

INVERSE MATROID OPTIMIZATION PROBLEM UNDER THE WEIGHTED HAMMING DISTANCES

Massoud AMAN¹, Hassan HASSANPOUR² and Javad TAYYEBI^{*3}

Abstract

Given a matroid equipped with a utility vector to define a matroid optimization problem, the corresponding inverse problem is to modify the utility vector as little as possible so that a given base of the matroid becomes optimal to the matroid optimization problem. The modifications can be measured by various distances. In this article, we consider the inverse matroid problem under the bottleneck-type and the sum-type weighted Hamming distances. In the sum-type case, the problem is converted into a minimum weighted node covering problem on a bipartite network and consequently, it can be solved in strongly polynomial time. In the bottleneck case, we propose an algorithm based on the binary search technique to solve the problem in strongly polynomial time.

Mathematics Subject Classification: 90C27, 90C35.

Key words: combinatorial optimization, matroids, inverse problems, Hamming distance.

1 Introduction

A matroid is a structure that generalizes the notion of linear independence in vector spaces [27]. Several combinatorial optimization problems can be formulated in terms of matroids such as shortest path problem (in undirected graphs), minimum spanning tree problem and bipartite matching problem [22]. This implies that matroids play an important role in combinatorial optimization. A pair $M = (S, \mathcal{J})$ is called a matroid if S is a finite set, $\mathcal{J} \subseteq 2^S$ and the following axioms hold [27]:

¹Department of Mathematics, Faculty of Mathematical Sciences and Statistics, *Birjand* University, Birjand, Iran, e-mail: mamann@birjand.ac.ir

²Department of Mathematics, Faculty of Mathematical Sciences and Statistics, *Birjand* University, Birjand, Iran, e-mail: hhasanpour@birjand.ac.ir

^{3*} *Corresponding author*, Department of Industrial Engineering, *Birjand* University of Technology, Birjand, Iran, e-mail: javadtayyebi@birjandut.ac.ir

Authors in order of their contribution: J. Tayyebi, M. Aman and H. Hassanpour

- $\emptyset \in \mathcal{J}$ and if $Y \subseteq X$ and $X \in \mathcal{J}$, then $Y \in \mathcal{J}$.
- If $X, Y \in \mathcal{J}$ and $|X| < |Y|$, then $X \cup \{y\} \in \mathcal{J}$ for some $y \in Y$.

For matroid $M = (S, \mathcal{J})$, the members of \mathcal{J} are called independent sets and the others dependent sets. An independent set with the maximum cardinality is said to be a base and a dependent set with the minimum cardinality is called a circuit. It is verified that all bases of a matroid have the same cardinality [3]. Suppose $S = \{1, 2, \dots, m\}$ and $\mathbf{c} = (c_1, c_2, \dots, c_m)$ is a nonnegative utility vector on the elements of S . A matroid (S, \mathcal{J}) equipped with a utility vector \mathbf{c} is denoted by $(S, \mathcal{J}, \mathbf{c})$. For each $X \subseteq S$, we denote the value $\sum_{i \in X} c_i$ by $\mathbf{c}(X)$. A matroid optimization problem is to find an independent set X such that $\mathbf{c}(X)$ is maximized [10]. Thus, the problem can be formulated as follows:

$$\max_{X \in \mathcal{J}} \mathbf{c}(X). \quad (1)$$

The problem (1) can be solved by a greedy algorithm [22]. Since the utility vector is nonnegative, there exists at least an optimal base to the problem. Therefore, we can reduce the problem to finding a base with maximum utility. We suppose that any matroid is given by an independence testing oracle which can decide whether $X \subseteq S$ is independent in $O(1)$ time. It is notable that the oracle is polynomial for many specific problems in combinatorial optimization [22].

For a given base $B \in \mathcal{J}$, the corresponding inverse matroid (IM) problem is to modify the utility vector \mathbf{c} as little as possible so that B becomes optimal to the problem (1). The modifications can be measured by different distances such as l_1, l_2 and l_∞ norms and also, the weighted Hamming distances. In spite of developing matroid theory, inverse matroid problems are rarely considered in literature [6, 7, 8, 10]. In [6, 7], the inverse matroid intersection problem under l_1 is studied and it is shown that the problem can be transformed into a minimum cost circulation problem. Hence, it can be solved by strongly polynomial time algorithms. Dell'Amico et al. [10] introduced a new matroid, called the base matroid, and discussed an efficient implementation of the greedy algorithm to solve the base-matroid optimization problem. Then, they considered the IM problem under l_1 and showed the IM problem can be converted into a base-matroid optimization problem and consequently it can be solved efficiently.

In this article, we consider the inverse matroid problem under the weighted Hamming distances. Both the sum-type and the bottleneck-type cases are studied. In the sum-type case, we look for a vector \mathbf{d} satisfying the following conditions for a given base B :

- (I) $\mathbf{d}(B') \leq \mathbf{d}(B)$ for each $B' \in \mathcal{J}$.
- (II) $-l_i \leq d_i - c_i \leq u_i$, for each $i \in S$ where $l_i \geq 0$ and $u_i \geq 0$ are respectively given bounds for decreasing and increasing the value c_i to d_i for each $i \in S$.
- (III) The value $\sum_{i \in S} w_i H(c_i, d_i)$ is minimized where for each $i \in S$, $w_i \geq 0$ is an associated penalty for modifying c_i to d_i and $H(c_i, d_i) = 0$ if $c_i = d_i$ and otherwise, $H(c_i, d_i) = 1$;

It is notable that the nonnegativity of the vector \mathbf{d} does not guarantee by (II). But one can redefine the lower bound vector \mathbf{l} as

$$l_i = \begin{cases} c_i & c_i - l_i < 0, \\ l_i & c_i - l_i \geq 0, \end{cases} \quad \forall i \in S,$$

which guarantees the nonnegativity condition. Therefore, this is not restrictive.

In the bottleneck-type case, we look for \mathbf{d} satisfying the conditions (I) and (II) together with the following condition instead of (III):

- (IV) The value $\max_{i \in S} w_i H(c_i, d_i)$ is minimized where the notations are defined as in (III).

The rest of this article is organized as follows. In Section 2, we review some papers on the inverse combinatorial optimization problems. In Section 3, the IM problem under the weighted sum-type Hamming distance is studied. It is shown that the problem can be converted into a minimum weighted node cover problem in an auxiliary bipartite network and consequently, it can be solved in strongly polynomial time. In Section 4, the IM problem under the weighted bottleneck-type Hamming distance is considered and an efficient algorithm is proposed. The proposed algorithm is based on the binary search technique. Finally, we conclude in Section 5.

2 Literature review

Suppose that a particular optimization problem is given together with a feasible solution to it. The corresponding inverse problem is to modify some parameters of the optimization problem as little as possible such that the feasible solution becomes optimal. The concept of inverse problems was first proposed by Tarantola in geophysical sciences [23]. In the context of optimization, Burton and Toint [4, 5] considered the inverse shortest paths problem where norm 2 is used to measure the modifications. They also introduce two applications of the inverse problem in traffic modeling and seismic tomography. Afterwards, many authors considered the inverse optimization problems under the l_1 and l_∞ norms (for a survey, see [14, 11]).

Over the past decade, the Hamming distances, both the sum-type (H_1) and the bottleneck-type (H_∞) cases, have received attention in the inverse problems. Since the Hamming distance is discontinuous and nonconvex, the known methods for the l_1, l_2 and l_∞ norms cannot be applied directly to the problems under the Hamming distance [25]. Hence, the inverse problems under the Hamming distance are studied separately from the inverse problems under the norms.

He et al. [13] considered three models of the inverse spanning tree problems: unbounded case, unbounded case with forbidden edges, and general bounded case. They showed that the problems can be solved in strongly polynomial time by reducing them to a minimum weighted node cover problem in a bipartite network.

Zhang et al. [29] studied the constrained inverse spanning tree problem under the bottleneck-type Hamming distance. The constraint is that the total modification cost cannot exceed a given upper bound. It is also shown that these inverse problem can be transformed into a minimum node cover problem on a bipartite graph. The inverse min-max spanning problems, the non-constraint case and the constraint case, are studied in [16, 17] and strongly polynomial-time algorithms are proposed to solve them. Since the inverse matroid optimization problem is an extension of the inverse spanning tree problem, in this article, we generalize the known results on the inverse spanning tree problems to the inverse matroid optimization problem.

Liu and Zhang [19] considered the inverse maximum flow problem under the sum-type and bottleneck-type Hamming distance and showed that the problems can be converted to minimum cut problems on an auxiliary network and can be solved in strongly polynomial time. In [9], the authors considered the inverse feasibility problem of an inverse maximum flow problem. Their considered problem is whether an inverse maximum flow is feasible or not; and if not, how the problem can be feasible by modifying the parameters.

The inverse and reverse shortest path problem under the H_1 distance is discussed in [26, 28]. It is shown that the problem is NP-hard due to the 3-SAT problem. Duin and Volgenant [12] considered some inverse network problems under the H_∞ such as the inverse shortest path problem. They presented algorithms based on the linear and binary search technique for solving the problems.

The inverse minimum cost flow problem under the Hamming distances is studied in [1, 2, 15]. It is shown that the inverse problem under H_1 is APX-hard because a special case of the problem is equivalent to the weighted feedback arc set problem. In the bottleneck-type case, we proposed a strongly polynomial-time algorithm for solving the inverse minimum cost flow problem [24]. In [18], it is shown that the inverse perfect matching problem under H_1 is NP-hard but the inverse problem under H_∞ can be solved in strongly polynomial time.

In this article, we consider the inverse matroid problem under the weighted Hamming distances. In the H_1 case, we show that the problem is reducible to a minimum node cover problem defined on an auxiliary bipartite network by using a proposed optimality condition of the matroid optimization problem (1). This result is a generalization of the known results on the inverse minimum spanning tree problems [13, 29]. In the H_∞ case, we present an algorithm based on the binary search technique for solving the problem. This technique is a popular method for solving the inverse problems under H_∞ and l_∞ distances [12, 20, 21].

3 Inverse problem under the sum-type Hamming distance

Let $M = (S, \mathcal{J}, \mathbf{c})$ be a weighted matroid, and B a given base of M . In this section, we consider the inverse matroid optimization problem under the sum-type

weighted Hamming distance as follows:

$$\min \sum_{i \in S} w_i H(c_i, d_i), \quad (2a)$$

$$s.t. \quad \mathbf{d}(B') \leq \mathbf{d}(B) \quad \forall B' \in \mathcal{J}, \quad (2b)$$

$$-l_i \leq d_i - c_i \leq u_i \quad \forall i \in S, \quad (2c)$$

where the notations are defined as in Section 1.

For satisfying the constraints (2b) in this form, it is required to enumerate all of matroid's independent sets. Hence, for avoiding this undesired constraints, we introduce necessary and sufficient conditions for the optimality of a base in the problem (1). By using the optimality conditions, we substitute the constraints (2b) by some equivalent constraints which are more convenient for investigating.

Lemma 1. [3] *Let B_1 and B_2 be two bases of a matroid. For each $i \in B_1 \setminus B_2$, there exists $j \in B_2 \setminus B_1$ so that the sets $(B_1 \setminus \{i\}) \cup \{j\}$ and $(B_2 \setminus \{j\}) \cup \{i\}$ are also bases.*

Lemma 1 implies that one can obtain a base from the other by replacing some element of the base by some suitable element of the other. For every $i \in B$, the set

$$\{j \in S : (B \setminus \{i\}) \cup \{j\} \text{ is a base}\}$$

contains obviously i and has no $j \in B \setminus \{i\}$ because any two bases have the same cardinality. For each base B and each $i \in B$, we define

$$S_B(i) = \{j \in S \setminus \{i\} : (B \setminus \{i\}) \cup \{j\} \text{ is a base}\}. \quad (3)$$

It is evident that $S_B(i) \cap B = \emptyset$, $i \in B$. We also define the set

$$\mathcal{B} = \{i \in B : S_B(i) \neq \emptyset\} \quad (4)$$

for restricting our attention to $S_B(i)$'s to be non-empty. We now state necessary and sufficient conditions for the optimality of the given base B to the problem (1).

Theorem 1. *Let B be a base of matroid $M = (S, \mathcal{J}, \mathbf{c})$. B is an optimal base to the problem (1) if and only if*

$$c_j \leq c_i \quad \forall i \in \mathcal{B}, \forall j \in S_B(i)$$

where the sets $S_B(i)$ and \mathcal{B} are defined in (3) and (4), respectively.

Proof. We first prove the necessity. Let B be an optimal base to the problem (1). For each $i \in \mathcal{B}$ and each $j \in S_B(i)$, define $B_{ij} = (B \setminus \{i\}) \cup \{j\}$. By definition, B_{ij} is a base. Since B is optimal, $\mathbf{c}(B_{ij}) \leq \mathbf{c}(B)$ and consequently, $c_j \leq c_i$.

Now, the sufficiency is proved. Suppose that B' is an optimal base to the problem (1). It is obvious that the cardinality of sets $B' \setminus B$ and $B \setminus B'$ are equal. We

construct the base B from B' by replacing elements of $B' \setminus B$ with elements of $B \setminus B'$ one by one. The assumption guarantees the optimality of B .

If the cardinal of $B \setminus B'$ is zero, then the result is immediate. Suppose that the cardinal of $B \setminus B'$ is equal to k . Choose an arbitrary element $i_1 \in B \setminus B'$. By using Lemma 1, there exists $j_1 \in (B' \setminus B) \cap S_B(i_1)$ so that $B'_1 = (B' \setminus \{j_1\}) \cup \{i_1\}$ is a base. Based on the assumption, $c_{i_1} \geq c_{j_1}$ and hence, $\mathbf{c}(B'_1) \geq \mathbf{c}(B')$. Therefore, B'_1 is an optimal base. Choose $i_2 \in B \setminus B'_1$. By using Lemma 1, $j_2 \in (B'_1 \setminus B) \cap S_B(i_2)$ exists so that $B'_2 = (B'_1 \setminus \{j_2\}) \cup \{i_2\}$ is a base. By the assumption, $\mathbf{c}(B'_2) \geq \mathbf{c}(B'_1)$ and consequently, B'_2 is also an optimal base. By repeating this process, we can construct a sequence of the optimal bases B'_1, B'_2, \dots, B'_k where $B'_l = (B'_{l-1} \setminus \{i_l\}) \cup \{j_l\}$, $l = 1, 2, \dots, k$, for some $i_l \in B \setminus B'_{l-1}$ and some $j_l \in B'_{l-1} \setminus B$. Since the cardinality of $B \setminus B'_l$, $l = 1, 2, \dots, k$, is equal to $k - l$, it follows that $B'_k = B$. This completes the proof. \square

Based on Theorem 1, we can rewrite the problem (2) as follows:

$$\min \quad \sum_{i \in S} w_i H(c_i, d_i), \quad (5a)$$

$$s.t. \quad d_j \leq d_i \quad \forall i \in \mathcal{B}, \forall j \in S_B(i), \quad (5b)$$

$$-l_i \leq d_i - c_i \leq u_i \quad \forall i \in S. \quad (5c)$$

Lemma 2. *If the problem (5) is feasible, then there exists an optimal solution $\bar{\mathbf{d}}$ so that*

- $\bar{d}_i \geq c_i \quad \forall i \in \mathcal{B};$
- $\bar{d}_i \leq c_i \quad \forall i \in \bigcup_{k \in \mathcal{B}} S_B(k);$
- $\bar{d}_i = c_i \quad \forall i \notin \mathcal{B} \cup \bigcup_{k \in \mathcal{B}} S_B(k).$

Proof. Since for each feasible solution, the objective value of the problem (5) belongs to a finite set, the feasibility of the problem implies that the problem has at least one optimal solution. Suppose that $\hat{\mathbf{d}}$ is an optimal solution to the problem (5) and $i_0 \in \mathcal{B}$ exists so that $\hat{d}_{i_0} < c_{i_0}$. Define $\bar{\mathbf{d}}$ as follows:

$$\bar{d}_i = \begin{cases} c_{i_0} & i = i_0, \\ \hat{d}_i & i \neq i_0. \end{cases}$$

Since $\hat{\mathbf{d}}$ is feasible,

$$\hat{d}_j \leq \hat{d}_{i_0} \quad \forall j \in S_B(i_0),$$

and because of $\bar{d}_{i_0} = c_{i_0} > \hat{d}_{i_0}$,

$$\hat{d}_j \leq \bar{d}_{i_0} \quad \forall j \in S_B(i_0).$$

This result together with $\bar{d}_i = \hat{d}_i$, $i \in S \setminus \{i_0\}$, guarantees $\bar{\mathbf{d}}$ is feasible to the problem (5). On the other hand,

$$\sum_{i \in S} w_i H(c_i, \hat{d}_i) - \sum_{i \in S} w_i H(c_i, \bar{d}_i) = w_{i_0} \geq 0,$$

then $\bar{\mathbf{d}}$ is also optimal. By repeating this process, we can obtain an optimal solution $\bar{\mathbf{d}}$ so that $\bar{d}_i \geq c_i$ for all $i \in \mathcal{B}$.

Similarly, the proof of the second case can be done.

The result of the third case is immediate, because the variable d_k does not appear in the constraints (5b) for each $k \notin \mathcal{B} \cup \bigcup_{i \in \mathcal{B}} S_B(i)$. \square

For each $i \in S$, we set $\alpha_i = |d_i - c_i|$. By Lemma 2, we can limit our attention to the special form of feasible solutions by satisfying

- $\alpha_i = d_i - c_i$ for $i \in \mathcal{B}$;
- $\alpha_i = c_i - d_i$ for $i \in \bigcup_{k \in \mathcal{B}} S_B(k)$;
- $\alpha_i = 0$ for $i \notin \mathcal{B} \cup \bigcup_{k \in \mathcal{B}} S_B(k)$.

By these settings, the problem (5) reduces to

$$\begin{aligned} \min \quad & \sum_{i \in \mathcal{B} \cup \bigcup_{k \in \mathcal{B}} S_B(k)} w_i H(0, \alpha_i), \\ \text{s.t.} \quad & c_j - c_i \leq \alpha_j + \alpha_i \quad \forall i \in \mathcal{B}, \forall j \in S_B(i), \\ & 0 \leq \alpha_i \leq u_i \quad \forall i \in \mathcal{B}, \\ & 0 \leq \alpha_i \leq l_i \quad \forall i \in \bigcup_{k \in \mathcal{B}} S_B(k). \end{aligned} \quad (6)$$

Now, we discuss about some results of the problem (6).

Lemma 3. *If $\bar{\alpha}$ is a feasible solution to the problem (6), then $\hat{\alpha}$ defined by*

$$\hat{\alpha}_i = \begin{cases} 0 & \bar{\alpha}_i = 0, \\ u_i & \bar{\alpha}_i \neq 0, i \in \mathcal{B}, \\ l_i & \bar{\alpha}_i \neq 0, i \in \bigcup_{k \in \mathcal{B}} S_B(k), \end{cases}$$

is also feasible and the objective values of both the feasible solutions are equal.

Proof. Based on the definition of $\hat{\alpha}$, $\bar{\alpha}_i \neq 0$ if and only if $\hat{\alpha}_i \neq 0$. This shows that the objective values of $\hat{\alpha}$ and $\bar{\alpha}$ are equal. From the definition of $\hat{\alpha}$, it follows that $\bar{\alpha}_i \leq \hat{\alpha}_i$ for each $i \in S$ and consequently,

$$c_j - c_i \leq \bar{\alpha}_j + \bar{\alpha}_i \leq \hat{\alpha}_j + \hat{\alpha}_i \quad \forall i \in \mathcal{B}, \forall j \in S_B(i).$$

This completes the proof. \square

Set $C = \{(i, j) \in S \times S : i \in \mathcal{B} \text{ and } j \in S_B(i)\}$. By using Lemma 3, the problem (6) can be reduced to the following combinatorial optimization problem:

$$\begin{aligned} \min \quad & \sum_{i \in \mathcal{B} \cup \bigcup_{k \in \mathcal{B}} S_B(k)} w_i H(0, \alpha_i), \\ \text{s.t.} \quad & c_j - c_i \leq \alpha_j + \alpha_i \quad \forall (i, j) \in C, \\ & \alpha_i = 0, u_i \quad \forall i \in \mathcal{B}, \\ & \alpha_i = 0, l_i \quad \forall i \in \bigcup_{k \in \mathcal{B}} S_B(k). \end{aligned} \quad (7)$$

Proposition 1. *The problem (7) is feasible if and only if*

$$c_j - c_i \leq l_j + u_i \quad \forall i \in \mathcal{B}, \forall j \in S_B(i).$$

Proof. The necessity case is obvious because if α is feasible, then

$$c_j - c_i \leq \alpha_j + \alpha_i \leq l_j + u_i,$$

for each $i \in \mathcal{B}$ and each $j \in S_B(i)$. The proof of sufficiency is immediate since α^0 defined by

$$\alpha_i^0 = \begin{cases} u_i & i \in \mathcal{B}, \\ l_i & i \in \bigcup_{k \in \mathcal{B}} S_B(k), \end{cases}$$

is feasible to the problem. □

It is remarkable that for some $i \in S$, the value of α_i is fixed in all feasible solutions and consequently, we can first set such α_i 's. During these preprocessing operations, some constraints are satisfied and can be deleted from the problem. Let us explain how to perform the preprocessing operations in more details. For each $(i, j) \in C$ corresponding to the constraint $c_j - c_i \leq \alpha_j + \alpha_i$, one of the following five cases holds:

- If $c_j - c_i \leq 0$, then the constraint $c_j - c_i \leq \alpha_j + \alpha_i$ is satisfied for any value of α_i and α_j .
- If $0 < c_j - c_i \leq \min\{l_j, u_i\}$, then it is possible to satisfy the corresponding constraint by setting $\alpha_i = u_i$ or $\alpha_j = l_j$. We denote the set $\{(i, j) \in C : 0 < c_j - c_i \leq \min\{l_j, u_i\}\}$ by C_1 .
- If $\min\{l_j, u_i\} < c_j - c_i \leq \max\{l_j, u_i\}$, then for satisfying the constraint $c_j - c_i \leq \alpha_j + \alpha_i$, it is necessary to set $\alpha_i = u_i$ if $u_i = \max\{u_i, l_j\}$ and $\alpha_j = l_j$ otherwise.
- If $\max\{l_j, u_i\} < c_j - c_i \leq l_j + u_i$, then it is necessary to set $\alpha_i = u_i$ and $\alpha_j = l_j$ for satisfying the associated constraint.
- If $c_j - c_i > l_j + u_i$, then the inverse problem is infeasible based on Proposition 1.

Suppose that the problem (7) is feasible. Based on the above argument, if α_i 's are set in cases 3 or 4, then any constraint corresponding to $(i, j) \in C \setminus C_1$ is satisfied and can be eliminated. We denote the set of indices of the variables which do not set in cases 3 or 4 by \bar{S} . It is notable that we also eliminate the constraints corresponding to $(i, j) \in C_1$ if $i \notin \bar{S}$ or $j \notin \bar{S}$. Because if $i \notin \bar{S}$, then $\alpha_i = u_i \geq \min\{l_j, u_i\} \geq c_j - c_i$ and similarly, if $j \notin \bar{S}$, then $\alpha_j = l_j \geq \min\{l_j, u_i\} \geq c_j - c_i$. We denote the set $\{(i, j) \in C_1 : i, j \in \bar{S}\}$ by \bar{C} . This argument shows that the

problem (7) can be reduced as follows:

$$\begin{aligned}
\min \quad & \sum_{i \in \bar{S}} w_i H(0, \alpha_i) + \sum_{i \in (\mathcal{B} \cup \bigcup_{k \in \mathcal{B}} S_B(k)) \setminus \bar{S}} w_i, \\
\text{s.t.} \quad & c_j - c_i \leq \alpha_j + \alpha_i \quad \forall (i, j) \in \bar{C}, \\
& \alpha_i = 0, u_i \quad \forall i \in \bar{S} \cap \mathcal{B}, \\
& \alpha_i = 0, l_i \quad \forall i \in \bar{S} \cap \bigcup_{k \in \mathcal{B}} S_B(k),
\end{aligned} \tag{8}$$

For each $i \in \bar{S}$, we define $y_i \in \{0, 1\}$ with the following properties:

$$\begin{aligned}
y_i = 1 & \Leftrightarrow \alpha_i = u_i \quad \forall i \in \bar{S} \cap \mathcal{B}, \\
y_j = 1 & \Leftrightarrow \alpha_j = l_j \quad \forall j \in \bar{S} \cap \bigcup_{k \in \mathcal{B}} S_B(k).
\end{aligned}$$

Since $0 < c_j - c_i \leq \min\{l_j, u_i\}$ for each $(i, j) \in \bar{C}$, it follows that the problem (8) is equivalent to the following problem:

$$\begin{aligned}
\min \quad & \sum_{i \in \bar{S}} w_i y_i + \sum_{i \in (\mathcal{B} \cup \bigcup_{k \in \mathcal{B}} S_B(k)) \setminus \bar{S}} w_i, \\
\text{s.t.} \quad & 1 \leq y_i + y_j \quad \forall (i, j) \in \bar{C}, \\
& y_i = 0, 1 \quad \forall i \in \bar{S}.
\end{aligned} \tag{9}$$

This problem is an instance of the well-known minimum weighted node cover problems defined on the bipartite network $G(\bar{S}, \bar{C}, \mathbf{w})$ where \bar{S} is the node set, \bar{C} is the arc set and finally, \mathbf{w} is the arc weight vector [22]. Now, we are ready to state our proposed algorithm for solving the inverse problem (4).

Algorithm 1.

Input: Matroid (S, I, \mathbf{c}) , a penalty vector \mathbf{w} , bound vectors \mathbf{l} and \mathbf{u} and a base $B \in I$.

Initialization: Set $\alpha = \mathbf{0}$, $z = 0$ and $M = \emptyset$.

Step 1: If $c_j - c_i > l_j + u_i$ for some $(i, j) \in C$, then the problem (5) is infeasible and stop; otherwise, go to Step 2.

Step 2: For each $(i, j) \in C$ with $\max\{l_j, u_i\} < c_j - c_i \leq l_j + u_i$, perform the following operations:

- If $i \notin M$, then set $\alpha_i = u_i$ and update $z = z + w_i$ and $M = M \cup \{i\}$.
- If $j \notin M$, then set $\alpha_j = l_j$ and update $z = z + w_j$ and $M = M \cup \{j\}$.

Step 3: For each $(i, j) \in C$ with $\min\{l_j, u_i\} < c_j - c_i \leq \max\{l_j, u_i\}$, perform the following operations:

- If $i \notin M$ and $u_i = \max\{l_j, u_i\}$, then set $\alpha_i = u_i$ and update $z = z + w_i$ and $M = M \cup \{i\}$.

- Else, if $j \notin M$ and $l_j = \max\{l_j, u_i\}$ then set $\alpha_j = l_j$ and update $z = z + w_j$ and $M = M \cup \{j\}$.

Step 4: Solve the minimum weighted node covering problem (9). Follow the following procedure:

- For each $i \in \bar{S} \cap \mathcal{B}$, if $y_i = 1$, then set $\alpha_i = u_i$ and $z = z + w_i$.
- For each $j \in \bar{S} \cap \bigcup_{k \in \mathcal{B}} S_B(k)$, if $y_j = 1$, then set $\alpha_j = l_j$ and $z = z + w_j$.
- Construct the vector \mathbf{d} as follows:

$$d_i = \begin{cases} c_i + \alpha_i & i \in \mathcal{B}, \\ c_i - \alpha_i & i \in \bigcup_{k \in \mathcal{B}} S_B(k), \\ c_i & \text{otherwise.} \end{cases}$$

Output: \mathbf{d} is optimal solution to the problem (5) with objective value z .

Now, we analyze the complexity of Algorithm 1. We first discuss about the complexity of solving the problem (9). Since $\mathcal{B} \cap \bigcup_{k \in \mathcal{B}} S_B(k) = \emptyset$ and $\bar{C} = \{(i, j) : i \in \bar{S} \cap \mathcal{B} \text{ and } j \in \bar{S} \cap \bigcup_{k \in \mathcal{B}} S_B(k)\}$, the network $G(\bar{S}, \bar{C}, \mathbf{w})$ is bipartite. Consequently, the problem (9) can be solved by solving a maximum flow problem in a transformed bipartite network in $O(n^3m)$ time where $n = |B| \geq |\bar{S} \cap \mathcal{B}|$ and $m = |S| \geq |\bar{S} \cap \bigcup_{k \in \mathcal{B}} S_B(k)|$ [13]. On the other hand, Steps 1, 2 and 3 are done in $O(mn)$ time. Hence, we establish the following result.

Theorem 2. *The inverse matroid problem (5) can be solved in $O(n^3m)$ time where $m = |S|$ and $n = |B|$.*

4 Inverse problem under the bottleneck-type Hamming distance

In this section, for a given base B of matroid $M = (S, \mathcal{J}, \mathbf{c})$, we consider the following problem:

$$\begin{aligned} \min \quad & \max_{i \in S} w_i H(c_i, d_i), \\ \text{s.t.} \quad & \mathbf{d}(B') \leq \mathbf{d}(B) \quad \forall B' \in \mathcal{J}, \\ & -l_i \leq d_i - c_i \leq u_i \quad \forall i \in S, \end{aligned} \tag{10}$$

where the notations are defined as in Section 1. Based on Theorem 1, we can rewrite the problem as follows:

$$\begin{aligned} \min \quad & \max_{i \in S} w_i H(c_i, d_i), \\ \text{s.t.} \quad & d_j \leq d_i \quad \forall i \in \mathcal{B}, \forall j \in S_B(i), \\ & -l_i \leq d_i - c_i \leq u_i \quad \forall i \in S. \end{aligned} \tag{11}$$

Since the feasible sets of both the inverse problems (5) and (11) are the same, the obtained feasibility results for the problem (5) are also valid for the problem (11). Each objective value of the problem (11) is a member of the finite set $\{w_1, w_2, \dots, w_m\}$. Hence, the feasibility of the problem implies at least the existence of an optimal solution. Now, we restrict our attention to a special form of feasible solutions and look for such an optimal solution.

Lemma 4. *If the problem (11) has a feasible solution with objective value less than or equal to w_k , then the solution $\mathbf{d}^{(k)}$ defined as*

$$d_i^{(k)} = \begin{cases} c_i + u_i & w_i \leq w_k, i \in \mathcal{B}, \\ c_i - l_i & w_i \leq w_k, i \in \bigcup_{j \in \mathcal{B}} S_B(j), \\ c_i & w_i > w_k \text{ or } i \notin \mathcal{B} \cup \bigcup_{j \in \mathcal{B}} S_B(j), \end{cases} \quad (12)$$

is also feasible with objective value less than or equal to w_k .

Proof. From the definition of $\mathbf{d}^{(k)}$, $H(d_i^{(k)}, c_i) = 0$ for each $i \in S$ with $w_i > w_k$. Consequently, the objective value of $\mathbf{d}^{(k)}$ is at most w_k . Let \mathbf{d} be a feasible solution with objective value less than or equal to w_k . Based on the definition of $\mathbf{d}^{(k)}$ and the bound constraints of the problem (11), the relations

$$\begin{aligned} d_i &= d_i^{(k)} && \text{for } i \in S \text{ with } w_i > w_k, \\ d_i &\leq d_i^{(k)} && \text{for } i \in \mathcal{B} \text{ with } w_i \leq w_k, \\ d_i &\geq d_i^{(k)} && \text{for } i \in \bigcup_{j \in \mathcal{B}} S_B(j) \text{ with } w_i \leq w_k, \end{aligned}$$

hold. These relations together with the feasibility of \mathbf{d} yield

$$d_j^{(k)} \leq d_j \leq d_i \leq d_i^{(k)} \quad \forall i \in \mathcal{B}, \forall j \in S_B(i).$$

Thus, the solution $\mathbf{d}^{(k)}$ is feasible. \square

Suppose we sort the arc penalties w_{ij} 's: let $w_1 \leq w_2 \leq \dots \leq w_m$ denote the sorted list of these penalties. In the following, two immediate consequences of Lemma 4 are given.

Corollary 1. *The solution $\mathbf{d}^{(m)}$, defined by (12) for $k = m$, is feasible to the problem (11) if the problem is feasible.*

Corollary 2. *Suppose that k is the least index so that $\mathbf{d}^{(k)}$ is feasible to the problem (11), then the solution $\mathbf{d}^{(k)}$ is optimal.*

By Corollary 2, the problem (11) can be reduced to finding the least index k so that $\mathbf{d}^{(k)}$ is feasible. We use the binary search technique to look for the index k .

Let us state our proposed algorithm formally.

Algorithm 2.

Input: Matroid $M = (S, I, \mathbf{c})$, a penalty vector \mathbf{w} , bound vectors \mathbf{l} and \mathbf{u} , and a base B .

Step 1: Sort all the arcs in nondecreasing order of their penalties.

Step 2: If $\mathbf{d}^{(m)}$ is feasible to the problem (11), then set $s = \lfloor \frac{m}{2} \rfloor$, $k = \lfloor \frac{m}{2} \rfloor$, $\mathbf{d}^* = \mathbf{d}^{(m)}$, $w^* = w_m$ and go to Step 3. Otherwise, the problem (11) is infeasible and stop (see Corollary 1).

Step 3: If $s = 0$, then go to Step 4 and otherwise, go to Step 5.

Step 4: If $\mathbf{d}^{(k)}$ is feasible to the problem (11), then set $s = \lfloor \frac{s}{2} \rfloor$, $k = k - s$, $\mathbf{d}^* = \mathbf{d}^{(k)}$, $w^* = w_k$ and otherwise, set $s = \lfloor \frac{s}{2} \rfloor$, $k = k + s$. Go to Step 3.

Step 5: Stop. \mathbf{d}^* is an optimal solution to the problem (11) with the objective value w^* .

In Algorithm 2, \mathbf{d}^* stores the latest feasible solution to be found by Algorithm and w^* is its objective value. Suppose that $m = |S|$ and $n = |B|$. Since the problem consists of at most $O(mn)$ inequalities as $d_j \leq d_i$, the feasibility of a given solution can be identified in $O(mn)$ time. On the other hand, the number of iterations is $O(\log m)$. We have thus established the following result.

Theorem 3. *The inverse matroid problem (11) can be solved in $O(mn \log m)$ time.*

5 Conclusion

In this article, we studied the inverse matroid optimization problem under the weighted Hamming distances, the sum-type case and the bottleneck-type case. For the former, we reduce the problem to a minimum weighted node cover problem on an auxiliary bipartite network. Therefore, the problem can be solved in $O(mn^3)$ time. For the latter, we applied the binary search technique for solving the problem in $O(mn \log m)$ time.

References

- [1] Aman, M, Hassanpour, H. and Tayyebi, J. *A modified algorithm for solving the inverse minimum cost flow problem under the bottleneck-type Hamming distance*, Bulletin of the Transilvania University of Braşov Series III **9 (58)** (2016), 97–110
- [2] Aman, M. and Tayyebi, J., *Capacity inverse minimum cost flow problem under the weighted Hamming distances*, Iranian Journal of Operations Research **5**, no. 2, (2014), 12–25.
- [3] Brylawski, T.L., *Some properties of basic families of subsets*, Discrete Mathematics **6** (1973), 333–341.

- [4] Burton, D. and Toint, P.L., *On an instance of the inverse shortest paths problem*, Mathematical Programming **53** (1992), 45–61.
- [5] Burton, D. and Toint, P.L., *On the use of an inverse shortest paths algorithm for recovering linearly correlated costs*, Mathematical Programming **63** (1994), 1–22.
- [6] Cai, M., *Inverse problems of matroid intersection*, Journal of Combinatorial Optimization **3** (1999), 465–474.
- [7] Cai, M. and Li, Y., *Inverse matroid intersection problem*, ZOR-Math. Methods Oper. Res. **45** (1997), 235–243.
- [8] Cai, M., Yang, X. and Li, Y., *Inverse polymatroidal flow*, J. Combin. Theory **3** (1999), 115–126.
- [9] Deaconu, A. and Ciurea, E., *Inverse feasibility problems of the inverse maximum flow problems*, Sādhanā, **38** (2013), 199–209.
- [10] Dell’Amico, M., Maffioli, F. and Malucelli, F., *The base-matroid and inverse combinatorial optimization problems*, Discrete Applied Mathematics **128** (2003), 337–353.
- [11] Demange, M. and Monnot, J., *An introduction to inverse combinatorial problems*. In: Vangelis Th. Paschos, Paradigms of Combinatorial Optimization (Problems and New approaches), Wiley, London-Hoboken 2010.
- [12] Duin, C. and Volgenant, A., *Some inverse optimization problems under the hamming distance*, European Journal of Operational Research **170** (2006), 887–899.
- [13] He, Y., Zhang, B. and Yao, E., *Weighted inverse minimum spanning tree problems under hamming distance*, Journal of Combinatorial Optimization **9** (2005), 91–100.
- [14] Heuberger, C., *Inverse optimization: A survey on problems, methods, and results*, Journal of Combinatorial Optimization **8** (2004), 329–361.
- [15] Jiang, Y. Liu, L., Wuc, B. and Yao, E., *Inverse minimum cost flow problems under the weighted Hamming distance*, European Journal of Operational Research **207** (2010), 50–54.
- [16] Liu, L. and Wang, Q., *Constrained inverse minmax spanning tree problems under the weighted hamming distance*, Journal of Global Optimization **43** (2009), 83–95.
- [17] Liu, L. and Yao, E., *Inverse minmax spanning tree problem under the weighted sum-type hamming distance*, Theoretical Computer Science **396** (2008), 28–34.

- [18] Liu, L. and Yao, E. *Weighted inverse maximum perfect matching problems under the hamming distance*, Journal of Global Optimization **55** (2013), 549–557.
- [19] Liu, L. and Zhang, J., *Inverse maximum flow problems under the weighted hamming distance*, Journal of Combinatorial Optimization **12** (2006), 395–408.
- [20] Liu, Z.H. and Zhang, J.Z., *On inverse problems of optimum perfect matching*, Journal of Combinatorial Optimization **7** (2003), 215–228.
- [21] Roland, J., Smeta, Y.D. and Figueira, J.R. *Inverse multi-objective combinatorial optimization*, Discrete Applied Mathematics **161** (2013), 2764–2771.
- [22] Schrijver, A., *Combinatorial optimization: polyhedra and efficiency*, Springer-Verlag Berlin Heidelberg, New York, 2003.
- [23] Tarantola, A., *Inverse problem theory: methods for data fitting and model parameter estimation*, Elsevier, Amsterdam 1987.
- [24] Tayyebi, J. and Aman, M., *Note on "inverse minimum cost flow problems under the weighted hamming distance"*, European Journal of Operational Research **234** (2014), 916–920.
- [25] Tayyebi, J. and Aman, M., *On inverse linear programming problems under the bottleneck-type weighted Hamming distance*, Discrete Applied Mathematics, DOI: 10.1016/j.dam.2015.12.017 (2015).
- [26] Tayyebi, J. and Aman, M., *Efficient algorithms for the reverse shortest path problem on trees under the hamming distance*, Yugoslav Journal of Operations Research, DOI: 10.2298/YJOR150624009T (2016).
- [27] Whitney, H., *On the abstract properties of linear dependence*, American Journal of Mathematics **57** (1935), 509–533.
- [28] Zhang, B., Zhang, J. and Qi, L., *The shortest path improvement problems under Hamming distance*, Journal of Combinatorial Optimization **12** (2006), 351–361.
- [29] Zhang, B., Zhang, J. and He, Y., *Constrained inverse minimum spanning tree problems under the bottleneck-type Hamming distance*, Journal of Global Optimization **34** (2006), 467–474.