EXPERIMENTAL INVESTIGATION
OF AN INTERIOR SEARCH METHOD
WITHIN A SIMPLEX FRAMEWORK

by

Gautam Mitra, Mehrdad Tamiz
And Joseph Yadegar
Contents

1. Introduction

2. Background and Overview
   2.1 Approaches Within Simplex Framework
   2.2 Approaches Outside Simplex Framework

3. Description of the Method
   3.1 Computing a Direction of Steepest Gradient
   3.2 Computing an Improved Non-Basic Feasible Solution
   3.3 Purification Procedure
   3.4 Statement of the Algorithm
   3.5 An Example

4. Experimental Results
   4.1 Alternative Solution Strategies
   4.2 Presentation and Discussion of Results
   4.3 Other Computational Considerations

5. Conclusions

6. Acknowledgments

7. References
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 in 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 $O(n^6L^2)$ 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 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.


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
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

\[
P_1: \text{minimise} \sum_{j=1}^{n} c_j x_j \]

subject to \[\sum_{j=1}^{n} a_{ij} x_j = b_i \quad i = 1, \ldots, m,
\]

\[1 \leq x_j \leq u_j \quad j = 1, \ldots, n,
\]

they transform it to a related (relaxed) quadratic problem

\[
P_2: \text{minimise} \sum_{j=1}^{n} c_j x_j + M \sum_{i=1}^{m} r_i^2 \]

subject to \[1 \leq x_j \leq u_j \quad j = 1, \ldots, n,
\]

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 \(M \to \infty\).

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 [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], Mangasarian reports solution of randomly generated LP problems of substantial dimensions, ranging from 500x1000 to 5000x20000.
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:

\[
\text{maximise } x_0 = \sum_{j=1}^{n} c_j x_j \tag{3.1}
\]

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

For the convenience of exposition, the above statement is also set out in vector notation as:
maximise \( x_0 \)
subject to \( \sum_{j=0}^{n} a_j x_j = b \) \( (3.3) \)

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:

\[
\sum_{j=0}^{n} \tilde{a}_j x_j = \tilde{b} \quad (3.4)
\]

Traditionally the first entry of \( \tilde{a}_j \) 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,

\[
S_B = \{0, i_1, \ldots, i_m\} \\
S_N = \{i_{m+1}, \ldots, i_n\} \\
\text{whereby} \\
S_B \cup S_N = \{0, 1, \ldots, n\}.
\]

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 \( d_j \) 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,

\[
Q = \{j | j \in S_N \text{ and } d_j < 0\} \quad (3.6)
\]
define $\alpha$, a trial direction of search which is expressed as:

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

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 $\alpha$, 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

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

then $t$ is the maximum value of a feasible step length in the new (steepest gradient) direction $\alpha$. 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 held at level $e_j$, where

$$e_j = -td_j \text{ for } j \in Q. \quad (3.9)$$

thus

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

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

\[ \bar{b}_i = \bar{b}_i - \bar{p}_j \bar{a}_{ij} \]

or in vector notation

\[
\begin{bmatrix}
\bar{b}_i \\
\vdots \\
\bar{b}_m
\end{bmatrix}
= \begin{bmatrix}
\bar{b}_i \\
\vdots \\
\bar{b}_m
\end{bmatrix}
- \bar{p}_j \begin{bmatrix}
\bar{a}_{ij} \\
\vdots \\
\bar{a}_{mj}
\end{bmatrix}
\]

It is easy to see that the variable \( x_{ip} \) in the \( p \)th row after update takes the value

\[ x_{ip} = \bar{b}_p = \bar{b}_p - \bar{p}_j \bar{a}_{ij} \alpha_p = 0. \] (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 \(| SB \cup Q| - 1 \) 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 \( e_j \) to a higher level by the standard ratio test of simplex.

Let

\[ \frac{\bar{b}_p}{\bar{a}_{pj}} = \min_i \left( \frac{\bar{b}_j}{\bar{a}_{ij}} \mid \bar{a}_{ij} > 0, \bar{b}_j > 0 \right) \] (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 \( x_j \) which pivots into the \( p \)th row is now updated by the simple upper bound algorithm. Thus \( x_j \) takes the value

\[ x_j = e_j + \frac{\bar{b}_p}{\bar{a}_{pj}} \]
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 $e_j$ to a lower level which is

$$x_j = \max \left[ 0, e_j + \frac{\bar{b}_p}{\bar{a}_{pj}} \right]$$

(3.13)

Where

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

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

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

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 steepest 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 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 \( x_1 + x_2 + x_3 \)

subject to

\[
\begin{align*}
3x_1 + 2x_2 - x_3 + x_4 &= 6 \\
3x_1 + 2x_2 + 4x_3 + x_5 &= 16 \\
3x_1 - 4x_3 + x_6 &= 3 \\
9/4x_1 + 4x_3 + 3x_3 + x_7 &= 17 \\
x_1 + 2x_2 + x_3 + x_8 &= 10 \\
x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, &> 0
\end{align*}
\]

where \( x_4 \) to \( x_8 \) are slack variables.

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

Minor Procedure 1

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

\[
\propto = \begin{bmatrix}
-3 \\
4 \\
9 \\
-1 \\
37/4 \\
4
\end{bmatrix}
\]

and initially \( b = \begin{bmatrix}
0 \\
6 \\
16 \\
17 \\
10
\end{bmatrix} \)

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

\[
\delta b = \begin{bmatrix}
0 \\
6 \\
16 \\
3 \\
17 \\
10
\end{bmatrix} - 6/4 \begin{bmatrix}
-3 \\
4 \\
9 \\
-1 \\
37/4 \\
4
\end{bmatrix}
\]

then \( x_0 = 9/2 \)

\[
\begin{align*}
x_4 &= 0 \\
x_5 &= 5/2 \\
x_6 &= 9/2 \\
x_7 &= 25/8 \\
x_8 &= 4
\end{align*}
\]

and the non-basic variables are held at \( x_1 = x_2 = x_3 = 1.5 \)
Figure 3.1

Legend:  
- variables made basic
- interior search moves
- simplex moves

<table>
<thead>
<tr>
<th></th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p^1$</td>
<td>1.5</td>
<td>1.5</td>
<td>1.5</td>
</tr>
<tr>
<td>$p^2$</td>
<td>1.5</td>
<td>1.5</td>
<td>1.5</td>
</tr>
<tr>
<td>$p^3$</td>
<td>0.66</td>
<td>2.75</td>
<td>1.5</td>
</tr>
<tr>
<td>$p^4$</td>
<td>1.33</td>
<td>2.0</td>
<td>2.0</td>
</tr>
</tbody>
</table>
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 $d_j$ 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.
### TABLE 4.1 TEST PROBLEMS

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

### 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

<table>
<thead>
<tr>
<th>PROBLEM</th>
<th>ALTERNATIVE STRATEGIES</th>
<th>NO OF FULL ITERATIONS</th>
<th>NO OF MINOR ITERATIONS</th>
<th>TOTAL TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>seconds</td>
</tr>
<tr>
<td>1</td>
<td>Simplex</td>
<td>2328</td>
<td>-</td>
<td>2045</td>
</tr>
<tr>
<td></td>
<td>All Dir.</td>
<td>90</td>
<td>2225</td>
<td>3275</td>
</tr>
<tr>
<td></td>
<td>10 Best Dir.</td>
<td>411</td>
<td>1952</td>
<td>2148</td>
</tr>
<tr>
<td></td>
<td>5/10 Best Dir</td>
<td>518</td>
<td>1609</td>
<td>1440</td>
</tr>
<tr>
<td>2</td>
<td>Simplex</td>
<td>399</td>
<td>-</td>
<td>68</td>
</tr>
<tr>
<td></td>
<td>All Dir.</td>
<td>36</td>
<td>342</td>
<td>182</td>
</tr>
<tr>
<td></td>
<td>10 Best Dir.</td>
<td>69</td>
<td>267</td>
<td>79</td>
</tr>
<tr>
<td></td>
<td>5/10 Best Dir</td>
<td>77</td>
<td>248</td>
<td>55</td>
</tr>
<tr>
<td>3</td>
<td>Simplex</td>
<td>444</td>
<td>-</td>
<td>129</td>
</tr>
<tr>
<td></td>
<td>All Dir.</td>
<td>15</td>
<td>690</td>
<td>236</td>
</tr>
<tr>
<td></td>
<td>10 Best Dir.</td>
<td>90</td>
<td>425</td>
<td>93</td>
</tr>
<tr>
<td></td>
<td>5/10 Best Dir</td>
<td>146</td>
<td>458</td>
<td>118</td>
</tr>
<tr>
<td>4</td>
<td>Simplex</td>
<td>161</td>
<td>-</td>
<td>4.4</td>
</tr>
<tr>
<td></td>
<td>All Dir.</td>
<td>3</td>
<td>67</td>
<td>4.7</td>
</tr>
<tr>
<td></td>
<td>10 Best Dir.</td>
<td>7</td>
<td>60</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td>5/10 Best Dir</td>
<td>12</td>
<td>60</td>
<td>3.9</td>
</tr>
<tr>
<td>5</td>
<td>Simplex</td>
<td>44</td>
<td>-</td>
<td>0.39</td>
</tr>
<tr>
<td></td>
<td>All Dir.</td>
<td>13</td>
<td>51</td>
<td>0.60</td>
</tr>
<tr>
<td></td>
<td>10 Best Dir.</td>
<td>14</td>
<td>39</td>
<td>0.54</td>
</tr>
<tr>
<td></td>
<td>5/10 Best Dir</td>
<td>19</td>
<td>44</td>
<td>0.73</td>
</tr>
</tbody>
</table>

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 \( i = 1, \ldots, m \), 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 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 \( d_j \) 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.

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

<table>
<thead>
<tr>
<th>PROBLEM</th>
<th>SUBROUTINE</th>
<th>SIMPLEX</th>
<th>All Dir.</th>
<th>10 Best Dir.</th>
<th>5/10 Best Dir.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>FTRAN</td>
<td>644</td>
<td>2252</td>
<td>1278</td>
<td>813</td>
</tr>
<tr>
<td></td>
<td>BTRAN</td>
<td>570</td>
<td>33</td>
<td>123</td>
<td>133</td>
</tr>
<tr>
<td></td>
<td>PRICE</td>
<td>101</td>
<td>0.4</td>
<td>26</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>CHUZRO</td>
<td>45</td>
<td>165</td>
<td>95</td>
<td>67</td>
</tr>
<tr>
<td>2</td>
<td>FTRAN</td>
<td>8</td>
<td>132</td>
<td>47</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>BTRAN</td>
<td>14</td>
<td>7</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>PRICE</td>
<td>5</td>
<td>0.03</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>CHUZRO</td>
<td>2</td>
<td>27</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>FTRAN</td>
<td>17</td>
<td>162</td>
<td>42</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>BTRAN</td>
<td>19</td>
<td>1</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>PRICE</td>
<td>79</td>
<td>2</td>
<td>22</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>CHUZRO</td>
<td>7</td>
<td>60</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>4</td>
<td>FTRAN</td>
<td>0.9</td>
<td>1.8</td>
<td>1.2</td>
<td>1.3</td>
</tr>
<tr>
<td></td>
<td>BTRAN</td>
<td>1.5</td>
<td>0.6</td>
<td>0.7</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td>PRICE</td>
<td>0.5</td>
<td>0.06</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>CHUZRO</td>
<td>0.5</td>
<td>1.2</td>
<td>0.6</td>
<td>0.7</td>
</tr>
<tr>
<td>5</td>
<td>FTRAN</td>
<td>0.06</td>
<td>0.27</td>
<td>0.17</td>
<td>0.23</td>
</tr>
<tr>
<td></td>
<td>BTRAN</td>
<td>0.11</td>
<td>0.05</td>
<td>0.04</td>
<td>0.07</td>
</tr>
<tr>
<td></td>
<td>PRICE</td>
<td>0.14</td>
<td>0.02</td>
<td>0.08</td>
<td>0.13</td>
</tr>
<tr>
<td></td>
<td>CHUZRO</td>
<td>0.07</td>
<td>0.27</td>
<td>0.24</td>
<td>0.22</td>
</tr>
</tbody>
</table>

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

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.
7. References


