



Partial inverse min–max spanning tree problem under the weighted bottleneck hamming distance

Qingzhen Dong¹ · Xianyue Li¹ · Yu Yang¹

Accepted: 31 October 2023 / Published online: 16 November 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

Min–max spanning tree problem is a classical problem in combinatorial optimization. Its purpose is to find a spanning tree to minimize its maximum edge in a given edge weighted graph. Given a connected graph G , an edge weight vector w and a forest F , the partial inverse min–max spanning tree problem (PIMMST) is to find a new weighted vector w^* , so that F can be extended into a min–max spanning tree with respect to w^* and the gap between w and w^* is minimized. In this paper, we research PIMMST under the weighted bottleneck Hamming distance. Firstly, we study PIMMST with value of optimal tree restriction, a variant of PIMMST, and show that this problem can be solved in strongly polynomial time. Then, by characterizing the properties of the value of optimal tree, we present first algorithm for PIMMST under the weighted bottleneck Hamming distance with running time $O(|E|^2 \log |E|)$, where E is the edge set of G . Finally, by giving a necessary and sufficient condition to determine the feasible solution of this problem, we present a better algorithm for this problem with running time $O(|E| \log |E|)$. Moreover, we show that these algorithms can be generalized to solve these problems with capacitated constraint.

Keywords Partial inverse problem · Min–max spanning tree · Weighted bottleneck hamming distance · Strongly polynomial-time algorithm

1 Introduction

Many classical 0-1 combinatorial optimization problems can be written as $P = (E, \mathcal{T}, w)$, where E is a ground set, $\mathcal{T} \subseteq 2^E$ is the set of feasible solutions, and

A preliminary version of the paper appeared in the 16th International Conference on Algorithmic Aspects in Information and Management (AAIM 2022).

✉ Xianyue Li
lixianyue@lzu.edu.cn

¹ School of Mathematics and Statistics, Lanzhou University, Lanzhou 730000, Gansu, People's Republic of China

w is a weight vector on E . The objective of (E, \mathcal{T}, w) is to find an optimal solution $T^* \in \mathcal{T}$ such that

$$w(T^*) = \min(\max)_{T \in \mathcal{T}} w(T).$$

Given a combinatorial optimization problem P and a partial solution T' which is included in some feasible solutions, the *partial inverse problem* on P is to find a new weight vector w^* , so that T' can be extended into an optimal solution with respect to w^* and the gap between w and w^* is minimized. Researchers usually use the Manhattan distance (l_1 -norm), Euclid distance (l_2 -norm), Chebyshev distance (l_∞ -norm) and Hamming distance to measure the gap. A partial inverse problem is called as *capacitated*, if for any element $e \in E$, $-l(e) \leq w^*(e) - w(e) \leq u(e)$, where $l, u : E \rightarrow \mathbb{R}^+ \cup \{0\}$ are decreasing and increasing bound vectors, respectively. Note that when partial solution is a feasible solution, a partial inverse problem degenerates into an inverse problem.

For the partial inverse combinatorial optimization problems, researchers (Ben-Ayed and Blair, 1990; Hansen et al., 1992) firstly studied the partial inverse linear programming problem, and showed that this problem is strongly NP-Hard by studying the relationship between it and bi-level linear programming. Then, a lot of specific partial inverse problems have been studied. For the partial inverse assignment problem, Yang (2001) firstly proved that this problem with capacitated constraint under the Manhattan distance is APX-hard, and then Yang and Zhang (2007a) presented a strongly polynomial time algorithm to solve this problem under the Manhattan distance. For the partial inverse minimum cut problem, Yang (2001) firstly proved that this problem with capacitated constraint under the Manhattan distance is APX-hard, and then Gassner (2010) showed that this problem under the Manhattan distance is also NP-Hard. For the partial inverse sorting problem, Yang and Zhang (2007b) showed that this problem under the Manhattan distance, Euclid distance and Chebyshev distance can be solved in polynomial time. For the special case when the partial solution has only one element, Lai and Orlin (2003) proved that under the Chebyshev norm, decision versions of the partial inverse shortest path problem on acyclic graphs, the partial inverse assignment problem, the partial inverse minimum cost arc (or vertex) disjoint cycle problem, and the partial inverse minimum cut problem are all NP-Complete.

As a classical problem in combinatorial optimization, the partial inverse problems of spanning tree have attracted widely attention from researchers. Researchers firstly studied the *inverse maximum spanning tree problem* (IMST), a special case of *partial inverse maximum spanning tree problem* (PIMST). For IMST under the Manhattan distance, Zhang et al. (1997) obtained the first strongly polynomial-time algorithm to solve it. Then, Sokkalingam et al. (1999) introduced a $O(n^3)$ -algorithm to solve it, where n is the number of vertices in the graph. In 2000, Ahuja and Orlin (2000) improved the complexity of this problem to $O(n^2 \log n)$. For IMST under the Chebyshev distance, Sokkalingam et al. (1999) showed that it can be solved in $O(n^2)$ time. In 2003, Hochbaum (2003) summarized the above two problems, and proposed new algorithms to solve them, respectively. For IMST under the Hamming distance, He et al. (2005) firstly showed that IMST with capacitated constraint (CIMST) can be solved in $O(n^3 m)$ time for the weighted sum type, where m is the number of edges

in the graph. Then, Zhang et al. (2006) proposed an algorithm with running time $O(n^3 m \log m)$ to solve CIMST for the weighted bottleneck type.

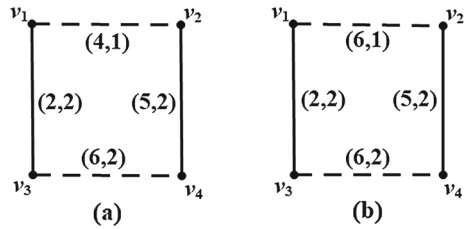
For the CPIMST, Cai et al. (2008) considered this problem with $l \equiv 0$, and showed that it can be solved in strongly polynomial-time under any norm. In 2016, Li et al. (2016) generalized the above method to CPIMST with $u \equiv 0$ under the Chebyshev norm. Then, Li et al. (2018) considered CPIMST with $u \equiv 0$ under the l_p -norm, they proved that it is APX-Hard when the partial solution has at least two edges, and it can be solved in strongly polynomial-time when the partial solution has only one edge. For CPIMST under the Hamming distance, Li et al. (2019) showed that it is APX-Hard for the weighted sum type and it can be solved in $O(m \log^2 m)$ time for the weighted bottleneck type. Recently, Li et al. (2020) proposed approximation algorithms for CPIMST under the weight l_p -norm and the weighted sum Hamming distance, and showed the approximation ratios are $\sqrt[k]{k}$ and k , respectively, where k is the number of edges in the partial solution.

Besides the maximum spanning tree problem, some partial inverse problems on other spanning tree problems have also been studied by researchers, such as min–max spanning tree problem and max+sum spanning tree problem, etc. For these problems, researchers mainly considered their inverse problems. Yang and Zhang (2007c) firstly studied inverse min–max spanning tree problem (IMMST) under the Manhattan distance and the Chebyshev distance, and showed that IMMST can be solved in strongly polynomial time. Liu and Wang (2009); Liu and Yao (2008) proposed strongly polynomial-time algorithms to solve IMMST under the weighted bottleneck Hamming distance, sum Hamming distance and their mixed distance. Guan et al. (2015) firstly studied inverse max+sum spanning tree problem (IMSST) under the Chebyshev norm, and showed that it can be solved in strongly polynomial-time. Then, Guan et al. showed that IMSST can be solved in strongly polynomial time under the weighted bottleneck Hamming distance, and IMSST under the weighted sum Hamming distance is NP-Hard (Guan et al., 2017). Finally, Guan et al. considered IMSST under the Manhattan distance and presented a column generation algorithm to solve its dual problem (Guan et al., 2018). For partial inverse problems, there are not many results on these problems at present. In 2020, Tayyebi and Sepasian (2020) proved that partial min–max spanning tree problem (PIMMST) under the Manhattan distance can be solved in $O(m^2 n^2 \log n)$ time.

In this paper, we study PIMMST and a its variant under the weighted bottleneck Hamming distance and show these two problems can be solved in strongly polynomial-time.

The paper is organized as follows. The definition of PIMMST under the weighted bottleneck Hamming distance and some useful basic results are introduced in Sect. 2. In Sect. 3, we consider PIMMST with the value of optimal tree restriction and propose a strongly polynomial-time to solve it. Then, we present two algorithms for PIMMST in Sect. 4. In detail, by obtaining the candidate set of the values of optimal tree, we firstly propose an algorithm with running time $O(m^2 \log m)$. Then, by characterizing a necessary and sufficient condition for feasible solutions of this problem, we propose a simpler algorithm with running time $O(m \log m)$. In Sect. 5, we make a conclusion.

Fig. 1 **a** An instance of PIMMST; **b** the unique optimal solution of this instance



2 Preliminary

In this section, the definitions of min–max spanning tree and PIMMST under the weighted bottleneck Hamming distance are presented, and some useful notations and basic results are introduced.

Given a connected graph $G = (V, E)$ and an edge weight vector w , a spanning tree is a *min–max spanning tree* (MMST) if its weight is the smallest among all spanning trees, where the weight of a spanning tree T is the maximum weight of all edges in T , i.e., $w(T) = \max_{e \in T} w(e)$. Clearly, every minimum spanning tree is a min–max spanning tree, but the opposite is not true. In 1978, Camerini (1978) presented a linear-time algorithm to solve MMST.

Definition 1 Given a connected graph $G = (V, E)$, a forest F , an edge weight vector w , and a norm function $c : E \rightarrow R^+$, the goal of Partial inverse min–max spanning tree problem (PIMMST) under the weighted bottleneck Hamming distance is to find a new weight vector w^* satisfying:

- (1) F is included in a min–max spanning tree of G with respect to w^* ;
- (2) $\|w^* - w\|_{WBH} = \max_{e \in E} \{c(e) \cdot H(w(e), w^*(e))\}$ is minimum,

where $H(\cdot)$ is the Hamming distance, that is, if $w(e) = w^*(e)$, $H(w(e), w^*(e)) = 0$; otherwise, $H(w(e), w^*(e)) = 1$.

For an instance $I = (G, F, w, c)$ of PIMMST, we call a new weight vector w' a *feasible solution* if it meets the first constraint, and a spanning tree T an *optimal tree* if T is min–max with respect to an optimal solution and $F \subseteq E(T)$.

In the study of PIMMST, the separation property of weight vector plays a very important role. Specifically, for any instance of this problem, there is always an optimal solution w^* such that $w^*(e) \leq w(e)$ for any edge $e \in F$ and $w^*(e) \geq w(e)$ for any edge $e \notin F$. Unfortunately, instances of PIMMST often do not have such properties. Figure 1 illustrates an example of above statement. The left figure is an instance of PIMMST under the weighted bottleneck Hamming distance, in which the two dash edges form the partial solution F and the two parameters on each edge e represent its weight $w(e)$ (the first value in parentheses) and norm $c(e)$ (the second value in parentheses), respectively. The right figure is the unique optimal solution of this instance. It can be seen that the weight of edge v_1v_2 increases from 4 to 6, which does not satisfy the separation property. The main reason is that, for a min–max spanning tree, if the weights of some non-maximum edges on the tree are increased to the weight of the tree, it is still a min–max spanning tree.

Fortunately, although PIMMST cannot satisfy the first half ($e \in F$) of the separation property, it still satisfies the second half ($e \notin F$) of the property. To show this, we firstly introduce some useful notations. Let $G = (V, E)$ be a connected graph, and w be any edge weight vector of G . Define $E_{w,r}^{\geq}$ as the subset of E with weight of edge at least r , that is, $E_{w,r}^{\geq} = \{e \in E | w(e) \geq r\}$. Let T be a spanning tree of G . For any edge $e \in E(T)$, define $C(T, e)$ as the set of edges connecting the two components of $T - e$, called the *fundamental cut* with respect to T and e . Clearly, $T - e + e'$ is also a spanning tree of G for any edge $e' \in C(T, e)$.

The following lemma gives a necessary and sufficient condition for a spanning tree T of G to be min–max.

Lemma 1 *Let $G = (V, E)$ be a connected graph, and w be an edge weight vector of G . A spanning tree T of G is a min–max with respect to w if and only if $G - E_{w,r}^{\geq}$ is disconnected, where $r = w(T) = \max_{e \in E(T)} w(e)$.*

Proof Let $G' = G - E_{w,r}^{\geq}$. For the necessity, suppose to the contrary that G' is connected. This implies that G' has a spanning tree T' with $w(T') < r$. Since T' is also a spanning tree of G , this contradicts that T is a min–max spanning tree.

For the sufficiency, since G' is disconnected, it implies that every spanning tree of G has at least on edge in $E_{w,r}^{\geq}$. Hence, $w(T') \geq r = w(T)$ for any spanning tree T' , that is, T is min–max. □

It should be mentioned that Guan and Zhang have provided similar conclusions for general bottleneck optimization problems, which can be found in Lemma 1 of Guan and Zhang (2007).

Theorem 1 not only proves that an optimal solution satisfies the second half of the separation property, but also gives its properties on the partial solution F .

Theorem 1 *Let $I = (G, F, w, c)$ be an instance of PIMMST under the weighted bottleneck Hamming distance. Then, there exists an optimal solution w^* and an optimal tree T^* , such that*

- (1) for any $e \in F$, if $w^*(e) \neq w(e)$, then $w^*(e) = w^*(T^*)$;
- (2) for any $e \notin F$, $w^*(e) \geq w(e)$.

Proof For the first part, let’s consider a new weight vector w_o as follows,

$$w_o(e) = \begin{cases} w^*(T^*), & e \in F \text{ and } w^*(e) \neq w(e), \\ w^*(e), & \text{otherwise.} \end{cases}$$

We can see that $w_o(T^*) = w^*(T^*)$, and $E_{w^*,w^*(T^*)}^{\geq} \subseteq E_{w_o,w_o(T^*)}^{\geq}$ since $F \subseteq E(T^*)$. By Lemma 1, we can conclude that T^* is also a min–max spanning tree with respect to w_o . Furthermore, $\|w_o - w\|_{WBH} \leq \|w^* - w\|_{WBH}$ since the two weight vectors differ only for the edges e with $w^*(e) \neq w(e)$. Hence, w_o is an optimal solution satisfying the first part.

For the second part, we prove it by contradiction. For any optimal solution w^* , suppose that there always exists an edge $e \notin F$ with $w^*(e) < w(e)$. In other words,

$E_{w^*,uf} \neq \emptyset$, where $E_{w^*,uf} = \{e \notin F | w^*(e) < w(e)\}$. Among all the optimal solutions, we choose the one with the smallest $|E_{w^*,uf}|$. Let $e' \in E_{w^*,uf}$ and

$$w_o = \begin{cases} w(e'), & e = e', \\ w^*(e), & e \neq e'. \end{cases}$$

Since the two weight vectors w^* and w_o differ only for the edge e' , we can see that

$$\|w_o - w\|_{WBH} \leq \|w^* - w\|_{WBH} \quad \text{and} \quad E_{w_o,uf} = E_{w^*,uf} \setminus \{e'\}.$$

Hence, if w_o is a feasible solution of I , this will contradict with the optimality of w^* or the choice of w^* . We shall prove the feasibility of w_o through the following two cases.

Case 1: $e' \notin E(T^*)$. In this case, $w_o(T^*) = w^*(T^*)$. If $e' \notin E_{w^*,w^*(T^*)}^{\geq}$, then $E_{w^*,w^*(T^*)}^{\geq} \subseteq E_{w_o,w_o(T^*)}^{\geq}$. If $e' \in E_{w^*,w^*(T^*)}^{\geq}$, then

$$w_o(T^*) = w^*(T^*) \leq w^*(e') < w(e') = w_o(e').$$

It implies that $e' \in E_{w_o,w_o(T^*)}^{\geq}$ and $E_{w^*,w^*(T^*)}^{\geq} = E_{w_o,w_o(T^*)}^{\geq}$. Thus, $E_{w^*,w^*(T^*)}^{\geq} \subseteq E_{w_o,w_o(T^*)}^{\geq}$ holds. By Lemma 1, we can see that T^* is a min-max spanning tree containing F with respect to w_o .

Case 2: $e' \in E(T^*)$ ($e' \in E(T^*) \setminus F$). Let $e'' = \arg \min\{w_o(e) | e \in C(T^*, e')\}$ and $T' = T^* - e' + e''$. Since $e' \notin F$, T' is a spanning tree containing F . If $w_o(e'') = w_o(T')$, by the definition of e'' , we can see that $w_o(e) \geq w_o(e'') = w_o(T')$, for any edge $e \in C(T^*, e')$. This implies that $C(T^*, e') \subseteq E_{w_o,w_o(T')}^{\geq}$, and T' is min-max with respect to w_o by Lemma 1. If $w_o(e'') < w_o(T')$, then $w_o(T') \leq w^*(T^*)$ since $T' - e'' = T^* - e'$. For any edge $e \in E_{w^*,w^*(T^*)}^{\geq}$, $w_o(e) = w(e) > w^*(e) = w^*(T^*) \geq w_o(T')$ if $e = e'$; and $w_o(e) = w^*(e) \geq w^*(T^*) \geq w_o(T')$, otherwise. This implies that $E_{w^*,w^*(T^*)}^{\geq} \subseteq E_{w_o,w_o(T')}^{\geq}$. By Lemma 1, we can see that T' is a min-max spanning tree containing F with respect to w_o . □

3 PIMMST with value of optimal tree constraint

In this section, we will introduce a variant of PIMMST, the partial inverse optimal value problem on min-max spanning tree (PIOV_MMST). By studying PIOV_MMST and giving the algorithm for this problem, we will obtain the first algorithm for PIMMST in the Sect. 4. In the previous work, there are some related results for this variant on inverse minimum spanning tree (IOV_MMST). Zhang et al. (2020) presented an algorithm for it under the Chebyshev norm with running time $O(mn)$, and Wang et al. (2021) presented an algorithm for it under the bottleneck Hamming distance with running time $O(mn \log n)$.

At the beginning, we give the definition of PIOV_MMST under the bottleneck Hamming distance.

Definition 2 Given an instance $I = (G, F, w, c)$ of PIMMST and a real number r^* , the goal of PIOV_MMST under the weighted bottleneck Hamming distance is to find a new weight vector w^* satisfying:

- (1) F is included in a min–max spanning tree T^* of G with respect to w^* and $w^*(T^*) = r^*$;
- (2) $\|w^* - w\|_{WBH} = \max_{e \in E} \{c(e) \cdot H(w(e), w^*(e))\}$ is minimum.

Easy to see that PIOV_MMST is obtained by adding the value of optimal tree constraint on PIMMST.

3.1 Decision version of PIOV_MMST

To deal with the PIOV_MMST, we firstly consider the decision version of PIOV_MMST. Let $I = (G, F, w, c; r^*)$ be an instance of PIOV_MMST under the bottleneck Hamming distance, and a positive real number R . The decision version of I with R is to ask whether there exists a feasible weight vector w' such that $\|w' - w\|_{WBH} \leq R$.

The following theorem shows that the decision version of PIOV_MMST can be transformed into a feasibility problem of a specific weight vector.

Theorem 2 *Let $I = (G, F, w, c; r^*, R)$ be an instance of decision version of PIOV_MMST under the weighted bottleneck Hamming distance. Then, the answer of I is “True” if and only if w_R is a feasible solution of $I' = (G, F, w, c; r^*)$ of PIOV_MMST, where*

$$w_R(e) = \begin{cases} r^*, & c(e) \leq R, \\ w(e), & \text{otherwise.} \end{cases} \tag{1}$$

Proof The sufficiency can be deduced from $\|w_R - w\|_{WBH} \leq R$. So, we will prove the necessity in the following.

Suppose that the answer of I is “True”, this means that there is a new weight vector w' such that F can be extended into a min–max spanning tree T' with respect to w' and $w'(T') = r^*$, and $\|w' - w\|_{WBH} \leq R$. Now, we shall show that $w_R(T') = r^*$ firstly. Clearly, $w_R(T') \leq r^*$ since for any edge $e \in E(T')$, $w_R(e) = r^*$ if $c(e) \leq R$ and $w_R(e) = w(e) = w'(e) \leq w'(T') = r^*$ otherwise. Let $e_M = \arg \max_{e \in T'} w'(e)$. Then $w'(e_M) = r^*$. If $c(e_M) \leq R$, we have $w_R(e_M) = r^*$; otherwise, $w_R(e_M) = w(e_M) = w'(e_M) = r^*$ since $\|w' - w\|_{WBH} \leq R$. Hence, we can obtain that $w_R(T') = r^*$.

Then, we shall show that T' is also min–max with respect to w_R . For any edge $e \in E_{w', w'(T')}^{\geq}$, we can see that $w_R(e) \geq r^*$, since $w_R(e) = r^*$ if $c(e) \leq R$ and $w_R(e) = w(e) = w'(e) \geq r^*$ otherwise. This implies that $E_{w', w'(T')}^{\geq} \subseteq E_{w_R, w_R(T')}^{\geq}$. Hence, by Lemma 1, we can see that T' is min–max with respect to w_R . □

By Theorem 2, we propose Alg. 1 to solve the decision version of PIOV_MMST.

Theorem 3 *Alg. 1 is a linear-time algorithm for decision version of PIOV_MMST under the weighted bottleneck Hamming distance.*

Algorithm 1 Decision version of PIOV_MMST under the WBH

Require: An instance $I = (G, F, w, c; r^*, R)$ of decision version of PIOV_MMST.

Ensure: “True” or “False”.

```

1: Calculate  $w_R$  by Eqn. (1);
2: Use the min–max spanning tree algorithm by Camerini (1978) on  $G$  and  $w_R$  to obtain  $T$ ;
3: if  $w_R(T) = r^*$  and  $\max_{e \in F} w_R(e) \leq r^*$  then
4:   return “True”;
5: else
6:   return “False”;
7: end if

```

Proof Clearly, Line 1 and Line 2 both cost $O(m)$ time and Line 3 costs $O(|F|)$ time. Hence, the running time of Alg. 1 is $O(m)$.

For the correctness, if $w_R(T) = r^*$ and $\max_{e \in F} w_R(e) \leq r^*$, it means that the weight of a min–max spanning tree of G with respect to w_R is r^* , and $w_R(e) \leq w_R(T)$ for any edge $e \in F \cup E(T)$. Since subgraph $G' = (V, F \cup E(T))$ is connected, there is a spanning tree T' of G' containing F and $w_R(T') \leq w_R(T)$. Thus, T' is also min–max with $w_R(T') = r^*$ and w_R is feasible. By Theorem 2, the answer should be “YES”. If $w_R(T) \neq r^*$, it means that the weight of a min–max spanning tree of G with respect to w_R is not equal to r^* . So, the answer should be “No”. If $\max_{e \in F} w_R(e) > r^*$, it implies that any spanning tree T' containing F has $w_R(T') \geq \max_{e \in F} w_R(e) > r^*$. Thus, the answer should be “No”. \square

3.2 Algorithm for PIOV_MMST

According to the results of the decision version, the key point to solving PIOV_MMST is to obtain the candidate set of the optimal value. From the characteristics of the weighted bottleneck Hamming distance, it is easy to see that if $I = (G, F, w, c; r^*, R_1)$ is true, then $I = (G, F, w, c; r^*, R_2)$ is also true, for any $R_2 > R_1$; and the optimal value of any $I = (G, F, w, c; r^*)$ is equal to $c(e)$ for some edge e . Hence, we can present the following strongly polynomial-time algorithm to solve PIOV_MMST by combining Alg. 1 with Binary Search Method.

Remark 1 The running time of Alg. 2 is $O(m \log m)$. In detail, Line 1 costs $O(m \log m)$ time and Line 11 costs $O(m)$ time. In Lines 2–10, the while-loop has $O(\log m)$ iterations (Binary Search Method) and each iteration costs $O(m)$ time. Thus, Lines 2–10 costs $O(m \log m)$ time. The whole running time is $O(m \log m) + O(m \log m) + O(m) = O(m \log m)$.

3.3 PIOV_MMST with capacitated constraint

In this section, we consider the PIOV_MMST with capacitated constraint (CPIOV_MMST) under the weighted bottleneck Hamming distance. CPIOV_MMST is obtained from PIOV_MMST by adding the capacitated constraint, that is, for any

Algorithm 2 PIOV_MMST under the weighted bottleneck Hamming distance

Require: An instance $I = (G, F, w, c; r^*)$ of PIOV_MMST.
Ensure: An optimal solution w_{r^*} and the optimal value R_{r^*} .
 1: Order the edges in E as $c(e_1) \leq c(e_2) \leq \dots \leq c(e_m)$;
 2: Set $l := 1$ and $u := m$; $\backslash \backslash$ Binary search to find the optimal value
 3: **while** $l < u$ **do**
 4: Set $i := \lfloor (l + u)/2 \rfloor$;
 5: **if** Alg. 1 on $(G, F, w, c; r^*, c(e_i))$ returns “True” **then**
 6: $u = i$;
 7: **else**
 8: $l = i + 1$;
 9: **end if**
 10: **end while**
 11: Calculate $w_{c(e_u)}$ by Eqn. (1);
 12: **return** $w_{r^*} = w_{c(e_u)}$ and $R_{r^*} = c(e_u)$.

edge $e \in E$,

$$-l(e) \leq w^*(e) - w(e) \leq u(e),$$

where l and u are two non-negative weight vectors, called as lower and upper bound vector, respectively.

As the PIOV_MMST, we can obtain the following result for the decision version of CPIOV_MMST.

Theorem 4 *Let $I = (G, F, w, c, l, u; r^*)$ be an instance of CPIOV_MMST under the weighted bottleneck Hamming distance, and R be a positive real number. There is a feasible solution w' of I with $\|w' - w\|_{WBH} \leq R$ if and only if w_R is a feasible solution of I , where $E_{R,r^*} = \{e \in E | c(e) \leq R \text{ and } w(e) - l(e) \leq r^* \leq w(e) + u(e)\}$, and*

$$w_R(e) = \begin{cases} r^*, & e \in E_{R,r^*}, \\ w(e), & \text{otherwise.} \end{cases} \tag{2}$$

Proof The sufficiency can be deduced from $\|w_R - w\|_{WBH} \leq R$. Now, let’s prove the necessity.

Let T' be a min–max spanning tree with respect to w' satisfying $F \subseteq E(T')$ and $w'(T') = r^*$. Firstly, we shall show that $w_R(T') = r^*$. For any $e \in E(T')$, if $e \in E_{R,r^*}$, then $w_R(e) = r^*$; otherwise, we can see that $c(e) > R$ or $c(e) \leq R$ and $w(e) + u(e) < r^*$. If $c(e) > R$, then $w_R(e) = w(e) = w'(e) \leq w'(T') = r^*$; if $c(e) \leq R$ and $w(e) + u(e) < r^*$, then $w_R(e) = w(e) < r^*$. Thus, we can conclude that $w_R(T') \leq r^*$. Let’s choose an edge $e_M \in E(T')$ with $w'(e_M) = r^*$. If $e_M \in E_{R,r^*}$, then $w_R(e_M) = r^*$; otherwise, we can see that $c(e) > R$ and $w_R(e_M) = w(e_M) = w'(e_M) = r^*$. Hence, we can obtain that $w_R(T') = r^*$.

Then, we shall show that T' is also min–max with respect to w_R . For any edge $e \in E_{w',w'(T')}$, if $e \in E_{R,r^*}$, then $w_R(e) = r^*$; otherwise, we can see that $c(e) > R$ or $c(e) \leq R$ and $w(e) - l(e) > r^*$. If $c(e) > R$, then $w_R(e) = w(e) = w'(e) \geq r^*$; if $c(e) \leq R$ and $w(e) - l(e) > r^*$, then $w_R(e) = w(e) > r^*$. This implies that

$E_{w', w'(T')}^{\geq} \subseteq E_{w_R, w_R(T')}^{\geq}$. Hence, by Lemma 1, we can see that T' is min–max with respect to w_R . □

Based on Theorem 4, we can modify Alg. 1 and Alg. 2 to solve the relative problems of CPIOV_MMST.

Algorithm 1' Decision version of CPIOV_MMST under the WBH

Require: An instance $I = (G, F, w, c, l, u; r^*, R)$ of CPIOV_MMST

Ensure: “True” or “False”.

- 1: Calculate w_R by Eqn. (2);
 - 2: Use the min–max spanning tree algorithm by Camerini (1978) on G and w_R to obtain T ;
 - 3: **if** $w_R(T) = r^*$ and $\max_{e \in F} w_R(e) \leq r^*$ **then**
 - 4: **return** “True”;
 - 5: **else**
 - 6: **return** “False”;
 - 7: **end if**
-

Algorithm 2' CPIOV_MMST under the weighted bottleneck Hamming distance

Require: An instance $I = (G, F, w, c, l, u; r^*)$ of CPIOV_MMST.

Ensure: An optimal solution $w_{r^*}^*$ and the optimal value R_{r^*} .

- 1: Order the edges in E as $c(e_1) \leq c(e_2) \leq \dots \leq c(e_m)$;
 - 2: Use Binary Search Method to find the minimum $c(e_i)$ such that Alg. 1' on $(G, F, w, c, l, u; r^*, c(e_i))$ returns “True”;
 - 3: Calculate $w_{c(e_i)}$ by Eqn. (2);
 - 4: **return** $w_{r^*}^* = w_{c(e_i)}$ and $R_{r^*} = c(e_i)$.
-

Remark 2 The running times of Alg. 1' and Alg. 2' are $O(m)$ and $O(m \log m)$, respectively.

4 PIMMST under the weighted bottleneck hamming distance

In this section, we will study PIMMST under the weighted bottleneck Hamming distance, and propose two algorithms to solve it.

4.1 Algorithm based on PIOV_MMST

From Sect. 3, we obtain a strongly polynomial-time algorithm to solve PIMMST with value of optimal tree constraint. Hence, if we can get the specific weight of optimal trees or a candidate set of weight of optimal trees in polynomial scale, we can solve this problem in strongly polynomial time. The following result gives a positive answer for above question.

Theorem 5 Let $I = (G, F, w, c)$ be an instance of PIMMST under the weighted bottleneck Hamming distance. There exists an optimal solution w^* and an optimal tree T^* such that $w^*(T^*) = w(e)$ for some edge $e \in E$.

Proof Let w^* be an arbitrary optimal solution of I , and T^* be an optimal tree with respect to w^* . Suppose to the contrary that $w^*(T^*) = r^* \neq w(e)$ for any edge e . Let $\bar{e} = \arg \min_{e \in E} w(e)$. We shall prove the result through the following two cases.

Case 1: $r^* < w(\bar{e})$. It implies that for any edge $e \in E(T^*)$, $w^*(e) \leq r^* < w(\bar{e}) \leq w(e)$. Now, let's consider a new weight vector

$$w'(e) = \begin{cases} w(\bar{e}), & e \in E(T^*), \\ w(e), & \text{otherwise.} \end{cases}$$

Clearly, $w'(T^*) = w(\bar{e})$ and T^* is a min-max spanning tree with respect to w' . Furthermore,

$$\begin{aligned} \|w' - w\|_{WBH} &= \max_{e \in E(T^*)} c(e)H(w(\bar{e}), w(e)) \\ &\leq \max_{e \in E(T^*)} c(e)H(w^*(e), w(e)) \leq \|w^* - w\|_{WBH}. \end{aligned}$$

Hence, w' is an optimal solution, T^* is an optimal tree and $w'(T^*) = w(\bar{e})$.

Case 2: $r^* > w(\bar{e})$. It implies there exist an edge e^* such that $w(e^*) < r^*$ and $w(e) \notin (w(e^*), r^*)$ for each edge $e \in E$. Now, let's consider a new weight vector

$$w'(e) = \begin{cases} w(e^*), & w(e^*) < w^*(e) \leq r^*, \\ w^*(e), & \text{otherwise.} \end{cases}$$

Firstly, We show that $w'(T^*) = w(e^*)$. For any $e \in T^*$, if $w^*(e) \leq w(e^*)$, then $w'(e) = w^*(e) \leq w(e^*)$; if $w^*(e) > w(e^*)$ (such edge always exists, since $w^*(T^*) = r^* > w(e^*)$), then $w'(e) = w(e^*)$. Thus, $w'(T^*) = w(e^*)$.

Then, we shall show that T^* is also a min-max spanning tree with respect w' . For any edge $e \in E_{w^*, w^*(T^*)}^{\geq}$, if $w^*(e) = w^*(T) = r^*$, then $w'(e) = w(e^*) = w'(T^*)$; if $w^*(e) > r^*$, then $w'(e) = w^*(e) > r^* > w(e^*) = w'(T^*)$. This implies that $E_{w^*, w^*(T^*)}^{\geq} \subseteq E_{w', w'(T^*)}^{\geq}$. By Lemma 1, T^* is a min-max spanning tree with respect w' .

Now, we can see that w' is a feasible solution of I . Since, for any edge e with $w(e^*) < w^*(e) \leq r^*$, $H(w'(e), w(e)) \leq 1 = H(w^*(e), w(e))$, we can obtain that

$$\|w' - w\|_{WBH} \leq \|w^* - w\|_{WBH}.$$

Hence, w' is an optimal solution, T^* is an optimal tree and $w'(T^*) = w(e^*)$. □

Combining this results with Alg. 2, we propose the following algorithm.

Remark 3 The running time of Alg. 3 is $O(m^2 \log m)$. In detail, Line 1 costs $O(m \log m)$ time and both of Line 5 and Line 6 cost $O(m)$ time. About the For-loop, it

Algorithm 3 PIMMST under the weighted bottleneck Hamming distance

Require: An instance $I = (G, F, w, c)$ of PIMMST.

Ensure: An optimal solution w^* .

- 1: Sorting the weights of all edges in E with $r_1 < r_2 < \dots < r_t$;
 - 2: **for** $j = 1$ to t **do**
 - 3: Execute Alg. 2 with $(G, F, w, c; r_j)$ to obtain R_{r_j} ;
 - 4: **end for**
 - 5: Set $R^* := \min_{1 \leq j \leq t} R_{r_j}$ and $j^* := \arg \min_{1 \leq j \leq t} R_{r_j}$;
 - 6: Calculate w_{R^*} by Eqn. (1) with parameters $r^* = r_{j^*}$;
 - 7: **return** $w^* = w_{R^*}$.
-

has at most $O(m)$ iterations and each iteration costs $O(m \log m)$ time. Thus, For-loop costs $O(m^2 \log m)$ time. The whole running time is $O(m \log m) + O(m^2 \log m) + O(m) = O(m^2 \log m)$.

4.2 A better algorithm

In this subsection, we will give a simpler algorithm for PIMMST under the weighted bottleneck Hamming distance with quasi-linear running time. To do this, we firstly give a necessary and sufficient condition for determining whether there is a min–max spanning tree including F . For any edge subset E' , define $w(E') = \max_{e \in E'} w(e)$.

Lemma 2 *Let $G = (V, E)$ be a connected graph, F be a forest and w be an edge weight vector. There exists a min–max spanning tree of G with respect to w containing F if and only if $G - E_{w,w(F)}^{\geq}$ is disconnected.*

Proof Let $G' = G - E_{w,w(F)}^{\geq}$. For the necessity, suppose to the contrary that G' is connected. This implies that G' has a spanning tree T' with $w(T') < w(F)$. Hence, F cannot be contained in any min–max spanning tree of G with respect to w .

For the sufficiency, since G' is disconnected, then the weight of any spanning tree of G is at least $w(F)$. Let T be a min–max spanning tree of G with respect to w . Consider the subgraph $G'' = (V, T \cup F)$, since G'' is connected, there is a spanning tree T_F containing F . Since $w(T) \geq w(F)$, we can see that

$$w(T_F) \leq w(T \cup F) = \max\{w(T), w(F)\} \leq w(T).$$

Hence, T_F is a min–max spanning tree of G with respect to w . □

From Lemma 2, we can obtain the following result for the decision version of PIMMST.

Theorem 6 *Let $I = (G, F, w, c)$ be an instance of PIMMST under the weighted bottleneck Hamming distance, and R be a positive real number. There is a feasible solution w' of I with $\|w' - w\|_{WBH} \leq R$ if and only if $F_R = \emptyset$ or w_R is a feasible solution of I , where $F_R = \{e \in F | c(e) > R\}$, and*

$$w_R(e) = \begin{cases} w(F_R), & c(e) \leq R, \\ w(e), & \text{otherwise.} \end{cases} \tag{3}$$

Proof Let’s prove the sufficiency firstly. If w_R is feasible, statement holds because of $\|w_R - w\|_{WBH} \leq R$. If $F_R = \emptyset$, it implies that $c(e) \leq R$ for each edge $e \in F$. Consider a new weight vector

$$w_m(e) = \begin{cases} w(\bar{e}), & e \in F, \\ w(e), & \text{otherwise,} \end{cases}$$

where $\bar{e} = \arg \min_{e \in E} w(e)$. Clearly,

$$\|w_m - w\|_{WBH} = \max_{e \in F} c(e)H(w(\bar{e}), w(e)) \leq \max_{e \in F} c(e) \leq R.$$

Furthermore, by Lemma 2, w_m is a feasible solution of I , since $E_{w_m, w_m(F)}^{\geq} = E$ and $G' = G - E_{w_m, w_m(F)}^{\geq}$ is an empty graph.

Then, we shall show the necessity by proving the feasibility of w_R when $F_R \neq \emptyset$. For any edge $e \in F$, if $c(e) \leq R$, $w_R(e) = w(F_R)$; otherwise, $w_R(e) = w(e) \leq w(F_R)$. Thus, $w_R(F) = w(F_R)$. For each edge $e \in E_{w', w'(F)}^{\geq}$, if $c(e) \leq R$, we obtain $w_R(e) = w(F_R) = w_R(F)$; otherwise, we have

$$w_R(e) = w(e) = w'(e) \geq w'(F) \geq w'(F_R) = w(F_R) = w_R(F).$$

This implies that $E_{w', w'(F)}^{\geq} \subseteq E_{w_R, w_R(F)}^{\geq}$. Hence, by Lemma 2, w_R is a feasible solution of I . □

By Theorem 6 and the characteristics of the weighted bottleneck Hamming distance. We can obtain the following quasi-linear time algorithm to solve PIMMST.

Algorithm 4 Simpler Algorithm for PIMMST under WBH

Require: An instance $I = (G, F, w, c)$ of PIMMST.

Ensure: An optimal solution w^* and the optimal value r^* .

1: Sort the norms of edges in E with $c_1 \leq c_2 \leq \dots \leq c_m$;

2: Use Binary Search Method to find the minimum c_j such that F_R is an empty set or $G' = G - E_{w_{c_j}, w_{c_j}(F)}^{\geq}$ is disconnected, where w_{c_j} is obtained by Eqn. (3);

3: **return** $w^* = w_{c_j}$ and $r^* = c_j$.

Remark 4 The running time of Alg. 4 is $O(m \log m)$. In detail, Line 1 costs $O(m \log m)$ time. In Line 2, it has $O(\log m)$ iterations and each iteration costs $O(m)$ time. Thus, Line 2 costs $O(m \log m)$ time. The whole running time is $O(m \log m) + O(m \log m) = O(m \log m)$.

4.3 PIMMST with capacitated constraint

Similar with PIMMST, we solve the decision version of CPIMMST firstly. Given an instance $I = (G, F, w, c, l, u)$ be an instance of CPIMMST under the weighted

bottleneck Hamming distance, and a real number $R > 0$, define $F_R = \{e \in F | c(e) > R\}$, $r_1 = w(F_R)$ and $r_2 = \max_{e \in F \setminus F_R} (w(e) - l(e))$. Note that r_1 or r_2 equals to $-\infty$ if $F_R = \emptyset$ or $F_R = F$, respectively.

Theorem 7 *There is a feasible solution w' of I with $\|w' - w\|_{WBH} \leq R$ if and only if w_R is a feasible solution of I , where $r = \max\{r_1, r_2\}$, $E_r = \{e \notin F | c(e) \leq R \text{ and } w(e) < r\}$ and*

$$w_R(e) = \begin{cases} \min\{r, w(e) + u(e)\}, & e \in (F \setminus F_R) \cup E_r, \\ w(e), & \text{otherwise.} \end{cases} \tag{4}$$

Proof Firstly, we show w_R satisfies the capacitated constraint. Clearly, for any edge $e \in (F \setminus F_R) \cup E_r$,

$$w_R(e) - w(e) = \min\{r, w(e) + u(e)\} - w(e) \leq u(e). \tag{5}$$

If $w_R(e) = w(e) + u(e)$, we can see that $w_R(e) \geq w(e) \geq w(e) - l(e)$, that is,

$$-l(e) \leq w_R(e) - w(e). \tag{6}$$

If $w_R(e) = r$ and $e \in E_r$, this implies that $w_R(e) = r > w(e)$ and

$$-l(e) \leq 0 < w_R(e) - w(e). \tag{7}$$

If $w_R(e) = r$ and $e \in F \setminus F_R$, this implies that $w(e) - l(e) \leq r_2 \leq r = w_R(e)$, that is,

$$-l(e) \leq w_R(e) - w(e). \tag{8}$$

By Eqs. (5), (6), (7) and (8), w_R satisfies the capacitated constraint. Furthermore, since $\|w_R - w\|_{WBH} \leq R$, the sufficiency is proved.

For the necessity, it is only to show that there is a min-max spanning tree with respect to w_R containing F . To do this, we firstly prove that $w_R(F) = r$. For any edge $e \in F$, if $e \in F_R$, we can see that $w_R(e) = w(e) \leq w(F_R) = r_1 \leq r$; if $e \in F \setminus F_R$, then $w_R(e) = \min\{r, w(e) + u(e)\} \leq r$. Thus, $w_R(F) \leq r$. Moreover, if $r_1 \geq r_2$, it implies that $F_R \neq \emptyset$. Let $e^* = \arg \max_{e \in F_R} w(e)$. Then,

$$w_R(e^*) = w(e^*) = w(F_R) = r_1 = r.$$

If $r_1 < r_2$, it implies that $F \setminus F_R \neq \emptyset$. Let $e^* = \arg \max_{e \in F \setminus F_R} w(e) - l(e)$. Then, $r = r_2 = w(e^*) - l(e^*) \leq w(e^*) + u(e^*)$ and

$$w_R(e^*) = \min\{r, w(e^*) + u(e^*)\} = r.$$

Hence, we can deduce that $w(F_R) = r$.

Next, we shall show that $w'(F) \geq r$. For any edge $e \in F_R$, since $\|w' - w\|_{WBH} \leq R$, we have $w(e) = w'(e) \leq w'(F)$. Thus,

$$r_1 = w(F_R) = \max_{e \in F_R} w(e) \leq w'(F).$$

For any edge $e \in F \setminus F_R$, since w' is feasible, we have $w(e) - l(e) \leq w'(e) \leq w'(F)$. Thus,

$$r_2 = \max_{e \in F \setminus F_R} w(e) - l(e) \leq w'(F).$$

So, we can obtain that

$$w'(F) \geq \max\{r_1, r_2\} \geq r.$$

Finally, we show that $E_{w',w'(F)}^{\geq} \subseteq E_{w_R,w_R(F)}^{\geq}$. For any edge $e \in E_{w',w'(F)}^{\geq}$, if $e \in (F \setminus F_R) \cup E_r$, we can see that $r \leq w'(F) \leq w'(e) \leq w(e) + u(e)$ and

$$w_R(e) = r.$$

If $c(e) > R$, we can obtain that

$$w_R(e) = w(e) = w'(e) \geq w'(F) \geq r.$$

If $c(e) \leq R$, $e \notin F$ and $w(e) \geq r$, clearly

$$w_R(e) = w(e) \geq r.$$

As above, we can conclude that $E_{w',w'(F)}^{\geq} \subseteq E_{w_R,w_R(F)}^{\geq}$. Furthermore, by Lemma 2, there is a min-max spanning tree with respect to w_R containing F . □

By Theorem 7, we can obtain the following quasi-linear time algorithm to solve CPIMMST.

Algorithm 4' CPIMMST under the Weighted Bottleneck Hamming Distance

Require: An instance $I = (G, F, w, c, l, u)$ of CPIMMST.

Ensure: An optimal solution w^* and the optimal value r^* .

1: Sort the norms of edges in E with $c_1 \leq c_2 \leq \dots \leq c_m$;

2: Use Binary Search Method to find the minimum c_j such that $G' = G - E_{w_{c_j}, w_{c_j}(F)}^{\geq}$ is disconnected,

where w_{c_j} is obtained by Eqn. (4);

3: **return** $w^* = w_{c_j}$ and $r^* = c_j$.

Remark 5 The running time of Alg. 4' is the same as that of Alg. 4, which is $O(m \log m)$.

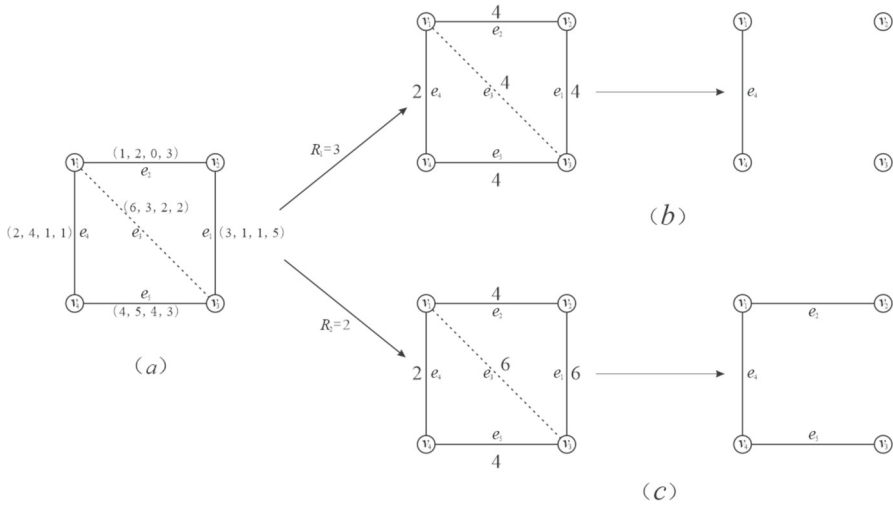


Fig. 2 **a** An instance of CPIMMST; **b** The case of $R = 3$, the left figure shows the new weight vector w_1 and the right one is the resulting graph after deleting $E_{w_1,w_1(F)}^{\geq}$; **c** The case of $R = 2$, the left figure shows the new weight vector w_2 and the right one is the resulting graph after deleting $E_{w_2,w_2(F)}^{\geq}$

Now, we shall give an example of Alg. 4'. Figure 2a is an instance of CPIMMST, in which the dash edge e_3 form the partial solution F . The four numbers in parentheses on each edge represent weight $w(e)$, norm $c(e)$, lower bound $l(e)$ and upper bound $u(e)$, respectively. The process of Alg. 4' is as follows.

Step 1: Sort the norms of edges in E with $1 < 2 < 3 < 4 < 5$;

Step 2: Use Binary Search Method to find optimal value.

Substep 2.1: $R_1 = 3$. In this case, $F_R = \emptyset$, then $r_1 = -\infty, r_2 = 6 - 2 = 4$ and $r = 4$. Hence, $E_r = \{e_1, e_2\}$ and the new weight vector $w_1 = (4, 4, 4, 2, 4)$ (left figure of Fig. 2b). Further, $E_{w_1,w_1(F)}^{\geq} = \{e_1, e_2, e_3, e_5\}$ and $G' = G - E_{w_1,w_1(F)}^{\geq}$ (right figure of Fig. 2b) is disconnected. Thus, w_1 is a feasible solution.

Substep 2.2: $R_2 = 2$. In this case, $F_R = \{e_3\}$, then $r_1 = 6, r_2 = -\infty$ and $r = 6$. Hence, $E_r = \{e_1, e_2\}$ and the new weight vector $w_2 = (6, 4, 6, 2, 4)$ (left figure of Fig. 2c). Further, $E_{w_2,w_2(F)}^{\geq} = \{e_1, e_3\}$ and $G'' = G - E_{w_2,w_2(F)}^{\geq}$ (right figure of Fig. 2c) is connected. Thus, w_2 is not a feasible solution.

Binary search complete, the optimal value is 3, and one optimal solution is $w_1 = (4, 4, 4, 2, 4)$.

Step 3: Return $w^* = w_1$ and $r^* = 3$.

5 Conclusion

In this paper, we study the partial inverse min–max tree problem under the weighted bottleneck Hamming distance. Firstly, we deal with PIOV_MMST, a variant of

PIMMST, and present an algorithm to solve it with running time $O(m \log m)$. Then, we present two algorithms to solve PIMMST. The first one is based on the algorithm for PIOV_MMST with running time $O(m^2 \log m)$. The second one is a better one with running time $O(m \log m)$. Finally, we generalize the algorithms to solve CPIOV_MMST and CPIMMST with the same running time.

Funding The work is supported in part by the National Natural Science Foundation of China [Nos.: 12271228, 12071194] and Basic Research Project of Qinghai (2021-ZJ-703).

Data Availability Enquiries about data availability should be directed to the authors.

Declarations

Conflict of interest The authors have not disclosed any competing interests.

References

- Ahuja RK, Orlin JB (2000) A faster algorithm for the inverse spanning tree problem. *J Algorithms* 34:177–193
- Ben-Ayed O, Blair CE (1990) Computational difficulties of bilevel linear programming. *Oper Res* 38(3):556–560
- Cai M-C, Duin CW, Yang X, Zhang J (2008) The partial inverse minimum spanning tree problem when weight increasing is forbidden. *Eur J Oper Res* 188:348–353
- Camerini PM (1978) The min–max spanning tree problem and some extensions. *Inf Process Lett* 7(1):10–14
- Gassner E (2010) The partial inverse minimum cut problem with L_1 -norm is strongly NP-hard. *RAIRO Oper Res* 44:241–249
- Guan X, He X, Pardalos PM, Zhang B (2017) Inverse max+sum spanning tree problem under Hamming distance by modifying the sum-cost vector. *J Global Optim* 69(4):911–925
- Guan X, Pardalos PM, Zhang B (2018) Inverse max+sum spanning tree problem under weighted l_1 norm by modifying the sum-cost vector. *Optim Lett* 12(5):1065–1077
- Guan X, Pardalos PM, Zuo X (2015) Inverse Max+Sum spanning tree problem by modifying the sum-cost vector under weighted l_∞ Norm. *J Global Optim* 61(1):165–182
- Guan X, Zhang J (2007) Inverse constrained bottleneck problems under weighted l_∞ norm. *Comput Oper Res* 34:3243–3254
- Hansen P, Jaumard B, Savard G (1992) New branch-and-bound rules for linear bilevel programming. *SIAM J Sci Stat Comput* 13:1194–1217
- He Y, Zhang B, Yao E (2005) Weighted inverse minimum spanning tree problems under hamming distance. *J Comb Optim* 9:91–100
- Hochbaum DS (2003) Efficient algorithms for the inverse spanning-tree problem. *Oper Res* 51(5):785–797
- Lai T, Orlin J (2003) The complexity of preprocessing. Research Report of Sloan School of Management. MIT, Cambridge
- Li S, Zhang Z, Lai H-J (2016) Algorithms for constraint partial inverse matroid problem with weight increase forbidden. *Theoret Comput Sci* 640:119–124
- Li X, Shu X, Huang H, Bai J (2019) Capacitated partial inverse maximum spanning tree under the weighted Hamming distance. *J Comb Optim* 38:1005–1018
- Li X, Zhang Z, Du D-Z (2018) Partial inverse maximum spanning tree in which weight can only be decreased under l_p -norm. *J Global Optim* 30:677–685
- Li X, Zhang Z, Yang R, Zhang H, Du D-Z (2020) Approximation algorithms for capacitated partial inverse maximum spanning tree problem. *J Global Optim* 77(2):319–340
- Liu L, Wang Q (2009) Constrained inverse min-max spanning tree problems under the weighted Hamming distance. *J Global Optim* 43:83–95
- Liu L, Yao E (2008) Inverse min-max spanning tree problem under the Weighted sum-type Hamming distance. *Theoret Comput Sci* 196:28–34

- Sokkalingam PT, Ahuja RK, Orlin JB (1999) Solving inverse spanning tree problems through network flow techniques. *Oper Res* 47:291–298
- Tayyebi J, Sepasian AR (2020) Partial inverse min-max spanning tree problem. *J Comb Optim* 40:1075–1091
- Wang H, Guan X, Zhang Q, Zhang B (2021) Capacitated inverse optimal value problem on minimum spanning tree under bottleneck Hamming distance. *J Comb Optim* 41:861–887
- Yang X (2001) Complexity of partial inverse assignment problem and partial inverse cut problem. *RAIRO Oper Res* 35:117–126
- Yang X, Zhang J (2007) Partial inverse assignment problem under l_1 norm. *Oper Res Lett* 35:23–28
- Yang X, Zhang J (2007) Inverse sorting problem by minimizing the total weighted number of changers and partial inverse sorting problem. *Comput Optim Appl* 36(1):55–66
- Yang X, Zhang J (2007) Some inverse min-max network problems under weighted l_1 and l_∞ norms with bound constraints on changes. *J Comb Optim* 13(2):123–135
- Zhang B, Guan X, Zhang Q (2020) Inverse optimal value problem on minimum spanning tree under unit l_∞ -norm. *Optim Lett* 14(8):2301–2322
- Zhang B, Zhang J, He Y (2006) Constrained inverse minimum spanning tree problems under the bottleneck-type Hamming distance. *J Global Optim* 34:467–474
- Zhang J, Xu S, Ma Z (1997) An algorithm for inverse minimum spanning tree problem. *Optim Methods Softw* 8(1):69–84
- Zhang Z, Li S, Lai H-J, Du D-Z (2016) Algorithms for the partial inverse matroid problem in which weights can only be increased. *J Global Optim* 65(4):801–811

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.