Operations Research Letters 12 (1992) 227–233
North-Holland

October 1992

Algorithms for the minimum cost circulation problem based on maximizing the mean improvement

Refael Hassin

School of Mathematical Sciences, Department of Statistics, Tel Aviv University, Tel Aviv 69978, Israel

Received March 1990
Revised February 1992

Several recent polynomial algorithms for the minimum cost circulation problem have the following in common: The solution, primal or dual, is changed in a way that the mean improvement of the objective function with respect to some measure is maximized. This note contains some new insight on such algorithms. In addition, it is shown that a dual algorithm which selects node-wise maximum mean cuts, is not polynomially bounded.

network flow; minimum cost circulation; dual algorithm

1. Introduction

Let $G=(N, A)$ be a directed graph. The minimum cost circulation problem is to

$$\text{Minimize} \sum_{(i,j)\in A} c_{ij}x_{ij} \tag{1.1}$$

subject to

$$\sum_{j|(i,j)\in A} x_{ij} - \sum_{j|(j,i)\in A} x_{ji} = 0, \quad i \in N,$$

$$l_{ij} \le x_{ij} \le u_{ij}, \quad (i,j)\in A.$$

In (1.1), x_{ij} is the flow along arc (i, j), c_{ij} is the unit cost of this flow, and l_{ij} and u_{ij} are lower and upper bounds on this flow. We refer to x in (1.1) as a *circulation*. The dual problem can be written as follows:

$$\text{Maximize} \sum_{(i,j)\in A} \left\{ l_{ij}(v_{ij})^{+} + u_{ij}(v_{ij})^{-} \right\}, \tag{1.2}$$

subject to

$$p_i - p_j + v_{ij} = c_{ij}, \quad (i, j)\in A,$$

where $(v)^+ = \max\{0, v\}$ and $(v)^- = \min\{0, v\}$. We refer to $p=(p_j)$ in (1.2) as a *price* vector.

Several recent polynomial algorithms for the minimum cost circulation problem have the following in common: The solution, primal or dual, is changed in a way that the mean improvement of the objective function with respect to some measure is maximized. This note contains some new insight on such algorithms. In addition, it is shown that a dual algorithm which selects node-wise maximum mean cuts, is not polynomially bounded.

2. The primal approach

Consider a given circulation $x \in R^{|A|}$. For simplicity, we assume in this section that $(i, j)\in A$ implies $(j, i)\notin A$. Let $A_+^x = \{(i, j)|(i, j)\in A,\ x_{ij} < u_{ij}\}$. Let $A_-^x = \{(i, j)|(j, i)\in A,\ x_{ji} > l_{ji}\}$. The *complementary slackness* theorem states that a necessary and sufficient condition for the opti-

Correspondence to: Refael Hassin, School of Mathematical Sciences, Department of Statistics, Tel Aviv University, Tel Aviv 69978, Israel.

mality of x is that there exists a price vector $p \in R^{|N|}$ satisfying

$$c_{ij} - p_i + p_j \geq 0, \quad (i, j) \in A_+^x,$$

$$c_{ij} - p_i + p_j \leq 0, \quad (j, i) \in A_-^x.$$

Consider the graph $G^x = (N, A^x)$, where $A^x = A_+^x \cup A_-^x$. Define (modified) arc-costs on A^x:

$$c_{ij}^x = c_{ij}, \quad (i, j) \in A_+^x,$$

$$c_{ij}^x = -c_{ji}, \quad (i, j) \in A_-^x. \tag{2.1}$$

Using the modified costs, the complementary slackness condition can be written as

$$c_{ij}^x - p_i + p_j \geq 0, \quad (i, j) \in A^x. \tag{2.2}$$

The *cost* of a (directed) cycle is the sum of its arc-costs. A cycle is *negative* if its cost is negative. The following classical theorem gives necessary and sufficient optimality conditions:

Theorem 2.1 (Busacker and Saaty, 1965). *A feasible solution to* (1.1) *is optimal if and only if G^x contains no negative cycles with respect to the modified costs c^x.*

Theorem 2.1 suggests a general approach for solving (1.1). At each iteration a negative cycle is computed and the flow of each of its arcs is increased by the same amount until some modified cost changes as a result of a change in the sets A_+^x, A_-^x, (see Klein, 1967). This approach is general in the sense that it leaves much freedom in the negative cycle selection part. An example by Zadeh (1973) proves that

Theorem 2.2. *The negative cycle algorithm may have an exponential number of iterations even if a most negative cycle is selected at each iteration.*

In view of Theorem 2.2, the question of what rules for selecting negative cycles yield polynomial algorithms is of interest. Let the *marginal cost* of a cycle be its cost times its (residual) capacity. Barahona and Tardos (1989) showed that if a cycle with the most negative marginal cost (a *most helpful cycle* in terms of Ervolina and McCormick 1990b) is chosen then the number of iterations is polynomially bounded. Since finding this cycle is NP-hard, they suggested a minor variation based on an algorithm by Weintraub (1974), which is polynomial. Goldberg and Tarjan (1989) found another rule which we discuss below.

For a feasible x define $\varepsilon(x)$ as follows:

$$\varepsilon(x) = \max\ z \tag{2.3}$$

subject to

$$c_{ij}^x - p_i + p_j \geq z, \quad (i, j) \in A^x,$$

where p_i $i \in N$ are unrestricted in sign.

Clearly, x is optimal if and only if $\varepsilon(x) \geq 0$. The *mean cost* of a cycle is its cost divided by the number of its arcs. Let $\mu(x)$ denote the minimum mean cost (with respect to the modified costs) of a cycle in G^x.

Theorem 2.3 (Engel and Schneider, 1975; Goldberg and Tarjan, 1989).

$$\varepsilon(x) = \mu(x).$$

We present a proof for Theorem 2.3, using linear programming duality. This proof will be used to motivate the analysis that will be developed later.

Proof. The dual to (2.3) is

$$\min \sum_{(i,j) \in A^x} c_{ij}^x f_{ij} \tag{2.4}$$

subject to

$$\sum_{j|(i,j) \in A^x} f_{ij} - \sum_{j|(j,i) \in A^x} f_{ji} = 0, \quad \forall i \in N,$$

$$\sum_{(i,j) \in A^x} f_{ij} = 1,$$

$$f \geq 0.$$

Since f is a circulation in G^x, it is composed of sum of flows in simple (directed) cycles in G^x. Let Γ denote the set of simple cycles in G^x. Let f_γ denote the flow value attached to a cycle γ. The cycle contributes in total $|\gamma| f_\gamma$ to $\Sigma_{(i,j) \in A^x} f_{ij} = 1$. Let $c_\gamma = \Sigma_{(i,j) \in \gamma} c_{ij}$. Then (2.4) is equivalent to

$$\min \sum_{\gamma \in \Gamma} c_\gamma f_\gamma \tag{2.5}$$

subject to

$$\sum_{\gamma \in \Gamma} |\gamma| f_\gamma = 1,$$

$$f \geq 0.$$

Program (2.5) is a simple continuous knapsack problem, and it is solved by setting $f_\gamma = 1$ for a cycle of minimum mean length, while $f_\gamma = 0$ for all of the other cycles. Thus (2.4) is, as already observed by Dantzig, Blattner and Rao (1967), a linear programming formulation for the minimum mean cycle problem. Theorem 2.3 follows immediately from this observation through the duality theorem of linear programming. □

Theorem 2.3 motivates the application of a minimum mean cycle selection rule in Klein's algorithm:

Theorem 2.4 (Goldberg and Tarjan, 1989). *The minimum mean cycle selection algorithm is (strongly) polynomial.*

3. The dual approach

We now describe some dual analogues to the ideas described in Section 2. Consider a given price vector $p \in R^{|N|}$. Let $c_{ij}^p = c_{ij} - p_i + p_j$. Let

$$A_0^p = \{(i, j) \in A \mid c_{ij}^p = 0\},$$

$$A_+^p = \{(i, j) \in A \mid c_{ij}^p > 0\},$$

$$A_-^p = \{(i, j) \in A \mid c_{ij}^p < 0\}.$$

The *complementary slackness* theorem states that a necessary and sufficient condition for the optimality of p is that there exists a circulation $x \in R^{|A|}$ satisfying

$$x_{ij} = l_{ij}, \quad (i, j) \in A_+^p,$$

$$l_{ij} \le x_{ij} \le u_{ij} \quad (i, j) \in A_0^p,$$

$$x_{ij} = u_{ij}, \quad (i, j) \in A_-^p.$$

Define modified bounds on the arc flows as follows:

$$(l_{ij}^p, u_{ij}^p) = \begin{cases} (l_{ij}, l_{ij}), & (i, j) \in A_+^p, \\ (l_{ij}, u_{ij}), & (i, j) \in A_0^p, \\ (u_{ij}, u_{ij}), & (i, j) \in A_-^p. \end{cases} \tag{3.1}$$

Using the modified bounds, the complementary slackness conditions can be written as

$$l_{ij}^p \le x_{ij} \le u_{ij}^p, \quad (i, j) \in A. \tag{3.2}$$

A set $M \subset N$, $M \ne \emptyset$, is called a *cut*. With respect to a given price vector p we define the *surplus*, $S^p(M)$, of a cut, M, as

$$S^p(M) = \sum_{\substack{(i,j) \in A \\ i \notin M, j \in M}} l_{ij}^p - \sum_{\substack{(i,j) \in A \\ i \in M, j \notin M}} u_{ij}^p.$$

A cut is *positive* if its surplus is positive. The following theorem gives necessary and sufficient optimality conditions:

Theorem 3.1 (Hassin, 1983). *A price vector p is optimal for* (1.2) *if and only if G contains no positive cuts with respect to the modified bounds* (l^p, u^p).

Theorem 3.1 suggests a general approach for solving (1.2). At each iteration a positive cut is computed and the price of each of its nodes is increased by the same amount until some modified bound changes as a result of a sign change in the sets A^p. This approach is general in the sense that it leaves much freedom in the positive cut selection part. Some well known special cases are the Dual Simplex, Primal–Dual, and Out-of-Kilter Algorithms (see Hassin, 1983). An extension to submodular flows is described by Chung and Tcha (1991). An example of Zadeh (1973) proves that

Theorem 3.2. *The (general) positive cut algorithm may have an exponential number of iterations.*

In view of Theorem 3.2, the question of whether there exists a rule for selecting positive cuts that yields a polynomial algorithm is of interest. In contrast with the difficulty associated with computing a most negative cycle with respect to the modified costs, a most positive cut with respect to the modified bounds can be polynomially computed (see, Hassin, 1983, McCormick and Ervolina, 1990). However, it does not result in a polynomial algorithm (see, Hassin, 1983). On the other hand, an algorithm which selects a *most helpful cut* (a dual analogue of the most helpful cycle algorithm mentioned above) was shown by Ervolina and McCormick (1990b) to be (weakly) polynomial.

The similarity of Theorems 2.1 and 3.1, and the analysis related to them, suggest that an attractive approach may be to select a cut that is maximum in some average sense. This is the subject of the rest of this note, where we continue to develop results that are analogous to those of Section 2.

For a price vector $p \in R^{|N|}$ define $\delta(p)$:

$$\delta(p) = \min z \tag{3.3}$$

subject to

$$\sum_{j|(j,i)\in A} f_{ji} - \sum_{j|(i,j)\in A} f_{ij} \le z, \quad i \in N,$$

$$l_{ij}^p \le f_{ij} \le u_{ij}^p, \quad (i,j) \in A.$$

A necessary and sufficient condition for f to be a circulation is that $\delta(p) = 0$. By (3.2), this is also a necessary and sufficient condition for the optimality of p.

The *node-wise mean surplus* of a cut M is its surplus divided by $|M|$. Let $\nu(p)$ denote the node-wise maximum mean surplus (with respect to the modified bounds) of a cut in G.

Theorem 3.3. $\delta(p) = \nu(p)$.

Proof. The dual to (3.3) is

$$\max \sum_{(i,j)\in A} w_{ij} l_{ij}^p - \sum_{(i,j)\in A} w'_{ij} u_{ij}^p$$

subject to

$$q_i - q_j + w_{ij} - w'_{ij} = 0, \quad (i,j) \in A,$$

$$\sum_{i\in N} q_i = 1,$$

$$q, w, w' \ge 0.$$

This problem can be reformulated as follows:

$$\max \sum_{(i,j)\in A} (q_j - q_i)^+ l_{ij}^p - \sum_{(i,j)\in A} (q_i - q_j)^+ u_{ij}^p \tag{3.4}$$

subject to

$$\sum_{i\in N} q_i = 1,$$

$$q \ge 0.$$

Consider a solution q to (3.4). Without loss of generality, assume that the nodes are indexed so that $q_i \le q_j$ $i < j$, and let $q_0 = 0$. Let $A_v = \{i \in N \mid i \ge v\}$, $v \in N$. Note that $q_i = \sum_{j\le i}(q_j - q_{j-1})$. The constraint in (3.4) can be written as $\sum_{j\in N}(q_j - q_{j-1})|A_j| = 1$. Applying the definition of $S^p(M)$ to A_v we obtain

$$S^p(A_v) = \sum_{\substack{(i,j)\in A\\ i<v, j\ge v}} l_{ij}^p - \sum_{\substack{(i,j)\in A\\ i\ge v, j<v}} u_{ij}^p.$$

The objective function of (3.4) can be written as

$$\sum_{\substack{(i,j)\in A\\ i<j}} \sum_{v=i+1}^{j} (q_v - q_{v-1}) l_{ij}^p - \sum_{\substack{(i,j)\in A\\ i>j}} \sum_{v=j+1}^{i} (q_v - q_{v-1}) u_{ij}^p$$

$$= \sum_{v\in N} (q_v - q_{v-1}) \sum_{\substack{(i,j)\in A\\ i<v\le j}} l_{ij}^p - \sum_{v\in N} (q_v - q_{v-1}) \sum_{\substack{(i,j)\in A\\ j<v\le i}} u_{ij}^p$$

$$= \sum_{v\in N} (q_v - q_{v-1}) S^p(A_v).$$

Consider the following program, with variables q_M defined on the subsets $M \subset N$:

$$\max \sum_{M\subset N} q_M S^p(M) \tag{3.5}$$

subject to

$$\sum_{M\subset N} |M| q_M = 1,$$

$$q_M \ge 0, \quad M \subset N.$$

Every feasible solution to (3.4) can be mapped into a distinct feasible solution to (3.5) by setting $q_{A_v} = q_v - q_{v-1}$, $v \in N$, and $q_M = 0$ for all other subsets of N. Moreover, as was the case with program (2.5), (3.5) is a (continuous) knapsack problem. It is solved by setting $q_{M^*} = 1/|M^*|$ for a set M^* with maximum average surplus (per node), and $q_M = 0$ otherwise. This solution solves (3.5) over a range that strictly contains the solutions corresponding to the feasible solutions of (3.4). Therefore, $q_i = q_{M^*}$ $i \in M^*$ and $q_i = 0$ $i \notin M^*$ solves (3.4). It comes out that (3.4) the value of Program (3.4) is exactly $\nu(p)$. Theorem 3.3 follows immediately from this observation through the duality theorem of linear programming. □

Theorem 3.3 motivates the application of a node-wise maximum mean cut selection rule in the dual algorithm. In view of the similarity of the results in this section and in Section 2, the following theorem is somewhat surprising:

Theorem 3.4. *The node-wise maximum mean cut selection algorithm may have an exponential number of iterations.*

Proof. The proof is by an example (Zadeh's examples do not apply here). We describe a family of problems that are constructed in such a way that when the node-wise maximum mean cut rule is applied to the n-th problem, it requires at least $3(2)^{n-2}$ iterations. Problem n consists of a graph (N_n, A_n), where $N_n = \{1, \dots, n+1\}$ and

$$A_n = \{(i, j) \mid i, j \in N_n, i < j\} \cup \{(n+1, j) \mid j = 1, \dots, n\}.$$

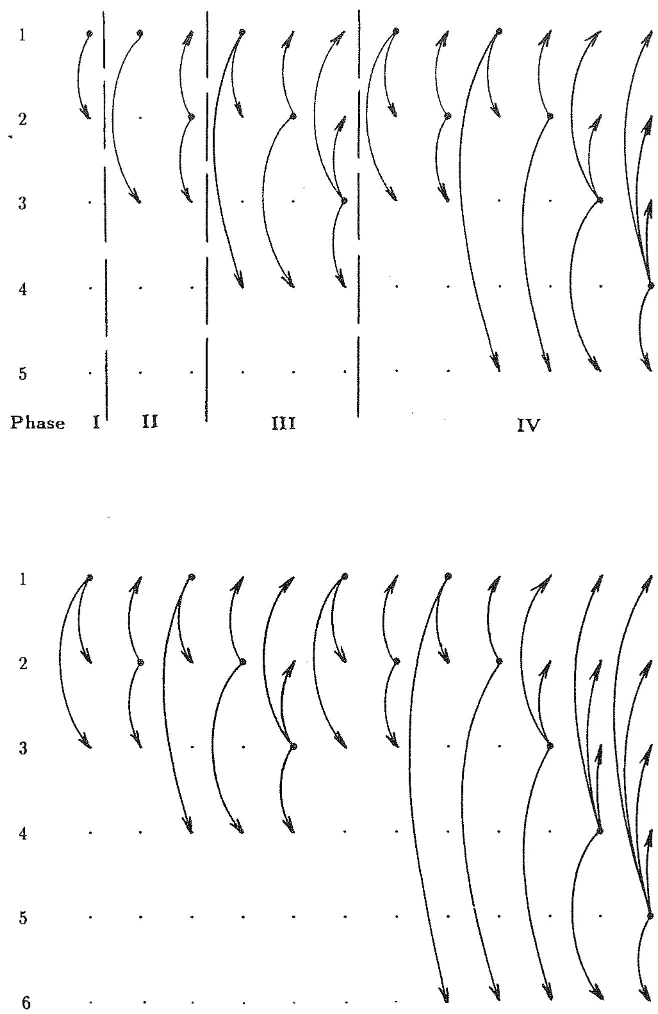
The arc costs are $c_{ij} = 2^{j-2}$ for $i < j$, and $c_{n+1,j} \equiv -C_n$, $j = 1, \dots, n$, where C_n is very large (e.g., 2^{2n}). The upper bounds are $u_{ij} = 2^{j-2}(j-1)!$ for $i < j$, and $u_{n+1,j} = U_n - 2^{j-1}j!$, $j = 1, \dots, n$, where U_n is sufficiently large (e.g., $2^{2n}n!$). The lower bounds are equal to zero.

The choice of the bounds is such that initially $S^p(\{i\}) > S^p(\{k\})$ for $i < k$, and this relation holds even when $S^p(\{i\})$ decreases by $\sum_{j=i+1}^{k} u_{ij}$ and $S^p(\{k\})$ increases by $(k-i)u_{ik}$. However, once $S^p(\{i\})$ decreases by $u_{i,k+1}$, the inequality is reversed.

We partition the iterations executed by the algorithm on Problem n to n phases indexed $r = 1, \dots, n$. Phase 1 contains a single iteration. For $r = 2, \dots, n-1$, Phase r contains $3(2)^{r-1}$ iterations.

Figure 1 illustrates the sequence of blocking arcs (those which join A_0) in Phases 1–5. In each column of this figure, a set of successively blocking arcs (i, j) with i fixed is given. The arcs with $j > i$ block one at a time in increasing order of j, and the arcs with $j < i$ block simultaneously in one iteration.

Figure 2 illustrates Phases 1–3 in Problem 4. Node 5 and the arcs incident with it are not shown. The maximum mean cuts are the single nodes marked by circles, the blocking arcs are marked by bold lines. Let $U \equiv U_4$. In Figure 2B the chosen cut contains only node 1 and its mean surplus is $u_{5,1} = U - 1$. In 2C the same cut is selected with mean surplus of $u_{5,1} - u_{1,2} = U - 2$.



Phase V

Fig. 1

For example, in 2C the cut consisting of $\{1, 2\}$ has mean surplus of $0.5(u_{5,1} + u_{5,2}) = U - 2.5$.

As can be seen, (Figure 2C, 2F, 2L) the initial state of Phase r is identical to the initial state of Phase 1, except that $c_{ir}^p = 0$ $\forall i < r$. Therefore, it first repeats Phases $1, 2, \dots, r-2$. Then, whenever in Phase $r-1$ arc (i, r) blocked, it is now $(i, r+1)$ which blocks. Then a state is reached (e.g., Figure 2J) where arcs $(i, r-1)$ have identical negative costs, $c_{i,r}^p = 0$ $i \neq r-1$, and $c_{r-1,r}^p$ still have its original value. Phase r terminates in two additional iterations, an iteration where all arcs $(i, r-1)$ block simultaneously, and a last iteration where $(r, r+1)$ blocks.

Denoting by $g(r)$ the number of iterations in Phase r, we obtain for $r > 1$ that $g(r) = g(1)$

$+\ldots+g(r-1)+2$. By induction, $g(r)=1+3(2^0+2^1+\ldots 2^{r-2})+2=3(2)^{r-1}$. □

4. Other dual rules

It makes no difference if we divide the cost of a cycle by the number of its nodes or by the number of its arcs since these numbers are identical. However, the mean surplus of a cut can be defined with respect to the number of arcs incident with it and a different measure results. Some possibilities will be discussed below.

One could modify (3.3) as follows. For a given price vector p define $\rho(p)$:

$$\rho(p)=\min\ z \tag{4.1}$$

subject to

$$\sum_{j\mid(j,i)\in A} f_{ji}-\sum_{j\mid(i,j)\in A} f_{ij}=0,\quad i\in N,$$

$$l_{ij}^p-z\le f_{ij}\le u_{ij}^p+z,\quad (i,j)\in A.$$

Clearly, p is optimal if and only if $\rho(p)\le 0$. Let us define the *arc-wise mean surplus* of a cut M as its surplus divided by the number of arcs in $\{(i,j)\mid i\in M,\ j\notin M\}\cup\{(i,j)\mid i\notin M,\ j\in M\}$. Let

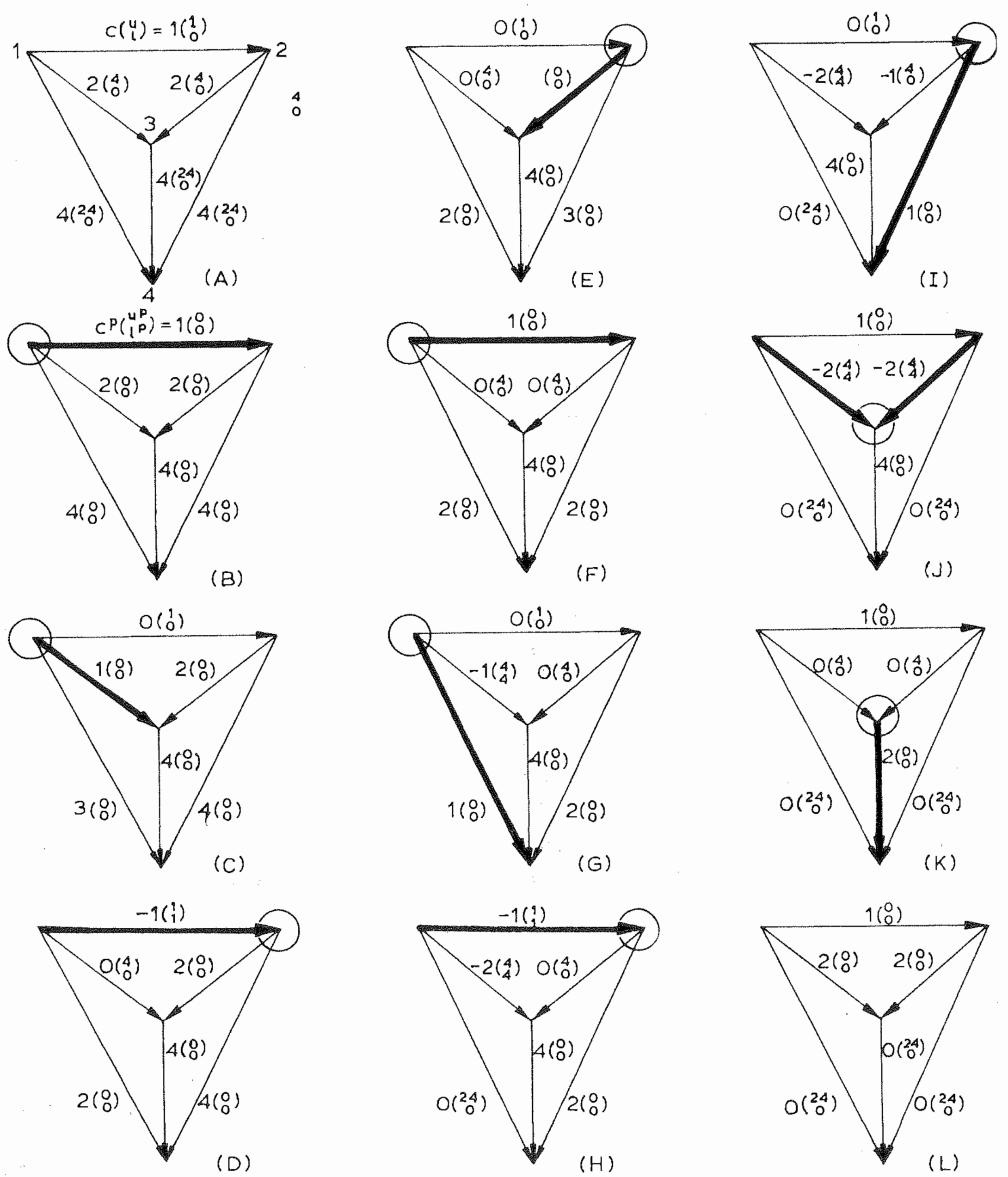


Fig. 2

$\pi(p)$ denote the arc-wise maximum mean surplus of a cut in G.

Theorem 4.1. $\rho(p)=\pi(p)$.

Proof. The dual to (4.1) is

$$\max \sum_{(i,j)\in A} w_{ij} l_{ij}^p - \sum_{(i,j)\in A} w'_{ij} u_{ij}^p \tag{4.2}$$

subject to

$$p_i - p_j + w_{ij} - w'_{ij} = 0, \quad (i,j)\in A,$$

$$\sum_{(i,j)\in A} (w_{ij} + w'_{ij}) = 1,$$

$$w, w' \geq 0.$$

Program (4.2) can be reformulated as

$$\max \sum_{(i,j)\in A} l_{ij}^p (p_j - p_i)^+ - \sum_{(i,j)\in A} u_{ij}^p (p_i - p_j)^+ \tag{4.3}$$

subject to

$$\sum_{(i,j)\in A} |p_j - p_i| = 1.$$

The rest of the proof is similar to that of Theorem 3.3. □

Theorem 4.2 (Ervolina and McCormick, 1990a; McCormick and Ervolina, 1990). *The arc-wise maximum mean surplus algorithm is (strongly) polynomial.*

Straightforward variations on the above analysis are possible. We may omit z from the right-hand-side (left-hand-side) of the bound constraints in (4.1) and modify the definition of mean surplus dividing the cut's surplus by the number of arcs in $\{(i,j) \mid i \notin M, j \in M\}$ ($\{(i,j) \mid i \in M, j \notin M\}$). McCormick has recently shown that these rules also yield polynomial algorithms (McCormick, 1991). Rote (1991) and Radzik (1992) obtained efficient algorithms for computing such mean surplus maximizing cuts.

References

F. Barahona and E. Tardos (1989), "Note on Weintraub's minimum cost circulation algorithm", *SIAM J. Comput.* **18**, 579–583.

R.G. Busacker and T.L. Saaty (1965), *Finite Graphs and Networks: An Introduction with Applications*, McGraw Hill, New York.

N. Chung and D. Tcha (1991), "A dual algorithm for submodular flow problems", *Oper. Res. Lett.* **10**, 489–495.

G.B. Danzig, W.O. Blattner, and M.R. Rao (1967), "Finding a cycle in a graph with minimum cost to time ratio with applications to a ship routing problem", in: *Theory of Graphs*, International Symposium Rome 1966, Gordon and Breach, New York, 77–83.

G.M. Engel and H. Schneider (1975), "Diagonal similarity and equivalence for matrices over groups with 0", *Czechoslovak Math. J.* **25**, 389–403.

T.R. Ervolina and S.T. McCormick (1990a), "A strongly polynomial maximum mean cut cancelling algorithm for minimum cost network flow", UBC Faculty of Commerce Working Paper 90-MSC-009.

T.R. Ervolina and S.T. McCormick (1990b), "Cancelling most helpful cuts for minimum cost network flow", UBC Faculty of Commerce Working Paper 90-MSC-018.

A.V. Goldberg and R.E. Tarjan (1989), "Finding minimum-cost circulations by canceling negative cycles", *J. Assoc. Comput. Mach.* **36**, 873–886.

R. Hassin (1983), "The minimum cost flow problem: A unifying approach to dual algorithms and a new tree-search algorithm", *Math. Programming* **25**, 228–239.

M. Klein (1967), "A primal method for minimal cost flows", *Management Sci.* **14**, 205–220.

S.T. McCormick (1991), Private communication.

S.T. McCormick and T.R. Ervolina (1990), "Computing maximum mean cuts", UBC Faculty of Commerce Working Paper 90-MSC-011.

T. Radzik (1992), "Minimizing capacity violations in a transshipment network", in: *Proceedings of the Third Annual ACM-SIAM Symposium on Discrete Algorithms*.

G. Rote (1991), "An improved time bound for computing maximum mean cuts", presented at the *14th International Symposium on Mathematical Programming*, Amsterdam.

A. Weintraub (1974), "A primal algorithm to solve network flow problems with convex costs", *Management Sci.* **21**, 87–97.

N. Zadeh (1973), "A bad network problem for the simplex method and other minimum cost flow algorithms", *Math. Programming* **5**, 255–266.