A FEASIBLE DIRECTION METHOD FOR LINEAR PROGRAMMING

K.G. MURTY *

Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI 48109, USA

Y. FATHI

Department of Industrial and Systems Engineering, University of Michigan - Dearborn, Dearborn, MI 48128, USA

Received December 1983
Revised April 1984

We discuss a finite method of feasible direction for linear programming problems. The method begins with a feasible basic vector for the problem, constructs a profitable direction to move using the updated column vectors of the nonbasic variables eligible to enter this basic vector. It then moves in this direction as far as possible, while retaining feasibility. This move in general takes it though the relative interior of a face of the set of feasible solutions. The final point, $\bar{x}$, obtained at the end of this move will not in general be a basic solution. Using $\bar{x}$, the method then constructs a basic feasible solution at which the objective value is better than, or the same as that at $\bar{x}$. The whole process repeats with the new basic feasible solution. We show that this method can be implemented using basis inverses. Initial computer runs of this method in comparison with the usual edge following primary simplex algorithms are very encouraging.

linear programming • primal simplex algorithm • edge directions • interior directions • degeneracy • finite termination

1. Introduction

We consider the linear program

$$\begin{aligned} &\text{minimize} && z(x) = cx \\ &\text{subject to} && Ax = b, \\ & && x \geqslant 0 \end{aligned} \tag{1}$$

where A, b, c are given matrices of orders $m \times n$, $m \times 1$ and $1 \times n$ respectively, and $\text{rank}(A) = m$. Let K denote the set of its feasible solutions. If D is any matrix, we denote its ith row by the symbol $D_{i.}$ and its jth column by the symbol $D_{.j}$. Let $x_B = (x_1, \ldots, x_m)$ be a feasible basic vector for this problem. $x_D = (x_{m+1}, \ldots, x_n)$ is the corresponding vector of nonbasic variables. The well-known primal simplex algorithm always begins with a feasible basic vector like this for the problem. The basic feasible solution (BFS) corresponding to this basic vector is $\bar{x} = (\bar{x}_B, \bar{x}_D) = (\bar{b}^T, 0)$.

If the relative cost coefficients $c_j \geqslant$ for all x_j nonbasic, the BFS $\bar{x}$ is optimal to the LP and we terminate. Otherwise, the nonbasic variable x_j is *eligible to enter the basic vector* x_B if $\bar{c}_j < 0$. The primal simplex algorithm selects exactly one of the eligible variables and tries to bring it into the basic vector. The selected variable is called *the entering variable*. Suppose it is x_s. Then it constructs a new solution by giving the entering variable x_s a value λ, leaving all other nonbasic variables equal to zero and then reevaluating the value of the basic variables. This leads to $x(\lambda) = (x_j(\lambda))$ where

$$\left[\begin{array}{l} i\text{th basic value, } x_i(\lambda) = \bar{b}_i - \lambda \bar{a}_{is},\ i = 1 \text{ to } m, \\ x_s = \lambda, \\ x_j = 0 \text{ for all nonbasic } x_j \neq x_s; \end{array}\right]$$

define $y = (y_i) \in \mathbf{R}^n$ by

$$\begin{aligned} y_i &= -\bar{a}_{is} && \text{for } i = 1 \text{ to } m, \\ &= 1 && \text{for } i = s, \\ &= 0 && \text{for all other } i. \end{aligned}$$

* The work of K.G. Murty has been partially supported by NSF grant No. ECS-8401081; also a portion of this work was carried out while on a visit to the Indian Statistical Institute, New Delhi, India under a travel grant INT-8318399 from the Division of International Programs of NSF.

Fig. 1.

Table 1
Canonical tableau with respect to the basic vector $x_B = (x_1, \ldots, x_m)$

basic variables	$x_1 \cdots x_m$	$\cdots$	x_j	$\cdots$	$-z$	b
x_1	I	$\cdots$	$\bar{A}_{.j}$	$\cdots$	0	$\bar{b}$
$\vdots$						
x_m						
$-z$	0	$\cdots$	$\bar{c}_j$	$\cdots$	1	$-\bar{z}$

Then $x(\lambda) = \bar{x} + \lambda y$. Let θ be the *minimum ratio* for bringing x_s into the basic vector x_B. This is a nondegenerate pivot step if $\theta > 0$. In this case we get the adjacent BFS $x(\theta) = \bar{x} + \theta y$ at the end of this pivot step. This new BFS $x(\theta)$ is obtained by moving from the current feasible solution $\bar{x}$ in the direction y to the maximum possible step length (while retaining feasibility) of θ. This is a walk along an edge of K, and hence the direction y used in this step is an *edge direction*. In the primal simplex algorithm the same thing happens in each nondegenerate pivot step, the algorithm moves from a BFS to an adjacent BFS along the edge of K joining them. So the primal simplex algorithm is a feasible direction method in which the directions chosen are always restricted to be edge directions.

Here we discuss a method which constructs a *profitable direction to move* from the current BFS, using a weighted combination of the updated columns of some or all the nonbasic columns in Table 1 which are eligible to enter the basis there. In general, such a move walks through the relative interior of a face of K, and therefore is likely to go much farther towards optimality than a walk along an edge. See Figure 1.

Such feasible direction methods are well known, in fact one was proposed in a 1951 paper by G.W. Brown and T.C. Koopmans [1] which appeared in the same volume as Dantzig's classical simplex paper [3]. But these straight forward feasible direction methods may not even have the finite termination property [5]. By combining these moves in profitable directions, with an efficient technique for finding a BFS with the same or better objective value as a given feasible solution, we get a method which terminates in a finite number of steps. We also show how this whole method can be implemented by maintaining a basis inverse at each stage. All the computer implementations of the usual edge following simplex algorithm also depend on maintaining a basis inverse in some form or other. Hence this suggested method can easily be adopted by existing large scale implementations of the usual simplex algorithm and take full advantage of the sparsity and any other advantageous structural properties that (1) may possess.

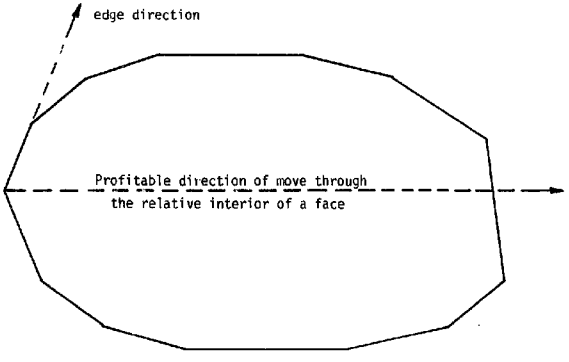

Fig. 1.

2. To get a BFS of better or same objective value from a given feasible solution

Suppose $\hat{x}$ is a feasible solution for (1), not necessarily basic. Define $J(\hat{x}) = \{j: \hat{x}_j > 0\}$. If the set $\{A_{.j}: j \in J(x)\}$ is linearly independent, x is itself a BFS. If $\{A_{.j}: j \in J(\hat{x})\}$ are linearly dependent, let a linear dependence relation among them be $\Sigma(\alpha_j A_{.j}: j \in (J(\hat{x})) = 0$. Define $x(\lambda) = (x_j(\lambda))$ by

$$x_j(\lambda) = \hat{x}_j + \lambda\alpha_j \quad \text{if } j \in J(\hat{x}),$$
$$= 0 \quad \text{otherwise.}$$

Also, define

$$\theta_1 = \min\{\hat{x}_j/(-\alpha): j \in J(\hat{x}) \text{ and } \alpha_j < 0\},$$

$$\theta_2 = \max\{\hat{x}_j/(-\alpha): j \in J(\hat{x}) \text{ and } \alpha_j > 0\}.$$

In computing θ_1, θ_2 define the minimum, maximum in the empty set to be $+\infty$, $-\infty$ respectively. If either $\Sigma(c_j\alpha_j: j \in J(\hat{x})) < 0$ and $\theta_1 = +\infty$ or $\Sigma(c_j\alpha_j: j \in J(\hat{x})) > 0$ and $\theta_2 = -\infty$, making $\lambda \to +\infty$ or $-\infty$, respectively, we get a feasible half-line along which $z(x) \to -\infty$; we terminate. If this unboundedness criterion is not satisfied, chose $\theta = \min\{|\theta_1|, |\theta_2|\}$ if $\Sigma(\alpha_j c_j: j \in J(\hat{x})) = 0$, $\theta = \theta_1$ if $\Sigma(\alpha_j c_j: j \in J(\hat{x})) < 0$, and $\theta = \theta_2$ if $\Sigma(\alpha_j c_j: j \in J(\hat{x})) > 0$. Then it can be verified that $x(\theta)$ is a feasible solution of (1) satisfying $z(x(\theta)) \leqslant z(\hat{x})$ and that $|J(x(\theta))| < |J(\hat{x})|$. We will call this process of obtaining this $x(\theta)$ beginning with the feasible solution $\hat{x}$, using a linear dependence relation among the columns $\{A_{.j}: j \in J(\hat{x})\}$, a *reduction step*. The process can be repeated with $x(\theta)$. After at most n reduction steps of this type we will either discover that $z(x)$ is unbounded below in (1) or find a BFS $\bar{x}$ for it satisfying $z(\bar{x}) \leqslant z(\hat{x})$.

This move from the feasible solution $\hat{x}$ which is not a BFS, to the BFS $\bar{x}$ can be carried out efficiently by the following procedure. Let $J(\hat{x}) = \{j: \hat{x}_j > 0\} = \{1, \ldots, m, m+1, \ldots, m+r\}$, say. If (1) is nondegenerate, the number of positive variables in a feasible solution which is not basic will always be strictly greater than m. We consider this nondegenerate case here. The changes to be made to resolve degeneracy are discussed in Section 5. The procedure continually maintains a basis B consisting of m columns from $J(\hat{x})$. Suppose the current basis B consists of columns $\{A_{.1}, \ldots, A_{.m}\}$. Let $(\alpha_1, \ldots, \alpha_m)^T = B^{-1}A_{.m+1}$, computed using the basis inverse B^{-1}. Then

$$\alpha = (-\alpha_1, \ldots, -\alpha_m, 1, 0, \ldots, 0)^T$$

are the coefficients in a linear dependence relation among $\{A_{.j}: j \in J(\hat{x})\}$.

Note 1. At this stage, the basis is $B = (A_{.1}, \ldots, A_{.m})$ and the columns $A_{.m+1}, \ldots, A_{.m+r}$ are those corresponding to variables which are positive in the present feasible solution, but not contained in the present basis. In general, one can determine weights, $\gamma_{m+1}, \ldots, \gamma_{m+r}$, associated with these columns, where $(\gamma_{m+1}, \ldots, \gamma_{m+r}) \neq 0$, define $(\alpha_1, \ldots, \alpha_m)^T = B^{-1}(\Sigma(\gamma_j A_{.j}: j = m+1 \text{ to } m+r))$, $\alpha = (-\alpha_1, \ldots, -\alpha_m, \gamma_{m+1}, \ldots, \gamma_{m+r})^T$, and use this α as the vector of coefficients in a linear dependence relation among $\{A_{.j}: j \in J(\hat{x})\}$.

Using a linear dependence relation obtained either way, carry out a reduction step and obtain the feasible solution $x(\theta)$ as discussed above. In $x(\theta)$, one or more of the variables x_j with $j \in J(\hat{x})$ have value zero. If all these j are among $m+1$ to $m+r$, drop all of them from further consideration and keep the same basis. On the other hand, if some of x_1 to x_m are zero in $x(\theta)$, replace the columns $A_{.j}$ corresponding to them from the basis, one at a time, by columns from $A_{.m+1}$ to $A_{.m+r}$ corresponding to variables which are positive in $x(\theta)$. Update the basis inverse and repeat the process with the new feasible solution $x(\theta)$, until a BFS is reached.

So this whole procedure of moving from a 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.

3. The feasible direction method

It starts with a BFS $\bar{x}$ for (1) associated with the feasible basic vector, $x_B = (x_1, \ldots, x_m)$, say, for (1) (if such a BFS for (1) is not available, it can be found, if (1) is feasible, by solving a Phase I problem corresponding to (1) using the same method). If the relative cost coefficients $\bar{c}_j$ are $\geqslant 0$ for all nonbasic x_j, $\bar{x}$ is optimal for (1) and we terminate. Otherwise, separate the nonbasic variables $x_{m+1}, \ldots, x_n$ into two parts, say, $x_{D_1} = (x_{m+1}, \ldots, x_{m+r})$ and $x_{D_2} = (x_{m+r+1}, \ldots, x_n)$, where the relative cost coefficient $\bar{c}_j < 0$ for $j =$

$m+1$ to $m+r$, and $\bar{c}_j \geqslant 0$ for $j = m+r+1$ to n. Let $\bar{A}_{.j}$ denote the updated column of x_j with respect to x_B. Define $y = (y_1, \ldots, y_n)^T$ by

$$\begin{aligned} y_j &= w_j \quad \text{for } j = m+1 \text{ to } m+r, \\ &= 0 \quad \text{for } j = m+r+1, \ldots, n, \end{aligned} \tag{2}$$

and

$$(y_1, \ldots, y_m)^T = -\Sigma(w_t \bar{A}_{.t} : t = m+1 \text{ to } m+r)$$

where the w_j's are positive weights whose choice is discussed in Section 6.

Now we move from $\bar{x}$ in the direction y, that is, look at points $\bar{x} + \lambda y$, $\lambda \geqslant 0$. This continues to satisfy the equality constraints in (1). The step length of the move is the maximum possible length determined so as to keep all the variables $\geqslant 0$. Define

$$\theta = \min\{\bar{x}_j/(-y_j) : j \text{ such that } y_j < 0\}. \tag{3}$$

If (1) is nondegenerate, this θ will be strictly positive, that is, we will take a step of positive length when we move in the direction y, and this results in a strict decrease in the objective value. If $\theta = +\infty$, $z(x)$ is unbounded below in (1) and we terminate. Otherwise this move in the direction y takes us either through the interior of K or the relative interior of a boundary face of K to the point $\bar{x} + \theta y = \hat{x}$, say. In general, $\hat{x}$ may not be a BFS of (1). Using $\hat{x}$ obtain a BFS $\bar{x}$ of (1) satisfying $z(\bar{x}) \leqslant z(\hat{x})$ by the procedure described in Section 2. Repeat the whole process again beginning with the BFS $\bar{x}$.

4. Finiteness under nondegeneracy

Each step in the feasible direction method discussed above begins with a BFS for (1), makes a move in a profitable direction to a feasible solution, then goes through several reduction steps until a new BFS is obtained. The process is repeated with the new BFS. Assume that (1) is nondegenerate. In this case, if $\bar{x}$ is the initial BFS associated with the basic vector $x_B = (x_1, \ldots, x_m)$ in a step of this method, we have $\bar{x}_j > 0$ for all $j = 1$ to m. Let $\{j : \bar{c}_j < 0$ in the canonical tableau with respect to $x_B\}$ be $\{m+1, \ldots, m+r\}$, say. From the definition of the profitable direction to move, y, in (2), we see that y_j can be < 0 only for $j \in \{1, 2, \ldots, m\}$. From these facts we conclude that $\theta > 0$ in (3). The change in the objective value as we move from the BFS $\bar{x}$ to $\bar{x} + \theta y = \hat{x}$ is $\theta(\Sigma(\bar{c}_j w_j : j = m+1 \text{ to } m+r))$ and since $\theta > 0$, $w_j > 0$ and $\bar{c}_j < 0$ for all $j = m+1$ *to* $m+r$, the objective value strictly decreases in this move in the profitable direction y. The reduction steps may result in lowering the objective value further. Thus, if $\bar{x}$ is the BFS obtained at the end of this step, we have $z(\bar{x}) < z(\hat{x})$. Since the objective value decreases in each step, a BFS cannot reappear in this method. Since there are only a finite number of BFS's for (1), the method must terminate after a finite number of steps.

5. Resolution of degeneracy

If (1) is degenerate, we might get a feasible basic vector x_B, for which some of the basic variables have zero value in the BFS associated with it. If this happens, the step length in the profitable direction of move computed as in (3) could be zero. This would result in no change in the objective value or even the feasible solution in a profitable direction of move, and the method could get stuck.

This problem under degeneracy in this method can be resolved using perturbations of the right-hand side constants vector in (1), just as it is done in the usual edge following primal simplex algorithm. It is well known that when b in (1) is replaced by $b(\varepsilon) = b + \nu$ where $\nu = (\varepsilon, \varepsilon^2, \varepsilon^3, \ldots, \varepsilon^m)^T$, and ε is a small positive number, it becomes nondegenerate for all ε positive but sufficiently small. It is not necessary to give a specific numerical value for ε, but it is left as a small positive parameter. In every feasible solution, $x = (x_j)$ of the perturbed problem, each variable x_j will have a value which is of the form $a_0^j + a_1^j\varepsilon + \ldots + a_m^j\varepsilon^m$ and we can represent this by the corresponding vector of coefficients $(a_0^j, a_1^j, \ldots, a_m^j)$. This solution is feasible for the perturbed problem for sufficiently small positive values of ε if the vector $(a_0^j, a_1^j, \ldots, a_m^j)$ is either zero or lexicopositive for all j. For example, if $\bar{x} = (\bar{x}_j)$ is the solution of the perturbed problem corresponding to the basic vector $x_B = (x_1, \ldots, x_m)$ associated with the basis B, and $\beta = B^{-1}$, $\bar{b} = B^{-1}b$, then

$$\begin{aligned} \bar{x}_i &= \bar{b}_i + \beta_{i.}\nu, \quad i = 1 \text{ to } m, \\ &= 0 \qquad\qquad\quad i = m+1 \text{ to } n, \end{aligned}$$

and so we represent $\bar{x}_i$ by the vector $(\bar{b}_i, \beta_{i.})$ for each $i = 1$ to m, and by the vector $0 \in \mathbb{R}^{m+1}$ for each $i = m+1$ to n.

We apply the method on the perturbed problem, treating ε as a sufficiently small positive parameter without giving any specific value to it, representing each variable in a solution by a vector in $\mathbb{R}^{m+1}$ in this manner, initiating the method with a lexicofeasible basic vector. Under this scheme, the computation of θ in (3) would involve taking the lexicominimum along the appropriate vectors instead of the usual minimum, and θ itself will be represented by a lexicopositive vector in $\mathbb{R}^{m+1}$. In this scheme, in carrying out reduction steps, the computation of θ_1, θ_2 will involve taking the lexicominimum, lexicomaximum respectively, instead of the usual minimum and maximum, and these θ_1, θ_2 will themselves be represented by the corresponding vectors in $\mathbb{R}^{m+1}$. Using this (essentially replacing the minimum, maximum in each place by the appropriate lexicominimum, lexicomaximum), the method can be carried out on the perturbed problem without giving any specific value to the parameter ε. Since the perturbed problem is nondegenerate for positive but sufficiently small values of the parameter ε, the method must terminate after a finite number of steps. At termination, we get the same termination condition for the original problem by setting the parameter $\varepsilon = 0$ in the terminal output for the perturbed problem.

6. Choice of weights

The natural choice for the weights w_j in (2) for determining the profitable direction of move seems to be $w_j = 1$ or $-\bar{c}_j$. The choice of $w_j = -\bar{c}_j$ is similar to the directions chosen in reduced gradient methods (see Wolfe [6]). It is not necessary to choose $w_j > 0$ for all $j \in J = \{j: j \text{ such that } \bar{c}_j < 0\}$. The method can be executed by choosing $w_j > 0$ for only a subset of $j \in J$, in particular subsets with cardinalities ranging from 2 to 5 may be more attractive than the option of making $w_j > 0$ for all $j \in J$.

In many packages for large scale linear programming, they usually select a number (about 5) of columns with significantly negative relative cost coefficients and process the subproblem consisting of the present basic columns and the selected 5 or so columns only, ignoring all the other columns temporarily (see Section 3.6.3, Multiple Pricing, in [4]). Once an optimum basis for the subproblem is obtained, the whole process is repeated with it. Instead of the usual edge following simplex algorithm, we can use the feasible direction method discussed above to solve each subproblem in this process, thereby improving the efficiency.

7. Alternate procedures

Consider BFS $\bar{x}$ for (1) associated with the feasible basic vector $x_B = (x_1, \ldots, x_m)$, say. As in Section 3, suppose the relative cost coefficients of x_{m+1} to x_{m+r} are < 0, and those of x_{m+r+1} to x_n are $\geqslant 0$, in the canonical tableau with respect to x_B. Define $A_{.n+1} = \Sigma(w_j A_{.j}: j = m+1 \text{ to } m+r)$, $c_{n+1} = \Sigma(w_j c_j: j = m+1 \text{ to } m+r)$. Introduce the *new variable* x_{n+1} associated with the column vector $A_{.n+1}$ and the cost coefficient c_{n+1} into (1). In the canonical tableau for the augmented problem with respect to x_B the updated column of x_{n+1} will be

$$\begin{pmatrix} \bar{A}_{.n+1} \\ \vdots \\ \bar{c}_{n+1} \end{pmatrix},$$

where $\bar{A}_{.n+1} = \Sigma(w_j \bar{A}_{.j}: j = m+1 \text{ to } m+r)$, $\bar{c}_{n+1} = \Sigma(w_j \bar{c}_j: j = m+1 \text{ to } m+r)$, with $\bar{A}_{.j}$, $\bar{c}_j$ being the updated column and the relative cost coefficient of x_j for $j = m+1$ to $m+r$. For this augmented problem, x_{n+1} is an eligible variable to enter the feasible basic vector x_B. When x_{n+1} is brought into the basic vector x_B in this augmented problem, suppose we get the new feasible solution $x^1 = (x_1^1, \ldots, x_n^1, x_{n+1}^1)^T$. It can be verified that the step length θ computed in (3) is actually x_{n+1}^1, and that the feasible solution for the original problem $\bar{x}$ obtained in Section 3 at the end of the move in direction y is $(x_1^1, \ldots, x_m^1, x_{m+1}^1 + w_{m+1}x_{n+1}^1, \ldots, x_{m+r}^1 + w_{m+r}x_{n+1}^1, x_{m+r+1}^1, \ldots, x_n^1)^T$. So the feasible solution x^1 for the augmented problem leads to the feasible solution $\hat{x}$ for the original problem by remembering that the *new variable* x_{n+1} corresponds to $\Sigma(w_j x_j: j = m+1 \text{ to } m+r)$. Now x^1 is of course a BFS for the augmented problem. If the current basic vector for the augmented problem satisfies the primal simplex optimality criterion in the augmented problem, x^1 is optimal for it and correspondingly $\hat{x}$ is optimal for the original problem. If the optimality criterion is not satisfied in the canonical tableau for the augmented problem with respect to the present basic

vector we can identify all nonbasic variables eligible to enter, and repeat the same procedure again.

It should be noted that the set of nonbasic variables eligible to enter the basic vector for this augmented problem is different from the set $\{x_{m+1},\ldots,x_{m+r}\}$, which was the set of eligible variables in the previous step. This is true because after the pivot step of bringing x_{n+1} into the basic vector x_B, the updated cost coefficient of x_{n+1} is zero, and from the definition of the column vector associated with x_{n+1} we notice that the updated cost coefficient of x_{n+1} will always be equal to $\Sigma(w_j$ [updated cost coefficient of x_j]: $j = m+1$ to $m+r)$. Therefore in every canonical tableau with x_{n+1} as a basic variable, at least one of the variables $x_{m+1},\ldots,x_{m+r}$ will have a nonnegative updated cost coefficient.

After several steps of this, we will have a canonical tableau for an augmented problem with one or more *new variables*, say, $x_{n+1},\ldots,x_{n+t}$. The column associated with any of these new variables is a positive combination of the columns of the nonbasic variables among $x_1,\ldots,x_n$, which were eligible to enter at the stage that this new variable was created. Whenever one of these *new variables* becomes the dropping variable in some step of this process, that *new variable* is totally discarded after the updating process in this pivot step. Because of this, all nonbasic variables at any stage will be original problem variables. Each of these *new variables* x_{n+1} to x_{n+t} in the augmented problem at this stage corresponds to a weighted combination of some original problem variables; using this we can obtain a feasible solution (not necessary basic) of the original problem from the current BFS of the present augmented problem.

Using this process of adding *new variables* to the problem we can identify two different procedures to solve the original problem. Both these procedures are guaranteed to terminate after a finite number of steps.

Alternate Procedure 1 – After some chosen number of steps of this process we can stop the process, identify the feasible solution $\hat{x}$ of the original problem corresponding to the present BFS of the current augmented problem, and go to the reduction steps with $\hat{x}$ as in Section 2. We then discard all *new variables*, and start all over again with the BFS of the original problem obtained at the end of these reduction steps. Finite termination of this procedure can be guaranteed using similar arguments as presented in Sections 4 and 5.

Alternate Procedure 2 – Continue the process of adding new columns until at some stage either the optimality criterion is satisfied for the current augmented problem, in which case an optimal solution to the original problem is readily available, or an unbounded ray is identified for the current augmented problem, in which case we can conclude that the original problem is unbounded. In either case terminate the procedure.

Finite termination of this algorithm can be guaranteed if all w_j's are taken to be equal to 1 (or any other preselected positive values) throughout the algorithm. To see this let $\mathcal{N} = \{N_1, N_2,\ldots,N_s\}$, where $s = -1 + 2^n$, be the set of all nonempty subsets of $N = \{1, 2,\ldots,n\}$, and consider the following maximal augmented LP corresponding to (1).

$$\begin{aligned} \text{minimize} \quad & z'(\zeta) = \sum_{k=1}^{s} \left(\Sigma_{t\in N_k} w_t c_t\right)\zeta_k \\ \text{subject to} \quad & \sum_{k=1}^{s} \left(\sum_{j\in N_k} w_j A_{\cdot j} \right)\zeta_k = b, \\ & \zeta_k \geqslant 0 \text{ for } k = 1 \text{ to } s. \end{aligned} \tag{4}$$

The variable ζ_k in (4) corresponds to $\Sigma_{j\in N_k} w_j x_j$ in (1). Clearly every new column that we may introduce in alternate procedure 2 is a column associated with a variable in (4), and every *new variable* introduced in the alternate procedure 2 is one of the ζ_k's, and thus every feasible basic vector obtained in this procedure, is a feasible basic vector for (4).

The total number of basic vectors for (4) is finite. Alternate procedure 2 is the same as executing the primal simplex algorithm on (4), and when it is executed using the lexico-dropping variable choice rules, it is guaranteed to terminate finitely.

8. Computational experiment

We are in the process of conducting a computational experiment comparing these methods with the usual edge following primal simplex algorithm on randomly generated problems of various sizes. Initial results are very encouraging. The experiment is also being designed to identify the best choices for various alternatives in the method. We

are also investigating better techniques for resolving degeneracy and ways of handling degeneracy in computer implementations of the method.

References

[1] G.W. Brown and T.C. Koopmans, "Computational suggestions for maximizing a linear function subject to linear inequalities", in: T.C. Koopmans, ed., *Activity Analysis of Production and Allocation*, Wiley, 1951, pp. 377–380.

[2] L. Cooper and J. Kennington, "Non-extreme point solution strategies for linear programs", *Naval Research Logistics Quarterly* **26**(3), 477–462 (1979).

[3] G.B. Dantzig, "Minimization of a linear function of variables subject to linear inequalities", in: T.C. Koopmans, ed., *Activity Analysis of Production and Allocation*, Wiley, 1951.

[4] B.A. Murtagh, *Advanced Linear Programming*, McGraw-Hill, New York, 1981.

[5] H.D. Sheralli, A.L. Soyster and S.G. Baines, "Nonadjacent extreme point methods for solving linear programs", *Naval Research Logistics Quarterly* **30**, 145–161 (1983).

[6] P. Wolfe, "Methods of nonlinear programming", in: R.L. Graves and P. Wolfe, eds., *Recent Advances in Mathematical Programming*, McGraw-Hill, New York, 1963.