Mathematical Programming 25 (1983) 228–239
North-Holland Publishing Company

THE MINIMUM COST FLOW PROBLEM: A UNIFYING APPROACH TO DUAL ALGORITHMS AND A NEW TREE-SEARCH ALGORITHM*

Refael HASSIN

Statistics Department, Tel-Aviv University, Tel-Aviv, Israel

Received 20 January 1981
Revised manuscript received 27 November 1981

This paper is concerned with the minimum cost flow problem. It is shown that the class of dual algorithms which solve this problem consists of different variants of a common general algorithm. We develop a new variant which is, in fact, a new form of the 'primal–dual algorithm' and which has several interesting properties. It uses, explicitly, only dual variables. The slope of the change in the (dual) objective is monotone. The bound on the maximum number of iterations to solve a problem with integral bounds on the flow is better than bounds for other algorithms.

Key words: Minimum Cost Network Flow, Tree-Search Algorithm, Primal–Dual Algorithm.

Let (N, A) be a directed network, where N is a finite set and where $A \subseteq N \times N$. We investigate the familiar minimal cost network flow problem:

(P)
$$\text{Minimize} \quad \sum_{(i,j)\in A} c_{ij}x_{ij},$$

$$\text{subject to} \quad \sum_{(i,j)\in A} x_{ij} - \sum_{(j,i)\in A} x_{ji} = 0, \quad i \in N, \tag{1}$$

$$d_{ij} \le x_{ij} \le k_{ij}, \quad (i, j) \in A. \tag{2}$$

In problem (P), x_{ij} is the flow along arc (i, j), c_{ij} is the unit cost of this flow, and d_{ij} and k_{ij} are lower and upper bounds on this flow, possibly with $k_{ij} = \infty$. The flow into each node equals the flow out of it, and therefore (P) is a circulation problem. The dual to (P) can be written as follows:

(D)
$$\text{Maximize} \quad \sum_{(i,j)\in A} \{d_{ij}(V_{ij})^+ + k_{ij}(V_{ij})^-\},$$

$$\text{subject to} \quad U_i - U_j + V_{ij} = c_{ij}, \quad (i, j) \in A, \tag{3}$$

$$U_i, V_{ij} \text{ unrestricted}, \quad i \in N, (i, j) \in A. \tag{4}$$

In (D) and throughout the paper $(x)^+ = \max\{0, x\}$ and $(x)^- = \min\{0, x\}$.

* This paper is part of the author's doctoral dissertation submitted at Yale University.

Numerous computational and theoretical works developed algorithms for solving the minimal cost network flow problem (cf. [1–6, 8–13]). Recently, Jensen and Barnes [9] classified minimum cost network algorithms. They define four classes; primal, dual-node-infeasible, dual-arc-infeasible, and primal–dual. In this paper we investigate the essential features of these algorithms in order to find the relations between them, and to evaluate the advantages each one has over the other. We divide the most common algorithms into two classes, the primal and the dual algorithms, each class consisting of different variants of a common general algorithm. Further we show that the so called primal–dual algorithm can be carried out in the same manner as a pure dual algorithm by using only dual variables and a slight modification of the well-known version of this algorithm. Finally, we show that this modification presents two valuable properties: the slope of the change in the (dual) objective is monotonous, and the bound on the maximum number of iterations is better (at least in some special cases) than the bounds for other algorithms.

We note that in [5] and [6] the superiority of the primal simplex procedure for the minimum cost problem was attested. However, it is important to have improved dual algorithms, as these methods are useful in some situations such as sensitivity analysis.

1. Notation and terminology

A *path* in (N, A) is a sequence $(\alpha_1, \dots, \alpha_n)$ of n $(n \geq 1)$ arcs having, for $m = 1, \dots, n$, arc $\alpha_m \in A$ and either $\alpha_m = (i_m, i_{m+1})$ or $\alpha_m = (i_{m+1}, i_m)$. This path is a *cycle* if $i_1 = i_{n+1}$. Arc α_m in this path has *positive* orientation if $\alpha_m = (i_m, i_{m+1})$ and *negative* orientation if $\alpha_m = (i_{m+1}, i_m)$. A cycle is a *directed* cycle if all its arcs have positive orientation. If arc (i, j) has cost c_{ij}, then the *cost* of this path is the sum of the costs of its positively oriented arcs less the sum of the costs of its negatively oriented arcs.

A *subgraph* (M, B) of (N, A) has $\emptyset \neq M \subseteq N$, $B \subseteq M \times M$ and $B \subseteq A$. (We allow a subgraph to have no arcs.) A subgraph (M, B) of (N, A) is *connected* if (M, B) contains a path from each node $i \in M$ to each node $j \in M$ having $j \neq i$. A subgraph is called a *tree* if it is connected and if it has no cylces. A set of node-disjoint trees is called a *forest*.

For sets $S \subseteq N$, $T \subseteq N$, $B \subseteq A$ and a function f, we have the following definitions:

$$S' = \{i \in N: i \notin S\},$$

$$(S, T) = \{(i, j) \in A: i \in S, j \in T\},$$

$$f(B) = \sum_{(i,j) \in B} f_{ij}.$$

2. Classification of algorithms

The most common algorithms for solving network flow problems can be classified to 'primal' and 'dual' algorithms, according to the method by which they improve solutions and the optimality criterion they use. We note that some algorithms which use both primal and dual variables may belong to both classes.

Primal Algorithms. For a given feasible circulation x_{ij}, define the modified network with the set of arcs A^m and the costs c^m as follows:

$$(i, j) \in A^m \text{ and } c^m_{ij} = c_{ij} \quad \text{if } (i, j) \in A \text{ and } x_{ij} < k_{ij}.$$
$$(i, j) \in A^m \text{ and } c^m_{ij} = -c_{ji} \quad \text{if } (j, i) \in A \text{ and } x_{ji} > d_{ji}.$$

To improve the solution, find in the modified network a directed cycle with negative cost (i.e., a *negative cycle*) and increase all the flow values of its arcs by the same amount until some x_{ij} not previously equal to one of its bounds becomes equal to it (and a modified cost changes). The following theorem gives a necessary and sufficient condition for the termination of the algorithm.

Theorem 1 (Busacker and Saaty [2]). *A feasible solution to* (P) *is optimal if and only if the modified network* (N, A^m) *has no negative cycles.*

Dual Algorithms. In our study of (D), we call U_i the *potential* of node i, and V_{ij} the *reduced cost* of arc (i, j). Each reduced cost V_{ij} appears in exactly one dual constraint. The potentials U_j and the reduced cost V_{ij} are not restricted in sign. Consequently, every set $\{U_j\}$ of potentials is dual feasible, since (3) is satisfied by taking $V_{ij} = c_{ij} - U_i + U_j$.

For a given feasible set of reduced costs V_{ij}, define the modified network (N, A) with upper (b_{ij}) and lower (a_{ij}) bounds as follows:

$$(a_{ij}, b_{ij}) = \begin{cases} (d_{ij}, d_{ij}) & \text{if } V_{ij} > 0, \\ (d_{ij}, k_{ij}) & \text{if } V_{ij} = 0, \\ (k_{ij}, k_{ij}) & \text{if } V_{ij} < 0. \end{cases}$$

Find a set $M \subset N$ with $I(M) > 0$, where

$$I(M) = a(M', M) - b(M, M'), \tag{5}$$

and increase all potentials in the set M by the same amount ϵ, until some V_{ij} previously not 0 becomes 0 (and a modified bound changes).

The effect of this step is to increase V_{ij} by ϵ for $(i, j) \in (M', M)$ and to decrease V_{ij} by ϵ for $(i, j) \in (M, M')$. It is an elementary matter to check that for

$0 \leq \epsilon \leq T(M)$ where

$$T(M) = \min \begin{cases} \infty, \\ \{-V_{ij}: (i, j) \in (M', M),\ V_{ij} < 0\}, \\ \{V_{ij}: (i, j) \in (M, M'),\ V_{ij} > 0\}, \end{cases} \tag{6}$$

the change in the objective of (D) is $I(M) \cdot \epsilon$. Since $I(M) > 0$ we obtained a new feasible solution with a greater objective value. Therefore we call M an *improving set*. If $T(M) = \infty$, then (D) is feasible and unbounded, which indicates that (P) is infeasible. The following theorem gives a necessary and sufficient condition for the termination of the algorithm.

Theorem 2. *A feasible solution to* (D) *is optimal if and only if the modified network has no improving sets.*

Proof. The condition is trivially necessary. To prove sufficiency, we use 'Hoffman's Existence Theorem for Circulations' [7; 13, p. 268]: A feasible circulation exists in a network (N, A) if and only if $k(M, M') \geq d(M', M)$ for every $M \subseteq N$.

Suppose no improving sets exist. By Hoffman's Theorem, there exists a feasible circulation with respect to the modified bounds. That is,

$$\begin{aligned} x_{ij} &= d_{ij} && \text{for } V_{ij} > 0, \\ d_{ij} \leq x_{ij} &\leq k_{ij} && \text{for } V_{ij} = 0, \\ x_{ij} &= k_{ij} && \text{for } V_{ij} < 0. \end{aligned}$$

By 'complementary slackness' this circulation is an optimal solution to (P) and the set of reduced costs constitutes an optimal solution to (D).

The most important part of the algorithm is the search for improving sets, and it is here that the various algorithms differ.

3. Some existing dual algorithms

In this section we demonstrate how some of the most common existing algorithms fit into the class of dual algorithms described in Section 2.

3.1. The out of kilter algorithm

The *deviation* of an arc is defined as:

$$\begin{array}{ll} d_{ij} - x_{ij} & \text{if } V_{ij} \geq 0,\ x_{ij} < d_{ij}, \\ x_{ij} - d_{ij} & \text{if } V_{ij} > 0,\ x_{ij} > d_{ij}, \\ x_{ij} - k_{ij} & \text{if } V_{ij} \leq 0,\ x_{ij} > k_{ij}, \\ k_{ij} - x_{ij} & \text{if } V_{ij} < 0,\ x_{ij} < k_{ij}, \text{ and} \\ \text{zero} & \text{otherwise.} \end{array}$$

Improving sets are chosen as follows:

Step 1: Arbitrarily choose an arc with a positive deviation.

Step 2: By any flow algorithm (e.g. the labeling algorithm), try to find a cycle which includes this arc, such that, by increasing all flows in arcs oriented in one way, and decreasing flows in arcs of opposite orientation, no deviation is increased (and the deviation of the original arc is decreased).

Step 3: By repeating Step 2, attempt to decrease the deviation of the arc to zero. If the attempt succeeds, go to Step 1. Else, let M be the set of labeled nodes; then $I(M)>0$.

The last assertion requires proof. If (i,j) is an arc for which the origin i is labeled, and the extremity j cannot be labeled, then either

$$V_{ij}>0 \text{ and } x_{ij}\geq d_{ij}, \quad \text{or} \quad V_{ij}\leq 0 \text{ and } x_{ij}\geq k_{ij}.$$

If (i,j) is an arc for which the extremity j is labeled, and the origin i is not labeled, then either $V_{ij}>0$ and $x_{ij}\leq d_{ij}$, or $V_{ij}<0$ and $x_{ij}\leq k_{ij}$.

Since x is a circulation,

$$\begin{aligned} b(M,M') &= d(\{(i,j)\in(M,M')\colon V_{ij}>0\})+k(\{(i,j)\in(M,M')\colon V_{ij}\leq 0\}) \\ &\leq x(M,M')=x(M',M) \\ &\leq d(\{(i,j)\in(M',M)\colon V_{ij}\geq 0\}) \\ &\quad + k(\{(i,j)\in(M',M)\colon V_{ij}<0\})=a(M',M). \end{aligned}$$

However, since at least one arc in $(M,M')\cup(M',M)$ has positive deviation (by construction), one of the inequalities is strict and, $b(M,M')<a(M',M)$ or, $I(M)>0$.

3.2. *The dual simplex algorithm*

The dual simplex algorithm maintains a circulation and a spanning tree T with $V=0$, such that all arcs not in the tree satisfy complementary slackness: $x_{ij}=d_{ij}$ if $V_{ij}>0$, $x_{ij}=k_{ij}$ if $V_{ij}<0$. For simplicity we assume that the current dual solution is nondegenerate. In this case, all arcs not in T have non-zero reduced costs. The algorithm chooses the arc $(m,n)\in A$ which has the maximum deviation from the feasible region ($d_{ij}-x_{ij}$ for $x_{ij}<d_{ij}$, $x_{ij}-k_{ij}$ for $x_{ij}>k_{ij}$). The tree is cut at this arc and its components are M, M'. One of them is an improving set. Its potentials are increased until a new tree is obtained. Then flow is sent through the unique path of T connecting the end nodes of the arc which blocks the change of potentials, to create a new primal solution.

To see that either $I(M)>0$ or $I(M')>0$, suppose for example that $(m,n)\in$

(M, M'). If $x_{mm} > k_{mm}$, then

$$\begin{aligned} I(M) &= a(M', M) - b(M, M') \\ &= [d(\{(i,j) \in (M', M)\colon V_{ij} > 0\}) + k(\{(i,j) \in (M', M)\colon V_{ij} < 0\})] \\ &\quad - [d(\{(i,j) \in (M, M')\colon V_{ij} > 0\}) \\ &\quad + k(\{(i,j) \in (M, M')\colon V_{ij} \le 0\}) + k_{mn}] \\ &> x(M', M) - x(M, M') = 0. \end{aligned}$$

Similarily, if $x_{mn} < d_{mn}$, then $I(M') > 0$.

3.3. *The primal–dual algorithm*

For each set of reduced costs, a flow is constructed such that complementary slackness holds, and the sum of 'node infeasibilities' is minimum. This requires solving the following 'restricted' primal problem:

$$\begin{aligned} \text{minimize} \quad & \sum_{i \in N} (Y_i^+ + Y_i^-), \\ \text{subject to} \quad & x(i, N) - x(N, i) + Y_i^+ - Y_i^- = 0, && i \in N, \\ & x_{ij} = k_{ij}, && V_{ij} < 0, \\ & x_{ij} = d_{ij}, && V_{ij} > 0, \\ & d_{ij} \le x_{ij} \le k_{ij}, && V_{ij} = 0, \\ & Y_i^+, Y_i^- \ge 0. \end{aligned}$$

If the solution equals zero, then the flow values x constitute a feasible circulation and complementary slackness holds. Hence, this circulation is optimal. Otherwise, let U be the corresponding optimal dual solution. Then the set of nodes i for which $U_i > 0$ is an improving set. In fact, for every vertex of the dual polyhedron $U_i \in \{+1, -1\}$, and the set $\{i\colon U_i = +1\}$ is an improving set.

4. A tree-search algorithm

The utility of Theorem 2 is enhanced if one finds efficient ways to determine improving sets. So far we described some common methods that perform the search. We describe below a more direct algorithm which is a modified version of the primal–dual algorithm. This algorithm solves the 'restricted' dual problem directly by using explicitly only dual variables.

Network (N, A) is said to have *independent costs* if it contains no simple cycle whose cost is 0. It is always possible to perturb the c_{ij}'s so that a network has this property [3, p. 231]. To simplify the exposition, we assume throughout that network (N, A) has independent costs. We note, however, the algorithm described in this section can be executed also with costs which are not independent. In this case, when some reduced costs become simultaneously zero, all except for one are assumed to retain their sign. Consequently it may happen that $T(m)=0$ in equation (6).

Consider any feasible solution to (D), and set

$$F=\{(i,j)\colon V_{ij}=0,\ d_{ij}<k_{ij}\},$$

so that F is the set of arcs whose reduced costs are (currently) 0. Suppose F contained a simple cycle. Sum (3) to see that the cost of this cycle equals 0. But this contradicts our assumption of independent costs. Hence, F has no cycle. Consequently, (N, F) is a forest.

Call M a *good* improving set if $I(M)>0$ and if $I(M)>I(S)$ for every proper subset S of M. Call M the *best* improving set if M is the unique good improving set such that $I(M)\geq I(S)$ for every $S\subseteq N$.

The tree-search algorithm locates the best improving set:

Step 1: Set $f(i)=0$ for each $i\in N$. Set $F=\{(i,j)\in A\colon V_{ij}=0,\ d_{ij}<k_{ij}\}$.

Step 2: For each $(i,j)\in F$, set $f(i)\leftarrow[f(i)+1]$ and $f(j)\leftarrow[f(j)+1]$. (Step 2 initializes $f(i)$ to the number of arcs in F which are incident with node i.)

Step 3: Set $S=N$, $M=\emptyset$ and $P(i)=\{i\}$, $J(i)=I(\{i\})$ for each $i\in N$. (This procedure is intended to terminate with $S=\emptyset$ and $M=$ best improving set. At each stage $J(i)$ is the 'effective' improvement for node i, and $P(i)$ is a set of nodes which belong to the best improving set if and only if node i belongs to the set.)

Step 4: Stop if $S=\emptyset$. Else find $i\in S$ such that $f(i)\leq 1$. Set $S\leftarrow[S-\{i\}]$.

Step 5: If $J(i)\leq 0$, go to Step 8. Else set $M\leftarrow[M\cup P(i)]$.

Step 6: If $f(i)=0$, go to Step 4. Else find the unique j such that either $(i,j)\in F$ or $(j,i)\in F$.

Step 7: Set $f(j)\leftarrow[f(j)-1]$. If $(i,j)\in F$, set $J(j)\leftarrow[J(j)+b_{ij}-a_{ij}]$, $F\leftarrow[F-\{(i,j)\}]$. Else set $J(j)\leftarrow[J(j)+b_{ji}-a_{ji}]$, $F\leftarrow[F-\{(j,i)\}]$. Go to Step 4. (The new value of $J(j)$ represents the change in (5) caused by joining j to M, given $i\in M$.)

Step 8: If $f(i)=0$, go to Step 4. Else find the unique j such that either $(i,j)\in F$ or $(j,i)\in F$. Set $f(j)\leftarrow[f(j)-1]$. If $(i,j)\in F$ set $\alpha\leftarrow(i,j)$, and if $(j,i)\in F$ set $\alpha\leftarrow(j,i)$. Set $F\leftarrow F-\alpha$. If $J(i)+b_{ij}-a_{ij}\leq 0$ and $\alpha=(i,j)$ or if $J(i)+b_{ji}-a_{ji}\leq 0$ and $\alpha=(j,i)$ go to Step 4.

Step 9: Set $P(j)\leftarrow[P(j)\cup P(i)]$. If $\alpha=(i,j)$ set $J(j)\leftarrow[J(j)+J(i)+b_{ij}-a_{ij}]$. Else set $J(j)\leftarrow[J(j)+J(i)+b_{ji}-a_{ji}]$. Go to Step 4.

Theorem 3. *The tree-search algorithm finds the best improving set.*

Proof. Denote by T the best improving set. Suppose that at a certain stage of the algorithm

(i) $P(i) \subseteq T$ or $P(i) \subseteq T'$, for every $i \in N$;

(ii) $M \subseteq T$, $N - S - M \subseteq T'$.

Note that since $i \in P(i)$, assumption (i) implies that for every $i \in N$:

(iii) $P(i) \subseteq T$ if, and only if, $i \in T$.

These assumptions clearly hold when Step 3 is executed, and we must show that they still hold after each execution of Steps 5 and 9.

In Step 5, if $J(i) > 0$, then joining i to any set which contains M increases the value of (5) of this set. Since by (ii) $M \subseteq T$ also $i \in T$, and by (iii) $P(i) \subseteq T$. If $J(i) \leq 0$ the value of (5) will not increase and $P(i) \subseteq T'$. We conclude that (ii) is preserved in Step 5.

In Step 9, if either $J(i) + b_{ij} - a_{ij} > 0$ and $(i, j) \in F$ or $J(i) + b_{ji} - a_{ji} > 0$ and $(j, i) \in F$, then joining i to any set containing $P(j)$ increases the value of (5) for this set. In any other case the value of (5) decreases. Together with (iii) this implies that (i) is preserved by this step.

Since (ii) holds until the algorithm terminates with $s = \emptyset$, the final set M satisfies $M \subseteq T$ and $M' \subseteq T'$ so that $M = T$.

An alternative search policy is to apply the tree-search algorithm until any (good but not necessarily the best) improving set is found, and then to change its potentials. This policy requires less computations in each iteration. However, when best improving sets are found, some theorems, including a bound on the number of iterations, can be proved. We now state and prove these theorems.

Theorem 4. *Let M_r be the best improving set at iteration r. Let $I(M, r)$ be the value of $I(M)$ at iteration r. Then $I(M_r, r)$ is nonincreasing in r.*

Proof. Suppose, for example that only arc $(i, j) \in (M_r, M_r')$ blocks the change of potentials at iteration r. (The same proof holds when more than one arc blocks the change.) The only decrease in $I(M_r)$ is caused by b_{ij} which was changed from d_{ij} to k_{ij}. (If $(j, i) \in (M_r', M_r)$, then the decrease is caused only by a_{ji} which was changed from k_{ji} to d_{ji}.)

(a) Suppose $i \in M_{r+1}$, then $M_{r+1} \supseteq M_r$.

If $M_{r+1} = M_r$, then

$$I(M_{r+1}, r+1) = I(M_r, r) + a_{ij} - b_{ij} < I(M_r, r).$$

If $M_{r+1} \neq M_r$, then $M_{r+1} \supset M_r$, $j \in M_{r+1}$ and

$$I(M_{r+1}, r+1) = I(M_{r+1}, r) \leq I(M_r, r).$$

(b) Suppose $i \not\subseteq M_{r+1}$, then $M_{r+1} \subset M_r$ and since M_r is a good improving set $I(M_{r+1}, r+1) = I(M_{r+1}, r) < I(M_r, r)$.

Note that by perturbation it is possible to ensure that only one arc blocks the change of potentials. In this case either $M_{r+1} \subseteq M_r$ or $M_{r+1} \supseteq M_r$ in each iteration.

Corollary 1. *If $I(M_r, r) = I(M_{r+1}, r+1)$, then $M_{r+1} \supset M_r$.*

Corollary 2. *The same value of $I(M_r, r)$ cannot recur more than $|N| - 1$ times.*

Corollary 3. *Suppose all bounds are integers, then no more than $|N| \cdot I(M_1, 1)$ iterations are needed to find an optimal solution.*

Since the order of work needed in each iteration is $|N|^2$, the bound is of order $|N|^3 \cdot I(M_1, 1)$. This bound is better than the bound for the out-of-kilter method which is $|N|^3$ times the sum of initial primal infeasibilities (cf. [12]).

Corollary 4. *The direction of change in the reduced cost of any arc cannot be opposite in two successive iterations.*

Corollary 5. *If the same arc blocks the change of potentials in iterations m and n, $m < n$, then $m \leq n - 3$.*

Proof for Corollary 5. Suppose that (i, j) blocks in iteration m. In iteration $m + 1$, V_{ij} may become nonzero. In iteration $m + 2$ the direction of change in V_{ij} cannot be opposite. Only in iteration $m + 3$, (i, j) can block again.

Theorem 5. *The algorithm converges to an optimal solution in a finite number of iterations.*

Proof. $I(M)$ is equal to a linear combination $\sum \epsilon_{ij} Z_{ij}$, where $\epsilon_{ij} \in \{0, 1, -1\}$, $Z_{ij} \in \{d_{ij}, k_{ij}\}$. Hence there is only a finite set of possible values of $I(M)$ for M = the best improving set. By Corollary 2 of Theorem 3, the algorithm terminates after a finite number of iterations.

Theorem 6. *Let M_r, $I(M, r)$ be as in Theorem 3. Then $\bigcap_r M_r \neq \emptyset$. (In other words, there exists at least one node which is included in all best improving sets.)*

Proof. Suppose the assertion is false. Then there exist r and M such that $I(M, r) > 0$, $I(S, r) \leq 0$ for all $S \subset M$ (i.e. M is a *minimal improving set* in iteration r) and $(\bigcap_{t=1}^{r-1} M_t) \cap M = \emptyset$.

Let $s = \max\{t\colon t < r$, M is not a minimal improving set in iteration $t\}$. Then, either (a) or (b) holds:

(a) For all $S \subseteq M$, $I(S, s) \leq 0$.

If $M_s \cap M = \emptyset$, then M_s is not the *best* improving set (since if we join M to M_s we obtain a better set).

If $M_s \cap M = M_s$, then $I(M, s) \geq I(M, s + 1)$. But this is a contradiction since we assumed $I(M, s + 1) > 0 \geq I(M, s)$.

Therefore, $M \cap M_s \neq \emptyset$ and $M \cap M'_s \neq \emptyset$ is incident to the blocking arc in iteration s. Since M is a *minimal* improving set in iteration $s + 1$, $I(M \cap M_s, s) =$

$I(M \cap M_s, s+1) \le 0$. But $I(M, s+1) > 0$, hence by joining $M \cap M_s'$ to M_s we obtain a better set. Again, this is a contradiction.

(b) There exists $S \subset M$ such that $I(S, s) > 0$. Replace M by S, r by s and restart the procedure.

Since $|N|$ is finite, the process must result in a contradiction. Therefore there exists at least one node included in all best improving sets.

Corollary. *If $c_{ij} \neq 0$ for all $(i, j) \in A$, then exactly one node is contained in all the best improving sets.*

Similarily the following theorem can be proved:

Theorem 7. *There exists at least one node, which is not included in any of the best improving sets.*

Note that the tree search algorithm and the related theorems can be devised similarly when a best improving set is defined to be the set M such that $I(M) \ge I(S)$ for all $S \subseteq N$ and if $I(S) = I(M)$, then $S \subset M$.

5. Example and some comments

There need not be a unique sequence of iterations when the primal–dual algorithm is applied in its common form. Uniqueness may be achieved when the capacities are perturbed. But in this case, the integrality of data and the bound of Corollary 3 of Theorem 3, are lost. However, for the cost perturbed problem when the tree search version is applied the sequence of iterations is unique.

The following theorem and example illustrate that at least in the special case presented there the use of the tree search version proves to be beneficial.

Theorem 8. *A problem with four nodes and no more than one arc connecting each pair of nodes, is always solved by no more than eight iterations.*

Proof. Suppose node 1 is in all sets, and node 4 is not included in any of them. Then only the arc connecting nodes 2 and 3 may block more than once. By Corollary 5 to Theorem 3, between any two iterations in which this arc blocks, there must be at least two other iterations in which the other five arcs block. Hence this arc blocks at most three times.

The example presented in Fig. 1 is solved by the primal–dual algorithm. Improving sets are marked by small circles. Nine improving sets are found and each of them strictly improves the dual objective. This is more than the bound given by Theorem 8. Note that the upper arc blocks four times. In the first five iterations, best improving sets are found. But, in the next three iterations, the

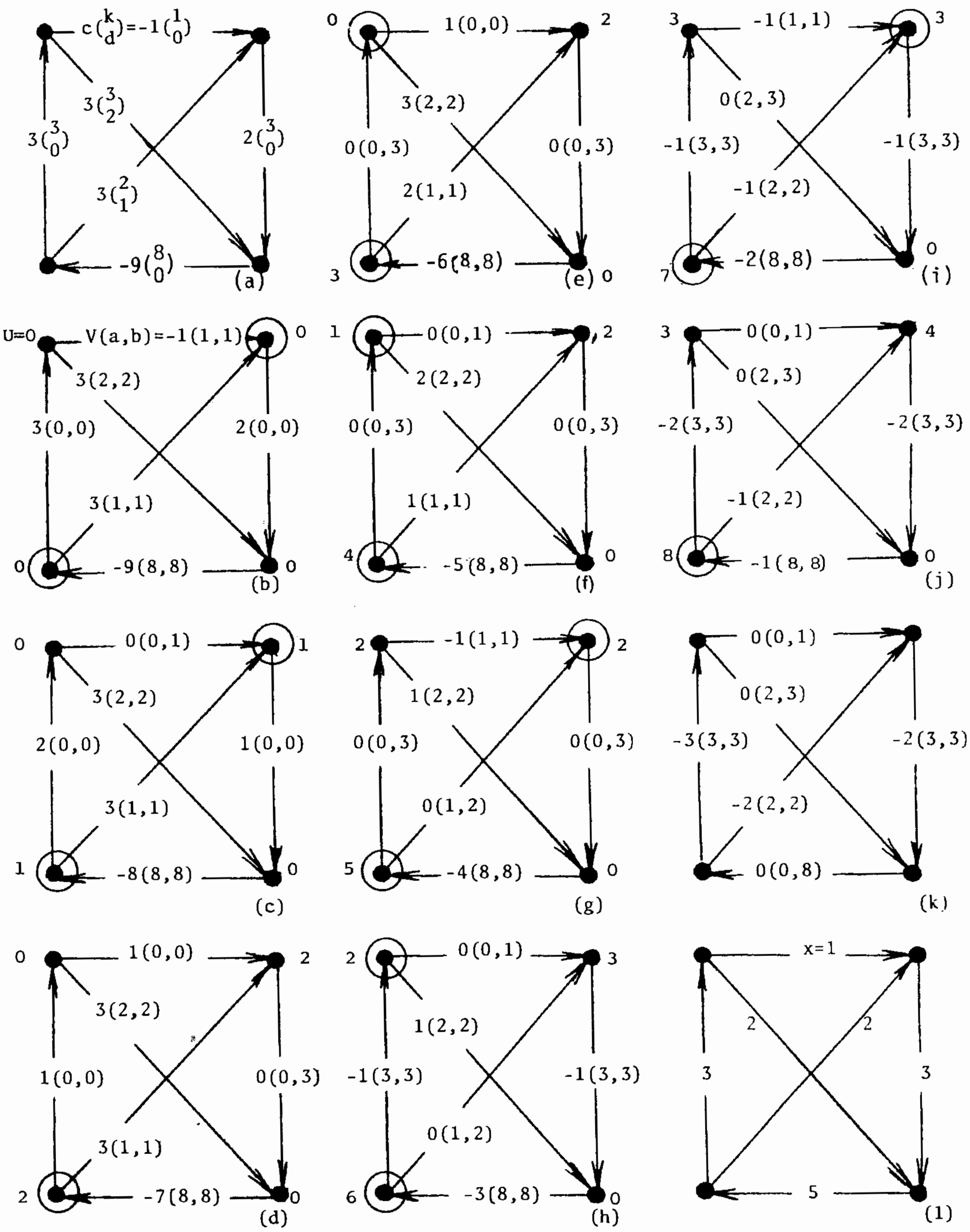

Fig. 1. The primal–dual algorithm.

best set consists of only the left lower node. Thus, Corollary 5 to Theorem 3 does not hold for the upper arc.

Note, however, that the number of iterations in the primal–dual method, even in its modified form does not have a polynomial upper bound, relative to $|N|$. This can be observed by using it to solve Zadeh's 'bad' problems [17]. These are simple transportation problems with n sources and n sinks, which require $2^n + 2^{n-2} - 2$ iterations.

There is no unique sequence of iterations, when applying the out-of-kilter algorithm. In this algorithm as opposed to the primal–dual algorithm, it is possible that the removal of certain nodes from the improving set increases its unit-improvement, $I(\cdot)$. Another possible dual algorithm consists of the following steps: (1) apply the tree-search procedure until any good (not necessarily best) improving set is determined; (2) change potentials. In this algorithm the removal of a node from the set decreases its unit-improvement (in fact the set ceases to be an improving set). This algorithm consists of shorter iterations but may lead to long sequences of improving sets before reaching the optimal solution.

Acknowledgements

My deepest gratitude to Professor A.J. Hoffman, my thesis director, for his support, guidance and inspiring enthusiasm for the subject.

References

[1] M.S. Bazarraa and J.J. Jarvis, *Linear programming and network flows* (Wiley, New York, 1977).
[2] R.G. Busacker and T.L. Saaty, *Finite graphs and networks: An introduction with applications* (McGraw Hill, New York, 1965).
[3] G.B. Dantzig, *Linear programming and extensions* (Princeton University Press, Princeton, NJ, 1963).
[4] D.R. Fulkerson, "An Out-of-Kilter Method for minimal cost flow problems", *Journal of the Society of Industrial and Applied Mathematics* 9 (1961) 18–27.
[5] F. Glover, D. Karney and D. Klingman, "Implementation and computational comparisons of primal, dual and primal–dual computer codes for minimum cost network flow problems", *Networks* 4 (1974) 191–212.
[6] F. Glover, D. Karney, D. Klingman and A. Napier, "A computational study of start procedures, basis change criteria and solution algorithms for transportation problems", *Management Science* 20 (1974) 793–819.
[7] A.J. Hoffman, "Some recent applications of the theory of linear inequalities to extremal combinatiorial analysis", in: R. Bellman and M. Hall Jr. (eds.), *Proceedings of symposia in applied mathematics*, Vol. X, *Combinatorial analysis* (American Mathematical Society, Providence, RI, 1960) 113–127.
[8] T.C. Hu, *Integer programming and network flows* (Addison-Wesley, Reading, MA, 1969).
[9] P.A. Jensen and J.W. Barnes, *Network flow programming* (Wiley, New York, 1980).
[10] E.L. Johnson, Chapter II-4, in: S.E. Elmaghraby and J.J. Moder (eds.), *Handbook of operations research, foundations and fundamentals* (Van Nostrand Reinhold, New York, 1978).
[11] M. Klein, "A primal method for minimal cost flows", *Management Science* 14 (1967) 205–220.
[12] J.F. Shapiro, "A note on the primal–dual and out-of-kilter algorithms for network optimization problems", *Networks* 7 (1977) 81–88.
[13] M. Simonnard, *Linear programming* (Prentice-Hall, Englewood Cliffs, NJ, 1966).
[14] N. Zadeh, "A bad network problem for the simplex method and other minimum cost-flow algorithms", *Mathematical Progrramming* 5 (1973) 255–266.