

Research Article

An Efficient Algorithm for Solving Minimum Cost Flow Problem with Complementarity Slack Conditions

Yongwen Hu ^{1,2}, Xiao Zhao,^{1,2} Jing Liu,^{1,3} Binyuan Liang,^{1,2} and Chao Ma ^{1,3}

¹Hubei Key Laboratory of Power System Design and Test for Electrical Vehicle, Hubei University of Arts and Science, Xiangyang 441053, China

²School of Mechanical Engineering, Hubei University of Arts and Science, Xiangyang 441053, China

³School of Automotive and Traffic Engineering, Hubei University of Arts and Science, Xiangyang 441053, China

Correspondence should be addressed to Chao Ma; [mma1123@163.com](mailto:mmma1123@163.com)

Received 10 July 2019; Revised 11 October 2019; Accepted 6 November 2019; Published 12 February 2020

Academic Editor: Marcello Pellicciari

Copyright © 2020 Yongwen Hu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents an algorithm for solving a minimum cost flow (MCF) problem with a dual approach. The algorithm holds the complementary slackness at each iteration and finds an augmenting path by updating node potential iteratively. Then, flow can be augmented at the original network. In contrast to other popular algorithms, the presented algorithm does not find a residual network, nor find a shortest path. Furthermore, our algorithm holds information of node potential at each iteration, and we update node potential within finite iterations for expanding the admissible network. The validity of our algorithm is given. Numerical experiments show that our algorithm is an efficient algorithm for the MCF problem, especially for the network with a small interval of cost of per unit flow.

1. Introduction

The minimum cost flow (MCF) problem is to find a minimal cost of a given amount flow from a set of supply nodes to a set of demand nodes in a directed network where each arc is assigned a cost and a capacity constraint. MCF problem plays a fundamental role in network optimization. On the one hand, many network flow problems, such as shortest path problem, maximal flow problem, and transportation problem, can be viewed as special cases of the MCF problem. On the other hand, the MCF problem has a remarkable wide range of applications [1] in the real world, for example, scheduling [2], agriculture [3], communications [4], health care [5, 6], and transportation [7].

Since the MCF problem is a linear program, we can also find the optimal solution with linear programming methodologies. Indeed, many of various optimality conditions about the MCF problem and linear programming are derived by researchers. Thus, we can develop efficient algorithms for the MCF problem with theory of duality.

Several algorithms have been designed for solving the MCF problem over past decades both from theoretical and

from practical aspects. Many algorithms combining ingredients of both shortest path and maximal flow are developed to solve the MCF problem. These algorithms find an optimal solution by solving a sequence of the shortest path problem within maximum flow-like residual networks and augmenting path, such as the cycle-canceling algorithm [8], successive shortest path algorithm [9], and primal-dual and out-of-kilter algorithms [10]. They are equivalent in the sense that these algorithms perform the same sequence of strategy, and the key step is defining the residual network iteratively. Moreover, both cost and capacity are to be considered for defining the residual network, and some more arcs are to be introduced with negative cost, which make the original problem much more difficult to solve. Although none of the algorithms mentioned above has been shown to be a well complexity, they usually behave much better in practice compared to their theoretical analysis. Edmonds and Karp [11] proposed the first polynomial algorithm. Currently, an efficient polynomial time algorithm for the MCF problem was proposed by Orlin [12] and Vygen [13]. Some theoretical results and algorithms are given in [14–18] for solving the MCF problem or its variants.

Recently, Vieira et al. [19] evaluated the practical performance of 4 algorithms on real networks. And Vernet et al. [20] used a success shortest path algorithm for dynamic graph. A recent survey on the MCF problem is given in [21].

Our purpose in this paper is to present a new algorithm for solving the MCF problem. We consider the dual problem of MCF problem and derive the optimality conditions. The algorithm can find an augmenting path directly in a directed network when some dual variables (nodes potentials) of the MCF problem are updated iteratively in finite steps. In contrast to other algorithms, we do not need to define any residual network.

This paper is organized as follows: Section 2 derives optimality conditions of MCF problem with duality theory. Section 3 gives an iterative algorithm for the MCF problem. Section 4 discusses validity of the presented algorithm. Numerical experiment results and analysis of computational experiments are reported in Section 5. Section 6 contains some concluding remarks on the proposed algorithm for solving the MCF problem.

2. Optimality Conditions

Let $G = (N, A, C, U)$ be a directed network with a node set N , an arc set A , a cost set C , and a capacity set U . Furthermore, a cost $c_{ij} \in C$ and a nonnegative capacity $u_{ij} \in U$ associated with each arc $(i, j) \in A, \forall i, j \in N$. And we associate with each node $i \in N$ a number b_i which indicates its supply or demand depending on whether $b_i < 0$ or $b_i > 0$. Let node s and node t be the source node and sink node, respectively. And we assume that $\sum_{i \in N, i \neq s, t} b_i = 0$. The MCF problem can be stated as follows:

$$\text{MCF} \left\{ \begin{array}{l} \min z(x) = \sum_{(i,j) \in A} c_{ij} x_{ij} \\ \sum_{j: (i,j) \in A} x_{ij} - \sum_{j: (j,i) \in A} x_{ji} = v^*, \quad i = s, \\ \sum_{j: (i,j) \in A} x_{ij} - \sum_{j: (j,i) \in A} x_{ji} = -v^*, \quad i = t, \\ \sum_{j: (i,j) \in A} x_{ij} - \sum_{j: (j,i) \in A} x_{ji} = 0, \quad \forall i \in N, i \neq s, t, \\ 0 \leq x_{ij} \leq u_{ij}, \quad \forall (i, j) \in A. \end{array} \right. \quad (1)$$

The dual problem of MCF is

$$\text{DMCF} \left\{ \begin{array}{l} \max \omega = v^* p_s - v^* p_t + \sum_{(i,j) \in A} u_{ij} p_{ij} \\ p_i - p_j + p_{ij} \leq c_{ij}, \quad \forall (i, j) \in A, \\ p_i \text{ is free}, \quad \forall i \in N, \\ p_{ij} \leq 0, \quad \forall (i, j) \in A, i \in N, \end{array} \right. \quad (2)$$

where p_i and p_{ij} are the dual variables of the MCF problem. And we refer to p_i and p_{ij} as the potential of node i and arc $(i, j), \forall i \in N, \forall (i, j) \in A$, respectively.

Let $x = \{x_{ij}\}$ and $p = \{p_i, p_{ij}\}$ be a feasible solution of MCF and DMCF, respectively. If feasible solution $x = \{x_{ij}\}$

and $p = \{p_i, p_{ij}\}$ are optimal solutions, according to theorem of complementary slackness, the following conditions will be satisfied:

$$x_{ij} = 0, \quad \text{if } p_i - p_j + p_{ij} < c_{ij}, \forall i \in N, (i, j) \in A, \quad (3)$$

$$x_{ij} = u_{ij}, \quad \text{if } p_{ij} < 0, \forall (i, j) \in A. \quad (4)$$

Since p_i is free $\forall i \in N$, let $p_{ij} = \min\{0, c_{ij} + p_j - p_i\}$, then we have $p_{ij} \leq 0$. Therefore, we can find a feasible solution $p = \{p_i, p_{ij}\}$ of DMCF where the value of p_{ij} is defined above $\forall i \in N, \forall (i, j) \in A$. Thus, the complementary slackness conditions (3) and (4) are equal to the following conditions:

$$x_{ij} = 0, \quad \text{if } p_i - p_j < c_{ij}, \forall i \in N, (i, j) \in A, \quad (5)$$

$$x_{ij} = u_{ij}, \quad \text{if } p_i - p_j > c_{ij}, \forall i \in N, (i, j) \in A. \quad (6)$$

Therefore, if feasible solution $x = \{x_{ij}\}$ is an optimal solution of MCF, then x will satisfy conditions (5) and (6). On the contrary, for the arc $(i, j) \in A$ satisfying $p_i - p_j = c_{ij}$, we have $0 \leq x_{ij} \leq u_{ij}$. In other words, we can find an augmenting path where each arc $(i, j) \in A$ on the path such that $p_i - p_j = c_{ij}$ when the value of flow is not the maximum flow in a network. We refer to arc $(i, j) \in A$ satisfying $p_i - p_j = c_{ij}$ as the admissible arc, and let $G' = (N', A', C', U')$ be an admissible network satisfying that $N' \subseteq N, A' \subseteq A, C' \subseteq C, U' \subseteq U, p_i - p_j = c_{ij}, \forall (i, j) \in A'$.

3. An Iterative Algorithm for Solving MCF

It is clear that $x_{ij} = 0, \forall (i, j) \in A$ is an optimal solution with a flow valued 0. In order to get an optimal solution of MCF with a given amount of flow, we iteratively augment flow holding the complementary slackness condition (5) and (6) at each iteration. And flow can be augmented after we find an augmenting path within the admissible network. Moreover, a new admissible network can be given by updating the potential of node when we cannot find an augmenting path and flow is not the maximal flow. Let V be a number of given amount of flow. We now describe formally our algorithm for solving the MCF problem (Algorithm 1).

Next, we will give the validity of the proposed algorithm.

4. Validity of the Proposed Algorithm

The validity of the above algorithm depends mainly from the following results:

Lemma 1. *When the current flow is not the maximal flow in a network, at least one admissible arc can be found after updating node potential.*

Proof. It is clear that there is at least one augmenting path if current flow is not the maximal flow in a directed network. Therefore, we have $x_{ij} < u_{ij}$ for some forward arc (i, j) or $x_{ji} > 0$ for some backward arc $(j, i), \forall i \in S, j \in \bar{S}$. Otherwise, node j will be labeled and $j \in S$ according to the presented

Initialization: initialize node potential $p_i = 0, \forall i \in N$. Let $x_{ij} = 0, (i, j) \in A$, be an initial optimal solution with a flow valued 0. Label source node $(0, +\infty)$. Set $S = \{s\}, \bar{S} = N - S$.

While $\sum_{i:(s,i) \in A} x_{si} \geq V$ or θ cannot be found

While $t \notin S$

For $i \in S$

$$\theta_i = \min \left\{ \begin{array}{l} \min_{\substack{j:(i,j) \in (S,\bar{S}) \cap A \\ p_i - p_j < c_{ij}}} (c_{ij} - p_i + p_j), \\ \min_{\substack{j:(j,i) \in (\bar{S},S) \cap A \\ p_j - p_i > c_{ij}}} (p_j - p_i - c_{ij}) \end{array} \right\}$$

End

$$\theta = \min_{i \in S} \{\theta_i\}$$

$$p_i = \begin{cases} p_i + \theta, & i \in S, \\ p_i, & i \in \bar{S}. \end{cases}$$

If $p_i - p_j = c_{ij}$ **then**
 $(i, j) \in R$

End

If $t \in S$ **then**

For $\forall i \in S, j \in \bar{S}$

If $(i, j) \in R$ and $x_{ij} < u_{ij}$
Label node j with a label (i, δ_j) , where $\delta_j = \min\{\delta_i, u_{ij} - x_{ij}\}, S = S \cup \{j\}$.

End

If $(j, i) \in R$ and $x_{ji} > 0$ **then**
Label node j with a label $(-i, \delta_j)$, where $\delta_j = \min\{\delta_i, x_{ji}\}, S = S \cup \{j\}$.

End

End

$$x_{ij} = \begin{cases} x_{ij} + \delta_i, & x_{ij} \in \mu^+, \\ x_{ij} - \delta_i, & x_{ij} \in \mu^-, \\ x_{ij}. & \end{cases}$$

End

End

End

ALGORITHM 1

algorithm. Then, we can find θ and update node potential by Algorithm 1. Thus, at least one new admissible forward arc $(i, j), p_i - p_j = c_{ij}$ or backward arc $(j, i), p_j - p_i = c_{ji}$ will be found. \square

Theorem 1. *The proposed algorithm finds an augmenting path in a finite iteration when the current flow is not the maximal flow.*

Proof. Let n be the number of nodes in a network $G = (N, A, C, U)$, and there are at most $n - 1$ arcs on a path μ from the source node to sink node. According to Lemma 1, the presented algorithm can find an augmenting path in $n - 1$ iterations, which implies Theorem 1. \square

Theorem 2. *After augmenting a flow of value v by our algorithm, then the new feasible flow valued $v' = v + \delta_i$ is the minimum cost flow valued v' .*

Proof. the algorithm begins a feasible flow $x_{ij} = 0, \forall (i, j) \in A$, and feasible flow $x_{ij} = 0, \forall (i, j) \in A$ is the minimum cost flow valued 0. Furthermore, the proposed algorithm holds the complementary slackness condition (5) and (6) at each iteration. Therefore, augmented flow updated by Algorithm 1 is the optimal solution of a minimum cost flow valued v' , which gives Theorem 2. \square

5. Numerical Experiment

In this section, the computational experimental results are presented in order to verify the performance of the presented algorithm. The presented algorithm has been coded with Matlab and run on PC with Intel® Core™ i5-7300HQ CPU (2.5 GHz) and RAM 8 GB. The tested problem is generated by the well-known Waxman network topology generator with the method proposed by Waxman [22]. All nodes of the generated problem are a Poisson process in the plane with scaled. Nodes i and j are connected with probability $p(i, j) = \alpha e^{-d(i,j)/\beta L}$, where α is the maximal link probability, β is a parameter to control length of the arcs, $d(i, j)$ is the Euclidean distance between node i and j , and L is the maximal distance between any two nodes. For the network generated by the Waxman generator, we randomly give a c_{ij} and u_{ij} for each arc $(i, j) \in A$. The computational experiments are performed on a network with different scales (n varied from 10 to 200). Let $n(m)$ be the number of nodes (arcs) of the generated network. And for a fixed n and m , a set of 10 instances are solved by our proposed algorithm and the successive shortest path algorithm (SSPA) [9] in order to make computational comparison. Furthermore, for a fixed n , our algorithm and SSPA are performed on a network with different number of arcs and $c_{ij}, i, j = 1, 2, \dots, n$. For the sake of simplicity, the algorithm proposed in [9] is named as Algorithm A1 and the algorithm

TABLE 1: CPU time on network with n varying from 10 to 200 and two different intervals of c_{ij} .

No. of nodes	No. of arcs (average)	AVER. CPU time (s), $c \in [1, 10]$		AVER. CPU time (s), $c \in [1, 1000]$	
		A1	A2	A1	A2
		10	32	7.964	0.050
10	43	8.245	0.059	8.306	0.064
15	62	8.430	0.060	8.517	0.078
15	99	8.642	0.079	8.726	0.085
20	86	8.578	0.069	8.469	0.086
20	184	8.860	0.102	8.872	0.131
30	210	8.954	0.087	8.936	0.164
30	418	9.014	0.166	9.201	0.258
40	371	8.976	0.092	9.005	0.255
40	763	9.356	0.278	9.428	0.423
50	572	9.169	0.127	9.225	0.397
50	1186	9.628	0.364	9.617	0.673
80	1578	9.725	0.247	9.829	1.056
80	3083	10.427	0.853	10.526	2.502
100	2616	10.320	0.344	10.431	1.734
100	4857	10.569	1.550	10.726	6.438
200	9488	18.351	2.069	19.251	37.181
200	19482	210.358	23.429	221.852	143.759

proposed in this paper as A2. The numerical experiment results are reported in Table 1. The column titled ‘‘AVER.’’ in Table 1 provides an average CPU time out of 10 runs in the execution.

The numerical experiments indicate the following observations.

A2 outperforms A1 in terms of CPU time in all cases. A2 runs approximately 9–140 times faster than A1 as ‘‘ n ’’ varies from 10 to 200 with $c_{ij} \in [1, 10], \forall (i, j) \in A$. For a larger interval of $c_{ij} \in [1, 1000], \forall (i, j) \in A$, A2 takes more CPU time compared with a smaller interval of $c_{ij} \in [1, 10], \forall (i, j) \in A$. As the proposed algorithm described, the proposed algorithm holds information of node potential at each iteration, and we update node potential for expanding the admissible network. Therefore, we may find several admissible arcs after updating the node potential in an iteration for a smaller interval of c_{ij} , which implies that we find augmenting path(s) more efficiently compared with a larger interval of $c_{ij}, \forall (i, j) \in A$.

Furthermore, the numerical experiments show that the increased CPU time of A2 is greater than A1 when the number of nodes increased. And the largest number of nodes in the network generated by the Waxman generator on our PC is 200. Therefore, for a larger number of nodes (great than 200) and a great interval of $c_{ij} \in [1, 1000], \forall (i, j) \in A$ in a network, A2 will take more CPU time compared to A1 for solving the MCF problem.

6. Conclusion

In this paper, we proposed an algorithm for solving the MCF problem with a dual approach which holds a complementary slackness conditions at each iteration. The key step of our algorithm is updating node potential, and flow can be augmented after updating node potentials in finite steps.

Unlike existing algorithms such as successive shortest path algorithm and primal-dual algorithm, our algorithm does not find the residual network, nor find the shortest path iteratively. We conduct a numerical experiment to investigate the performance of the proposed algorithm. The results of the numerical experiment show that our algorithm reduces CPU time greatly compared with SSPA.

In this paper, the cost of flow is fixed, and we plan to develop an algorithm for solving the MCF problem with fuzzy cost and constraints. In practice, it is worthwhile to study the case that we have a close-to-optimal solution for the MCF problem. It is fundamentally important to transform the multiobject assignment problem and multiobject transportation problem into the MCF problem, which could be our further study.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This work was supported by the Hubei Superior and Distinctive Discipline Group of ‘‘Mechatronics and Automobiles’’ under grant XKQ2019005, the Youth Project of Humanities and Social Sciences financed by the Ministry of Education under grant 17YJC630084, and the Starting Research Fund from the Hubei University of Arts and Science. The authors sincerely appreciate all the supports.

References

- [1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms and Applications*, Prentice-Hall, New Jersey, NJ, USA, 1993.
- [2] H. C. Lawler, ‘‘The preparation of a soluble acetylcholinesterase from brain,’’ *Biochimica et Biophysica Acta (BBA)—Specialized Section on Enzymological Subjects*, vol. 81, no. 2, pp. 280–288, 1964.
- [3] W. S. Jewell, ‘‘Warehousing and distribution of a seasonal product1,’’ *Naval Research Logistics Quarterly*, vol. 4, no. 1, pp. 29–34, 1957.
- [4] M. Segal, ‘‘The operator-scheduling problem: a network-flow approach,’’ *Operations Research*, vol. 22, no. 4, pp. 808–823, 1974.
- [5] C. H. Slump and J. J. Gerbrands, ‘‘A network flow approach to reconstruction of the left ventricle from two projections,’’ *Computer Graphics and Image Processing*, vol. 18, no. 1, pp. 18–36, 1982.
- [6] C. Sapountzis, ‘‘Allocating blood to hospitals from a central blood bank,’’ *European Journal of Operational Research*, vol. 16, no. 2, pp. 157–162, 1984.
- [7] J. Edmonds and E. L. Johnson, ‘‘Matching, Euler tours and the Chinese postman,’’ *Mathematical Programming*, vol. 5, no. 1, pp. 88–124, 1973.

- [8] M. Klein, "A primal method for minimal cost flows with applications to the assignment and transportation problems," *Management Science*, vol. 14, no. 3, pp. 205–220, 1967.
- [9] R. G. Busaker and P. J. Gowen, "A procedure for determining minimal-cost network flow patterns," in *Operational Research Office*, John Hopkins University, Baltimore, MD, USA, 1961.
- [10] D. R. Fulkerson, "An out-of-kilter method for minimal-cost flow problems," *Journal of the Society for Industrial and Applied Mathematics*, vol. 9, no. 1, pp. 18–27, 1961.
- [11] J. Edmonds and R. M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *Journal of the ACM*, vol. 19, no. 2, pp. 248–264, 1972.
- [12] J. B. Orlin, "A faster strongly polynomial minimum cost flow algorithm," *Operations Research*, vol. 41, no. 2, pp. 338–350, 1993.
- [13] J. Vygen, "On dual minimum cost flow algorithms," *Mathematical Methods of Operations Research (ZOR)*, vol. 56, no. 1, pp. 101–126, 2002.
- [14] N. A. El-Sherbeny, "Algorithm of fuzzy minimum cost flow problem with fuzzy time-windows," *Global Journal of Pure and Applied Mathematics*, vol. 14, no. 2, pp. 219–231, 2018.
- [15] Z. R. Bogdanowicz, "A new efficient algorithm for optimal assignment of smart weapons to targets," *Computers & Mathematics with Applications*, vol. 58, no. 10, pp. 1965–1969, 2009.
- [16] A. Quaglino and R. Krause, "Towards a multigrid method for the minimum-cost flow problem," 2016, <https://arxiv.org/abs/1612.00201>.
- [17] A. Chassein and A. Kinscherff, "Complexity of strict robust integer minimum cost flow problems: an overview and further results," *Computers & Operations Research*, vol. 104, pp. 228–238, 2019.
- [18] A. V. Goldberg, S. Hed, H. Kaplan, and R. E. Tarjan, "Minimum-cost flows in unit-capacity networks," *Theory of Computing Systems*, vol. 61, no. 4, pp. 987–1010, 2017.
- [19] C. L. D. S. Vieira, M. M. M. Luna, and J. M. Azevedo, "Minimum-cost flow algorithms: a performance evaluation using the Brazilian road network," *World Review of Intermodal Transportation Research*, vol. 8, no. 1, pp. 3–21, 2019.
- [20] M. Vernet, M. Drozdowski, Y. Pigné, and E. Sanlaville, "Successive shortest path algorithm for flows in dynamic graphs," in *Proceedings of the 16th Cologne-Twente Workshop on Graphs and Combinatorial Optimization*, Paris, France, June 2018.
- [21] A. Sifaleras, "Minimum cost network flows: problems, algorithms, and software," *Yugoslav Journal of Operations Research*, vol. 23, no. 1, pp. 3–17, 2013.
- [22] B. M. Waxman, "Routing of multipoint connections," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 9, pp. 1617–1622, 1988.