Mathematical Programming 11 (1976) 105–116.
North-Holland Publishing Company

# A NETWORK SIMPLEX METHOD

W.H. CUNNINGHAM*
*The Johns Hopkins University, Baltimore, Md., U.S.A.*

Received 3 February 1975
Revised manuscript received 8 September 1975

Simple combinatorial modifications are given which ensure finiteness in the primal simplex method for the transshipment problem and the upper-bounded primal simplex method for the minimum cost flow problem. The modifications involve keeping "strongly feasible" bases. An efficient algorithm is given for converting any feasible basis into a strongly feasible basis. Strong feasibility is preserved by a rule for choosing the leaving basic variable at each simplex iteration. The method presented is closely related to a new perturbation technique and to previously known degeneracy modifications for shortest path problems and maximum flow problems.

## 1. Introduction

Let $G$ be a finite directed graph. We denote its vertex-set by $V$ and its edge-set by $E$. For $e \in E$ write $h(e)$, $t(e)$ to denote, respectively, the head and tail of $e$. Let $A_{m \times n}$ be the incidence matrix of $G$; that is, $A = (a_{ij}: i \in V, j \in E)$ where

$$a_{ij} = \begin{cases} 1 & \text{if } i = h(j); \\ -1 & \text{if } i = t(j); \\ 0 & \text{otherwise.} \end{cases}$$

Now let $c = (c_j: j \in E)$ and $b = (b_i: i \in V)$ be real vectors and let $u = (u_j: j \in E)$ be a positive real vector. The linear programs (1) and (2) below are usually called the transshipment problem and the minimum cost flow problem, respectively.

$$\begin{array}{rl} \min & z = cx, \\ \text{subject to} & Ax = b, \quad x \geqslant 0. \end{array} \tag{1}$$

$$\begin{array}{rl} \min & z = cx, \\ \text{subject to} & Ax = b, \quad 0 \leqslant x \leqslant u. \end{array} \tag{2}$$

The simplex method for (1) and the upper-bounded simplex method for (2) may be implemented as algorithms operating combinatorially on $G$ rather than arithmetically on matrices ([2, Chapters 17 and 18]). Provided that cycling does

* The author holds a National Research Council of Canada Post-Doctorate Fellowship.

not occur the algorithms are finite and, in fact, practically efficient. In particular, when the linear programs are non-degenerate, the algorithms are finite. Unfortunately, instances of (1) and (2) are often highly degenerate, especially when $b$ (and $u$) consists of integers. An example of (1) for which the simplex method does cycle has been discovered by Gassner [8].

In the case of the Hitchcock problem, which is a special case of (1), degeneracy has been handled by perturbing $b$ to create a new non-degenerate problem such that every feasible (optimal) solution of the new problem yields a feasible (optimal) solution of the original [1, 11]. These methods involve doing extra arithmetic and, if the perturbation is actually carried out, have the undesirable property of converting integer data to non-integer data.

Johnson [10] has given a combinatorial degeneracy modification for the upper-bounded simplex method for the maximal flow problem ([6; 2, Chapter 19]). As he points out, this can be used to ensure finiteness of the primal-dual algorithm [3] applied to the minimum cost flow problem.

The main purpose here is to provide simple combinatorial modifications to prevent cycling in the primal simplex algorithm for (1) and the upper-bounded primal simplex algorithm for (2). The method involves keeping "strongly feasible" bases, which arise from spanning trees of $G$ providing basic feasible solutions and whose edges satisfy certain additional orientation requirements. We give an efficient algorithm for constructing such bases from ordinary feasible bases. The rule for keeping successive bases of the simplex algorithm strongly feasible is a simple graph-theoretic rule for deciding at each iteration which basic edge becomes non-basic.

The method presented here bears an interesting relation to a perturbation device and also to previously known degeneracy modifications for shortest path problems and maximum flow problems.

## 2. Preliminaries

Elementary terms used but not defined here can be found in [5].

Where $p = (p_j : j \in J)$ is a real vector and $I \subseteq J$, we abbreviate $\Sigma\,(p_j : j \in I)$ to $p(I)$. Where $S \subseteq V$, we denote $V - S$ by $\bar{S}$. The set $\delta(S)$ is defined to be $\{e \in E : t(e) \in S, h(e) \in \bar{S}\}$. The following proposition is easily demonstrated.

**Proposition 1.** *For any $S \subseteq V$ and any solution $x^0$ of $Ax = b$,*

$$b(S) = x^0(\delta(\bar{S})) - x^0(\delta(S)). \tag{3}$$

It follows that $b(V) = 0$ is a necessary condition for (1) and (2) to have solutions. Thus we assume that $b(V) = 0$.

If $G$ is not connected, it is easy to see that (1) (or (2)) decomposes into two smaller problems of the same kind. Hence we may assume that $G$ is connected. For this case the following proposition is well-known.

**Proposition 2.** *The set of columns of A indexed by $T \subseteq E$ is a column-basis for A if and only if T is the edge-set of a spanning tree of G.*

Henceforth, we abbreviate "edge-set of a spanning tree" to "spanning tree." An important property of a spanning tree of $G$ is that there is a unique path from any vertex of $G$ to any other using only edges of the tree. Where $e$ is an edge not in a spanning tree $T$ and $G$, we define $C(T, e)$ to be the subset of $E$ consisting of $e$ together with the edges of the path $P$ from $h(e)$ to $t(e)$ in $T$. We define $\alpha(T, e) = (\alpha_j(T, e)\colon j \in E)$ by

$$\alpha_j(T, e) = \begin{cases} -1 & \text{if } j = e \text{ or } j \text{ is a forward edge of } P; \\ 1 & \text{if } j \text{ is a reverse edge of } P; \\ 0 & \text{otherwise.} \end{cases}$$

The set $C(T, e)$ may be traversed as a cycle in a number of ways, depending on direction and initial vertex. If we choose such a cycle in which $e$ is a forward (reverse) edge, we say that $C(T, e)$ is traversed in the *direction of e* (*direction opposite to e*).

The *cost* (with respect to $c$) of a path $P$ of $G$ is defined to be $\Sigma\,(c_j\colon j$ a forward edge of $P) - \Sigma\,(c_j\colon j$ a reverse edge of $P$). Let $T$ be a spanning tree of $G$; let $d(v, w)$ denote the cost of the path in $T$ from $v$ to $w$ for $v, w \in V$. Then for $v_1, v_2, v_3 \in V$,

$$d(v_1, v_2) + d(v_2, v_3) = d(v_1, v_3). \tag{4}$$

Throughout the paper, $r$ will denote a fixed vertex of $G$. Let $f$ be an edge and $T$ be a spanning tree of $G$. We define $R(T, f)$ to be $\{v \in V\colon$ the path from $r$ to $v$ in $T$ does not contain $f\}$.

## 3. The transshipment problem

We see from Proposition 2 that every basic solution of (1) is the unique solution $x^0$ of a system ($Ax = b$, $x_j = 0$ for $j \notin T$) for some spanning tree $T$ of $G$. We say that $T$ is a *feasible basis* of (1) if its associated $x^0$ is feasible to (1). The following fact is a consequence of Proposition 2. For $f \in T$,

$$x_f^0 = \begin{cases} b(R(T, f)) & \text{if } h(f) \in R(T, f); \\ -b(R(T, f)) & \text{otherwise.} \end{cases} \tag{5}$$

We say that $f$ is *directed toward r in T* if $h(f) \in R(T, f)$; otherwise we say that $f$ is *directed away from r in T*.

In (i) to (iv) below we give a description of the simplex algorithm for (1). We assume that we have a feasible basis $T$ with associated basic solution $x^0$ with which to initiate the algorithm. The problem of finding a feasible basis is treated in the next section.

*Simplex algorithm*

(i) Let $\pi_i$, $i \in V$, be the cost of the path from $r$ to $i$ in $T$. For $e \notin T$, let $\bar{c}_e = c_e + \pi_{t(e)} - \pi_{h(e)}$.

(ii) Find $e \notin T$ such that $\bar{c}_e < 0$. If no such $e$ exists, stop; $x^0$ is optimal. Otherwise go to (iii).

(iii) Let $\theta = \min(x_j^0 \colon \alpha_j(T, e) = 1)$. If this minimum does not exist, stop; $C(T, e)$ (traversed in the direction of $e$) is a uniformly directed cycle of negative cost and (1) is unbounded. Otherwise, go to (iv).

(iv) Let $\hat{x}^0 = x^0 - \theta\alpha(T, e)$. Choose $f \in F = \{j \colon \alpha_j(T, e) = 1,\ x_j^0 = \theta\}$. Let $\hat{T} = (T \cup \{e\}) - \{f\}$. Replace $x^0$ by $\hat{x}^0$ and $T$ by $\hat{T}$ and go to (i).

(To see that this algorithm is the simplex method, one observes that $(\pi_i : i \in V)$ defined in (i) satisfies complementary slackness together with $x^0$, and that

$$\sum (\alpha_j(T, e)a_{ij} : j \in T) = a_{ie} \quad \text{for each } i \in V).$$

Some details of efficient handling of the data for computer implementation can be found in [9] and [10]. Recent experimental evidence indicates that, for most instances of (1), the network simplex method is, at the least, competitive with other solution algorithms, such as the methods of [5].

It follows from (4) and the definition of $\hat{x}^0$ in (iv) that $c\hat{x}^0 = cx^0 + \theta\bar{c}_e$. Thus the value of $z$ strictly decreases for each iteration of (iv) for which $\theta \neq 0$. Since the number of feasible bases is finite, the algorithm is finite unless it "cycles," that is, encounters repeatedly the same sequence of feasible bases, all having the same associated basic solution $x^0$.

Let $T$ be a feasible basis and $x^0$ its associated basic solution.

**Definition 1.** We say that $T$ is *strongly feasible* if every $f \in T$ with $x_f^0 = 0$ is directed away from $r$ in $T$.

We modify the simplex algorithm (i)–(iv) above by making two refinements. We initiate the algorithm with a strongly feasible basis and we make a specific choice of $f$ in (iv) as follows.

(v) Let $s$ be the first common vertex in the paths in $T$ from $h(e)$ and $t(e)$ to $r$. Choose $f$ to be the first member of $F$ encountered in traversing $C(T, e)$ in the direction of $e$ beginning at $s$.

As an illustration, consider the graph in Fig. 1, where only $e$ and edges of $T$ are shown. If $F = \{d, g, h\}$, then the rule (v) would choose $f = d$.

We denote by MSA (modified simplex algorithm) the simplex algorithm with these two refinements.

**Theorem 1.** *MSA is finite.*

**Lemma 1.** *MSA encounters only strongly feasible bases.*

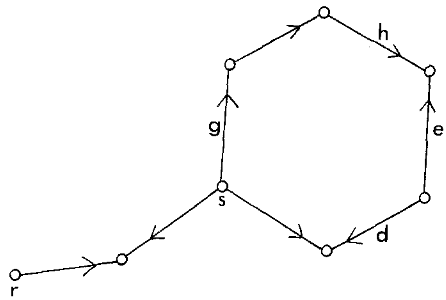


Fig. 1. Illustration of the rule (v).

**Proof.** Let $T$ be a strongly feasible basis of (1) and let $\hat{T}=(T\cup\{e\})-\{f\}$ be the next feasible basis generated by MSA. Let $x^0$, $\hat{x}^0$ be the associated basic solutions. It suffices to show that $\hat{T}$ is strongly feasible.

We remark that edges of $T$ not in $C(T,e)$ are directed away from $r$ in $\hat{T}$ if and only if they are so directed in $T$. Thus we restrict consideration to edges of $C(T,e)$. There are two cases.

*Case* 1: $\theta>0$. The edges $j\in C(T,e)\cap\hat{T}$ such that $\hat{x}^0_j=0$ are precisely the edges of $F-\{f\}$. Clearly rule (v) will ensure that each of these is directed away from $r$ in $\hat{T}$.

*Case* 2: $\theta=0$. Then $\hat{x}^0=x^0$. Since $T$ is strongly feasible, each element of $F$ is a forward edge of the path in $T$ from $s$ to $h(e)$. The other elements $j$ of $C(T,e)$ such that $\hat{x}^0_j=0$ are $e$ and certain forward edges of the path in $T$ from $s$ to $t(e)$. It is clear that (v) ensures that all of these edges will be directed away from $r$ in $\hat{T}$.

Thus $\hat{T}$ is strongly feasible and the lemma is proved.

It will be useful to have the following explicit formula for the node numbers $(\hat{\pi}_i: i\in V)$ for the spanning tree $\hat{T}=(T\cup\{e\})-\{f\}$ in terms of the node numbers $(\pi_i: i\in V)$ for $T$.

This formula follows from (4) and the definition of $\bar{c}_e$.

$$\begin{aligned}&\text{If } i\in R(T,f), \quad \text{then } \hat{\pi}_i=\pi_i.\\ &\text{If } i\notin R(T,f), \quad \text{then } \begin{cases}\hat{\pi}_i=\pi_i+\bar{c}_e & \text{if } t(e)\in R(T,f)\\ \hat{\pi}_i=\pi_i-\bar{c}_e & \text{otherwise.}\end{cases}\end{aligned}\tag{6}$$

**Proof of Theorem 1.** We show that MSA cannot cycle by showing that if $\hat{T}$ is obtained from $T$ by an iteration of MSA for which $\theta=0$, then $\Sigma(\hat{\pi}_i: i\in V)<\Sigma(\pi_i: i\in V)$. Since $T$ is strongly feasible, $f$ is a forward edge of the path in $T$ from $s$ to $h(e)$. Therefore, $t(e)\in R(T,f)$ and so by (15) and the facts that $\bar{c}_e<0$ and $R(T,f)\neq V$, we have $\Sigma(\hat{\pi}_i: i\in V)<\Sigma(\pi_i: i\in V)$, which completes the proof.

## 4. Initial solutions and feasibility

In this section we present an efficient algorithm for transforming a feasible basis of (1) into a strongly feasible basis having the same associated basic solution. We also describe a Phase 1 application of MSA for finding a feasible basis. This latter device provides a proof of a well-known feasibility theorem for (1).

Let $T$ be a feasible basis for (1) with associated basic solution $x^0$. Suppose that $j \in T$ is badly directed, that is, $j$ is directed toward $r$ in $T$ and $x_j^0 = 0$. If $\delta(R(T,j)) = \phi$, it is easy to see from Proposition 1 that (1) decomposes into two smaller transshipment problems. Moreover, $T$ yields a feasible basis for each of the smaller problems, and thus we could apply the present procedure to each of them. Otherwise, choose $e \in \delta(R(T,j))$. Let $f$ be the first edge with $x_f^0 = 0$ in the path $P$ in $T$ from $h(e)$ to $h(j)$. Let $\hat{T} = (T \cup \{e\}) - \{f\}$. (Notice that the basic solution associated with $\hat{T}$ is $x^0$.) Then the edges of $T \cap \hat{T}$ whose orientation with respect to $r$ is different in $T$ and $\hat{T}$ are just the edges of $P$ between $e$ and $f$. If $f$ is forward in $P$, then it is directed toward $r$ in $T$, and $\hat{T}$ has one fewer badly directed edge than $T$. If $f$ is reverse in $P$, then $\hat{T}$ has the same badly directed edges as $T$ (including $j$), but $R(\hat{T},j) \supset R(T,j)$. Thus we will obtain a tree having fewer badly directed edges than $T$ within $(m-1)$ iterations, and so we will obtain a strongly feasible basis within $k(m-1)$ iterations, where $k$ is the number of badly directed edges in the initial feasible basis $T$.

The above algorithm bears a remarkable resemblance to MSA itself. However, the amount of computation required to implement it is clearly bounded by a polynomial in $m$ and $n$, while the simplex method for (1) is known not to be similarly "good" [4, 12].

We now describe a procedure for finding a feasible basis for (1). Consider a tree $T$ having an edge directed from $r$ to $i$ for each $i \in V - \{r\}$ with $b_i \geq 0$ and an edge directed from $i$ to $r$ for each $i \in V - \{r\}$ with $b_i < 0$. Wherever such an edge is not available in $G$, we create an *artificial* edge; the set of such edges is denoted by $A$. We denote by $G'$ the graph so obtained. Now $T$ is a strongly feasible basis for a transshipment problem on $G'$ having the same $b$-vector. The cost vector for the new problem is $d = (d_j : j \in E \cup A)$, where $d_j = 0$ for $j \in E$ and $d_j = 1$ for $j \in A$.

The application of MSA to this problem yields (by Theorem 1, since the problem is bounded) a spanning tree $T'$ of $G'$ with associated optimal basic solution $(x_j^0 : j \in E \cup A)$. If $x_j^0 = 0$ for $j \in A$, then $x^0 = (x_j^0 : j \in E)$ is a feasible solution to (1). Moreover, a feasible basis to (1) is easily obtained from $T'$ in this case. Any edge $j \in A \cap T'$ may be replaced by an edge

$$f \in \delta(R'(T',j)) \cup \delta(\overline{R'(T',j)})$$

where $R'(T',j)$ denotes

$$\{v \in V \colon \text{the path in } T' \text{ from } r \text{ to } v \text{ does not contain } j\},$$

since $b(R'(T',j)) = x_f^0 = x_j^0 = 0$.

Such an $f$ exists, since $G$ is connected. Therefore, this Phase 1 procedure, together with the algorithm at the beginning of the section, can be used to find a strongly feasible basis.

On the other hand, suppose that there exists $j \in A \cap T'$ with $x_j^0 > 0$. We assume that $j$ is directed away from $r$ in $T'$. (The other case is similar.) Let $S = \cap\{R'(T',f) \colon f \in A \cap T'\}$. Then by Proposition 1, $b(S) < 0$. Let $\pi'_i$, for $i \in V$, be the cost (with respect to $d$) of the path in $T'$ from $r$ to $i$. Then $\pi'_i = 1$ for $i \in \bar{S}$ and $\pi'_i \leq 0$ for $i \in S$. Now suppose that there exists $e \in \delta(S)$. Then $d_e - \pi'_{h(e)} + \pi'_{t(e)} < 0$, contradicting the optimality of $T'$ by (ii). Thus $\delta(S) = \phi$. The following result, a special case of a theorem of Gale [7], has therefore been proved. (The necessity of the condition follows from Proposition 1).

**Theorem 2.** *There is a feasible solution to* (1) *if and only if there does not exist a set* $S \subseteq V$ *with* $b(S) < 0$ *and* $\delta(S) = \phi$.

## 5. A perturbation

Let $\epsilon$ be a positive real number. Define $b' = (b'_i : i \in V)$ by $b'_i = b_i + \epsilon$ for $i \neq r$ and $b'_r = b_r - (m-1)\epsilon$. Consider the following transshipment problem.

$$\begin{aligned} &\text{minimize} \quad z = cx, \\ &\text{subject to} \quad Ax = b', x \geq 0. \end{aligned} \tag{7}$$

**Theorem 3.** *If $\epsilon$ is sufficiently small, then* (7) *is non-degenerate and a basis is feasible to* (7) *if and only if it is strongly feasible to* (1).

It is a consequence of Theorem 3 that MSA is equivalent to the ordinary simplex algorithm applied to a non-degenerate problem, in the sense that the same sequence of bases is generated from the same initial basis. However, the perturbation approach lacks the elegance of the combinatorial method and, as previously remarked, involves extra numerical computation. (It should be noted that, in the special case of the Hitchcock problem, the perturbation (7) is different from that given in [1] and [11].)

**Proof of Theorem 3.** Let $T$ be a spanning tree of $G$ and let $x^0, x^1$ be the basic solutions associated with $T$ in (1) and (7), respectively. We apply (5) and the definition of $b'$ to evaluate $x^1_f$ for any $f \in T$. If $h(f) \in R(T, f)$, then

$$\begin{aligned} x^1_f &= b'(R(T,f)) \\ &= b(R(T,f)) - \epsilon(m-1) + \epsilon(|R(T,f)| - 1) \\ &= x^0_f - \epsilon(m - |R(T,f)|). \end{aligned}$$

If $t(f) \in R(T, f)$, then

$$\begin{aligned} x^1_f &= -b'(R(T,f)) \\ &= -b(R(T,f)) + \epsilon(m-1) - \epsilon(|R(T,f)| - 1) \\ &= x^0_f + \epsilon(m - |R(T,f)|). \end{aligned}$$

Let $\epsilon_0 = \min(m^{-1}|b(S)|: S \subseteq V,\ b(S) \neq 0)$. Since $1 \leq m - |R(T,f)| \leq m - 1$ for each $f \in T$, we have, for $\epsilon \leq \epsilon_0$:

(a) If $x^0_f < 0$, then $x^1_f < 0$;

(b) If $x^0_f > 0$, then $x^1_f > 0$;

(c) If $x_f^0 = 0$, then $x_f^1 > 0$ if $f$ is directed away from $r$ in $T$, and $x_f^1 < 0$ otherwise. This completes the proof.

## 6. The shortest path problem

By the "shortest path problem" we mean the problem of finding a least cost uniformly directed path from $r \in V$ to $t \in V$. We assume that $G$ is connected and has no uniformly directed cycle of negative cost.

Let $T$ be an optimal basis of (1) where $b_r = -1$, $b_t = 1$ and $b_i = 0$ for $i \neq r, t$. It is well-known that the path from $r$ to $t$ in $T$ is an optimal solution to the shortest path problem. This instance of (1) is obviously highly degenerate. (In fact an example of cycling for such a problem can be constructed, somewhat artificially, by using Gassner's example.)

The problem of finding a least cost uniformly directed path from $r$ to every $v \in V$ can also be solved by solving a linear program, namely, (1) with $b_r = -(m-1)$ and $b_i = 1$ for $i \neq r$. This linear program is obviously non-degenerate. Moreover, it is well-known [2, Ch. 17; 4], and easy to see from Proposition 2 and 5, that its feasible bases are the "branchings rooted at $r$," that is, spanning trees in which every edge is directed away from $r$. One might regard the consideration of this more general problem to be a device for handling the degeneracy of the linear programming formulation of the shortest path problem. The effect of the device in the simplex solution of the shortest path problem is to allow only strongly feasible bases. Thus MSA could be regarded as a generalization of this known method. (This observation was made to the author by Jack Edmonds.)

## 7. The minimum cost flow problem

In this section we present results for the minimum cost flow problem which are analogous to the results for the transshipment problem in Sections 3–5. In many cases proofs are similar to those previously given, and so are not given in detail.

We see from Proposition 2 that every basic solution of (2) is the unique solution $x^0$ of a system ($Ax = b$, $x_j = u_j$ for $j \in U$, $x_j = 0$ for $j \notin T \cup U$) where $T$ is a spanning tree of $G$ and $U \subseteq E - T$. In this section we call $(T, U)$ a *basis* of (2) and say that $(T, U)$ is a *feasible basis* if its associated $x^0$ is feasible to (2). In this section $L$ will always denote $E - (T \cup U)$. The following fact is a consequence of Proposition 1.

$$x_f^0 = b(R(T,f)) + u(U \cap \delta(R(T,f))) - u(U \cap \delta(\overline{R(T,f)})) \tag{8a}$$

where $f$ is directed toward $r$ in $T$, and similarly

$$x_f^0 = -b(R(T,f)) - u(U \cap \delta(R(T,f))) + u(U \cap \delta(\overline{R(T,f)})) \tag{8b}$$

when $f$ is directed away from $r$ in $T$.

In (i′)–(iv′) below we give a description of the upper-bounded primal simplex method for (2). We assume that we have a feasible basis $(T, U)$ with associated basic solution $x^0$ with which to initiate the algorithm.

*Upper-bounded simplex algorithm*

(i′) Let $\pi_i$, $i \in V$, be the cost of the path from $r$ to $i$ in $T$. For $e \notin T$, let $\bar{c}_e = c_e + \pi_{t(e)} - \pi_{h(e)}$.

(ii′) Find $e \in L$ such that $\bar{c}_e < 0$ and go to (iii′) or $e \in U$ such that $\bar{c}_e > 0$ and go to (iv′). If no edge of either kind exists, stop; $x^0$ is optimal.

(iii′) Let $\theta = \min(\{x_j^0 \colon \alpha_j(T, e) = 1\} \cup \{u_j - x_j^0 \colon \alpha_j(T, e) = -1\})$. Let $\hat{x}^0 = x^0 - \theta\alpha(T, e)$. Choose

$$f \in F = \{j \colon \alpha_j(T, e) = 1, x_j^0 = \theta\} \cup \{j \colon \alpha_j(T, e) = -1, u_j - x_j^0 = \theta\}.$$

Let $\hat{T} = (T \cup \{e\}) - \{f\}$. If $\hat{x}_f^0 = 0$, let $\hat{U} = U$; if $\hat{x}_f^0 = u_f$, let $\hat{U} = U \cup \{f\}$. Replace $T$ by $\hat{T}$, $U$ by $\hat{U}$, and $x^0$ by $\hat{x}^0$ and go to (i′).

(iv′) Let $\theta = \min(\{x_j^0 \colon \alpha_j(T, e) = -1\} \cup \{u_j - x_j^0 \colon \alpha_j(T, e) = 1\})$. Let $\hat{x}^0 = x^0 + \theta\alpha(T, e)$. Choose

$$f \in F = \{j \colon \alpha_j(T, e) = -1, x_j^0 = \theta\} \cup \{j \colon \alpha_j(T, e) = 1, u_j - x_j^0 = \theta\}.$$

Let $\hat{T} = (T \cup \{e\}) - \{f\}$. If $\hat{x}_f^0 = 0$, let $\hat{U} = U - \{e\}$; if $\hat{x}_f^0 = u_f$, let $\hat{U} = (U - \{e\}) \cup \{f\}$.

Replace $T$ by $\hat{T}$, $U$ by $\hat{U}$, and $x^0$ by $\hat{x}^0$ and go to (i′).

Using (4) we can show that, in (iii′), $c\hat{x}^0 = cx^0 + \theta\bar{c}_e$ and that, in (iv′), $c\hat{x}^0 = cx^0 - \theta\bar{c}_e$. It follows that the algorithm is finite unless it cycles.

**Definition 2.** A feasible basis $(T, U)$ of (2), with associated basic solution $x^0$, will be called *strongly feasible* if each $f \in T$ with $x_f^0 = 0$ is directed away from $r$ in $T$, and each $f \in T$ with $x_f^0 = u_f$ is directed toward $r$ in $T$.

We refine the Upper Bounded Simplex Algorithm (i′)–(iv′) by initiating it with a strongly feasible basis and by making a specific choice of $f \in F$ in (iii′) and (iv′), according to the following rule.

**Rule 1.** Let $s$ be the first common vertex in the paths in $T$ from $h(e)$ and $t(e)$ to $r$. Choose $f$ to be the first member of $F$ encountered in traversing $C(T, e)$ beginning at $s$, where $C(T, e)$ is traversed in the direction of $e$ for (iii′) and in the direction opposite to $e$ for (iv′).

We denote by MUSA the algorithm (i′)–(iv′) with these two refinements.

**Theorem 4.** *MUSA is finite.*

**Lemma 2.** *MUSA encounters only strongly feasible bases.*

The proof of Lemma 2 is similar to the proof of Lemma 1 except that there are twice as many cases. Using Lemma 2, one can prove Theorem 4 by showing, as in the proof of Theorem 1, that at any iteration of MUSA for which $\theta = 0$, in either (iii′) or (iv′), $\Sigma\,(\pi_i : i \in V)$ strictly decreases.

The algorithm for converting a feasible basis of (2) into a strongly feasible basis is only slightly more complicated than its analog in Section 4. Let $(T, U)$ be a feasible basis of (2) with associated basic solution $x^0$. If $j \in T$ is badly directed (violates Definition 2), we choose $e \in L \cap \delta(R(T, j))$ or $e \in U \cap \delta(\overline{R(T, j)})$. If no such $e$ exists, it is easy to see that (2) decomposes into two smaller minimum cost flow problems. Otherwise, let $v$ be the end (head or tail) of $e$ which is not in $R(T, j)$ and let $w$ be the end of $j$ which is in $R(T, j)$. Choose $f$ to be the first edge of the path $P$ in $T$ from $v$ to $w$ such that $x_f^0 = 0$ or $u_f$.

Let $\hat{T} = (T \cup \{e\}) - \{f\}$, and define $\hat{U}$ according to the values of $x_e^0, x_f^0$. As before, $(\hat{T}, \hat{U})$ has associated basic solution $x^0$, and $\hat{T}$ either has fewer badly directed edges than $T$, or has the same badly directed edges as $T$ and has $R(\hat{T}, j) \supset R(T, j)$. This algorithm has the same efficiency as its analog in Section 4.

The Phase 1 application of MUSA to obtain a feasible basis for (2) is similar to that described in Section 4. This procedure also yields a proof of the following theorem of Gale [7].

**Theorem 5.** *There is a feasible solution to* (2) *if and only if there does not exist a set* $S \subseteq V$ *such that* $b(S) + u(\delta(S)) < 0$.

Finally we consider the following perturbed version of (2) where $b'$ is as defined in Section 5.

$$\begin{aligned} &\text{minimize} && z = cx, \\ &\text{subject to} && Ax = b', \qquad 0 \le x \le u. \end{aligned} \tag{9}$$

**Theorem 6.** *If $\epsilon$ is sufficiently small, then* (9) *is non-degenerate and a basis is feasible to* (9) *if and only if it is strongly feasible to* (2).

The proof of Theorem 6 is similar to the proof of Theorem 3, using (8a, b) instead of (5), except that $\epsilon_0$ must be chosen more carefully. One way to do this is as follows. Where $S \subseteq V$, $U_1 \subseteq \delta(S)$, and $U_2 \subseteq \delta(\bar{S})$, let $\Delta(S, U_1, U_2)$ denote $b(S) - u(U_1) + u(U_2)$. Let $\epsilon_0$ be the minimum over all $(S, U_1, U_2)$, such that $\Delta(S, U_1, U_2) \neq 0$, of $m^{-1}|\Delta(S, U_1, U_2)|$.

## 8. The maximum flow problem

Let $g$ be a special "return" edge of $G$ and let $h(g) = r$ and $t(g) = q$. Consider the instance of (2) in which $b_v = 0$ for all $v \in V$, $c_j = 0$ for $j \in E - \{g\}$, $c_g = -1$, and $x_g$ has no upper bound (or, equivalently, $u_g$ is sufficiently large that, for any feasible solution $x^0$ of (2), $x_g^0 < u_g$). It is easy to see that this instance of (2) is equivalent to the problem of finding a maximum flow from $r$ to $q$ [5].

In the upper-bounded simplex method for the present problem, we may assume that $g$ is an element of each basis, since $u_g > x_g^0 > 0$ for every solution $x^0$ for which $cx^0 \neq 0$. Thus a basis may be regarded as $g$ together with trees $T_r$, $T_q$ having vertex sets $V_r$, $V_q$ such that $r \in V_r$, $q \in V_q$ and $\{V_r, V_q\}$ is a partition of $V$. An edge $e \in L$ has $\bar{c}_e < 0$ if and only if $e \in \delta(V_r)$; an edge $e \in U$ has $\bar{c}_e > 0$ if and only if $e \in \delta(V_q)$. Such an edge $e$ identifies a path $P$ from $r$ to $q$; if $\theta > 0$, $P$ is a "flow-augmenting" path in the usual sense [5]. To maintain a simplex basis some edge of $P$ is deleted; the candidates (members of $F$) are the forward edges of $P$ whose new $x$-value is at its upper bound and the reverse edges of $P$ whose new $x$-value is at zero. In case there is more than one such edge, a perturbation method [6] can be used to select an edge to become non-basic; Rule 1 simply says that the candidate closest to $r$ in $P$ should be selected. (Notice that the vertex $s$ of Rule 1 will always be $r$.) In this instance of (2) it is also particularly easy to construct an initial strongly feasible basis, corresponding to the solution $x = 0$.

For the maximum flow problem, MUSA is closely related to the method proposed by Johnson [10]. The method of [10] does not require strong feasibility, but a degenerate "basis" may have fewer than the usual number of edges. However, if a "full" basis is encountered, it will be strongly feasible. Johnson's method and MUSA for the maximum flow problem share an attractive property, noted in [10]: there can be at most $m-2$ consecutive degenerate pivots. To see this for MUSA, one observes that, if $\theta = 0$, the members of $F$ in (iii') are edges of $T_q$, so the new $T_r$ will have at least one more edge than the old $T_r$. Such a bound does not seem to be available for more general instances of (2).

## Added in proof

When MSA is applied to another special case of (1), the optimal assignment problem, the resulting algorithm is equivalent to one discovered recently by Barr, Glover, and Klingman [13]. In this instance of (1), strongly feasible bases have a particularly simple structure, and a number of valuable computational advantages are detailed in [13].

Another method for preventing cycling, which extends to the simplex method for general linear programs, is a beautiful technique due to Bland [14]. A practical disadvantage of Bland's method is that it includes a specific choice of the new basic variable, and thus, unlike the present method, is not compatible with pivot choice rules which have been shown experimentally to improve the efficiency of the simplex algorithm.

## References

[1] G.B. Dantzig, "Application of the simplex method to a transportation problem", in: T.C. Koopmans, ed., *Activity analysis of production and allocation* (Wiley, New York, 1951).

[2] G.B. Dantzig, *Linear programming and extensions* (Princeton University Press, Princeton, 1963).

[3] G.B. Dantzig, L.R. Ford, Jr., and D.R. Fulkerson, "A primal-dual algorithm for linear programs", in: H.W. Kuhn and A.W. Tucker, eds., *Linear inequalities and related systems*, Annals of Mathematics Study 38 (Princeton University Press, Princeton, 1956).

[4] Jack Edmonds, "Exponential growth of the simplex method for shortest path problems", University of Waterloo, (1970) unpublished.
[5] L.R. Ford, Jr. and D.R. Fulkerson, *Flows in networks* (Princeton University Press, Princeton, 1962).
[6] D.R. Fulkerson and G.B. Dantzig, "Computations of maximal flows in networks", *Naval Research Logistics Quarterly* 2 (1955) 277–283.
[7] D. Gale, "A theorem on flows in networks", *Pacific Journal of Mathematics* 7 (1957) 1073–1082.
[8] B.J. Gassner, "Cycling in the transportation problem", *Naval Research Logistics Quarterly* 11 (1964) 43–58.
[9] Fred Glover, D. Karney, and D. Klingman, "The augmented predecessor index method for locating stepping-stone paths and assigning dual prices in distribution problems", *Transportation Science* 6 (1972) 171–179.
[10] Ellis Johnson, "Networks and basic solutions", *Operations Research* 14 (1966) 619–623.
[11] Alex Orden, "The transshipment problem", *Management Science* 2 (1956) 276–285.
[12] Norman Zadeh, "A bad network problem for the simplex method and other minimum cost flow algorithms", *Mathematical Programming* 5 (1973) 255–266.

## *Additional references*

[13] R.S. Barr, F. Glover and D. Klingman, "The alternating basis algorithm for assignment problems", Mathematical Programming, to appear.
[14] R.G. Bland, "New finite pivoting rules for the simplex method", CORE Discussion Paper, Université de Louvain, Belgium (1976).