Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 11, No. 4, pp. 593–603, October 2015

SOME NEW RESULTS ON INVERSE/REVERSE OPTIMIZATION PROBLEMS UNDER THE WEIGHTED HAMMING DISTANCE*

Longcheng Liu†, Yong Chen, Biao Wu and Enyu Yao

Abstract: The idea of the inverse/reverse optimization is to adjust the values of the parameters so that the observed solutions or objective values are indeed optimal or meet some requirements. The adjustments are measured in terms of a penalty function, and the goal is to adjust the parameters while minimizing the penalty.

In this paper, we consider the inverse minimum cut problem under the weighted sum-type Hamming distance and the reverse shortest path problems under the weighted Hamming distance. For the inverse minimum cut problem, we show the general case is NP-hard due to the Knapsack Problem. For the reverse shortest path problems, we first show that the general case under the weighted sum-type Hamming distance is strongly NP-hard due to the Three Dimensional Matching Problem, and then we present a strongly polynomial algorithm for the general problem under the weighted bottleneck-type Hamming distance.

Key words: *minimum cut, shortest path, inverse problems, reverse problems, Hamming distance*

Mathematics Subject Classification: *90B10, 68R05, 68W40, 68Q25, 03D15*

1 Introduction

In this paper, we study two inverse/reverse network optimization problems. For the first problem, a digraph and a cut is given. It is required to modify the capacity of some arcs such that the given cut is a minimal cut of the current network. The objective is to minimize the total weight of the arcs whose capacities are changed. We show that this problem is NP-hard in general case due to the Knapsack Problem. For the second problem, a undirect graph and some thresholds are given. It is required to modify the length of some edges such that the length of the shortest paths between the given pairs of vertices in the current graph are no more than the corresponding thresholds. The objective is to minimize the total weight of the edges whose lengths are changed or to minimize the maximize weight among the edges whose lengths are changed. We first show the problem with the first objective is NP-hard due to the Three Dimensional Matching Problem, and then present a strongly polynomial algorithm for the problem with the second objective.

*This research is supported by the National Natural Science Foundation of China (Grant No. 11001232), Fujian Provincial Natural Science Foundation of China (Grant No. 2012J01021) and Fundamental Research Funds for the Central Universities (Grant No. 2010121004). A preliminary version of this paper has been published in Lecture Notes in Computer Science Vol. 7285, 26-35, 2012.

†Corresponding author

In general, in an inverse optimization problem, a feasible solution is given which is not optimal under the current parameter values, and it is required to modify some parameters with minimum modification cost such that the given feasible solution becomes an optimal solution. And for the reverse optimization problem, the aim is to adjust the values of the parameters as little as possible so that the objective values meet some requirements. Burton and Toint [2] were the first who investigate the inverse version of the shortest path problem. Since then, different inverse/reverse optimization problems have been considered by various authors when the modification cost is measured by (weighted) l_1, l_2 and l_∞ norms. For detail, readers may refer to the survey paper [6] and papers cited therein. In these studies, a basic assumption is that the modifications of the weights of edges (or arcs) can be any number in the specified intervals, i.e., the weights of edges (or arcs) can be continuously modified from the current values to their maximum or minimum values. Recently, inverse/reverse problems under the weighted Hamming distance also received attention. The Hamming distance $H(c, d)$ between two given values c and d can be defined as follows:

$$H(c,d) = \begin{cases} 1, & \text{if } c \neq d, \\ 0, & \text{otherwise.} \end{cases}$$

He et al. [5] were the first who investigate the inverse version of the minimum spanning tree problem under the Hamming distance. In fact the weighted Hamming distance corresponds to the situation in which we might care about only whether the parameter of an arc is changed, but without considering the magnitude of its change as long as the adjustment is restricted to a certain interval. We may find applications of the inverse/reverse optimization problems under the weighted Hamming distance in real world. For example, in practice, we often wish to reduce the traveling time (or increase the runoff) through a road by widening the road, and in fact only some places of the road are narrow and to be widened (rather than every place of the road is narrow). In order to widen these roads, it is possible that the main cost is spent on demolishing some buildings and rebuilding them elsewhere. Such cost for road (i, j) may be a fixed amount w_{ij} instead of being $w_{ij}|c_{ij} - d_{ij}|$. On the other hand, if we want to reduce the runoff of some roads, we often construct one or some toll stations (or traffic lights system) in these roads, then the cost for road (i, j) may be fixed amount w_{ij}. Another example, with the development of the computer network, users' demand on network changes persistently. More kinds of information transmitted and higher quality of services (Qos) provided are greatly required. To meet those requirements, we need to modify the exist network, often to use the new Transport Materials or to construct one or some network servers, then the cost may be fixed amount. So, it is meaningful to consider the inverse/reverse optimization problems under the weighted Hamming distance. Noting that not like the l_1, l_2 and l_∞ norms which are all convex and continuous about the modification, the Hamming distance is discontinuous and nonconvex, which makes the known methods for l_1, l_2 and l_∞ unable to be applied directly to the problems under such distance measure.

The paper is organized as follows. Section 2 considers the inverse minimum cut problem under the sum-type Hamming distance. Sections 3 considers the reverse shortest path problems under the weighted Hamming distance. Some final remarks are made in Section 4.

2 Inverse Minimum Cut Problem Under the Weighted Sum-Type Hamming Distance

Let $N(V, A, c)$ be a connected and directed network, where $V = \{1, 2, \ldots, n\}$ is the node set, A is the arc set ($|A| = m$) and c is the capacity vector for the arcs. Each component c_{ij}

of c is called the *capacity* of arc (i, j). There are two special nodes in V: the source node s and the sink node t. Let X and $\overline{X} = V \setminus X$ be a partition of all vertices such that $s \in X$ and $t \in \overline{X}$. An $s - t$ cut, denoted by $\{X, \overline{X}\}$, is the set of arcs with one endpoint in X and another endpoint in $\overline{X}$. We further use $(X, \overline{X})$ to express the set of forward arcs from a vertex in X to a vertex in $\overline{X}$ and use $(\overline{X}, X)$ to express the set of all backward arcs in the $s - t$ cut. As we know, the capacity of the $s - t$ cut $\{X, \overline{X}\}$, denoted by $c(\{X, \overline{X}\})$, is the sum of the capacities of all forward arcs, i.e.,

$$c(\{X, \overline{X}\}) = \sum_{(i,j)\in(X,\overline{X})} c_{ij}.$$

The *minimum-cut problem* is to determine an $s - t$ cut with minimum capacity. It is a classical network optimization problem that has many applications. It is well-known that the minimum cut problem can be solved in strongly polynomial time [1].

Conversely, an inverse minimum cut problem is to modify the arc capacity vector as little as possible such that a given $s-t$ cut can form a minimum cut. Yang et al. [9] showed that the inverse minimum cut problem with one $s - t$ cut is given under the l_1 norm is strongly polynomial time solvable by transform it into a minimum cost flow problem. Zhang and Cai [11] further studied the more general inverse minimum cut problem in which multiple cuts are given. They transformed the problem into a minimum cost circulation problem and hence the problem can be solved efficiently by strongly polynomial algorithm. Liu and Yao [7] showed that the weighted inverse minimum cut problem under the bottleneck type Hamming distance is also strongly polynomial time solvable. In this paper, we consider the inverse minimum cut problem under the weighted sum-type Hamming distance.

Let each arc (i, j) have an associated capacity modification cost $w_{ij} \geq 0$, and let w denote the arc modification cost vector. Let $\{X^0, \overline{X}^0\}$ be a given $s - t$ cut in the network $N(V, A, c)$. Then for the general weighted inverse minimum cut problem under the sum-type Hamming distance, we look for an arc capacity vector d such that

(a) $\{X^0, \overline{X}^0\}$ is a minimum cut of the network $N(V, A, d)$;

(b) for each $(i, j) \in A$, $-l_{ij} \leq d_{ij} - c_{ij} \leq u_{ij}$, where $0 \leq l_{ij} \leq c_{ij}, 0 \leq u_{ij}$ are respectively given bounds for decreasing and increasing capacity;

(c) the total modification cost for changing capacities of all arcs, i.e., $\sum_{(i,j)\in A} w_{ij} H(c_{ij}, d_{ij})$, is minimized, where $H(c_{ij}, d_{ij})$ is the Hamming distance between c_{ij} and d_{ij}.

Hence, the general weighted inverse minimum cut problem under the sum-type Hamming distance can be formulated as follows:

$$\begin{aligned} \min \; & \sum_{(i,j)\in A} w_{ij} H(c_{ij}, d_{ij}) \\ \text{s.t. } & \text{Cut } \{X^0, \overline{X}^0\} \text{ is a minimum cut of } N(V, A, d); \\ & -l_{ij} \leq d_{ij} - c_{ij} \leq u_{ij}, \text{ for each } (i, j) \in A. \end{aligned} \tag{2.1}$$

The following result is well known [1].

Lemma 2.1. *An $s - t$ cut $\{X, \overline{X}\}$ of the network $N(V, A, c)$ is a minimum cut if and only if there exists a feasible flow f from node s to node t that "saturates" the cut $\{X, \overline{X}\}$, i.e., there exists a feasible flow f such that*

$$f_{ij} = c_{ij}, \quad if \quad (i, j) \in (X, \overline{X}),$$

$$f_{ji} = 0, \quad if \quad (j, i) \in (\overline{X}, X).$$

In such case the flow f must be a maximum flow of the network $N(V, A, c)$.

Based on the Lemma 2.1, the next lemma can be proved by an argument similar to the proof of the Lemma 2.2 in [7].

Lemma 2.2. *If the problem (2.1) has a feasible solution, then there exists an optimal solution* d^* *such that*

(I) $d^*_{ij} \le c_{ij}$ *for* $(i,j) \in (X^0, \overline{X}^0)$.

(II) $d^*_{ij} = c_{ij}$ *for* $(i,j) \in (\overline{X}^0, X^0)$.

(III) $d^*_{ij} \ge c_{ij}$ *for other* (i,j).

Next, we will show the problem (2.1) is NP-hard. To show this result, we transfer the decision version of the Knapsack Problem, that is a well-known NP-hard problem[4], into a special case of the decision version of the problem (2.1).

The decision version of Knapsack Problem(DVKP):

Given a knapsack of capacity $C > 0$ and n items. Each item has value $p_i > 0$ and weight $w_i > 0$. For a given threshold K, whether there is a selection of items ($\theta_i = 1$ if item i be selected and 0 otherwise) satisfying the following two conditions:

(a) $\sum\limits_{i=1}^{n} \theta_i \cdot w_i \le C$;

(b) $\sum\limits_{i=1}^{n} \theta_i \cdot p_i \ge K$.

Theorem 2.3. *Even if the modified arcs are restricted in* $(X^0, \overline{X}^0)$, *i.e.,* $l_{ij} = u_{ij} = 0$ *if* $(i,j) \notin (X^0, \overline{X}^0)$, *the problem (2.1) is NP-hard.*

Proof. The decision version of the problem (2.1) is as follows:

For a given threshold C, whether there is a solution d satisfying the following conditions:

(a) $\{X^0, \overline{X}^0\}$ is a minimum cut of the network $N(V, A, d)$;

(b) for each $(i,j) \in A$, $-l_{ij} \le d_{ij} - c_{ij} \le u_{ij}$;

(c) the total modification cost for changing capacities of all arcs is not greater than C, i.e.,

$$\sum_{(i,j)\in A} w_{ij} H(c_{ij}, d_{ij}) \le C.$$

For a given instance of DVKP $\{n, w_i, p_i, C, K\}$, we construct a network N and an instance of the decision version of the problem (2.1) as follows:

The network N has $2n+3$ nodes: $\{s, t, 1, 2, \cdots, 2n+1\}$ and $3n+1$ arcs: $a_i = (s,i), i = 1, 2, \cdots, n; b_i = (i, i+n), i = 1, 2, \cdots, n; c_i = (i+n, 2n+1), i = 1, 2, \cdots, n; d = (2n+1, t)$. An illustration of the network N is shown in the Figure 1.

Set the $\{w_{ij}, c_{ij}, l_{ij}, u_{ij}\}$ as follows:

(1) If $(i,j) \in \{a_1, a_2, \ldots, a_n, c_1, c_2, \ldots, c_n\}$, then $l_{ij} = u_{ij} = 0$, $c_{ij} = w_{ij} = +\infty$.

(2) If $(i,j) = d$, then $l_{ij} = u_{ij} = 0$, $c_{ij} = \sum\limits_{i=1}^{n} p_i - K, w_{ij} = +\infty$.

(3) For the arc b_i in $\{b_1, b_2, \ldots, b_n\}$, we set the capacity equals the item's value p_i, set the upper bound u_{ij} equals 0, set the lower bound l_{ij} equals the item's value p_i, set the modification cost equals the item's weight w_i.

It is clear that the minimum cut of the constructed network N is $\{X, \overline{X}\} = \{d\}$.

At last, we set the given cut as $\{X^0, \overline{X}^0\} = \{b_1, b_2, \ldots, b_n\}$.

It is clear that the construction of the network N can be done in polynomial time.

Due to the Lemma 2.2, the property of the Hamming distance and the definition of $\{w_{ij}, c_{ij}, l_{ij}, u_{ij}\}$, the constructed instance of the decision version of the problem (2.1) is:

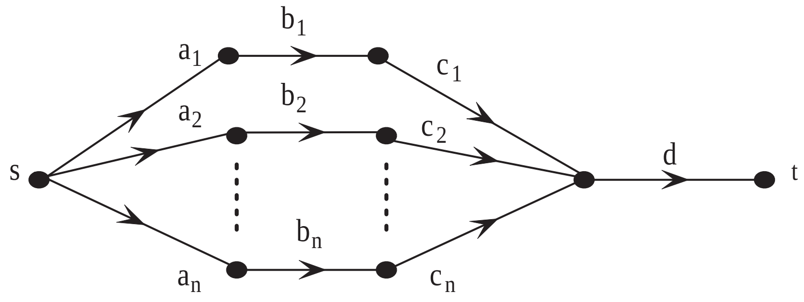

Figure 1: An illustration of the network N

For the given threshold C (the capacity of the knapsack), whether there is an arc set $\Omega \subseteq \{b_1, b_2, \ldots, b_n\}$ satisfying the following conditions:

(a) $\sum\limits_{b_i \in \Omega} p_i \geq K$;

(b) $\sum\limits_{b_i \in \Omega} w_i \leq C$ (where C is the capacity of the knapsack).

We next show that the answer of the given instance of the DVKP is *Yes* if and only if the answer of the constructed instance of the decision version of the problem (2.1) is *Yes*.

First, we assume that the answer of the given instance of the DVKP is *Yes*, i.e., there is a selection of items ($\theta_i = 1$ if item i be selected and 0 otherwise) satisfying the following two consitions:

(a) $\sum\limits_{i=1}^{n} \theta_i \cdot w_i \leq C$;

(b) $\sum\limits_{i=1}^{n} \theta_i \cdot p_i \geq K$.

Set $\Omega = \{b_i | \theta_i = 1, 1 \leq i \leq n\}$. Then we have $\Omega \subseteq \{b_1, b_2, \ldots, b_n\}$ and the following conditions are satisfied:

(a) $\sum\limits_{b_i \in \Omega} p_i \geq K$;

(b) $\sum\limits_{b_i \in \Omega} w_i \leq C$.

Which means the answer of the constructed instance of the decision version of the problem (2.1) is *Yes*.

Conversely, if the answer of the constructed instance of the decision version of the problem (2.1) is *Yes*, i.e., there is an arc set $\Omega \subseteq \{b_1, b_2, \ldots, b_n\}$ satisfying the following conditions:

(a) $\sum\limits_{b_i \in \Omega} p_i \geq K$;

(b) $\sum\limits_{b_i \in \Omega} w_i \leq C$.

Set $\theta_i = 1$ if $b_i \in \Omega$ and $\theta_i = 0$ otherwise, hence we have

(a) $\sum\limits_{i=1}^{n} \theta_i \cdot w_i \leq C$;

(b) $\sum\limits_{i=1}^{n} \theta_i \cdot p_i \geq K$.

Which means the answer of the given instance of the DVKP is *Yes*. $\square$

3 Reverse Shortest Path Problems Under the Weighted Hamming Distance

Let $G = (V, E, w)$ be a connected undirected network consisting of the vertex set $V = \{v_1, v_2, \ldots, v_n\}$ and the edge set $E = \{e_1, e_2, \ldots, e_m\}$. $w = (w_1, w_2, \ldots, w_m)$ is the length vector associated with the edges, where $w_i \geq 0$ is the length of edge e_i, $1 \leq i \leq m$. Let $(s_k, t_k)(k = 1, 2, \ldots, r)$ be r origin-destination pairs. Then, the shortest path problem is to find a path P_k from s_k to t_k such that $\sum_{p \in P_k} w(p)$ is minimum among all paths connecting s_k and t_k.

Due to its fundamental characteristic and wide range of applications, the shortest path problem plays an important role in the network optimization. In practice, many practical problems can be transformed to the shortest path problem. On the other hand, efficient algorithms for the shortest path problem are often called as a subprogram of the algorithms of some other network optimization problems. In fact, the shortest path problem is extensively applied in communication, computer systems, transportation networks and many other practical problems.

Conversely, a reverse shortest path problem is to modify the edge length vector as little as possible such that the shortest path cannot exceed a given bound. Duin and Volgenant [3] discussed the unbounded case of the inverse minimum spanning tree problem under the bottleneck-type Hamming distance. They presented an improved algorithm with a time complexity $O(n^2)$. They further extended the results to the inverse shortest path tree problem and the linear assignment problem. Xu and Zhang [8] discussed an inverse weighted shortest path problem with a single required pair and a single path. They first showed that the set of the feasible solution is a polyhedral cone and gave a sufficient and necessary condition for it. And then they found the algebraic characters of the extreme directions of the feasible set and showed a graphic character of the extreme directions. Zhang et al. [10] considered the shortest path improvement problems under the Hamming distance. They used a complicated reduction to show that the shortest path improvement problem under the sum-type Hamming distance is strongly NP-hard. Zhang and Lin [12] considered the reverse shortest path problem under l_1 norm. They first showed the general case is strongly NP-complete. And then they presented polynomial time algorithms for two special cases: the reverse shortest path problem with single source and single terminal, the reverse shortest path problem on tree network with single source. Zhang et al. [13] considered the inverse shortest path problem under the l_1 norm. They first developed a column generation algorithm which can get an optimal solution in finitely many steps. And then they used some numerical results to show the algorithm has a good performance. In this paper, we consider the weighted reverse shortest path problems under the Hamming distance, which can be described as follows: Let $l = (l_1, l_2, \ldots, l_m)(l_i \geq 0)$ be the lower bound vector of the edge length, that is, for each $i = 1, 2, \ldots, m$, the length of edge e_i cannot be less than l_i. Let $c = (c_1, c_2, \ldots, c_m)$ be the cost vector for modifying the edge lengths. Let $d_i(i = 1, 2, \ldots, r)$ be the upper bound of the length of the shortest path between s_i and t_i, that is, the length of the shortest path between vertices s_i and t_i cannot exceed d_i. We express the shortest distance between the vertices s_i and t_i under the length vector w as $d_w(s_i, t_i)$. Then for the weighted reverse shortest path problem under the sum-type Hamming distance, we look for a new edge length vector $w^* = (w_1^*, w_2^*, \ldots, w_m^*)$ such that

(a) For $i = 1, 2, \ldots, r$, $d_{w^*}(s_i, t_i) \leq d_i$;

(b) for $i = 1, 2, \ldots, m$, $l_i \leq w_i^* \leq w_i$;

(c) the total edge modification cost, i.e., $\sum_{i=1}^{m} c_i H(w_i^*, w_i)$, is minimized, where $H(w_i^*, w_i)$ is the Hamming distance between w_i^* and w_i.

For the weighted reverse shortest path problem under the bottleneck-type Hamming distance, we look for an edge length vector $w^* = (w_1^*, w_2^*, \ldots, w_m^*)$ such that the constraints (a) and (b) hold and

(c′) the maximum edge modification cost, i.e., $\max_{e_i \in E} c_i H(w_i^*, w_i)$, is minimized.

3.1 The complexity of the general problem under the sum-type Hamming distance

In this subsection, we consider the problem under the sum-type Hamming distance, which can be formulated as follows:

$$\begin{aligned} \min \ & \sum_{i=1}^{m} c_i H(w_i^*, w_i) \\ \text{s.t.} \ & d_{w^*}(s_i, t_i) \le d_i, \quad \text{for} \quad i = 1, 2, \ldots, r; \\ & l_i \le w_i^* \le w_i, \quad \text{for} \quad i = 1, 2, \ldots, m. \end{aligned} \tag{3.1}$$

We will show that the problem (3.1) is strongly NP-hard. To show this result, we introduce a strongly NP-complete problem as follows:

Three dimensional matching problem (3DM)

Given three disjoint sets U, V, W with $|U| = |V| = |W| = K$ and a subset Q of $U \times V \times W$, is there a subset M (called a perfect matching, if it exists) of Q with $|M| = K$ such that whenever (u, v, w) and (u', v', w') are distinct triples in M, $u \ne u'$, $v \ne v'$,$w \ne w'$?

It is well-known that the three dimensional matching problem is regarded as one of the six basic NP-complete problems [4].

Theorem 3.1. *Even if $s_i = s$ for each $i = 1, 2, \ldots, r$ and $c_i = 1$ for each $i = 1, 2, \ldots, m$, the problem (3.1) is strongly NP-hard.*

Proof. The decision version of the problem (3.1) is as follows: For a given threshold b, whether there is a solution $w^* = (w_1^*, w_2^*, \ldots, w_m^*)$ satisfying the following conditions:

(a) $d_{w^*}(s_i, t_i) \le d_i$, for $i = 1, 2, \ldots, r$;

(b) $l_i \le w_i^* \le w_i$, for $i = 1, 2, \ldots, m$;

(c) $\sum_{i=1}^{m} c_i H(w_i^*, w_i) \le b$.

For a given $w^* = (w_1^*, w_2^*, \ldots, w_m^*)$, checking conditions (a), (b) and (c) can be completed in polynomial time because there are polynomial time algorithms for the original shortest path problem. Therefore, the decision version of the problem (3.1) is in the class *NP*.

For a given instance of 3DM problem (U, V, W, Q), we construct a network N and an instance of the decision version of the problem (3.1) as follows:

(1) The vertex set of N is $U \cup V \cup W \cup Q \cup \{s\}$, where the new vertex s is a source and all vertices associated with the elements in $U \cup V \cup W$ are terminals;

(2) for every triple $q = (u, v, w) \in Q$, we join edges $(q, u), (q, v), (q, w)$, where $u \in U$, $v \in V$, $w \in W$, we also join (s, q) for all $q \in Q$;

(3) for edges $(q, u), (q, v)$ and (q, w), let the lengths of them are equal to 0; for edges (s, q), let the lengths of them are equal to 1; and for all edges in N, let the lower length bounds of them are equal to 0 and the modification costs of them are equal to 1. Moreover, for all pairs of sources and terminals $(s, u), (s, v)$ and (s, w), let the requested upper bound for these distance be $d_i = 0$ and let the threshold $b = K$.

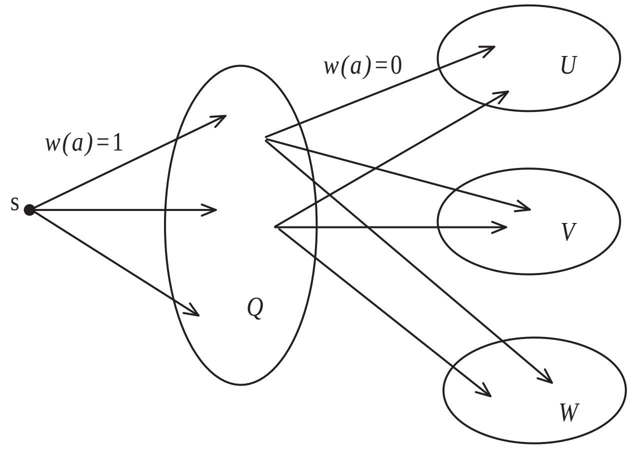

Figure 2: An illustration of the network N.

An illustration of the network N is shown in the Figure 2.

It is clear that the construction of this network N can be done in polynomial time. We next show that the instance of 3DM problem has a perfect matching M if and only if the constructed instance of the decision version of the problem (3.1) has a feasible solution $w^* = (w_1^*, w_2^*, \ldots, w_m^*)$ such that $\sum_{i=1}^{m} c_i H(w_i^*, w_i) \leq K$.

First, we assume that the 3DM problem has a perfect matching $M \subseteq Q$. We let

$$w^*(a) = \begin{cases} 1, & \text{if } a = (s, q) \text{ and } q \notin M, \\ 0, & \text{otherwise.} \end{cases}$$

Since K vertices of M are adjacent to all $3K$ vertices of $U \cup V \cup W$, it follows that the source s can reach all terminals by paths of length 0. Hence, $w^* = (w_1^*, w_2^*, \ldots, w_m^*)$ is a feasible solution with $\sum_{i=1}^{m} c_i H(w_i^*, w_i) = |M| = K$.

Conversely, let $w^* = (w_1^*, w_2^*, \ldots, w_m^*)$ be a feasible solution of the constructed instance with $\sum_{i=1}^{m} c_i H(w_i^*, w_i) \leq K$. By the condition (a) and the request that all $d_i = 0$, we can see that the source s and each terminal in $U \cup V \cup W$ have to be connected by a path of length 0. Therefore, there must be some edges $a = (s, q), q \in Q$ having $w^*(a) = 0$. Let $M = \{q \in Q \mid w^*(a) = 0 \text{ and } a = (s, q)\}$. Then the vertices of M can reach $3K$ vertices of $U \cup V \cup W$. Thus, $|M| \geq K$. Since $|M| \leq \sum_{i=1}^{m} c_i H(w_i^*, w_i) \leq K$, we have $|M| = K$. It implies that M is a perfect matching. □

3.2 The problem under the bottleneck-type Hamming distance

The problem considered in this subsection is the weighted reverse shortest path problem under the bottleneck-type Hamming distance which can be formulated as follows:

$$\begin{aligned} \min \max_{e_i \in E} \; & c_i H(w_i^*, w_i) \\ \text{s.t.} \quad & d_{w^*}(s_i, t_i) \leq d_i, \quad \text{for } i = 1, 2, \ldots, r; \\ & l_i \leq w_i^* \leq w_i, \quad \text{for } i = 1, 2, \ldots, m. \end{aligned} \tag{3.2}$$

Lemma 3.2. *Set* $w^{'}$ *as* $w_i^{'} = l_i$ *for* $i = 1, 2, \ldots, m$. *If* $w^{'}$ *is not a feasible solution of the problem (3.2), then the problem (3.2) is infeasible.*

The above result is straightforward since when we set $w_i^{'} = l_i$ for $i = 1, 2, \ldots, m$, we can not modify the edge length any more. But the $w^{'}$ is still not a feasible solution of the problem (3.2), hence the problem (3.2) is infeasible.

Now, we are going to present an algorithm to solve the problem (3.2) in strongly polynomial time due to the Lemma 3.2 and the property of the Hamming distance.

Algorithm 1.

Step 1 Set $w_i^{'} = l_i$ for $i = 1, 2, \ldots, m$ and run the shortest path algorithm to check whether the $w^{'}$ is a feasible solution of the problem (3.2). If not, output the problem (3.2) is infeasible and stop. Otherwise go to Step 2.

Step 2 Set $\Omega = \Delta = \emptyset$ and $\Gamma = E$, go to Step 3.

Step 3 Find

$$c_s = \min\{c_i \mid e_i \in \Gamma\}.$$

And let

$$\Delta = \{e_i \in \Gamma \mid c_i = c_s\},$$

$$\Omega = \Omega \cup \Delta, \Gamma = \Gamma \setminus \Delta,$$

$$w_i = \begin{cases} l_i, & \text{if } e_i \in \Delta, \\ w_i, & \text{otherwise.} \end{cases}$$

Go to Step 4.

Step 4 Run the shortest path algorithm to check whether the current length vector w is a feasible solution of the problem (3.2). If yes, then go to Step 5. Otherwise set $\Delta = \emptyset$ and go back to Step 3.

Step 5 Output the optimal solution w^* of the problem (3.2) as

$$w_i^* = \begin{cases} l_i, & \text{if } e_i \in \Omega, \\ w_i, & \text{otherwise,} \end{cases}$$

and the associate optimal objective value is $\max\{c_i \mid e_i \in \Omega\}$.

Theorem 3.3. *Algorithm 1 solves the problem (3.2) with a time complexity* $O(mn^2)$.

Proof. If the algorithm stops at Step 1, then the problem (3.2) has no feasible solution.

Otherwise the problem (3.2) has at least one optimal solution, i.e., the algorithm will stop at Step 5, which implies the output length vector w^* is a feasible solution of the problem (3.2). Furthermore, we claim that it is an optimal solution of the problem (3.2). If not, there exists another feasible solution $\overline{w}$ of the problem (3.2) satisfies

$$c_p < c_q,$$

where $c_p = \max_{e_i \in E} c_i H(w_i, \overline{w}_i)$ and $c_q = \max\{c_i \mid e_i \in \Omega\}$.

Let

$$D_p = \{e_i \in E \mid c_i \leq c_p\}$$

and

$$D_q = \{e_i \in E \mid c_i < c_q\}.$$

We refer the process starting from the Step 3 to the next Step 3 as one iteration. Then from the algorithm, there exists an iteration, denoted as I_q, in which the current length vector w is not a feasible solution of the problem (3.2) and we have $c_q = \min\{c_i \mid e_i \in \Gamma\}$, which yields that even if we changed all the costs of the edges in D_q, the current w can not meet the first constraint of the problem (3.2). Due to $D_p \subseteq D_q$ from $c_p < c_q$, we can conclude that even if we change all the costs of the edges in D_p, the new length vector can not meet the first constraint of the problem (3.2), which is contradictory to the fact that $\overline{w}$ is a feasible solution of the problem (3.2).

Finally, we consider the time complexity of the Algorithm 1. It is clear that the main computation of the Step 1 and the Step 2 is running the shortest path algorithm once, which can be finished in $O(n^2)$ time[1]. In each iteration, the main computation is running the shortest path algorithm once which can be finished in $O(n^2)$ time[1]. Since we change at least one arc length in each iteration, the algorithm will stop after at most m iterations. Hence the Algorithm 1 runs in $O(n^2) + O(mn^2) = O(mn^2)$ time in the worst case, and it is a strongly polynomial time algorithm.

□

4 Concluding Remarks

In this paper we studied the inverse minimum cut problem under the sum-type weighted Hamming distance and the reverse shortest path problems under the weighted Hamming distance. For the inverse minimum cut problem, we showed it is NP-hard due to the Knapsack Problem. For the reverse shortest path problems, we first showed the problem under the sum-type case is strongly NP-hard due to the Three Dimensional Matching problem, and then we presented an strongly polynomial time algorithm for the bottleneck-type case.

As a future research topic, it will be meaningful to consider other inverse combinational optimization problems under Hamming distance. Studying computational complexity results and proposing optimal/approximation algorithms are promising.

Acknowledgements

The authors wish to thank the Guest Editors, Lizhi Liao and Xiaoming Yuan, as well as anonymous referees whose valuable comments allowed us to improve the paper.

References

[1] R.K. Ahuja, T.L. Magnanti and J.B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice-Hall, Englewood Cliffs, 1993.

[2] D. Burton and P.L Toint, On an instance of the inverse shortest paths problem, *Mathematical Programming* 53 (1992) 45–61.

[3] C.W. Duin and A. Volgenant, Some inverse optimization problems under the Hamming distance, *European Journal of Operational Research* 170 (2006) 887–899.

[4] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the theory of NP-Completeness*, Freeman, San Francisco, CA, 1979.

[5] Y. He, B.W. Zhang and E.Y. Yao, Weighted inverse minimum spanning tree problems under Hamming distance, *Journal of Combinatorial Optimization* 9 (2005) 91–100.

[6] C. Heuberger, Inverse Optimization: A survey on problems, methods, and results, *Journal of Combinatorial Optimization* 8 (2004) 329–361.

[7] L.C. Liu and E.Y. Yao, A weighted inverse minimum cut problem under the bottleneck type Hamming distance, *Asia-Pacific Journal of Operational Research* 24 (2007) 725–736.

[8] S.J. Xu and J.Z. Zhang, An inverse problem of the weighted shortest path problem, *Japan Journal of Industrial and Applied Mathematics* 12 (1995) 47–59.

[9] C. Yang, J.Z. Zhang and Z.F. Ma, Inverse maximum flow and minimum cut problems, *Optimization* 40 (1997) 147–170.

[10] B.W. Zhang, J.Z. Zhang and L.Q. Qi, The shortest path improvement problems under Hamming distance, *Journal of Combinatorial Optimization* 12 (2006) 35-1-361.

[11] J.Z. Zhang and M.C. Cai, Inverse problem of minimum cuts, *Mathematical Methods of Operations Research* 47 (1998) 51–58.

[12] J.Z. Zhang and Y.X. Lin, Computation of the reverse shortest-path problem, *Journal of Global Optimization* 25 (2003) 243–261.

[13] J.Z. Zhang, Z.F. Ma and C. Yang, A column generation method for inverse shortest path problems, *ZOR-Mathematical Methods of Operations Research* 41 (1995) 347–358.

Manuscript received 3 June 3 2014
revised 1 December 2014
accepted for publication 1 December 2014

Longcheng Liu
School of Mathematical Sciences
Xiamen University, Xiamen 361005, P.R. China
E-mail address: longchengliu@xmu.edu.cn

Yong Chen
Institute of Operational Research & Cybernetics
Hangzhou Dianzi University, Hangzhou 310018, P.R. China
E-mail address: chenyong@hdu.edu.cn

Biao Wu
Department of Mathematics, Zhejiang University
Hangzhou 310027, P.R. China
E-mail address: wubiao@zju.edu.cn

Enyu Yao
Department of Mathematics, Zhejiang University
Hangzhou 310027, P.R. China
E-mail address: eyyao@zju.edu.cn