 79] Cooper, L., and Kennington J., "Non-extreme Point Solution Strategies for Linear Programs", Naval Research Logistics Quarterly, vol. 26, No. 3, pp. 447-461 (1979).

[DANT 51] Dantzing, G.B., "Minimisation of a Linear Function of Variables Subject to Linear Inequalities", in Activity Analysis of Production and Allocation, edited by Koopnians, T.C., Wiley, New York, Chapter XXI (1951).

[DANT 63] Dantzig, G.B., "Linear Programming and Extensions", Princeton University Press, New Jersey (1963).

[FOTO 72] 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, vol. 2, pp. 263-278 (1972).

[HARR 73] Harris, P.M.J., "Pivot Selection Methods of the Devex LP Code" Mathematical Programming, vol. 5, pp. 1-28 (1973).

[KARM 84] Karmarker, N., "A New Polynomial-Time Algorithm for Linear Programming" Combinatorica, vol. 4, No. 4, pp. 373-395 (1984).

[KHAC 79] Khachian, L.G., "A Polynomial Algorithm in Linear Programming", Doklady Akademii Nauk, USSR, Nova Seria, Vol. 244, No. 5, pp 1093-1096 (1979), translated in Soviet Mathematics Doklady, vol. 20, pp. 191-194 (1979).

[KLMI 72] Klee, V., and G. Minty, "How Good is the Simplex Algorithm?" in Inequalities III, edited by Shisha, O., Academic Press, New York, pp.159-179 (1972).

[LEMK 61] Lemke, C.E., "The Constrained Gradient Method of Linear Programming", Journal of the Society for Industrial and Applied Mathematics, vol. 9, pp 1-17 (1961).

[MANG 81] Mangasarian, O.L., "Iterative Solution of Linear Programs", SIAM Journal of Numerical Analysis, vol. 18, No. 4, pp 606-614 (1981).

[MANG 83] Mangasarian, O.L., "Normal Solutions of Linear Programs", Computer Sciences Technical REport 498, University of Wisconsin-Madison (1983).

[MITY 85] Mitra, G., Tamiz, M., and Yadegar, J., "FORTLP": A Linear, Integer and Nonlinear Programming System, Preliminary User Manual, Brunei University, July 1985.

[MPSX 71] MPSX, "Mathematical Programming System Extended", Program number 5734 XM4, IBM Trade Corporation, New York, USA (1971).

[MUAT 87] Murtagh, B.A., "Advanced Linear Programming: Computation and Practice", McGraww-Hill, New York (1981).

[MUFA 84] Murty, K.G. and Fathi, Y., "A Feasible Direction Method for Linear Programming", Operations Research Letters, vol. 3, No. 3, pp. 121-127 (1984).

[MURT 85] Murty, K.G., "A New Interior Variant of the Gradient Projection Method for Linear Programming", Technical Paper 85-18, Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI 48109-2117, USA.

[NIRO 85] Nickels, W.W., Rodder, L.X.U., and Zimmermann, H.J., "Intelligent Gradient Search in Linear Programming", European Journal of Operational Research, vol. 22, pp 293-303 (1985).

[ROSE 60] Rosen, J.B. "The Gradient Projection Method for Nonlinear Programming: Part I, Linear Constraints", Journal of the Society for Industrial and Applied Mathematics, vol. 8, pp. 181-217 (1960).

[SCIC 78] SCICON, "SCICONIC User Manual" 1978.

[SHSO 83] Sherali, H.D., Soyster, A.L., and Baines, S.G., "Nonadjacent Extreme Point Methods for Solving Linear Programs", Naval Research Logistics Quarterly, vol. 30, pp, 145-161 (1983).

[SIAM 85] SIAM Newsletter, "The Karmarkar Algorithm: Is It For Real?", SIAM-NEWS, vol. 18, No. 6, November 1985.

[TAMI 86] Tamiz, M., "Design, implementation and testing of a Large Scale Linear Programming System", Ph.D. Thesis, Department of Mathematics and Statistics, Brunel University, (under preparation), 1986.

[WOLF 63] Wolfe, P., "Methods of Nonlinear Programming", in Recent Advances in Mathematical Programming, edited by Graves R.L., and Wolfe P., Mc-Graw-Hill, New York (1963).

[ZOUT 60] Zoutendijk, G., "Methods of Feasible Directions", Elsevier Publishing Company, Amsterdam (1960).

[ZOUT 76] Zoutendijk, G., "Mathematical Programming Methods", North Holland Publishing Company, Amsterdam (1976).