



Inverse Multi-objective Shortest Path Problem Under the Bottleneck Type Weighted Hamming Distance

Mobarakeh Karimi, Massoud Aman, Ardeshir Dolati

► To cite this version:

Mobarakeh Karimi, Massoud Aman, Ardeshir Dolati. Inverse Multi-objective Shortest Path Problem Under the Bottleneck Type Weighted Hamming Distance. 2nd International Conference on Topics in Theoretical Computer Science (TTCS), Sep 2017, Tehran, Iran. pp.34-40, 10.1007/978-3-319-68953-1_4 . hal-01760645

HAL Id: hal-01760645

<https://inria.hal.science/hal-01760645>

Submitted on 6 Apr 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Inverse Multi-Objective Shortest Path Problem Under the Bottleneck Type Weighted Hamming Distance

Mobarakeh Karimi¹, Massoud Aman^{2*} and Ardeshir Dolati³

¹ Department of Mathematics, University of Birjand, Birjand, Iran
mobarake.karimi@birjand.ac.ir

² Department of Mathematics, University of Birjand, Birjand, Iran
mamann@birjand.ac.ir

³ Department of Computer Science, Shahed University, Tehran, Iran
dolati@shahed.ac.ir

Abstract. Given a network $G = (N, A, C)$ and a directed path P^0 from the source node s to the sink node t , an inverse multi-objective shortest path problem is to modify the cost matrix C so that P^0 becomes an efficient path and the modification is minimized. In this paper, the modification is measured by the bottleneck type weighted Hamming distance and is proposed an algorithm to solve the inverse problem. Our proposed algorithm can be applied for some other inverse multiobjective problem. As an example, we will mention how the algorithm is used to solve the inverse multi-objective minimum spanning tree problem under the bottleneck type weighted Hamming distance.

Keywords: Multi-objective optimization, Shortest path problem, Inverse problem, Hamming distance

1 Introduction

The inverse shortest path problem (ISPP) is one of the most typical problems of the inverse optimization, which is to make a predetermined solution becomes the optimal solution after modifications. This problem has attracted many attentions recently due to its broad applications in practice such as the traffic modeling and the seismic tomography (see, e.g., [5] and [10]). For example, assume that in a road network, we would like to modify the costs of the crossing such that a special path between two given nodes becomes optimum in order that, for some reason, the users select this path. To do this, we need to solve an ISPP.

In 1992, Burton and Toint [2] first formulated the ISPP using the l_2 norm to measure the modification. Zhang et. al [13] showed that the ISPP is equivalent to solving a minimum weight circulation problem when the modifications are measured by the l_1 norm. In [12], a column generation scheme is developed to solve the ISPP under the l_1 norm. Ahuja and Orlin [1] showed that the

* corresponding author.

ISPP under the l_1 norm can be solved by solving a new shortest path problem. For the l_∞ norm, they showed that the problem reduces to a minimum mean cycle problem. In [11], it is shown that all feasible solutions of the ISPP form a polyhedral cone and the relationship between this problem and the minimum cut problem is discussed. Duin and Volgenant [3] proposed an efficient algorithm based on the binary search technique to solve the ISPP under the bottleneck type Hamming distance (BWHD). In [9], Tayyebi and Aman extended their method to solve the inverse minimum cost flow problem and the inverse linear programming problem.

As with most real-world optimization problems, there is usually more than one objective that has to be taken into account, thus leading to multi-objective optimization problems (MOP) and inverse multi-objective optimization problems (IMOP). IMOP consists of finding a minimal adjustment of the objective functions coefficients such that a given feasible solution becomes efficient. Ronald et. al. [7] proposed an algorithm to solve the inverse multi-objective combinatorial optimization problems under the l_∞ norm.

In this paper, we propose an algorithm to solve the inverse multi-objective shortest path problem under the BWHD. Our proposed algorithm can be used for solving the inverse of the multi-objective version of some problems under the BWHD. As an example, we apply the algorithm for the inverse multi-objective minimum spanning tree problem under the BWHD.

2 Preliminaries

The notations and definitions used in this paper are given in this section. Let $x, y \in \mathbb{R}^q$ be two vectors. $x \leq y$ iff $x_k \leq y_k$ for every $k \in \{1, \dots, q\}$ and $x \neq y$. Let $G = (N, A, C)$ be a directed network consisting of a set of nodes $N = \{1, 2, \dots, n\}$, a set of arcs $A \subseteq N \times N$ with $|A| = m$ and a cost matrix $C \in \mathbb{R}^{m \times q}$. In the matrix C , we denote the row corresponding to the arc $a \in A$ by the vector $C(a)$. This vector is called the cost of the arc a . The element k of $C(a)$ is denoted by $C^k(a)$. For each $i_1, i_r \in N$ a directed path from i_1 to i_r in G is a sequence of nodes and arcs $i_1 - a_1 - i_2 - a_2 - \dots - i_{r-1} - a_{r-1} - i_r$ satisfying the properties that for all $1 \leq k \leq r-1$, $(i_k, i_{k+1}) \in A$ and for all $k, l \in \{1, \dots, r\}$, $i_k \neq i_l$ if $k \neq l$. For each path P in G , the cost of P is defined as $C(P) = \sum_{a \in P} C(a)$. A path P from i to j is called an efficient path if there is no other path P' from i to j such that $C(P') \leq C(P)$. Let $s, t \in N$ be two given nodes called the source and sink node, respectively. The multi-objective shortest path problem (MSPP) is to find all efficient directed paths from s to t .

Theorem 1. [8] *The bicriterion shortest path problem is NP-complete.*

In [4] the multi-objective label setting algorithm is presented in the case that the cost of the arcs is nonnegative. Also the multi-objective label correcting algorithm is presented in the other case.

For a given path P^0 from s to t in G , The inverse multi-objective shortest path problem (IMSP) is to find a matrix $D \in \mathbb{R}^{m \times q}$ such that

- (a) P^0 is an efficient path in the network $G = (N, A, D)$;
- (b) For each $a \in A$ and $k \in \{1, \dots, q\}$, $-L^k(a) \leq D^k(a) - C^k(a) \leq U^k(a)$, where $L^k(a), U^k(a) \geq 0$ are given bounds for modifying cost $C^k(a)$;
- (c) The distance between C and D is minimized.

The distance between C and D can be measured by various matrix norms. Also these matrices can be converted to two vectors, by a vectorization method, and a vector norm is used. Let each arc a has an associated penalty $w(a) \in \mathbb{R}_{\geq}^q$. In this paper, we use the BWHD defined as follows:

$$H_w(C, D) = \max_{a \in A, k \in \{1, \dots, q\}} w^k(a) \cdot H(C^k(a), D^k(a)), \quad (1)$$

where $H(C^k(a), D^k(a))$ is the Hamming distance, i.e.

$$H(C^k(a), D^k(a)) = \begin{cases} 1 & \text{if } C^k(a) \neq D^k(a), \\ 0 & \text{if } C^k(a) = D^k(a). \end{cases} \quad (2)$$

3 The IMSPP under the BWHD

In this section, the IMSPP under the BWHD is considered and an algorithm is proposed to solve it. Let $G = (N, A, C)$ be a network with a source node s and a sink node t . Assume that P^0 is a given directed path from s to t in G . We can write the IMSPP under the BWHD as follows:

$$\begin{aligned} \min \quad & H_w(C, D), \\ \text{s.t.} \quad & P^0 \text{ is an efficient path from } s \text{ to } t \text{ in } G = (N, A, D), \\ & -L^k(a) \leq D^k(a) - C^k(a) \leq U^k(a), \quad \forall a \in A, \quad \forall k \in \{1, \dots, q\} \\ & D \in \mathbb{R}^{m \times q}, \end{aligned} \quad (3)$$

where $w : A \rightarrow \mathbb{R}^q$ is the arc penalties function and for each $a \in A$ and $k \in \{1, \dots, q\}$, $L^k(a)$ and $U^k(a)$ are the bounds for modifying cost $C^k(a)$. Assume that $w_1 \leq w_2 \leq \dots \leq w_{qm}$ denote the sorted list of the arc penalties. For each $k \in \{1, \dots, q\}$ and $r \in \{1, 2, \dots, qm\}$, we define $A_r^k = \{a \in A : w^k(a) \leq w_r\}$ and the matrix D_r with the following elements is defined:

$$D_r^k(a) = \begin{cases} C^k(a) & \text{if } a \in A \setminus A_r^k, \\ C^k(a) + U^k(a) & \text{if } a \in A_r^k \setminus P^0, \\ C^k(a) - L^k(a) & \text{if } a \in A_r^k \cap P^0. \end{cases} \quad (4)$$

The following theorem provides a helpful result for presenting our algorithm.

Theorem 2. *If D is a feasible solution to the problem (3) with the objective value w_r , then D_r defined in (4) is also feasible whose objective value is less than or equal to w_r .*

Proof. It is easily seen that D_r satisfies the bound constraints and its objective value is not greater than w_r . On the contrary, suppose that P^0 is not efficient

in $G = (N, A, D_r)$. This means that there exists a path P from s to t such that $D_r(P) \leq D_r(P^0)$. We prove that $D(P) \leq D(P^0)$. Hence P^0 is dominated by P in $G = (N, A, D)$ which contradicts the feasibility of D for (3).

The inequality $D_r(P) - D_r(P^0) \leq 0$ implies that

$$\begin{aligned} D(P) - D(P^0) &\leq D_r(P^0) - D_r(P) + D(P) - D(P^0) \\ &= \sum_{a \in P^0} (D_r(a) - D(a)) + \sum_{a \in P} (D(a) - D_r(a)) \\ &= \sum_{a \in (P^0 \setminus P)} (D_r(a) - D(a)) + \sum_{a \in (P \setminus P^0)} (D(a) - D_r(a)). \end{aligned} \quad (5)$$

By the definition of D_r and feasibility of D for (3), all the phrases of the right hand side of (5) are nonpositive. Therefore $D(P) - D(P^0) \leq 0$, which completes the proof.

The following theorem is concluded immediately from Theorem 2.

Corollary 1. *If the optimal objective value of (3) is w_r , then D_r defined in (4) is an optimal solution to (3).*

The next theorem help us to find the optimal solution by a binary search on the set of the penalties.

Theorem 3. *If D_r is a feasible solution to the problem (3), then D_{r+1} is also feasible.*

Proof. On the contrary, suppose that D_{r+1} is not feasible to (3). Hence P^0 is not efficient in $G = (N, A, D_{r+1})$. Thus there exists a path P from s to t such that $D_{r+1}(P) \leq D_{r+1}(P^0)$. Analysis similar to that in the proof of Theorem 2 shows that $D_r(P) \leq D_r(P^0)$ which contradicts the feasibility of D_r for (3).

Based on the previous results, we propose an algorithm to solve the IMSPP under the BWHD. We find the minimum value of $r \in \{1, \dots, qm\}$ such that P^0 is an efficient path in $G = (N, A, D_r)$. For checking this condition, we can use the proposed algorithm in [4] to find all efficient paths from s to t in $G = (N, A, D_r)$. According to Theorem 3, the minimum value of r can be found by a binary

search on the set of the penalties. We now state our proposed algorithm formally.

Algorithm 1

Step 1. Sort the arc penalties. Suppose $w_1 \leq w_2 \leq \dots \leq w_{qm}$ is the sorted list.
Step 2. Set $i = \lfloor \frac{qm}{2} \rfloor$ and $r = qm$.
Step 3. construct the matrix D_r defined in (4).
Step 4. If P^0 is an efficient path in $G = (N, A, D_r)$, then go to Step 5. Otherwise, go to Step 6.
Step 5. If $i > 0$, then update $r = r - i$, $i = \lfloor \frac{i}{2} \rfloor$ and go to Step 3. Otherwise, go to Step 8.
Step 6. If $r = qm$, then the problem (3) is infeasible and stop. Otherwise update $r = r + i$, $i = \lfloor \frac{i}{2} \rfloor$ and go to Step 7.
Step 7. If $i > 0$ go to Step 3. Otherwise, go to Step 8.
Step 8. Stop. D_r is an optimal solution to (3).

To analyze the complexity of the algorithm, note that the number of the iterations is $O(\log(qm)) = O(\log(qn))$ and in each iteration an MSPP is solved. Hence if an MSPP can be solved in T time, then the complexity of the algorithm is $O(T\log(qn))$.

Theorem 4. *Algorithm 1 solves the IMSPP under the BWHD in $O(T\log(qn))$ time.*

4 Inverse multi-objective minimum spanning tree problem under the BWHD

The algorithm proposed in the previous section can be used for the inverse of the others multi-objective combinatorial optimization problems under the BWHD. For instance, consider the inverse multi-objective minimum spanning tree problem (IMMSTP). Let $G = (V, E, C)$ be a graph with $|V| = n$ nodes, $|E| = m$ edges and the cost matrix $C \in \mathbb{Z}^{m \times q}$. Assume that T^0 be a given spanning tree of G . The IMMSTP under the BWHD can be written as follows:

$$\begin{aligned} \min \quad & H_w(C, D), \\ \text{s.t.} \quad & T^0 \text{ is an efficient spanning tree of } G = (V, E, D), \\ & -L^k(a) \leq D^k(a) - C^k(a) \leq U^k(a), \quad \forall a \in E, \quad \forall k \in \{1, \dots, q\} \\ & D \in \mathbb{Z}^{m \times q}. \end{aligned} \tag{6}$$

For each $k \in \{1, \dots, q\}$ and $r \in \{1, \dots, qm\}$, the set A_r^k is exactly the same as the previous section and The matrix D_r is defined similar to (4) as follows:

$$D_r^k(a) = \begin{cases} C^k(a) & \text{if } a \in E \setminus A_r^k, \\ C^k(a) + U^k(a) & \text{if } a \in A_r^k \setminus T^0, \\ C^k(a) - L^k(a) & \text{if } a \in A_r^k \cap T^0. \end{cases} \tag{7}$$

Similarly, Theorem 2, Corollary 1 and Theorem 3 can be concluded for the problem (6). Consequently, Algorithm 1 can be applied for IMMSTP under the BWHD with this difference that in Step 4 we must investigate the efficiency of the spanning tree T^0 for $G = (V, E, D_r)$. It can be done by solving a multi-objective minimum spanning tree problem. We can use the Prim's spanning tree algorithm presented in [6].

5 Conclusion

In this article, the Inverse multi-objective shortest path problem under the bottleneck type weighted Hamming distance is considered. We proposed an algorithm based on the binary search technique to solve the inverse problem.

This work can be extended in different ways. For instance, other distances can be used. It is also possible to apply our proposed algorithm to solve the inverse of other problems.

Acknowledgments

The authors would like to thank the anonymous referees for their valuable comments to improve the paper.

References

1. R.K. Ahuja, J.B. Orlin, Inverse optimization, *Operation Research* 49, pp. 771-783, 2001.
2. D. Burton, Ph. L. Toint, On an instance of the inverse shortest paths problem, *Mathematical Programming* 53, pp. 45-61, 1992.
3. C.W. Duin, A. Volgenant, Some inverse optimization problems under the Hamming distance, *European Journal of Operational Research* 170, pp. 887-899, 2006.
4. M. Ehrgott, *Multicriteria optimization*, Springer-Verlage, Berlin, Germany, 2005.
5. A. Farago, A. Szentesi, and B. Szviatovszki, Inverse optimization in high-speed networks, *Discrete Applied Mathematics* 129, pp. 83-98, 2003.
6. J. Prim, Shortest connection networks and some generalizations, *Bell System Technicals Journal* 36, pp. 1389-1401.
7. J. Roland, Y. D. Smet, J. R. Figueira, Inverse multi-objective combinatorial optimization, *Discrete Applied Mathematics* 161, pp. 2764-2771, 2013.
8. P. Serafini, Some considerations about computational complexity for multi objective combinatorial problems, *Recent advances and historical development of vector optimization*, volume 294 of *Lecture Notes in Economics and Mathematical Systems*, pp. 222-232, 1986.
9. J. Tayyebi, M. Aman, On inverse linear programming problems under the bottleneck-type weighted Hamming distance, *Discrete Applied Mathematics*, DOI: 10.1016/j.dam.2015.12.017, 2015.
10. D.C. Wei, An optimized Floyd algorithm for the shortest path problem, *Journal of Networks* 5, pp. 1496-1504, 2010.

11. S. Xu, J. Zhang, An Inverse Problem of the Weighted Shortest Path Problem, Japan Journal of Industrial Applied Mathematics 12, pp. 47-59, 1995.
12. J. Z. Zhang, Z. Ma, C. Yang, A column generation method for inverse shortest path problems, Zeitschrift fr Operations Research 41, pp. 347-358, 1995.
13. J. Z. Zhang, Z. Ma, A network flow method for solving some inverse combinatorial optimization problems, Optimization: A Journal of Mathematical Programming and Operations Research 37, pp. 59-72, 1996.