



Research Note

A geometrical explanation for the optimality concept of minimum cost flows

M. Ghiyasvand*

Department of Mathematics, Faculty of Science, Bu-Ali Sina University, Hamedan, Iran.

Received 17 December 2014; received in revised form 3 September 2015; accepted 19 October 2015

KEYWORDS

The minimum cost-flow problem;
 Out-of-kilter;
 Most positive cut canceling.

Abstract. The algorithm proposed by Shigeno et al. (2000) is a scaling method to solve the minimum cost-flow problem. In each phase, they applied the most positive cut canceling idea. In this paper, we present a new approach to solve the problem, which uses the scaling method of Shigeno et al. (2000); but, in each phase, we apply the out-of-kilter idea instead of the most positive cut canceling idea. Our algorithm is inspired by Ghiyasvand (2012). The algorithm gives a geometrical explanation for the optimality concept. For a network with n nodes and m arcs, the algorithm performs $O(\log(nU))$ phases and runs in time $O(m(m + n \log n) \log(nU))$ (where U is the largest absolute arc bound), which is $O(m(m + n \log n) \log n)$ under the similarity assumption. This time is the current best strongly polynomial time to solve the minimum cost-flow problem presented by Orlin (1993) and Vygen (2002).

© 2016 Sharif University of Technology. All rights reserved.

1. Introduction

The minimum cost-flow problem is one of the most fundamental network flow problems. Different algorithms have been proposed to solve the problem. Some known algorithms for this problem are Edmond et al. [1], Goldberg and Tarjan [2], Orlin [3], Ahuja et al. [4], Hassin [5], Shigeno et al. [6], and Vygen [7]. Extensive discussion of this problem and its applications can be found in the books of Ahuja et al. [8] and Ford and Fulkerson [9]. Recently, a new algorithm was presented by Paparrizos et al. [10].

Classical algorithms for the minimum cost-flow problem are the out-of-kilter algorithm [11,12] and the cheapest path augmentation [13,14]. The out-of-kilter algorithm is the sort of algorithm that computers can do easily, but people must be careful in using it. This algorithm uses the complementary slackness optimality condition, selects arcs that do not satisfy this condition,

and changes flow and potential to enforce the condition.

Algorithms for the minimum cost-flow problem can be classified into primal and dual algorithms. Primal algorithms always have a feasible solution and work towards eliminating negative-cost circuits in the residual graph. Dual algorithms maintain a feasible potential in the residual graph and work towards primal feasibility. We denote the number of nodes, number of arcs, maximum arc capacity, and maximum absolute value of an arc cost by n , m , U , and ψ , respectively. The best running times for the minimum cost-flow problem are the $O((m \log U)(m + n \log n))$ time method of Edmonds et al. [1], the $O(nm \log(n^2/m) \log(n\psi))$ time method of Goldberg and Tarjan [2], the $O((m \log n)(m + n \log n))$ time method of Orlin [3] and Vygen [7], and the $O(nm(\log \log U) \log(n\psi))$ time of Ahuja et al. [4]. Each of these algorithms is the best for a different ranges of parameters n , m , U , and ψ . Computational and experimental implementations of minimum cost-flow algorithms have been presented in [15-20].

*. Tel./Fax: 081-38380987
 E-mail address: mghiayasvand@basu.ac.ir

Shigeno et al. [6] relax some arc bounds by δ , which starts with $\delta = U$ and in each repetition, decreases δ to $\delta/2$, and tries to find a $\delta/2$ -optimal circulation using the most positive cut cancelling idea. In this paper, we use the scaling method of [6]; but, in each phase, a $\delta/2$ -optimal circulation is computed using the out-of-kilter idea. Thus, the framework of our algorithm is similar to the algorithm of Shigeno et al. [6]; but, in each phase, two different methods are applied, which are the most positive cut cancelling and out-of-kilter. Our algorithm is inspired by Ghiyasvand [21].

Our algorithm gives a geometrical explanation for the optimality concept and describes a polynomial-time implementation of the out-of-kilter idea. It is a scaling algorithm, which in each phase transforms all out-of-kilter arcs into in-kilter arcs. The case that the network is infeasible is diagnosed by the algorithm. Our algorithm is called *Geometric-DMCF algorithm* (Geometric-Dual Minimum Cost Flow algorithm) and runs in time $O(m(m + n \log n) \log(nU))$. Similarity assumption [22] says that the bounds are at most a fixed power of n , namely $\log(U) = O(\log n)$. This assumption usually makes sense in practice and leads to lower asymptotic running times. Thus, under the similarity assumption, our algorithm runs in $O(m(m + n \log n) \log n)$, which is the current best strongly polynomial time to solve the minimum cost-flow problem presented by Orlin (1993) and Vygen (2002).

This paper consists of three sections in addition to the Introduction Section. Section 2 presents network notation and reviews some results used in the subsequent sections. We show the framework of our algorithm in Section 3.1. In Section 3.2, our procedure for the improvement approximation is described, which runs in time $O(m(m + n \log n))$ using a shortest-path computation. Finally, Section 4 concludes the paper.

1.1. Contribution of this paper

Section 5.3 of Shigeno et al. [6] is a scaling algorithm that, in each phase, computes a δ -optimal potential using at most m shortest path computations. In this section, our algorithm is compared with the algorithm of Shigeno et al.. We show that our algorithm is a modified version of the scaling method of Shigeno et al.'s algorithm; but, in each phase, our algorithm applies the out-of-kilter idea instead of the most positive cut canceling idea.

In the algorithm presented in Section 5.3 of [6], the most positive cut canceling method is used to compute a δ -optimal. In this algorithm, by adding two nodes s and t , a maximum flow problem (called $MFN(l^{\pi, \delta}, u^{\pi, \delta})$) is defined. The sets N_s, N_t (as nodes reachable from s and t), and strongly connected components of the residual graph by $N - (N_s \cup N_t)$ are computed. Each component $N_s, N_1, N_2, \dots, N_k, N_t$ is contracted to a single node, obtaining a network

H . By defining the length of each arc on H , let β_v denote shortest-path distance from N_t to N_v . Then, the algorithm updates node potentials by $\pi'_i := \pi_i + \beta_v$ for each $i \in N_v$. By the result of Lemma 5.10 in [6], this method decreases the most positive cut value by at least δ . Lemma 5.6 of [6] says the most positive cut value in a scaling phase is at most $m\delta$; thus, after at most m iterations, the most positive cut value is zero and the scaling phase is finished. Thus, in each phase, the algorithm of Shigeno et al. is a most positive canceling method to compute a δ -optimal potential.

In each phase of our algorithm, an out-of-kilter idea is extended to compute a δ -optimal potential. In this algorithm, using the 2δ -optimal potential computed in the last scaling phase, sets B^+, B^-, R, G^+ , and G^- are defined. Each arc in sets $B^+ \cup G^+$ and $B^- \cup G^- \cup R$ is called out-of-kilter and in-kilter, respectively. In each iteration of a scaling phase, at least one out-of-kilter arc is transformed into an in-kilter arc; therefore, the number of iterations in each scaling phase is at most m . For each $w \rightarrow v \in B^+ \cup G^+$, the algorithm finds a cycle (which is called $C_{w \rightarrow v}$) containing the arc $w \rightarrow v$, using one shortest-path computation. Then, δ units of flow are pushed in the cycle $C_{w \rightarrow v}$ such that the arc $w \rightarrow v$ leaves set $B^+ \cup G^+$ and enters into set $B^- \cup G^- \cup R$. Hence, in each phase, our algorithm transforms all out-of-kilter arcs into in-kilter arcs to compute a δ -optimal potential.

2. Preliminaries

2.1. Notations and definitions

A directed graph, D , is a pair $D = (N, A)$, where N is a set of nodes and A is a set of ordered pairs of nodes, called arcs. We denote an arc from node i to node j by $i \rightarrow j$ and define the cost on arc $i \rightarrow j$ by c_{ij} . A *simple cycle* C in a directed graph is a sequence, i_1, i_2, \dots, i_k , of distinct nodes of N such that either $i_r \rightarrow i_{r+1} \in A$ (a *forward arc* in C) or $i_{r+1} \rightarrow i_r \in A$ (a *backward arc* in C) for $r = 1, 2, \dots, k$ (where we interpret i_{k+1} as i_1). A *directed cycle* is a simple cycle with all forward arcs. *Simple path* and *directed path* are same as simple cycle and directed cycle, respectively, without arc $i_k \rightarrow i_1$. If W is a non-trivial subset of N (i.e. $W \neq \emptyset, W \neq N$) and $\bar{W} = N - W$, then we define $(W, \bar{W}) = \{i \rightarrow j | i \in W, j \notin W\}$ and $(\bar{W}, W) = \{i \rightarrow j | i \notin W, j \in W\}$. The arc subset (W, \bar{W}) (and (\bar{W}, W)) is called a *cut*.

Theorem 2.1. (Shortest-path optimality conditions [8]). For every node $r \in N$, let d_r denote the length of some directed path from a special node (as source node) to node r . Then, the numbers d_r represent distances of the shortest paths (from the source node to node r) with respect to distances c_{ij} if and only if they satisfy the following conditions:

$$d_j \leq d_i + c_{ij} \text{ for all } (i, j) \in A.$$

Consider a shortest path from the source to some node r ; then, for each arc $i \rightarrow j$ that belongs to this path, we have $d_j = d_i + c_{ij}$. ■

2.2. Minimum cost-flow problem

Let $D = (N, A)$ be a directed graph with $|N| = n$ and $|A| = m$. Let $c \in R^A$ be a cost function on A and l, u be lower and upper bounds on A , with $l_{ij} \leq u_{ij}$ for each arc $i \rightarrow j \in A$. The primal problem has flow x_{ij} on arc $i \rightarrow j$. The primal linear programming formulation (*the primal problem*) of the minimum cost-flow problem is as follows:

$$\min \sum_{i \rightarrow j \in A} c_{ij} x_{ij}, \tag{1}$$

$$\text{s.t.} \quad \sum_{j \in N} x_{ij} - \sum_{j \in N} x_{ji} = 0, \quad i \in N,$$

$$l_{ij} \leq x_{ij} \leq u_{ij}, \quad i \rightarrow j \in A. \tag{2}$$

We call $x \in R^A$ a *circulation* if only (1) is required and a *bounded circulation* if both Relations (1) and (2) are required. The primal problem is feasible if and only if there exists flow x satisfying Relations (1) and (2). The following theorem is a well-known result for the feasibility of the primal problem.

Theorem 2.2. (Hoffman’s theorem [23]) The primal problem is feasible if and only if for every cut (S, \bar{S}) , $\sum_{i \rightarrow j \in (S, \bar{S})} l_{ij} - \sum_{i \rightarrow j \in (\bar{S}, S)} u_{ij} \leq 0$. ■

We associate a node potential π_i on each node i and define the reduced cost of an arc $i \rightarrow j$ as:

$$c_{ij}^\pi = c_{ij} - \pi_i + \pi_j. \tag{3}$$

The dual linear program of the minimum cost-flow problem (*the dual problem*) is:

$$\max \sum_{i \rightarrow j \in A} (c_{ij}^\pi)^+ l_{ij} - \sum_{i \rightarrow j \in A} (c_{ij}^\pi)^- u_{ij},$$

$$\text{s.t.} \quad \pi_j + c_{ij} - \pi_i = c_{ij}^\pi \quad i \rightarrow j \in A,$$

$$\pi_i \text{ free} \quad i \in N,$$

where $(c^\pi)^+ = \max(0, c^\pi)$ and $(c^\pi)^- = \max(0, -c^\pi)$. Potential π is feasible for the dual problem if it satisfies the constraints of the dual problem.

The complementary slackness conditions for general linear programming (when specialized in minimum cost-flow problem) result in the following characterization of optimal primal and dual solutions for the primal and dual problems.

Theorem 2.3. The pair x and π are jointly optimal

for the primal and dual problems if and only if:

- (i) x is feasible for the primal problem;
- (ii) π is feasible for the dual problem; and
- (iii) For every $i \rightarrow j \in A$

$$\text{if } c_{ij}^\pi < 0, \text{ then } x_{ij} = u_{ij}, \tag{4a}$$

and:

$$\text{if } c_{ij}^\pi > 0, \text{ then } x_{ij} = l_{ij}. \blacksquare \tag{4b}$$

Given the node potentials π , for each arc $i \rightarrow j$, define lower bound l_{ij}^π and upper bound u_{ij}^π by:

$$\text{if } c_{ij}^\pi > 0, \text{ then } l_{ij}^\pi = u_{ij}^\pi = l_{ij}, \tag{5a}$$

$$\text{if } c_{ij}^\pi = 0, \text{ then } l_{ij}^\pi = l_{ij}, \quad u_{ij}^\pi = u_{ij}, \tag{5b}$$

and:

$$\text{if } c_{ij}^\pi < 0, \text{ then } l_{ij}^\pi = u_{ij}^\pi = u_{ij}. \tag{5c}$$

The following theorem is a result of Relations (4).

Theorem 2.4 [5]. Feasible node potentials π are optimal to the dual problem if and only if the primal network with nodes N , arcs A , lower bounds l^π , and upper bounds u^π has a bounded circulation flow. ■

Like [6] (Page 93), for $\delta \geq 0$, potential π is called *δ -optimal* if there exists a circulation x such that for all arcs $i \rightarrow j$:

$$\text{if } c_{ij}^\pi > 0, \text{ then } l_{ij} - \delta \leq x_{ij} \leq l_{ij} + \delta, \tag{6a}$$

$$\text{if } c_{ij}^\pi = 0, \text{ then } l_{ij} - \delta \leq x_{ij} \leq u_{ij} + \delta, \tag{6b}$$

and:

$$\text{if } c_{ij}^\pi < 0, \text{ then } u_{ij} - \delta \leq x_{ij} \leq u_{ij} + \delta. \tag{6c}$$

By Theorem 2.4, potential π is optimal if and only if it is 0-optimal. In our algorithm, we start with a large δ and drive δ toward zero. The following lemma says that δ does not need to start out too big, and does not need to end up too small. We define U as the largest absolute arc bound.

Lemma 2.5 [6]. Any node potentials π are U -optimal. Moreover, when $\delta < \frac{1}{m}$, all δ -optimal node potentials are optimal to the dual problem. ■

Consider node potential π ; we call a network with nodes N , arcs A , and Constraints (1) and (6) *the δ -network corresponding to π* . We say x is a feasible flow for the δ -network corresponding to π if it satisfies Relations (1) and (6). Thus, potential π is a δ -optimal set of node potentials if and only if there exists a feasible flow for δ -network corresponding to π .

```

Begin
  Define  $D'$  using Eqs. (12), (13), and (14);
  Do while  $B^+ \cup G^+ \neq \emptyset$ 
    Begin
      Select an arc  $w \rightarrow v \in B^+ \cup G^+$ ;
      If  $w \rightarrow v \in B^+$  (resp.  $w \rightarrow v \in G^+$ ) then determine the shortest path distance  $d(\cdot)$  from
      node  $w$  (resp. node  $v$ ) to all nodes in  $D'$  with respect to the distances  $\theta_{ij}$  and  $\theta_{ji}$ ;
      Let  $P$  be a shortest path from node  $w$  (resp. node  $v$ ) to node  $v$  (resp. node  $w$ ) in  $D'$ ;
      If for some node  $i$ ,  $d_i = \infty$  then write the primal problem is infeasible and break;
      Else
        Begin
          Compute  $\bar{P}$  and let  $C_{w \rightarrow v} = \bar{P} \cup \{w \rightarrow v\}$ ;
          Update  $\pi := \pi + d$ ;
          Augment  $\delta$  units of flow in  $C_{w \rightarrow v}$  using Augmentation  $(\delta, x, C)$ ;
        End;
      Update  $B, G, R$ , and  $D'$ ;
      Go to 1;
    End;
  End.
  
```

Figure 1. The method of Procedure 1.

```

Begin
   $\pi := 0, x := 0$  and  $\delta := U$ ;
  While  $\delta \geq \frac{1}{m}$  do
    Begin
      Put  $\delta := \delta/2$ ;
      Procedure 1  $(\delta, \pi, x)$ ;
    End;
   $\pi$  is optimal for the dual problem;
End.
  
```

Algorithm 1. The framework of the geometric-DMCF algorithm.

3. The Geometric-DMCF algorithm

3.1. The framework of the Geometric-DMCF algorithm

Our algorithm treats δ as a parameter and iteratively obtains δ -optimal potential for successively smaller values of δ . Initially, $\delta = U$, $x = 0$, and $\pi = 0$. The algorithm executes scaling phases, where each scaling phase cuts δ in half and applies Procedure 1 (Figure 1) that transforms a 2δ -optimal set of node potentials into a δ -optimal set of node potentials. This continues until $\delta < \frac{1}{m}$, the point at which Lemma 2.5 says that we are finished, having done $O(\log(nU))$ phases. Algorithm 1 describes the framework of our algorithm (the geometric-DMCF algorithm).

The algorithm of Shigeno et al. [6] has the same framework. In each phase of their algorithm, the most positive cut canceling method is used to compute a δ -optimal potential. In this paper, a δ -optimal potential is computed using the out-of-kilter method.

3.2. The method of Procedure 1

The essential part of each phase of our algorithm is Procedure 1. The input to the Procedure 1 (δ, π, x) is a 2δ -optimal set of node potentials π and a feasible flow x for the 2δ -network corresponding to π and its output is a δ -optimal set of node potentials π' and a feasible flow x' for the δ -network corresponding to π' .

Supposing that x is a feasible flow for 2δ -network corresponding to π , we color in G green, in B black, and in R red, where G, B , and R are defined as follows.

Let $G = G^- \cup G^+$ such that:

$$G^+ = \begin{cases} \{i \rightarrow j \in A | u_{ij} + \delta < x_{ij} \leq u_{ij} + 2\delta\}, \\ \text{if } c_{ij}^\pi \leq 0, \\ \{i \rightarrow j \in A | l_{ij} + \delta < x_{ij} \leq l_{ij} + 2\delta\}, \\ \text{if } c_{ij}^\pi > 0, \end{cases} \quad (7)$$

and:

$$G^- = \begin{cases} \{i \rightarrow j \in A | u_{ij} < x_{ij} \leq u_{ij} + \delta\}, \\ \text{if } c_{ij}^\pi \leq 0, \\ \{i \rightarrow j \in A | l_{ij} < x_{ij} \leq l_{ij} + \delta\}, \\ \text{if } c_{ij}^\pi > 0. \end{cases} \quad (8)$$

Also, let $B = B^+ \cup B^-$ such that:

$$B^+ = \begin{cases} \{i \rightarrow j \in A | u_{ij} - 2\delta \leq x_{ij} < u_{ij} - \delta\}, \\ \text{if } c_{ij}^\pi < 0, \\ \{i \rightarrow j \in A | l_{ij} - 2\delta \leq x_{ij} < l_{ij} - \delta\}, \\ \text{if } c_{ij}^\pi \geq 0, \end{cases} \quad (9)$$

and:

$$B^- = \begin{cases} \{i \rightarrow j \in A | u_{ij} - \delta \leq x_{ij} < u_{ij}\}, \\ \text{if } c_{ij}^\pi < 0, \\ \{i \rightarrow j \in A | l_{ij} - \delta \leq x_{ij} < l_{ij}\}, \\ \text{if } c_{ij}^\pi \geq 0. \end{cases} \quad (10)$$

Define:

$$R = \begin{cases} \{i \rightarrow j \in A | x_{ij} = u_{ij}\}, & \text{if } c_{ij}^\pi < 0, \\ \{i \rightarrow j \in A | x_{ij} = l_{ij}\}, & \text{if } c_{ij}^\pi > 0, \\ \{i \rightarrow j \in A | l_{ij} \leq x_{ij} \leq u_{ij}\}, & \text{if } c_{ij}^\pi = 0. \end{cases} \quad (11)$$

We call each arc in set $B^+ \cup G^+$ as an out-of-kilter arc, and each arc in set $B^- \cup G^- \cup R$ as an in-kilter arc. Procedure 1 successively transforms all the out-of-kilter arcs into in-kilter arcs, maintaining all in-kilter arcs as in-kilter arcs. The procedure terminates when all arcs are in-kilter. The following conclusion is obvious according to the definitions.

Conclusion 3.1.

- (a) In case $i \rightarrow j \in B^-$, if we increase x_{ij} by δ , then $i \rightarrow j \notin B^+ \cup G^+$;
- (b) In case $i \rightarrow j \in G^-$, if we decrease x_{ij} by δ , then $i \rightarrow j \notin B^+ \cup G^+$;
- (c) In case $i \rightarrow j \in R$, if we increase or decrease x_{ij} by δ , then $i \rightarrow j \notin B^+ \cup G^+$;
- (d) In case $i \rightarrow j \in B^+$, if we increase x_{ij} by δ , then $i \rightarrow j \notin B^+ \cup G^+$;
- (e) In case $i \rightarrow j \in G^+$, if we decrease x_{ij} by δ , then $i \rightarrow j \notin B^+ \cup G^+$.

If $B^+ \cup G^+ = \emptyset$, then x is a feasible flow for δ -network corresponding to π (also, potential π is a δ -optimal set of node potentials). But if $B^+ \cup G^+ \neq \emptyset$, we select an out-of-kilter arc $w \rightarrow v$ (i.e., $w \rightarrow v \in B^+ \cup G^+$) and transform it into an in-kilter arc (i.e., arc $w \rightarrow v$ is entered into set $B^- \cup G^- \cup R$) maintaining all the in-kilter arcs as in-kilter arcs (i.e., no arc in set $B^- \cup G^- \cup R$ enters into set $B^+ \cup G^+$).

Given a 2δ -optimal set of node potentials π and a feasible flow x for the 2δ -network corresponding to π . We define a network D' on the same node set as D . We assume that for any pair of nodes i and j , either $i \rightarrow j \in A$ or $j \rightarrow i \in A$, but not both (we can easily satisfy this assumption by performing a simple transformation, but the assumption is not needed in any case). Every arc $p \rightarrow q \in D$ becomes a pair of arcs, $p \rightarrow q$ and $q \rightarrow p$, in D' with distances θ_{pq} and θ_{qp} . We denote all arcs in D' by A' ; thus, $D' = (N, A')$. If $X, Y \subset N$ form a nontrivial partition of N , then, we define a cut in D by:

$$(X, Y)_{A'} = \{i \rightarrow j \in A' | i \in X, j \in Y\}.$$

Distances θ_{pq} and θ_{qp} are computed as follows:

If $p \rightarrow q \in G$, then:

$$\theta_{pq} = 0, \quad (12a)$$

and:

$$\theta_{qp} = \begin{cases} c_{pq}^\pi, & c_{pq}^\pi > 0, x_{pq} \leq u_{pq}, \\ \infty, & \text{otherwise.} \end{cases} \quad (12b)$$

If $p \rightarrow q \in B$, then:

$$\theta_{qp} = 0, \quad (13a)$$

and:

$$\theta_{pq} = \begin{cases} -c_{pq}^\pi, & c_{pq}^\pi < 0, x_{pq} \geq l_{pq}, \\ \infty, & \text{otherwise.} \end{cases} \quad (13b)$$

If $p \rightarrow q \in R$, then:

$$\theta_{pq} = \theta_{qp} = 0. \quad (14)$$

Supposing that $w \rightarrow v \in B^+ \cup G^+$. If $w \rightarrow v \in B^+$ (with respect to $w \rightarrow v \in G^+$), then, in D' , we determine a shortest path from node w (with respect to node v) to all other nodes with respect to θ . Consider a node r , we denote a shortest path to node r by p_r , i.e. if $w \rightarrow v \in B^+$ (with respect to $w \rightarrow v \in G^+$), then p_r is a shortest path from node w (with respect to node v). Note that p_r is found in D' and is a directed path. Let d_r be the shortest-path distance to node r , i.e.:

$$d_r = \sum_{i \rightarrow j \in p_r} \theta_{ij}.$$

By definitions, if $d_r = \infty$, then there exists at least one $i \rightarrow j \in p_r$ with $\theta_{ij} = \infty$. The following lemma shows that if there exists one node r with $d_r = \infty$, then the primal problem is infeasible.

Theorem 3.1. If there exists one node r in D' with $d_r = \infty$, then the primal problem is infeasible.

Proof. We use the idea of Dijkstra's shortest-path algorithm [24]. Without loss of generality, suppose that $d_1 \leq d_2 \leq \dots \leq d_k \leq d_l \leq \dots \leq d_n$ such that $1, 2, \dots, n$ are all nodes; also, let node l be the first node with $d_l = \infty$ (i.e. $d_l = \dots = d_n = \infty$ and $d_1 \leq d_2 \leq \dots \leq d_k < \infty$). Thus, Dijkstra's algorithm computes d_1, d_2, \dots, d_k before d_l . We denote nodes $1, 2, \dots, k$ by S and $\bar{S} = N - S$. By definitions of S and \bar{S} , we get:

$$\theta_{ij} = \infty \quad \text{for ever } i \rightarrow j \in (S, \bar{S})_{A'}. \quad (15)$$

Thus, by Eq. (14), we have:

$$((S, \bar{S}) \cup (\bar{S}, S)) \cap R = \emptyset. \quad (16)$$

Using Eqs. (13a) and (15), we get:

$$(\bar{S}, S) \cap B = \infty.$$

Hence, Eq. (16) says $i \rightarrow j \in G$ for every $i \rightarrow j \in (\bar{S}, S)$. By Eqs. (12b) and (15), there does not exist an arc $i \rightarrow j \in (\bar{S}, S)$ such that $i \rightarrow j \in G$, $c_{ij}^\pi > 0$ and $x_{ij} \leq u_{ij}$. Thus:

$$\sum_{i \rightarrow j \in (\bar{S}, S)} x_{ij} > \sum_{i \rightarrow j \in (\bar{S}, S)} u_{ij}. \tag{17}$$

Also, according to Eqs. (12a) and (15), we have $(S, \bar{S}) \cap G = \emptyset$. Hence, by Eqs. (15), (13b), and (16), we get:

$$\sum_{i \rightarrow j \in (S, \bar{S})} x_{ij} < \sum_{i \rightarrow j \in (S, \bar{S})} l_{ij}. \tag{18}$$

Since x is a circulation, we have:

$$\sum_{i \rightarrow j \in (S, \bar{S})} x_{ij} - \sum_{i \rightarrow j \in (\bar{S}, S)} x_{ij} = 0.$$

Thus, by Relations (17) and (18), we get:

$$\sum_{i \rightarrow j \in (S, \bar{S})} l_{ij} - \sum_{i \rightarrow j \in (\bar{S}, S)} u_{ij} > 0.$$

Therefore, by Theorem 2.2, the primal problem is infeasible. ■

If $w \rightarrow v \in B^+$ (with respect to $w \rightarrow v \in G^+$, then we denote a shortest path from node w (with respect to node v) to node v (with respect to node w) in D' by P (i.e., if $w \rightarrow v \in B^+$, then $P = p_v$ and if $w \rightarrow v \in G^+$, then $P = p_w$). Note that P is a directed path in D' . By the definition of D' , each arc $i \rightarrow j \in D'$ has a corresponding arc $i \rightarrow j$ or $j \rightarrow i$ in D ; thus, the directed path P in D' has a corresponding simple path \bar{P} in D .

We define $C_{w \rightarrow v} = \bar{P} \cup \{w \rightarrow v\}$ (note that $C_{w \rightarrow v}$ is a simple cycle in D). Our implementation lets $\pi'_r = \pi_r + d_r$ for each node r and augments δ units of flow along $C_{w \rightarrow v}$ according to the augmentation (δ, x, C) method (see Figure 2). Thus, by Conclusion 3.1 (d, e), arc $w \rightarrow v$ leaves $B^+ \cup G^+$. We prove that the implementation in Procedure 1 maintains all in-kilter arcs $B^- \cup G^- \cup R$.

Supposing that $w \rightarrow v \in B^+$ (with respect to $w \rightarrow v \in G^+$), the path P is a shortest path from node

w (with respect to node v) to node v (with respect to node w) in D' . According to Figure 1, we send δ units of flow in the direction of node w (with respect to node v) to node v (with respect to node w) in simple cycle $C_{w \rightarrow v}$. Hence, the direction of sending δ units flow in $C_{w \rightarrow v}$ is opposite to the direction of path P . Note that P is a directed path in D' , and \bar{P} is a simple path in D . Thus, we get the following conclusion.

Conclusion 3.2. Augmentation (δ, x, C) procedure sends δ units flow, in the simple cycle $C_{w \rightarrow v}$, in the opposite direction of directed path P .

Consider a node r and path p_r , for each arc $i \rightarrow j$ in D ; we allocate a Boolean variable $\text{Use}(p_r, i \rightarrow j)$. We say $\text{Use}(p_r, i \rightarrow j) = \text{True}$ if (in D') $i \rightarrow j \in p_r$ or $j \rightarrow i \in p_r$. Otherwise, $\text{Use}(p_r, i \rightarrow j) = \text{False}$.

Lemma 3.2. Suppose that the primal problem is feasible. Consider a node r and a path p_r ; the following operations do not send any arc $i \rightarrow j \in B^- \cup G^- \cup R$ into set $B^+ \cup G^+$:

- (a) $\pi'_i = \pi_i + d_i$, for each $i \in N$;
- (b) After Operation (a), for each arc $m \rightarrow n \in p_r$, sending δ units of flows is sent in D from node n to node m .

Proof. Consider an arc $i \rightarrow j \in B^- \cup G^- \cup R$; we prove that after Operations (a) and (b), we get $i \rightarrow j \notin B^+ \cup G^+$.

- (i) $\text{Use}(p_r, i \rightarrow j) = \text{True}$.

By the definition of $\text{Use}(p_r, i \rightarrow j) = \text{True}$, we consider two cases $i \rightarrow j \in p_r$ and $j \rightarrow i \in p_r$:

- (i.1) $i \rightarrow j \in B^-$.
 - (i.1.1) $i \rightarrow j \in p_r$.

The primal problem is feasible; thus, by Theorem 3.1 and Eq. (13b), we have $\theta_{ij} = -c_{ij}^\pi$ and $c_{ij}^\pi < 0$.

Thus, by Theorem 2.1, we get $d_j = d_i - c_{ij}^\pi$, which means

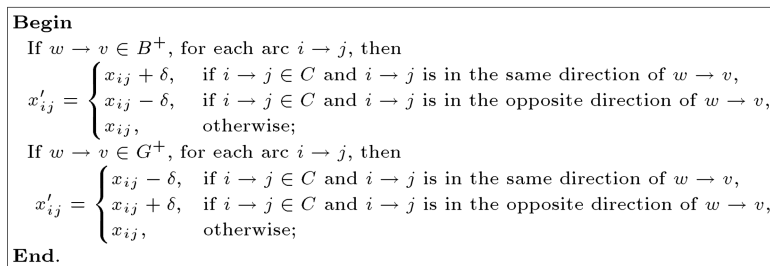


Figure 2. Augmentation (δ, x, C) procedure.

$c_{ij}^{\pi'} = 0$. Therefore, after Operation (a), we have $i \rightarrow j \in R$. By Conclusion 3.1(c), after sending δ units flow from node j to node i , the arc $i \rightarrow j$ does not enter into set $B^+ \cup G^+$;

$$(i.1.2) \quad j \rightarrow i \in p_r.$$

By Theorem 2.1 and Eq. (13a), we get:

$$d_i = d_j + \theta_{ji} = d_j + 0 = d_j.$$

Hence, $c_{ij}^{\pi'} = c_{ij}^{\pi}$, which means arc $i \rightarrow j$ remains in set B^- . Thus, by Conclusion 3.1(a), Operation (b) does not send the arc $i \rightarrow j$ into set $B^+ \cup G^+$.

$$(i.2) \quad i \rightarrow j \in G^-.$$

This case is proved in a similar way to Case (i-1);

$$(i.3) \quad i \rightarrow j \in R.$$

$$(i.3.1) \quad i \rightarrow j \in p_r.$$

The following is obtained by Theorem 2.1 and Eq. (14):

$$d_j = d_i + \theta_{ij} = d_i + 0 = d_i.$$

Thus, $c_{ij}^{\pi'} = c_{ij}^{\pi}$, which means $i \rightarrow j$ does not leave set R . Thus, by Conclusion 3.1(c), we have $i \rightarrow j \notin B^+ \cup G^+$;

$$(i.3.2) \quad j \rightarrow i \in p_r.$$

This case is proved in a similar way to Case (i-3-1).

$$(ii) \quad \text{Use}(p_r, i \rightarrow j) = \text{False}.$$

By $i \rightarrow j \notin p_r$ and $j \rightarrow i \notin p_r$, we only need to focus on Operation(b). In D' , there exist two arcs $i \rightarrow j$ and $j \rightarrow i$. By Theorem 2.1, we have:

$$d_j \leq d_i + \theta_{ij}, \tag{19}$$

$$d_i \leq d_j + \theta_{ji}. \tag{20}$$

$$(ii.1) \quad i \rightarrow j \in B^-.$$

By Eqs. (13a) and (20), we get $d_i \leq d_j$, or:

$$d_j - d_i \geq 0. \tag{21}$$

By $c_{ij}^{\pi'} = c_{ij} - (\pi_i + d_i) + (\pi_j + d_j) = c_{ij} + d_j - d_i$ and Relation (21), we conclude

$c_{ij}^{\pi'} \geq c_{ij}^{\pi}$. Therefore, Operation (a) increases the reduced cost of $i \rightarrow j$. We consider two cases $c_{ij}^{\pi} \geq 0$ and $c_{ij}^{\pi} < 0$:

$$(ii.1.1) \quad c_{ij}^{\pi} \geq 0.$$

We have $i \rightarrow j \in B$ and $c_{ij}^{\pi} > 0$; thus, if c_{ij}^{π} increases, then arc $i \rightarrow j$ remains in B^- ;

$$(ii.1.2) \quad c_{ij}^{\pi} < 0.$$

If $x_{ij} \geq l_{ij}$, then Eq. (13b) says $\theta_{ij} = -c_{ij}^{\pi}$; thus, by Relation (19), $d_j \leq d_i - c_{ij}^{\pi}$ or $c_{ij}^{\pi'} \leq 0$. Thus, the reduced cost of $i \rightarrow j$ does not change from case $c_{ij}^{\pi} < 0$ to case $c_{ij}^{\pi'} > 0$. Therefore, after Operation (a), $i \rightarrow j \notin B^+ \cup G^+$. Also, if $x_{ij} < l_{ij}$, then c_{ij}^{π} is changed from case $c_{ij}^{\pi} < 0$ to case $c_{ij}^{\pi} \geq 0$, which means $i \rightarrow j \in B^-$.

$$(ii.2) \quad i \rightarrow j \in G^-.$$

This case is proved in a similar way to Case (ii-1);

$$(ii.3) \quad i \rightarrow j \in R.$$

Using Relations (14), (19), and (20), we get $d_i = d_j$, which means $c_{ij}^{\pi'} = c_{ij}^{\pi}$. Hence, arc $i \rightarrow j$ remains in R .

From the above discussions, we get the next theorem.

Theorem 3.3. In Procedure 1, the following operations do not send any arc of set $B^- \cup G^- \cup R$ into set $B^+ \cup G^+$:

- (a) $\pi'_i = \pi_i + d_i$, for each node i ;
- (b) After Operation (a), δ units of flow is sent along cycle $C_{w \rightarrow v}$.

Proof. By Lemma 3.2, this theorem is true for Operation (a). Now, we prove the claim for Operation (b). The procedure sends δ units flow along $C_{w \rightarrow v} = \bar{P} \cup \{w \rightarrow v\}$ according to Procedure 1; therefore, arc $w \rightarrow v$ leaves set $B^+ \cup G^+$. Thus, we focus on the arcs of \bar{P} . Conclusion 3.2 says that the procedure sends δ units flow in the direction opposite to the direction of path P , and, by the definition of P , we have $P = p_v$ or $P = p_w$. Therefore, by Lemma 3.2, Operation (b) does not send any arc $i \rightarrow j \in B^- \cup G^- \cup R$ into set $B^+ \cup G^+$. ■

Theorem 3.3 says that Procedure 1 does not send any arc $i \rightarrow j \in B^- \cup G^- \cup R$ into set $B^+ \cup G^+$.

Also, in each iteration, there exists at least one arc which leaves set $B^+ \cup G^+$. Thus, after at most m iterations, we have $B^+ \cup G^+ = \emptyset$. The next lemma gives the running time of the procedure.

Lemma 3.4. Procedure 1 runs in $O(m(m + n \log n))$ time.

Proof. By $|B^+ \cup G^+| \leq m$, the number of iterations is at most m . In each iteration, the procedure solves a shortest path problem, which is solved in time $O(m + n \log n)$ [25]. Other operations in each iteration are done in $O(m)$. Hence, the running time of the procedure is $O(m(m + n \log n))$. ■

Putting Lemmas 2.5. and 3.4. together yields our bound on the running time of the Geometric-DMCF algorithm.

Theorem 3.5. The Geometric-DMCF algorithm runs in time $O(m(m + n \log n) \log(nU))$. ■

4. Conclusion

Shigeno et al. [6] is a minimum cost-flow algorithm, which in each phase computes a δ -optimal potential using at most m shortest-path computations. This paper is inspired by Ghiyasvand (2012), which uses the scaling method of Shigeno et al. (2000); but, in each phase, it applies the out-of-kilter idea instead of the most positive cut canceling idea. Our algorithm, in each phase, transforms all out-of-kilter arcs into in-kilter arcs to compute a δ -optimal potential. The algorithm runs in time $O(m(m + n \log n) \log(nU))$, which is $O(m(m + n \log n) \log n)$ under the similarity assumption. This time is the running time of the algorithms by Orlin [3] and Vygen [7], which are the current best strongly polynomial-time algorithms to solve the minimum cost-flow problem.

Acknowledgements

I would like to express great appreciation to the reviewers for their valuable comments and suggestions, which have helped to improve the quality and presentation of this paper.

References

- Edmonds, I. and Karp, R.M. "Theoretical improvements in algorithmic efficiency for network flow problems", *Journal of the Association on Computing Machinery*, **19**, pp. 248-264 (1972).
- Goldberg, A.V. and Tarjan, R.E. "Finding minimum-cost circulations by successive approximation", *Math. Oper. Res.*, **16**, pp. 430-466 (1990).
- Orlin, J.B. "A faster strongly polynomial minimum cost flow algorithm", *Oper. Res.*, **41**, pp. 338-350 (1993).
- Ahuja, R.K., Goldberg, A.V., Orlin, J.B. and Tarjan, R.E. "Finding minimum-cost flows by double scaling", *Mathematical Programming*, **53**, pp. 243-266 (1992).
- Hassin, R. "The minimum cost flow problem: A unifying approach to dual algorithms and a new tree-search algorithm", *Math. Programming*, **25**, pp. 228-239 (1983).
- Shigeno, M., Iwata, S. and McCormick, S.T. "Relaxed most negative cycle and most positive cut canceling algorithms for minimum cost flow", *Mathematics of Operations Research*, **25**, pp. 76-104 (2000).
- Vygen, J. "On dual minimum cost flow algorithms", *Mathematical Methods of Operations Research*, **56**, pp. 101-126 (2002).
- Ahuja, R.K., Magnanti, T.L. and Orlin, J.B., *Network flows: Theory, Algorithms, and Applications*, Prentice-Hall, Englewood Cliffs, NJ (1993).
- Ford, L.R. and Fulkerson, D.R., *Flows in Networks*, Princeton University Press, Princeton, NJ (1962).
- Paparrizos, K., Samaras, N. and Sifaleras, A. "An exterior simplex type algorithm for the minimum cost network flow problem", *Computers and operations research*, **36**, pp. 1176-1190 (2009).
- Fulkerson, D.R. "An out-of-kilter method for minimal cost flow problems", *SIAM J. Appl. Math.*, **9**, pp. 18-27 (1961).
- Minty, G.J. "Monotone networks", *Proc. Roy. Soc. London*, **257**, pp. 194-212 (1960).
- Busaker, R.G. and Gowen, P.J. "A procedure for determining a family of minimal-cost network flow patterns", Technical Report, **15**, O.R.O. (1961).
- Jewell, W.S. "Optimal flow through networks", Technical Report, **8**, M.I.T. (1958).
- Frangioni, A. and Manca, A. "A computational study of cost reoptimization for min-cost flow problems", *INFORMS J. Comput.*, **18**, pp. 61-70, (2006).
- Georgiadis, L., Goldberg, A.V., Tarjan, R.E. and Werneck, R.F. "An experimental study of minimum mean cycle algorithms", *Proc. 6th International Workshop on Algorithm Engineering and Experiments*, *SIAM*, pp. 1-13 (2009).
- Geranis, G., Paparrizos, K. and Sifaleras, A. "On a dual network exterior point simplex type algorithm and its computational behavior", *RAIRO-Operations Research*, **46**(3), pp. 211-234 (2012).
- Kiraly, Z. and Kovacs, P. "Efficient implementations of minimum-cost flow algorithms", *Acta Universitatis Sapientiae*, **4**, pp. 67-118 (2012).
- Kovacs, P. "Minimum-cost flow algorithms: An experimental evaluation", *Optimization Methods and Software*, **30**(1), pp. 94-127 (2015).

20. Sifaleras, A. “Minimum cost network flows: Problems, algorithms, and software”, *Yugoslav Journal of Operations Research*, **23**(1), pp. 3-17 (2013).
21. Ghiyasvand, M. “An $O(m(m + n \log n) \log(nC))$ -time algorithm to solve the minimum cost tension problem”, *Theoretical Computer Science*, **448**, pp. 47-55 (2012).
22. Gabow, H.N. “Scaling algorithms for network problems”, *Journal of Computer and System Science*, **31**, pp. 148-168 (1985).
23. Hoffman, A.J. “Some recent applications of the theory of linear inequalities to extremal combinatorial analysis”, In *American Mathematical Society*, R. Bellman and M. Hall, Eds., pp. 113-127 (1960).
24. Dijkstra, E.W. “A note on two problems in connection with graphs”, *Numerical Mathematics*, **1**, pp. 269-271 (1959).
25. Fredman, M.L. and Tarjan, R.E. “Fibonacci heaps and their uses in improved network optimization algorithms”, *J. Assoc. Comput. Mach.*, **34**, pp. 596-615 (1987).

Biography

Mehdi Ghiyasvand was born in Hamedan, Iran, in 1975. He obtained MS and PhD degrees in Operations Research from Tehran University, Iran, and spent a post-doctoral term at MIT, USA. He is now Associate Professor at Bu-Ali Sina University.