J Glob Optim (2016) 65:801–811
DOI 10.1007/s10898-016-0412-x

Algorithms for the partial inverse matroid problem in which weights can only be increased

Zhao Zhang[1] · Shuangshuang Li[2] · Hong-Jian Lai[3] · Ding-Zhu Du[4]

Received: 29 August 2015 / Accepted: 8 February 2016 / Published online: 15 February 2016

Abstract In a partial inverse combinatorial problem, given a partial solution, the goal is to modify data as small as possible such that there exists an optimal solution containing the given partial solution. In this paper, we study a constraint version of the partial inverse matroid problem in which the weight can only be increased. Two polynomial time algorithms are presented for this problem.

Keywords Partial inverse optimization problem · Matroid · Weight constraint · Polynomial time algorithm

1 Introduction

In an inverse problem, one is given an optimization problem with a weight function w, as well as a feasible solution X_0 which might not be optimal with respect to w, the goal is to modify w to a new weight function $\bar{w}$ as small as possible such that X_0 becomes optimal with respect to $\bar{w}$. The amount modified is measured by some norm $\|\cdot\|$. Some widely used norms include the l_p-norm $\|\cdot\|_p$ for $p = 1, 2$, or ∞ and Hamming distance $\|\cdot\|_H$. While these norms are defined on vectors, every weight function can be viewed as a vector whose coordinates are the weights of the elements. For $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$, we write $x \geq y$ if for each $1 \leq i \leq n$, $x_i \geq y_i$. A norm $\|\cdot\|$ is nondecreasing if $\|x\| \geq \|y\|$

✉ Zhao Zhang
hxhzz@sina.com

1 College of Mathematics Physics and Information Engineering, Zhejiang Normal University, Jinhua 321004, Zhejiang, China

2 College of Mathematics and System Sciences, Xinjiang University, Ürümqi 830046, Xinjiang, China

3 Department of Mathematics, West Virginia University, Morgantown, WV 26506, USA

4 Department of Computer Science, University of Texas at Dallas, Richardson, TX 75080, USA

whenever $x \geq y$. In this paper, we assume that $\|\cdot\|$ is nondecreasing and study a constraint partial inverse matroid problem.

Let S be a finite set and $\mathcal{I} \subseteq 2^S$ be a family of subsets of S. If $\mathcal{I} \neq \emptyset$ and $\mathcal{I}$ satisfies the *hereditary property* (that is, for any $I \in \mathcal{I}$, any subset of I is also in $\mathcal{I}$), then $M = (S, \mathcal{I})$ is called an *independent system*. An independent system $M = (S, \mathcal{I})$ is a *matroid* if for any $I, I' \in \mathcal{I}$ with $|I| < |I'|$, there exists an element $x \in I' \backslash I$ such that $I \cup \{x\} \in \mathcal{I}$. This property is called the *augmenting property* of matroid. Subsets in $\mathcal{I}$ are called *independent sets* and subsets of S which are not in $\mathcal{I}$ are called *dependent*. Maximal independent sets are called *bases*. It is well-known that all bases of a matroid have the same cardinality, which is called the *rank* of the matroid, denoted as r_M. It is also well-known that a matroid $M = (S, \mathcal{I})$ satisfies the *exchange property*, that is, for any distinct bases B_1, B_2 of M and any element $x \in B_1 \backslash B_2$, there is an element $y \in B_2 \backslash B_1$ such that $B_1 - x + y$ is a base. In a *matroid optimization* problem, we are given a matroid $M = (S, \mathcal{I})$ and a weight function $w : S \mapsto \mathbb{R}^+$, the goal is to find a base B with the maximum weight, where the weight of a subset $X \subseteq S$ is $w(X) = \sum_{x \in X} w(x)$. A base with the maximum weight under w is called w-*maximum*. The constraint partial inverse matroid problem (CPIM) studied in this paper is defined as follows.

Definition 1.1 ($CPIM^+$) Given a matroid $M = (S, \mathcal{I})$, a nonnegative weight function w, a nonnegative constraint function b, and an independent set $I_0 \in \mathcal{I}$, the goal is to increase weight function w to a new weight function $\bar{w}$ such that

(i) $0 \leq \bar{w}(x) - w(x) \leq b(x)$ for every $x \in S$,
(ii) I_0 is contained in a $\bar{w}$-maximum base of M,
(iii) $\|\bar{w} - w\|$ is minimized.

1.1 Related work

Burton and Toint [2] first studied the inverse shortest paths problem, which has an application in earthquake prediction [13]. After that, there have been a large quantities of studies on various inverse combinatorial problems. For comprehensive studies on this topic, the readers may refer to the surveys of Heuberger [11] and Demange and Monnot [5]. In the following, we only present some results which are closely related to this paper, namely studies on inverse matroid problem and partial inverse problem.

It has long been known that the inverse matroid problem can be solved in polynomial time. In fact, Ahuja and Orlin [1] proved that if the underlying optimization problem for each linear cost function is solvable in polynomial time, then its inverse problems under l_1 and l_∞-norms are also polynomially solvable. The result was achieved by modeling the inverse problem as a linear programming, and using an algorithm for the underlying optimization problem as a separation oracle. The inverse matroid problem falls into this realm. Notice that solving linear program is time consuming. Making use of structural properties of the problem, faster algorithm exists. For example, DellAmico et al. [4] proposed an $O(nr_M + r_M^3 + n\varphi)$ time algorithm for the inverse matroid problem, where $n = |S|$ and φ is the time complexity of finding the unique circuit when an element is added to a base, where circuit is a minimal dependent set of the matroid.

Partial inverse problem has been incorporated into several classical combinatorial optimization problems. Gentry [10] presented quadratic and MIP formulations for the partial inverse linear program and its extension. Lai and Orlin [12] studied a special partial inverse optimization problem called preprocessing problem, in which the partial solution contains only one element. They showed that the preprocessing problem under l_∞-norm is NP-hard if the underlying combinatorial optimization problem is the shortest path problem on an acyclic network, the assignment problem, the minimum cut problem, or the minimum cost arc (or node) disjoint cycle problem, while the preprocessing problem is solvable in polynomial time for matroid or lattice. Strong NP-hardness for the partial inverse minimum cut problem under unweighted l_1-norm was proved by Gassner [9]. Yang [18] proved that the partial inverse assignment problem and the partial inverse minimum cut problem are NP-hard if there are bound constraints on the modification of weights. On the other hand, the partial inverse assignment problem under l_1-norm without bound constraints can be solved in time $O(n^3)$ [19]. Yang and Zhang [20] showed that the partial inverse sorting problem can be solved in strong polynomial time under l_1, l_2 and l_∞ norms. Cai et al. [3] gave an efficient algorithm for the partial inverse minimum spanning tree problem when weight increase is forbidden. Recently, Wang [16] considered the partial inverse most unbalanced spanning tree problem and gave strongly polynomial time algorithms under weighted Hamming distance and weighted l_1-norm.

There are also a lot of variants on partial inverse problem. Yang [17] proposed the *robust partial inverse problem* in which the weights are modified as little as possible such that *all* full solutions containing the partial solution become optimal. He showed that the robust partial inverse spanning tree problem can be formulated as a linear program, and both the robust partial inverse minimum cut problem and the robust partial inverse assignment problem can be solved by combinatorial strongly polynomial algorithms. Gassner [8] introduced the *partial anti-inverse network problem* in which weights are modified as small as possible such that there is an optimal solution containing no elements in a prescribed set, and showed that partial anti-inverse shortest path/minimum cut/assignment problem under weighted l_1- or weighted l_∞-norm is strongly NP-hard.

1.2 Our contribution

In this paper, we study the constraint partial inverse matroid problem CPIM$^+$. We prove that the optimal solution (that is, the new weight $\bar{w}$) has an explicit formula. Though the formula cannot be used to find the optimal solution in polynomial time, it gives us a clue to model the CPIM$^+$ problem as a linear program which can be solved in polynomial time. The formula also gives us another clue leading to two more efficient algorithms which can be executed in time $O(k\log k + k\varphi_1 + |I_0|\varphi_2)$ and $O(n\log n + k\varphi_1 + r_M\varphi)$, respectively, where φ_1 is the time to determine whether a set is independent, φ is the time to find a fundamental circuit, and φ_2 is the time to find a fundamental cocircuit.

The paper is organized as follows. In Sect. 2, we introduce some terminologies and derive some properties which will be used in the algorithms. Polynomial time algorithms for CPIM$^+$ are studied in Sect. 3. Some discussions are given in Sect. 4.

2 Preliminaries and properties

In this section, we introduce some symbols and results which will be used in the algorithms as well as their analysis. We refer to [6,14,15] for the concepts of matroid.

A *circuit* in a matroid $M = (S, \mathcal{I})$ is a minimal dependent set. By the hereditary property of matroid, no independent set can contain a circuit. For a base B and any element $x \in S\backslash B$, there is a unique circuit in $B + x$, which is called the *fundamental circuit* with respect to B and x, and denoted as $C(B, x)$. By the uniqueness of $C(B, x)$,

$$\text{an element } y \text{ is in } C(B, x) \text{ if and only if } B + x - y \text{ is a base.} \tag{1}$$

Notice that

$$x \text{ is the only element of } C(B, x) \text{ which does not belong to } B. \tag{2}$$

For a base B and an element $x \in B$, define

$$K(B, x) = \{y \colon B - x + y \text{ is a base}\}.$$

It should be remarked that $K(B, x)$ is equivalent with *fundamental cocircuit* defined in Exercise 10 on page 78 of [14]. Since the definition of fundamental cocircuit in [14] needs more terminologies including comatriod, closure, and hyperplane, which will be a long story to tell, we choose to define $K(B, x)$ in the above simpler way and will call it fundamental cocircuit in the following. Notice that

$$K(B, x) \cap B = \{x\}. \tag{3}$$

For any $y \in K(B, x)$, $B' = B - x + y$ is a base. Notice that $B' - y + z = B - x + z$. Hence

$$K(B, x) = K(B', y). \tag{4}$$

For an element $x \in S$, let $\mathcal{C}_x$ be the set of circuits containing x. Denote by $\mathcal{P}_x = \{x\} \cup \{C - x \colon C \in \mathcal{C}_x\}$. The next lemma plays a fundamental role in our analysis.

Lemma 2.1 *For any base B, any element $x \in B$, and any $P \in \mathcal{P}_x$, $K(B, x) \cap P \neq \emptyset$.*

Proof If $P = \{x\}$, then $K(B, x) \cap P = \{x\} \neq \emptyset$. Suppose $P \neq \{x\}$. Then, $P = C - x$ for some $C \in \mathcal{C}_x$. Notice that $P \in \mathcal{I}$, $B - x \in \mathcal{I}$, and $|B - x| = r_M - 1$. We claim that there exists a subset $Q \subseteq B - x$ such that $P \cup Q$ is a base. In fact, if P is a base itself, we may take $Q = \emptyset$. If P is not a base, then $|P| < r_M = |B|$. By the augmenting property of matroid, there is a subset $Q \subseteq B \backslash P$ such that $P \cup Q$ is a base. Notice that x cannot be in Q since otherwise base $P \cup Q$ will contain a circuit $P + x$, which is impossible. Hence $Q \subseteq B - x$. The claim is proved. Then by $|B - x| < r_M = |P \cup Q|$, there is a $y \in (P \cup Q) \backslash (B - x)$ such that $B - x + y$ is a base. Since $Q \subseteq B - x$, such y can only be in P. It follows that $K(B, x) \cap P \supseteq \{y\} \neq \emptyset$. □

We call a base B to be a *w-maximum extension of I_0* if B contains the given independent set I_0 and $w(B \backslash I_0)$ is maximized. Notice that a w-maximum extension of I_0 may not be a w-maximum base. The following lemma gives a necessary and sufficient condition for a base to be a w-maximum extension.

Lemma 2.2 *For a matroid $M = (S, \mathcal{I})$, an independent set I_0, and a base B containing I_0, B is a w-maximum extension of I_0 if and only if one of the following conditions holds:*

(i) for any $x \in B \backslash I_0$, $w(x) = \max\{w(y) \colon y \in K(B, x)\}$;
(ii) for any $x \in S \backslash B$, $w(x) = \min\{w(y) \colon y \in C(B, x) \backslash I_0\}$.

Proof (i) For any $x \in B \backslash I_0$ and any $y \in K(B, x)$, $B - x + y$ is a base containing I_0. So, if B is a w-maximum extension of I_0, then $w(B) \geq w(B - x + y) = w(B) - w(x) + w(y)$, and thus $w(x) \geq w(y)$. It follows that $w(x) \geq \max\{w(y) \colon y \in K(B, x)\}$. Equality holds because $x \in K(B, x)$.

Conversely, suppose B satisfies condition (i). Let B^* be a w-maximum extension of I_0 with $|B^* \cap B|$ as large as possible. If $B \neq B^*$, let x be an element in $B \backslash B^*$, and let $P = C(B^*, x) - x$. Then $P \in \mathcal{P}_x$. By Lemma 2.1, there is an element $y \in K(B, x) \cap P$. By condition (i), we have $w(x) \geq w(y)$. By $y \in P \subseteq C(B^*, x)$ and by (1), $B' = B^* + x - y$ is a base. Since $I_0 \subseteq B^* \cap B$, we have $x \in B \backslash I_0$. Then, by (3) and the fact $I_0 \subseteq B$, we have $y \notin I_0$. It follows that B' is a base containing I_0. So, $w(B') \leq w(B^*)$. Combining this with $w(B') = w(B^*) + w(x) - w(y) \geq w(B^*)$, we conclude that B' is also a w-maximum extension of I_0. However, $|B' \cap B| > |B^* \cap B|$, contrary to the choice of B^*. So, $B = B^*$ is a w-maximum extension of I_0.

(ii) The proof for the necessity of (ii) is similar to the above by observing that for any $x \in S \backslash B$ and any $y \in C(B, x) \backslash I_0$, $B + x - y$ is a base containing I_0 and $x \in C(B, x) \backslash I_0$.

To prove the sufficiency, suppose B satisfies condition (ii) and let B^* be a w-maximum extension of I_0 with $|B^* \cap B|$ as large as possible. If $B \neq B^*$, let x be an element of $B^* \backslash B$ and let $P = C(B, x) - x$. By Lemma 2.1, $K(B^*, x) \cap P \neq \emptyset$. Let $y \in K(B^*, x) \cap P$. As $x \notin P$, we have $y \neq x$. By (3), $K(B^*, x) \cap B^* = \{x\}$. Then by $I_0 \subseteq B^*$, we have $y \notin I_0$. So, $y \in P \backslash I_0 \subseteq C(B, x) \backslash I_0$, and thus $w(x) \leq w(y)$ by condition (ii). Notice that $x \notin I_0$ since $I_0 \subseteq B$, so $B' = B^* - x + y$ is a base containing I_0, and thus $w(B') \leq w(B^*)$. Combining this with $w(B') = w(B^*) - w(x) + w(y) \geq w(B^*)$, we see that B' is also a w-maximum extension of I_0. However, $|B' \cap B| > |B^* \cap B|$, contradicting the choice of B^*. So, $B = B^*$ is a w-maximum extension of I_0. □

In particular, taking $I_0 = \emptyset$ in Lemma 2.2, we have the following characterization of w-maximum base. The second condition is exactly the one in [7].

Corollary 2.3 *For a matroid $M = (S, \mathcal{I})$, a base B is w-maximum if and only if one of the following conditions holds:*

(i) for any $x \in B$, $w(x) = \max\{w(y) \colon y \in K(B, x)\}$;
(ii) for any $x \in S \backslash B$, $w(x) = \min\{w(y) \colon y \in C(B, x)\}$.

With the same argument in the proof of Lemma 2.2, it can be seen that the conditions in Lemma 2.2 and Corollary 2.3 are equivalent to those conditions in the following remark, which are sometimes more convenient to be applied.

Remark 2.4 The two conditions in Lemma 2.2 are equivalent to

$(i)'$ for any $x \in B \backslash I_0$, $w(x) \geq \max\{w(y) \colon y \in K(B, x) \backslash \{x\}\}$;
$(ii)'$ for any $x \in S \backslash B$, $w(x) \leq \min\{w(y) \colon y \in C(B, x) \backslash (I_0 \cup \{x\}\}$.

The two conditions in Corollary 2.3 are equivalent to

$(i)''$ for any $x \in B$, $w(x) \geq \max\{w(y) \colon y \in K(B, x) \backslash \{x\}\}$;
$(ii)''$ for any $x \in S \backslash B$, $w(x) \leq \min\{w(y) \colon y \in C(B, x) \backslash \{x\}\}$.

Notice that a w-maximum extension of I_0 can be found in polynomial time, which is described in Algorithm 1.

Lemma 2.5 *The output of Algorithm 1 is indeed a w-maximum extension of I_0.*

Proof Suppose the output of Algorithm 1 is $B_0 = I_0 \cup \{x_{j_1}, \ldots, x_{j_r}\}$, where $j_1 < j_2 < \ldots < j_r$. Denote by $(B_0)_i = I_0 \cup \{x_{j_1}, \ldots, x_{j_i}\}$ the independent set obtained after the i-th iteration. We shall prove by induction on i that

$$\text{there exists a } w\text{-maximum extension of } I_0 \text{ containing } (B_0)_i. \tag{5}$$

Algorithm 1 greedy algorithm for w-maximum extension

Input: A matroid $M = (S, \mathcal{I})$ with weight function w and an independent set $I_0 \in \mathcal{I}$.
Output: A base B_0 of M which is a w-maximum extension of I_0.

Label elements of $S \backslash I_0$ as $x_1, x_2, \ldots, x_k$ such that $w(x_1) \geq w(x_2) \geq \cdots \geq w(x_k)$.
Set $B_0 \leftarrow I_0$.
for $i = 1, \ldots, k$ **do**
  if $B_0 + x_i \in \mathcal{I}$ **then**
    $B_0 \leftarrow B_0 + x_i$.
  end if
end for
Return B_0.

The basic step of $i = 0$ is obvious. Suppose the claim is true for some i. Let B' be a w-maximum extension of I_0 containing $(B_0)_i$. If $x_{j_{i+1}} \in B'$, then $(B_0)_{i+1} \subseteq B'$ and we are done. So, suppose $x_{j_{i+1}} \notin B'$. By Lemma 2.2 (ii),

$$\text{for any element } x \in C(B', x_{j_{i+1}}) \backslash I_0,\ w(x) \geq w(x_{j_{i+1}}). \tag{6}$$

For any element $x \in C(B', x_{j_{i+1}})$ which is not $x_{j_{i+1}}$, since $x \in B'$ and $(B_0)_i \subseteq B'$, we have $(B_0)_i \cup \{x\} \subseteq B'$. By the hereditary property of matroid, $(B_0)_i \cup \{x\} \in \mathcal{I}$. So, x can only be ordered after $x_{j_{i+1}}$ (otherwise x will be selected before $x_{j_{i+1}}$). It follows that

$$w(x) \leq w(x_{j_{i+1}}) \text{ for any } x \in C(B', x_{j_{i+1}}) \backslash (B_0)_i. \tag{7}$$

Combining (6) and (7), we have

$$w(x) = w(x_{j_{i+1}}) \text{ for any } x \in C(B', x_{j_{i+1}}) \backslash (B_0)_{i+1}. \tag{8}$$

Notice that $C(B', x_{j_{i+1}}) \backslash (B_0)_{i+1} \neq \emptyset$ because $(B_0)_{i+1}$ is an independent set and $C(B', x_{j_{i+1}})$ is a circuit. Let x be an element in $C(B', x_{j_{i+1}}) \backslash (B_0)_{i+1}$. Then $B'' = B' + x_{j_{i+1}} - x$ is a base containing $(B_0)_{i+1}$ and $w(B'') = w(B') + w(x_{j_{i+1}}) - w(x) = w(B')$. This finishes the induction step. □

3 Efficient algorithms for CPIM$^+$

3.1 Formula for the optimal solution

Define

$$\bar{w}(x) = \begin{cases} \max_{P \in \mathcal{P}_x} \min\{w(z) : z \in P \backslash (I_0 \backslash x)\}, & x \in I_0, \\ w(x), & \text{otherwise.} \end{cases} \tag{9}$$

Notice that $\bar{w}(x) \geq w(x)$ since $P = \{x\}$ is in $\mathcal{P}_x$. We shall show that as long as $\bar{w}(x) - w(x) \leq b(x)$ holds for any element $x \in I_0$, then $\bar{w}$ is an optimal solution to the CPIM$^+$ problem. This is proved through the following series of lemmas.

Lemma 3.1 *Any feasible solution w' to the CPIM$^+$ problem satisfies $w'(x) \geq \bar{w}(x)$ for any element $x \in S$.*

Proof If $x \notin I_0$, then $\bar{w}(x) = w(x) \leq w'(x)$. Next, consider an element $x \in I_0$. Suppose B' is a w'-maximum base containing I_0. By Corollary 2.3,

$$w'(x) = \max\{w'(y) : y \in K(B', x)\} \geq \max\{w(y) : y \in K(B', x)\}. \tag{10}$$

For each $P \in \mathcal{P}_x$, by Lemma 2.1, there exists an element $y_P \in K(B', x) \cap P$. If $y_P \in I_0$, then by (3), $y_P \in K(B', x) \cap I_0 \subseteq K(B', x) \cap B' = \{x\}$, and so $y_P = x$. It follows that no matter whether $y_P \in I_0$ or $y_P \in P - I_0$, we always have $y_P \in P \backslash (I_0 \backslash x)$, and thus

$$\max\{w(y) \colon y \in K(B', x)\} \geq w(y_P) \geq \min\{w(z) \colon z \in P \backslash (I_0 \backslash x)\}. \tag{11}$$

Taking the maximum over all $P \in \mathcal{P}_x$ in the above inequality and considering (10), we have $w'(x) \geq \bar{w}(x)$. □

Lemma 3.2 *Let B_0 be a w-maximum extension of I_0. Then B_0 is a $\bar{w}$-maximum base.*

Proof In view of Remark 2.4 $(i)''$, it suffices to show that for any $x \in B_0$,

$$\bar{w}(x) \geq \max\{\bar{w}(y) \colon y \in K(B_0, x) \backslash \{x\}\}. \tag{12}$$

Since $K(B_0, x) \cap B_0 = \{x\}$ (see (3)) and $I_0 \subseteq B_0$, we have $\bar{w}(y) = w(y)$ for any $y \in K(B_0, x) \backslash \{x\}$. So, proving (12) is equivalent to showing that for any $x \in B_0$,

$$\bar{w}(x) \geq \max\{w(y) \colon y \in K(B_0, x) \backslash \{x\}\}. \tag{13}$$

For $x \in B_0 \backslash I_0$, we have $\bar{w}(x) = w(x)$, and thus (13) follows from Remark 2.4 $(i)'$ and from the fact that B_0 is a w-maximum extension of I_0.

In the following, assume $x \in I_0$. For any $y \in K(B_0, x) \backslash \{x\}$, by (3), $y \notin B_0$, and so circuit $C(B_0, y)$ exists. Since $y \in K(B_0, x)$ implies that $B_0 - x + y$ is a base, we have $x \in C(B_0, y)$. Let $P_y = C(B_0, y) - x$. Then $P_y \in \mathcal{P}_x$. For any element $z \in P_y \backslash (I_0 \backslash x) = P_y \backslash I_0$ (notice that $x \notin P_y$), by the fact that $z \in P_y \subseteq C(B_0, y)$ and by (1), $B' = B_0 + y - z$ is a base containing I_0. It follows that $w(B') \leq w(B_0)$ and thus $w(y) \leq w(z)$. By the arbitrariness of z, we have

$$w(y) \leq \min\{w(z) \colon z \in P_y \backslash (I_0 \backslash x)\} \leq \max_{P \in \mathcal{P}_x} \min\{w(z) \colon z \in P \backslash (I_0 \backslash x)\} = \bar{w}(x).$$

Taking maximum over all $y \in K(B_0, x) \backslash \{x\}$, inequality (13) is proved. □

Recall that the norm $\|\cdot\|$ has been assumed to be nondecreasing. So by Lemma 3.2 and Lemma 3.1, we have the following result.

Theorem 3.3 *An instance of CPIM$^+$ has a feasible solution if and only if*

$$\bar{w}(x) - w(x) \leq b(x) \text{ for any element } x \in S. \tag{14}$$

Moreover, if the instance is feasible, then $\bar{w}$ is an optimal solution, and a w-maximum extension of I_0 is a $\bar{w}$-maximum base containing I_0.

Note that although the optimal solution $\bar{w}$ to the CPIM$^+$ problem has an explicit formula as presented in (9), it can not be used to calculate $\bar{w}$ efficiently since determining $\mathcal{P}_x$ in the formula is exponential. In view of Theorem 3.3, the CPIM$^+$ problem can be transformed to the inverse matroid problem as follows: find a w-maximum extension B_0 of I_0 and solve the inverse matroid problem which increases the weights of elements in I_0 as little as possible (the weight of element $x \in I_0$ is increased in the range of $[0, b(x)]$) to make B_0 a maximum weight base with respect to the new weight. The second step can be done by solving the following LP:

$$\min \sum_{x\in I_0} u_x$$
$$s.t. \sum_{x\in I_0\setminus B} u_x \geq \sum_{x\in B} w(x) - \sum_{x\in B_0} w(x), \ \forall \text{ base } B,$$
$$0 \leq u_x \leq b(x).$$

However, computing LP is time consuming. In next subsection, we present a combinatorial algorithm which is more efficient.

3.2 Combinatorial algorithm

Using the same argument in the proof of Lemma 3.1 with B' being replaced by B_0 (which is a $\bar{w}$-maximum base containing I_0 by Theorem 3.3), we conclude that

$$\bar{w}(x) = \max\{w(y)\colon y \in K(B_0, x)\} \text{ for any } x \in I_0. \quad (15)$$

This expression is much simpler than (9). It should be pointed out that before knowing that B_0 is indeed $\bar{w}$-maximum, we cannot use this expression to define $\bar{w}$ directly. The optimality is proved using definition (9). With the aid of the simpler expression (15), CPIM$^+$ can be calculated by Algorithm 2.

Algorithm 2 Algorithm for CPIM$^+$ Using Fundamental Cut

Input: A matroid $M = (S, \mathcal{I})$, a weight function $w : S \mapsto \mathbb{R}^+$, a bound function $b : S \mapsto \mathbb{R}^+$, an independent set I_0.
Output: Either claim that the instance for the CPIM$^+$ problem is infeasible, or output an optimal solution $\bar{w}$.
1: Find a w-maximum extension B_0 of I_0.
2: Set $\bar{w}(x) = w(x)$ for each $x \in S\setminus I_0$ and set $\bar{w}(x) = \max\{w(y)\colon y \in K(B_0, x)\}$ for each $x \in I_0$.
3: **if** there is an $x \in I_0$ with $\bar{w}(x) > w(x) + b(x)$ **then**
4: Return “infeasible”
5: **else**
6: Return $\bar{w}$.
7: **end if**

Theorem 3.4 *Algorithm 2 is an $O(k \log k + k\varphi_1 + |I_0|\varphi_2)$ time algorithm for the CPIM$^+$ problem, where $k = n - |I_0|$, φ_1 is the time to judge whether a set is independent and φ_2 is the time to find a fundamental cocircuit.*

Proof It should be noticed that for each $x \in I_0$, x is the only element of $K(B_0, x)$ whose weight might be changed. So, Step 2 of Algorithm 2 is well-defined. In other words, the change of weight of one element in I_0 is independent of the change of weights of other elements in I_0. Then the correctness can be seen from Theorem 3.3 and expression (15).

For the time complexity, $O(k \log k + k\varphi_1)$ is the time to find a w-maximum extension of I_0, and $O(|I_0|\varphi_2)$ is the time for Step 2 of Algorithm 2. □

Another way to realize formula (15) is to use fundamental circuit, which is described in Algorithm 3. To ease the understanding of the idea of the algorithm, we start from a coarser idea and then show how to refine it.

Order elements in $S\setminus B_0$ as $x_1, x_2, \ldots, x_k$ such that $w(x_1) \geq w(x_2) \geq \cdots \geq w(x_k)$. Recall that only elements in I_0 may have their weights changed. Consider an element $x \in I_0$.

Suppose $K(B_0, x) = \{x, x_{i_1}, \ldots, x_{i_t}\}$ such that $i_1 < \cdots < i_t$. Then, $w(x_{i_1}) \geq \cdots \geq w(x_{i_t})$. Combining this with expression (15), we have

$$\bar{w}(x) = \max\{w(x), w(x_{i_1})\}. \tag{16}$$

By observation (1) and the definition of fundamental cocircuit, we see that

$$x \in \bigcap_{j=1}^{t} C(B_0, x_{i_j}) \text{ and } x \notin C(B_0, y) \text{ for any } y \notin K(B_0, x). \tag{17}$$

A coarse idea is to consider elements $x_1, x_2, \ldots$ sequentially, and change weights of elements in $C(B_0, x_i) \cap I_0$ during the i-th iteration. Notice that the weight of x will remain intact before x_{i_1} is considered (because $x \notin C(B_0, x_i)$ for $i < i_1$ by (17)). In the i_1-th iteration, we may use (16) to set the value of $\bar{w}(x)$, and thus realizing (15). The reason why we consider elements of $S \backslash B_0$ in their decreasing order of weights is that the new weight of x can be determined at the first time an element in $K(B_0, x)$ is considered.

Notice that $(K(B_0, x) \backslash \{x\}) \cap B_0 = \emptyset$. Hence the above $x_{i_1} \notin B_0$. This is why only elements in $S \backslash B_0$ need to be considered. In the following, we show that it is sufficient to consider elements in $B \backslash B_0$ for some maximum w-base B. For this purpose, we first prove that for any w-maximum base B,

$$\exists z \in B \text{ such that } z \in K(B_0, x) \text{ and } w(z) = \max\{w(y) \colon y \in K(B_0, x)\}. \tag{18}$$

Suppose y_0 is an element in $K(B_0, x)$ such that $w(y_0) = \max\{w(y) \colon y \in K(B_0, x)\}$. If $y_0 \in B$, then $z = y_0$ satisfies (18). If $y_0 \notin B$, let $P = C(B, y_0) - y_0$ and $B_1 = B_0 - x + y_0$. Then $P \in \mathcal{P}_{y_0}$, and $P \cap K(B_1, y_0) \neq \emptyset$ (by Lemma 2.1). By (4), $K(B_0, x) = K(B_1, y_0)$. Hence $P \cap K(B_0, x) \neq \emptyset$. Let $z \in P \cap K(B_0, x)$. Since $z \in P \subseteq C(B, y_0)$, we see that $z \in B$ and $B + y_0 - z$ is a base (by (1)). Since B is w-maximum, we have $w(B + y_0 - z) \leq w(B)$, and thus $w(y_0) \leq w(z)$. On the other hand, $z \in K(B_0, x)$ implies $w(z) \leq w(y_0)$. Hence $w(z) = w(y_0) = \max\{w(y) \colon y \in K(B_0, x)\}$. Property (18) is proved.

Property (18) implies that the element z in (18) can play the role of x_{i_1} in (16). Hence only elements in $B \backslash B_0$ need to be considered (this is reflected in line 2 of Algorithm 3). In Algorithm 3, all elements outside of I_0 have their weights intact, and the set R is used to record those elements whose weights are to be determined. Initially, $R = I_0$. When some element $x \in I_0$ is considered for the first time in the inner for loop, by the ordering of elements, the new weight of x is determined once for all, and thus x can be removed from R permanently after that. When the algorithm jumps out of the outer for loop, the set R might be nonempty. In fact, an element $x \in I_0$ which is the only element in $B \cap K(B_0, x)$ will never appear in $C(B_0, x_i)$ for any $x_i \in B \backslash B_0 \subseteq B \backslash I_0$. For such an x, it is easy to see that $\bar{w}(x) = w(x)$, the new weight of x is set in line 15 of Algorithm 3.

Theorem 3.5 *Algorithm 3 is an $O(n \log n + n\varphi_1 + r\varphi)$ time algorithm for the CPIM$^+$ problem, where φ_1 is the time to judge whether a set is independent, r is the rank of the matroid, and φ is the time to find a fundamental circuit.*

Proof The correctness is the result of the argument above. The time complexity follows from the observation that finding B and B_0 takes time $O(n \log n + n\varphi_1)$, line 2 of Algorithm 3 takes time $O(r \log r)$ (notice that $m = |B \backslash B_0| \leq r$), the outer for loop is iterated $m \leq r$ times and a fundamental circuit has to be found in each iteration, all other steps take constant time. □

Algorithm 3 Algorithm for CPIM$^+$ Using Fundamental Circuit

Input: A matroid $M = (S, \mathcal{I})$, a weight function $w : S \mapsto \mathbb{R}^+$, a bound function $b : S \mapsto \mathbb{R}^+$, an independent set I_0.
Output: Either claim that the instance for the CPIM$^+$ problem is infeasible, or output an optimal solution $\bar{w}$.

Find a w-maximum extension B_0 of I_0 and a w-maximum base B.
Order elements of $B \backslash B_0$ as $x_1, x_2, \cdots, x_m$ such that $w(x_1) \geq w(x_2) \geq \cdots \geq w(x_m)$.
Set $\bar{w}(x) = w(x)$ for each $x \in S \backslash I_0$.
$R \leftarrow I_0$.
for $i = 1, \ldots, m$ **do**
    for each element $x \in C(B_0, x_i) \cap R$ **do**
        Set $\bar{w}(x) = \max\{w(x), w(x_i)\}$.
        if $\bar{w}(x) > w(x) + b(x)$ **then**
            Return "infeasible".
        else
            $R \leftarrow R \backslash \{x\}$.
        end if
    end for
end for
Set $\bar{w}(x) = w(x)$ for all remaining elements $x \in R$.
Return $\bar{w}$.

4 Conclusion

In this paper, we studied the partial inverse matroid problem under the constraint that weights can only be increased (CPIM$^+$). Efficient algorithms are given for CPIM$^+$ which are valid for any monotone norm. One interesting question is how about the constraint that weights can only be decreased? This small modification of the statement greatly increases the difficulty of the problem. The reason is that there are too many elements outside of I_0 the changes of whose weights inter-related with each other so that it is difficult to determine an explicit expression for the optimal solution. This will be an topic of our future research.

Acknowledgments This research is supported by NSFC (61222201, 11531011), SRFDP (20126501110001), and Xingjiang Talent Youth Project (2013711011).

References

1. Ahuja, R.K., Orlin, J.B.: Inverse optimization. Oper. Res. **49**(5), 771–783 (2001)
2. Burton, D., Toint, P.L.: On an instance of the inverse shortest path problem. Math. Program. **53**, 45–61 (1992)
3. Cai, M., Duin, C.W., Yang, X., Zhang, J.: The partial inverse minimum spanning tree problem when weight increase is forbidden. Eur. J. Oper. Res. **188**, 348–353 (2008)
4. DellAmico, M., Maffioli, F., Malucelli, F.: The base-matroid and inverse combinatorial optimization problems. Discrete Appl. Math. **128**, 337–353 (2003)
5. Demange, M., Monnot, J.: An introduction to inverse combinatorial problems. In: Paschos, V.T. (ed.) Paradigms of Combinatorial Optimization (Problems and New approaches). Wiley, London (2010)
6. Floudas, C.A., Pardalos, P.M. (eds.): Encyclopedia of optimization. Springer, New York (2009)
7. Frank, A.: A weighted matroid intersection algorithm. J. Algorithm **2**, 328–336 (1981)
8. Gassner, E.: Selected Partial Inverse Combinatirial Optimization Problems with Forbidden Elements. Technical Report 2009-15, Graz University of Technology (2009)
9. Gassner, E.: The partial inverse minimum cut problem with L_1-norm is strongly NP-hard. RAIRO Oper. Res. **44**, 241–249 (2010)
10. Gentry, S.: Partial Inverse Linear Programming. MIT Lab for Information and Decision Systems Report, LIDS-P-2532 (2001)

11. Heuberger, C.: Inverse optimization: a survey on problems, methods, and results. J. Combin. Optim. **8**, 329–361 (2004)
12. Lai, T., Orlin, J.: The Complexity of Preprocessing. Research Report of Sloan School of Management. MIT (2003)
13. Moser, T.J.: Shortest paths calculation of seismic rays. Geophysics **56**, 59–67 (1991)
14. Oxley, J.G.: Matroid Theory. Oxford University Press, Oxford (1992)
15. Pitsoulis, L.S.: Topics in matroid theory. Springer, New York (2014)
16. Wang, Q.: Partial inverse most unbalanced spanning tree problem. Prz. Elektrotech. **88**(1b), 111–114 (2012)
17. Yang, X.: Robust partial inverse network flow problems. Appl. Math. J. Chin. Univ. Ser. B **16**(2), 185–194 (2001)
18. Yang, X.: Complexity of partial inverse assignment problem and partial inverse cut problem. RAIRO Oper. Res. **35**, 117–126 (2001)
19. Yang, X., Zhang, J.: Partial inverse assignment problems under l_1 norm. Oper. Res. Lett. **35**, 23–28 (2007)
20. Yang, X., Zhang, J.: Inverse sorting problem by minimizing the total weighted number of changes and partial inverse sorting problems. Comput. Optim. Appl. **36**, 55–66 (2007)