Bulletin of the *Transilvania* University of Braşov • Vol 12(61), No. 1 - 2019
Series III: Mathematics, Informatics, Physics, 107-112
https://doi.org/10.31926/but.mif.2019.12.61.1.9

MINIMUM COST FLOW IN A NETWORK WITH AN OVERESTIMATED ARC CAPACITY

Laura CIUPALĂ*[1] and Adrian DEACONU[2]

Dedicated to the 75th birthday of Professor Eleonor Ciurea

Abstract

There are many situations in which minimum cost flows are solutions to real world problems. Sometimes, in these problems, minor data changes may occur and these will lead to corresponding changes in the networks, in which the practical problems are modeled as minimum cost flow problems. For instance, the capacity of an arc may increase or decrease in time.

In this paper, the case in which the capacity of a given arc decreases by a given value a is studied. We focus on the problem of finding a minimum cost flow in a network that differs from the initial network G, in which a minimum cost flow has been already determined, only by the capacity of a given arc (k, l) which has been decreased by a units. We develop an algorithm for establishing a minimum cost flow in the modified network in $O(anm)$ time, starting from the minimum cost flow in the original network.

2000 *Mathematics Subject Classification:* 90B10, 90C90.
Key words: network flow, minimum cost flow, incremental algorithm.

1 Introduction

The literature on network flow problems is extensive, one of the reasons for this being the widespread and diverse applications of these network optimization problems. From the late 1940s through the 1950s, researchers designed many fundamental algorithms for network flow, maximum flow algorithms and minimum cost flow algorithms being included here. In the past 60 years researchers have made continuous improvements to network flow algorithms, whose computational complexity has been significantly reduced by using enhanced data structures, techniques of scaling the problem data etc.

[1]* *Corresponding author*, Faculty of Mathematics and Informatics, *Transilvania* University of Braşov, Romania, e-mail: laura_ciupala@yahoo.com
[2]Faculty of Mathematics and Informatics, *Transilvania* University of Braşov, Romania, e-mail: a.deaconu@unitbv.ro

The minimum cost flow problem is a complex network flow problem that combines the shortest paths with the maximum flow techniques and it has been studied extensively in the past several decades. The importance of the minimum cost flow problem is also due to the fact that it arises in almost all industries, including agriculture, communications, defense, education, energy, health care, medicine, manufacturing, retailing and transportation. Changes in the networks that appear in these real world problems are likely to happen in time and these changes lead to corresponding changes in the networks in which these practical problems are modeled as minimum cost flow problems.

Let $G = (N, A)$ be a directed graph, defined by a set N of n nodes and a set A of m arcs. Each arc $(x, y) \in A$ has a capacity $c(x, y)$ and a cost $b(x, y)$. We associate with each node $x \in N$ a number $v(x)$ which indicates its supply or demand depending on whether $v(x) > 0$ or $v(x) < 0$. In the directed network $G = (N, A, c, b, v)$, the minimum cost flow problem is to determine the flow $f(x, y)$ on each arc $(x, y) \in A$ which

$$minimize \sum_{(x,y)\in A} b(x, y)f(x, y) \tag{1}$$

subject to

$$\sum_{y|(x,y)\in A} f(x, y) - \sum_{y|(y,x)\in A} f(y, x) = v(x), \qquad \forall x \in N \tag{2}$$

$$0 \leq f(x, y) \leq c(x, y), \qquad \forall (x, y) \in A. \tag{3}$$

A flow f satisfying conditions 2 and 3 is a *feasible* flow.

The value $\sum_{(x,y)\in A} b(x, y)f(x, y)$ is the *cost* of the flow and will be denoted by c_f.

A *pseudoflow* is a function $f : A \to \Re_+$ satisfying only condition 3. For any pseudoflow f the *imbalance* of node x is defined as $e(x) = f(N, x) - f(x, N)$, for all $x \in N$. If $e(x) > 0$ for some node x, we refer to $e(x)$ as the *excess* of node x; if $e(x) < 0$, we refer to $-e(x)$ as the *deficit* of node x. If $e(x) = 0$ for some node x, we refer to node x as *balanced.* Consequently, a flow is a particular case of pseudoflow.

The residual network $G(f) = (N, A(f))$ corresponding to a pseudoflow f is defined as follows. For each arc $(x, y) \in A$ one creates two arcs (x, y) and (y, x). The arc (x, y) has the cost $b(x, y)$ and the residual capacity $r(x, y) = c(x, y) - f(x, y)$ and the arc (y, x) has the cost $b(y, x) = -b(x, y)$ and the residual capacity $r(y, x) = f(x, y)$. The residual network consists only of those arcs with positive residual capacity.

We shall assume that the minimum cost flow problem satisfies the following assumptions:

1. All data (cost, supply/demand and capacity) are integral.

2. The network contains no directed negative cost cycle of infinite capacity.

3. All arc costs are nonnegative.

4. The supplies/demands at the nodes satisfy the condition $\sum_{x \in N} v(x) = 0$ and the minimum cost flow problem has a feasible solution.

5. The network contains an uncapacitated directed path (i.e. each arc in the path has infinite capacity) between every pair of nodes.

All these assumptions can be made without any loss of generality (for details see [1]).

We associate a real number $\pi(x)$ with each node $x \in N$. We refer to $\pi(x)$ as the potential of node x. For a given set of node potentials π, we define the reduced cost of an arc (x, y) as

$$b^{\pi}(x, y) = b(x, y) - \pi(x) + \pi(y).$$

The reduced costs are applicable to the residual network as well as to the original network.

Theorem 1. *[1](a) For any directed path P from node w to node z we have*
$\sum_{(x,y)\in P} b^{\pi}(x, y) = \sum_{(x,y)\in P} b(x, y) - \pi(w) + \pi(z)$
(b) For any directed cycle W we have
$\sum_{(x,y)\in W} b^{\pi}(x, y) = \sum_{(x,y)\in W} b(x, y)$

Theorem 2. *(Negative Cycle Optimality Conditions) [1] A feasible solution f is an optimal solution of the minimum cost flow problem if and only if it satisfies the following negative cycle optimality conditions:*
the residual network $G(f)$ contains no negative cost directed cycle.

Theorem 3. *(Reduced Costs Optimality Conditions) [1] A feasible solution f is an optimal solution of the minimum cost flow problem if and only if some set of node potentials π satisfy the following reduced cost optimality conditions:*
$b^{\pi}(x, y) \geq 0$ *for every arc (x, y) in $G(f)$*

There are two types of classical minimum cost flow algorithms:
1. those that maintain feasible solutions and strive toward optimality
2. those that maintain infeasible solutions that satisfy optimality conditions and strive toward feasibility.

Algorithms of both types have been developed after 1950 and can be found in [1].

The best known classical algorithms of the first type are the cycle-canceling algorithm and the out-of-kilter algorithm. Both of them maintain a feasible flow during their execution. The difference between them is that, while the cycle-canceling algorithm augments flow along negative cycle in the residual network and terminates when there is no more negative cycle in the residual network, which means that the flow is a minimum cost flow, the out-of-kilter algorithm augments flow along the shortest path in order to reach optimality.

The most known classical algorithms of the second type are the successive shortest path algorithm and the primal-dual algorithm. Both of them maintain an optimal pseudoflow. The successive shortest path algorithm augments the flow along shortest paths from excess nodes to deficit nodes in the residual network in order to convert the pseudoflow into a flow. The primal-dual algorithm solves maximum flow problems in order to convert the pseudoflow into a flow.

Starting from the classical algorithms for minimum cost flow, several polynomial-time algorithms have been developed. Most of them have been obtained by using the scaling technique. By capacity scaling, by cost scaling or by capacity and cost scaling, the following polynomial-time algorithms have been developed: capacity scaling algorithm, cost scaling algorithm, double scaling algorithm, repeated capacity scaling algorithm and enhanced capacity scaling algorithm.

Another way to obtain polynomial-time algorithms is by improving the running time of the cycle-canceling algorithm by imposing different rules for selecting the negative cycles, for instance selecting the negative cycle with maximum improvement or the negative cycle with minimum cost.

Because minimum cost flow problems are actually linear programs, another approach for solving them is to use linear programming methodologies. For instance, the relaxation algorithm and simplex method for determining a minimum cost flow have been developed in this manner.

2 Determining a minimum cost flow in a network with an overestimated arc capacity

In real life problems that can be modeled and solved as minimum cost flow problems data may vary in time. For instance, a usual variation in real life problems might imply a decrease of the capacity of one arc in the corresponding network. Suppose that a minimum cost flow has been already determined in the original network (by applying one of the algorithms mentioned in the previous section). In this case, instead of applying in the modified network a known minimum cost algorithm, which starts from a zero flow, one can use the minimum flow cost already established in the original network as a starting point. We focus on the second approach because it is more efficient than the first one.

Let f be a minimum cost flow of cost c_f in the original network $G = (N, A, c, b, v)$. We need to find a minimum cost flow in a network $G' = (N, A, c', b, v)$ having the same structure and the same capacities and costs, except one arc capacity, as G. So, the only difference between these two networks is the capacity of a given arc (k, l), which is by a units smaller in G' than it is in G. So, $c'(x, y) = c(x, y), \forall (x, y) \in A \backslash \{(k, l)\}$ and $c'(k, l) = c(k, l) - a$, where $a > 0$.

Obviously, a smaller capacity of the arc (k, l) may imply that the minimum cost flow in the network G' has a cost greater than c_f, but this is not mandatory. There are two cases that might occur:

Case 1: The residual network $G(f)$ with respect to the minimum cost flow f contains the arc (k, l) having a residual capacity of at least a units. In this case

f is a minimum cost flow in G' too.

Case 2: The residual network $G(f)$ with respect to the minimum cost flow f does not contain the arc (k, l) or it contains the arc (k, l) but its residual capacity is less than a. In this case, decreasing the capacity of the arc (k, l) by a units implies decreasing the flow on the arc (k, l) by $f(k, l) - c(k, l) + a$, which is a positive value, in order to meet the boundary constraints 3 in the modified network G'. This means that the node k will have an excess equal to $f(k, l) - c(k, l) + a$ and the node l will have a deficit equal to $-(f(k, l) - c(k, l) + a)$. So, in this way an optimal pseudoflow in the network G' is obtained. Consequently, we will use an algorithm that maintains the optimality of the pseudoflow and strives for its feasibility, which is the following:

OverestArcCap Algorithm;
Begin
  $\pi = 0$;
  $f' = f$;
  if $f'(k, l) > c(k, l) - a$ **then**
  begin
    $f'(k, l) = c(k, l) - a$;
    determine the residual network $G'(f')$;
    while $e(k) > 0$ **do**
    begin
      determine shortest path distances d from node k to all nodes in
        $G'(f')$ with respect to the reduced cost b^{π};
      let P be a shortest path from node k to node l;
      $\pi = \pi - d$;
      augment $\min\{e(k), \min\{r(x, y) | (x, y) \in P\}\}$ units of flow along
        the path P;
      update the residual network $G'(f')$;
    end;
  end;
end.

Theorem 4. *The OverestArcCap algorithm determines a minimum cost flow problem in a network with an overestimated arc capacity G' starting with a minimum cost flow in the original network G in $O(anm)$ time.*

Proof. A minimum cost flow f in the network G is also a minimum cost flow in G' if $f(k, l) \leq c(k, l) - a$; otherwise the flow on the arc (k, l) has to be decreased by $f(k, l) - c(k, l) + a$ units in order to meet the boundary constraints 3. In this way an optimal pseudoflow in the network G' is obtained, where the node k has an excess equal to $f(k, l) - c(k, l) + a$ and the node l has a deficit equal to $-(f(k, l) - c(k, l) + a)$. By sending flow along the shortest paths, with respect to reduced costs, from k to l, both the excess of k and the absolute value of the deficit of l are reduced. Since we assumed that all the data are integral, those decrements are by at least one unit each, which means that after $O(a)$ iterations all the nodes will be balanced and the optimal pseudoflow will become a minimum cost flow. Since determining a shortest path gives the complexity of an iteration and it can be computed in $O(nm)$ time using Bellman and Ford's algorithm, it follows that the algorithm complexity is $O(anm)$.

□

References

[1] Ahuja, R., Magnanti, T., Orlin, J., *Network Flow. Theory, Algorithms and Applications*, Prentice Hall, New Jersey, 1993.

[2] Bang-Jensen, J., Gutin, G., *Digraphs, Theory, Algorithms and Applications*, Springer-Verlag, London, 2001.

[3] Ciupala, L., *Maximum Flow in a Network with an Underestimated Arc Capacity*, Bulletin of the *Transilvania* University of Braşov, **10(59)** (2017), no. 1, 203-206.