The Base-Matroid and Inverse Combinatorial Optimization Problems

Mauro Dell'Amico* Francesco Maffioli† Federico Malucelli†

Abstract

A new kind of matroid is introduced: this matroid is defined starting from any matroid and one of its bases, hence we call it *Base-Matroid*. Besides some properties of the base-matroid, a non trivial algorithm for the solution of the related matroid optimization problem is devised. The new matroid has application in the field of inverse combinatorial optimization problems.

Key words: Matroids, Inverse Optimization, Combinatorial Optimization

1 Introduction

We introduce a new matroid, which we call *Base-Matroid*: the name is motivated by the fact that it is defined starting from any matroid and one of its bases. We refer to [12] for fundamentals of Matroid Theory.

Given a matroid M defined over a ground set E and having $\mathcal{F}$ as its family of independent sets, let B be one of the bases of M. Any closed set θ such that the cardinality of its intersection with B is equal to its rank is called *saturated closed set*. We denote by $\mathcal{F}_B$ the family of subsets S of E having the property that the cardinality of the intersection of S with any saturated closed set θ is not greater than the rank of θ. We prove in Section 2 that $M_B = (E, \mathcal{F}_B)$ is indeed a matroid on E. Therefore given a non-negative weighting of the elements of E, the problem of finding a base of M_B such that the sum of the weights of its elements is maximum can be solved by the greedy algorithm. A straightforward implementation of this algorithm would have an exponential complexity. Section 3 presents a more efficient implementation of the greedy algorithm requiring only $O(mn + n^3 + m\varphi)$ time, where $m = |E|$, n is the rank of E in M and φ is the complexity of finding the unique circuit formed by adding to the given base an element not in the base.

The motivation for introducing base-matroids came from studying *Inverse Combinatorial Optimization Problems* (ICOP). In general these problems ask for the "smallest" perturbation of the weighting of the elements of the ground set E which would make a given feasible subset of E optimal. Many different inverse problems have been addressed in the recent literature [1, 2, 4, 5, 6, 9]. Their applications range from traffic control to seismic tomography (see [4], [10]). We focus on one of the most fundamental ICOPs, namely the *inverse matroid problem*. Given a matroid $M = (E, \mathcal{F})$ and a target base B of M, this problem looks for perturbation parameters δ to be added to the weight c of the elements of E so that B becomes optimal for the (direct) matroid optimization problem with the new weighting $w_e = c_e + \delta_e, \forall e \in E$, and a given function of the perturbation

[*] Dipartimento di Scienze e Metodi dell'Ingegneria - Università di Modena e Reggio Emilia, Viale Allegri 13, 42100 Reggio Emilia - Italy

[†] Dipartimento di Elettronica e Informazione - Politecnico di Milano, Via Ponzio 34/5, 20133 Milano - Italy

parameters is minimized. We consider the case in which such function is given by the sum of the absolute values of δ_e's. We show in Section 4 that this problems can be solved by suitably exploiting an ad-hoc base-matroid.

2 Base-matroids

Consider a *matroid* $M = (E, \mathcal{F})$ defined by a *ground set of elements* E and a *family of independent sets* $\mathcal{F} \subseteq 2^E$. The three following axioms define a matroid [12]:

(a.1) $\emptyset \in \mathcal{F}$;

(a.2) $X \in \mathcal{F} \Longrightarrow \forall X' \subseteq X, X' \in \mathcal{F}$;

(a.3) $X \in \mathcal{F}$, $Y \in \mathcal{F}$ and $|X| > |Y| \Longrightarrow \exists x \in X \backslash Y$ such that $Y \cup \{x\} \in \mathcal{F}$.

Given a weighting function $c : E \rightarrow \mathbb{R}^+$, let $C : \mathcal{F} \rightarrow \mathbb{R}^+$, be defined as $C(S) = \sum_{e \in S} c_e$. The *matroid optimization problem* is to determine

$$\max_{S \in \mathcal{F}} C(S) \tag{1}$$

The *rank* r of a set $S \subseteq E$ is the cardinality of the largest $S' \subseteq S$ such that $S' \in \mathcal{F}$. Given a set $S \in \mathcal{F}$, let us denote with $\sigma(S)$ the *closure* of S, i.e. the superset of S obtained by adding to S all elements e such that $r(S \cup \{e\}) = r(S)$. A set θ is *closed* if $\theta = \sigma(\theta)$, i.e. $r(\theta \cup \{e\}) = r(\theta) + 1$ for all $e \in E \backslash \theta$. In the following we denote by Θ the set of all closed sets of M.

A *base* of matroid M is a set $B \in \mathcal{F}$ of maximum cardinality. The optimal solution of a matroid optimization problem is a base. Note that all the bases of a matroid have the same cardinality equal to $r(E)$. Let us define $m = |E|$ and $n = r(E) = |B|$. The following definitions refer to a given a matroid $M = (E, \mathcal{F})$ and a base B.

Definition 2.1 *We call* saturated *a set $\theta \subseteq E$ such that $|\theta \cap B| = r(\theta)$. If $\theta \in \Theta$, we have a* saturated closed *set. The set of all the saturated closed sets of M, with respect to base B, is denoted by Θ_B.*

Note that given any saturated closed set θ we have $\sigma(\theta \cap B) = \theta$.

Definition 2.2 *Given a closed set $\theta \in \Theta_B$, we call $\theta \cap B$ the* skeleton *of θ, with respect to base B.*

A *circuit* is a minimal dependent set, i.e. a set $S \notin \mathcal{F}$ such that for each $i \in S$, $S \backslash \{i\} \in \mathcal{F}$. Given a base B and an element $i \in E \backslash B$, the *fundamental circuit* of i is the minimal subset of $B \cup \{i\}$ which is not in $\mathcal{F}$ (note that i always belongs to its fundamental circuit). More specifically, calling $\gamma(i)$ the unique minimal subset of B such that $\gamma(i) \cup \{i\} \notin \mathcal{F}$, then $\gamma(i) \cup \{i\}$ is a fundamental circuit. Moreover let $\theta_B(i) = \sigma(\gamma(i) \cup \{i\})$ denote the closure of the fundamental circuit $\gamma(i) \cup \{i\}$. We call $\theta_B(i)$ the *fundamental closed set* associated with the base B and the element $i \in E$ (note that $\theta_B(i) = \{i\}$ when $i \in B$.) Finally observe that the rank of a fundamental closed set is $r(\theta_B(i)) = r(\gamma(i))$, thus any fundamental closed set is also saturated.

The following definition introduces a new matroid called *base-matroid*. The name is due to the fact that the new matroid is obtained from a given matroid, by considering a subset of constraints, in particular those saturated by a given base.

Definition 2.3 *Given a matroid $M = (E, \mathcal{F})$ and a base B let $\mathcal{F}_B = \{S \subseteq E : |S \cap \theta| \leq r(\theta), \forall \theta \in \Theta_B\}$, then $M_B = (E, \mathcal{F}_B)$ is the* base-matroid *induced by base B.*

One can easily prove that axioms (a.1) and (a.2) hold, i.e. that M_B is a *System of Independence*. In the following we prove that also axiom (a.3) holds, hence obtaining

Theorem 2.1 *$M_B = (E, \mathcal{F}_B)$ is a matroid.*

In order to prove Theorem 2.1 (whose proof is postponed to Section 2.2), we need to introduce several general properties related to closed sets and their skeletons. Some of these properties are also used to devise an efficient algorithm for the base-matroid optimization problem (see Section 3).

2.1 General properties

The first property relates fundamental circuits and saturated closed sets.

Property 2.1 *Given any saturated closed set $\theta \in \Theta_B$ and any $e \in \theta \backslash B$, then $\gamma(e) \subseteq \theta$.*

Proof. By definition of saturated closed set $r(\theta) = |\theta \cap B|$, hence θ can be obtained as the closure of the skeleton $\theta \cap B$. It immediately follows that $e \in \theta$ implies that $r(\theta \cap B) = r(\theta \cap B \cup \{e\})$, thus $\gamma(e) \subseteq \theta \cap B$. □

Note that this property does not hold for $\theta \in \Theta \backslash \Theta_B$. For instance consider the matric matroid whose elements are the columns of the following matrix:

$$\begin{pmatrix} 5 & 3 & 1 & 4 & 8 \\ 3 & 1 & 1 & 2 & 4 \\ 0 & 2 & 0 & 2 & 2 \end{pmatrix}$$

and independence is over the field of real. Consider the base $B = \{e_1, e_2, e_3\}$ and the closed set $\theta = \{e_4, e_5\}$ which is not saturated. For both elements $e_4, e_5 \in \theta \backslash B$ the corresponding γ sets are $\gamma(e_4) = \{e_2, e_3\}$ and $\gamma(e_5) = \{e_1, e_2\}$ (in fact $e_4 = e_2 + e_3$ and $e_5 = e_1 + e_2$), which are not contained in θ.

The following property deals with the intersection of closed sets, whereas the next three ones consider the union of closed sets.

Note that given two closed sets θ_1, θ_2, then also $\theta = \theta_1 \cap \theta_2$ is a closed set, indeed by definition of closure, $\theta \subseteq \theta_\ell \Rightarrow \sigma(\theta) \subseteq \sigma(\theta_\ell) = \theta_\ell$, for $\ell = 1, 2$ (see [12], chapter 1.2). Hence $\sigma(\theta) = \theta$.

Property 2.2 *The intersection of two saturated closed sets is a saturated closed set.*

Proof. Consider two saturated closed sets $\theta_1, \theta_2 \in \Theta_B$, and consider the closed set $I = \theta_1 \cap \theta_2$. ¿From Property 2.1 we know that the fundamental circuit of any element $e \notin B$ belongs to all the saturated closed sets containing e, hence for each element $e \in I \backslash B$ it is $\gamma(e) \subseteq I$. This implies that $r(I) \geq |B \cap I|$, but it also implies that adding any element $e \in I \backslash B$ to $B \cap I$ we cannot obtain a set with rank larger than $|B \cap I|$, therefore $r(I) = |B \cap I|$ and I is a saturated closed set whose skeleton is $B \cap I$. □

Note that, being the fundamental closed sets particular cases of saturated closed sets, also the intersection of two fundamental closed sets is saturated.

Property 2.3 *A set $U \subseteq E$ which is the union of saturated sets is saturated.*

Proof. Consider a family of saturated sets $\theta_1, \ldots, \theta_h$ and their skeletons $S_1, \ldots, S_h$. Let $U = \cup_{i=1,\ldots,h} \theta_i$, $S = \cup_{i=1,\ldots,h} S_i$ and observe that $B \cap U = S$. Given any $e \in U \backslash S$ there is a skeleton S_i such that $S_i \cup \{e\} \notin \mathcal{F}$, therefore $S \cup \{e\} \notin \mathcal{F}$ and S is the skeleton of U. It immediately follows that $r(U) = |S| = |B \cap U|$, i.e. U is saturated. □

Property 2.4 *Given a set $U \subseteq E$ which is the union of saturated closed sets, then $\sigma(U) \backslash U \subseteq E \backslash B$.*

Proof. Let us suppose, by contradiction, that there exists $e \in \sigma(U) \backslash U$ such that $e \in B$. Then $|B \cap \sigma(U)| \geq |B \cap U| + 1$ thus $r(\sigma(U)) > |B \cap U|$. The fact that $r(\sigma(U)) = |B \cap U|$ (Property 2.3) leads to a contradiction, and the thesis holds. □

Property 2.5 *A saturated closed set is the closure of the union of fundamental closed sets.*

Proof. Given a saturated closed set $\theta \in \Theta_B$ consider any element $e \in \theta \backslash B$. We know (see Property 2.1) that $\gamma(e)$ is a subset of θ, therefore all the elements that added to $\gamma(e)$ determine a circuit belong to the closed set θ. It immediately follows that $\theta_B(e) \subseteq \theta$. By applying the same reasoning to all $e \in \theta \backslash B$, the thesis follows. □

Note that given two fundamental closed sets θ_1 and θ_2, the closure of their union may fail to be a fundamental set. For instance consider the graphic matroid given by the graph in Figure 1, the base $B = \{a, b, c, d, e, h\}$ and the sets $\theta_1 = \theta_B(f) = \{a, b, c, f\}$ and $\theta_2 = \theta_B(g) = \{d, b, e, g\}$ which are fundamental closed sets. The set $S = \sigma(\theta_1 \cup \theta_2) = \{a, b, c, d, e, f, g, i\}$ is a saturated closed set (see Property 2.3), but it is not fundamental since there is no element $x \in S$ such that $r(S) = r(\gamma(x))$.

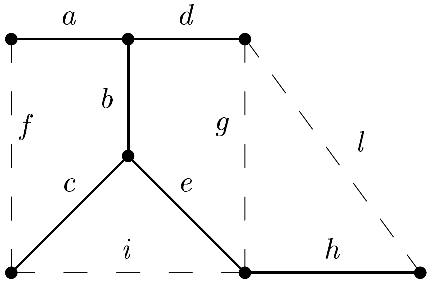

Figure 1: A graphic matroid (base elements are drawn as continuous lines)

2.2 Proof of Theorem 2.1

We are now ready to prove the main theorem.

Proof. M_B is a matroid on the ground set E if and only if it satisfies the three axioms (a.1)-(a.3). It is not difficult to prove that (a.1) and (a.2) hold. Using a proof by contradiction we show that also axiom (a.3) holds. Consider two sets X and $Y \in \mathcal{F}_B$ with $|X| = |Y| + 1$ and recall that, by definition of $\mathcal{F}_B$, for all saturated closed sets $\theta \in \Theta_B$ we have $|X \cap \theta| \leq r(\theta)$ and $|Y \cap \theta| \leq r(\theta)$. Let $X = \{e_1, \ldots, e_{|X|}\}$, and suppose that axiom (a.3) does not hold, that is, even if $|X| > |Y|$, $\forall e \in X \backslash Y$: $Y \cup \{e\} \notin \mathcal{F}_B$. Hence for each $e_i \in X$ there is a saturated closed set $\theta_i \in \Theta_B$ such that

$$|Y \cup \{e_i\} \cap \theta_i| > r(\theta_i) \tag{2}$$

Let us assume that θ_i (for $i = 1, \ldots, |X|$) denotes the maximum cardinality closed set of Θ_B which satisfies (2). We observe that:

(i) $e_i \in \theta_i$ and $e_i \notin Y$, otherwise $|Y \cup \{e_i\} \cap \theta_i| = |Y \cap \theta_i| \leq r(\theta_i)$ contradicting (2);

(ii) $|Y \cap \theta_i| = r(\theta_i)$, since by adding element e_i to $Y \cap \theta_i$ the cardinality of the new set would exceed $r(\theta_i)$.

Condition (ii) implies that in each closed set θ_i there is at least one element of Y, therefore if the closed sets $\theta_i, i = 1, \ldots, |X|$, are mutually disjoint it results $|Y| \geq |X|$, contradicting the hypothesis. Hence we consider the case in which some of the closed sets θ_i (for $i = 1, \ldots, |X|$) do intersect. In particular let θ_h and θ_k, with $h, k \in \{1, \ldots, |X|\}$, be two closed sets such that $\theta_h \cap \theta_k \neq \emptyset$. To conclude the proof we will use the following

Claim 2.1 *If $\theta_h \cap \theta_k \neq \emptyset$, then $\theta_h \equiv \theta_k$.*

Proof. Let us define $I = \theta_h \cap \theta_k$ and $U = \sigma(\theta_h \cup \theta_k)$, and recall that both I and U are saturated closed sets (see Properties 2.2 and 2.3). First observe that

$$|Y \cap U| \geq |Y \cap (\theta_h \cup \theta_k)| = |Y \cap \theta_h| + |Y \cap \theta_k| - |Y \cap I| \tag{3}$$

then from the facts that I and U are saturated closed sets, and that $B \cap U = B \cap (\theta_h \cup \theta_k)$

$$r(U) = |B \cap U| = |B \cap \theta_h| + |B \cap \theta_k| - |B \cap I| \tag{4}$$

Moreover since Y is an independent set we have

$$r(U) \geq |Y \cap U| \tag{5}$$

Finally recalling that $|Y \cap \theta_\ell| = r(\theta_\ell) = |B \cap \theta_\ell|$, for $\ell = h, k$ and using (3) - (5) we obtain

$$|Y \cap I| \geq |B \cap I|$$

But this inequality must hold with the '=' sign since Y is an independent set and $|B \cap I| = r(I)$. Using this fact and (3)-(5) we have $|Y \cap U| \leq r(U) = |B \cap \theta_h| + |B \cap \theta_k| - |B \cap I| = |Y \cap \theta_h| + |Y \cap \theta_k| - |Y \cap I| \leq |Y \cap U|$, hence $|Y \cap U| = r(U)$.

Finally from condition (i) above and the definition of U we know that $e_\ell \in U$ and $e_\ell \notin Y$, for $\ell = h, k$, thus $|Y \cup \{e_\ell\} \cap U| > |Y \cap U|$ ($\ell = h, k$), and (2) holds both for e_h and e_k, when U is used instead of θ_h and θ_k, respectively. But each closed set θ_i (for $i = 1, \ldots, |X|$) has been chosen as the largest closed set satisfying (2), hence $\theta_h = U$ and $\theta_k = U$ and the claim holds. □

We have thus proved that given a pair of closed sets in $\{\theta_i : i = 1, \ldots, |X|\}$, they either are disjoint or coincide. Let us identify with $\theta'_1, \ldots, \theta'_q$ ($q \leq |X|$), the disjoint closed sets in $\{\theta_i : i = 1, \ldots, |X|\}$. Recalling that: (a) for the feasibility of X then $|X \cap \theta'_i| \leq r(\theta'_i)$ $\forall i$; (b) each element $e_i \in X$ belongs to one and only one closed set θ'_i; and (c) $|Y \cap \theta'_i| = r(\theta'_i)$ $\forall i$ (see (ii) above), we conclude that

$$|X| = \sum_{i=1}^{q} |X \cap \theta'_i| \leq \sum_{i=1}^{q} r(\theta'_i) = \sum_{i=1}^{q} |Y \cap \theta'_i| \leq |Y|$$

which contradicts the hypothesis and proves the theorem. □

3 An algorithm for the base-matroid optimization problem

Being M_B a matroid, the corresponding optimization problem can be solved by means of a greedy algorithm which, starting from an empty solution, iteratively selects the elements by non-increasing value of c_e and checks whether the union of the current solution with the selected element is still independent. The key issue in the implementation of such an algorithm is to efficiently test the independence of a given set S of elements. A straightforward implementation of this test requires to scan all the $|\Theta_B|$ constraints. In this Section we propose a more efficient method allowing to implement the greedy algorithm so that it runs in $O(mn + n^3 + m\varphi)$, where φ is the computational complexity of a procedure which determines $\gamma(e)$, for a given $e \in E \backslash B$.

The main idea we use is to define a mapping associating the elements of a set with the elements of the target base B. More formally, given a set $S \subseteq E$ let us call *base-mapping* a function $a : S \rightarrow B$ such that $a(i) \in \gamma(i)$ if $i \in S \backslash B$, $a(i) = i$ if $i \in B$, and $a(i) \neq a(j)$ $i \neq j$. The existence of a base-mapping is directly related to the independence of S. In the following, we denote by $a(S)$ the mapping of a set S into B (i.e. $a(S) = \{a(i) : i \in S\}$).

Theorem 3.1 *Given a base-matroid $M_B = (E, \mathcal{F}_B)$ and a set $S \subseteq E$, there exists a base-mapping $a : S \rightarrow B$ if and only if S is an independent set of M_B.*

Proof. First we prove that if there is a base-mapping $a : S \rightarrow B$ then $S \in \mathcal{F}_B$. Consider any saturated closed set $\theta \in \Theta_B$. Since there exists the base-mapping for S, then $|S \cap \theta| = |a(S \cap \theta)|$. To prove the independence of S we show that $|a(S \cap \theta)| \leq r(\theta)$. We first note that $a(S \cap \theta) \subseteq \theta$, indeed, by definition of base-mapping: (**i**) $a(i) = i$ for each $i \in S \cap \theta \cap B$; (**ii**) $a(i) \in \gamma(i)$ for each $i \in (S \cap \theta) \backslash B$, but $\gamma(i) \in \theta$ for each element $i \in \theta \backslash B$ (see Property 2.1). Recalling that $a(i) \in B$ for all $i \in S$ we obtain $|a(S \cap \theta)| \leq |B \cap \theta| = r(\theta)$ and the independence of S follows.

Now we prove the second part of the thesis. Suppose that $S \in \mathcal{F}_B$ and consider a bipartite graph $G = (V_1 \cup V_2, L)$ where V_1 has one vertex v'_i for each element $i \in S$, V_2 has one vertex v''_j for each element $j \in B$, and the edge set L has one edge (v'_i, v''_i) for each $i \in S \cap B$ and one edge (v'_i, v''_j) for each $i \in S \backslash B$ and $j \in \gamma(i)$. One can see that each matching $\mathcal{M}$ of G with $|\mathcal{M}| = |V_1|$ corresponds to a base-mapping for S obtained by setting $a(i) = j$ for each edge $(v'_i, v''_j) \in \mathcal{M}$. To conclude the proof it is sufficient to show that G has always a matching $\mathcal{M}$ with cardinality of $|V_1|$. According to a well known result of König such a matching exists if and only if for each $H \subseteq V_1$ and $N(H) = \{v''_j : (v'_i, v''_j) \in L, v'_i \in H\}$, then $|H| \leq |N(H)|$. Consider the set $U = \cup_{i : v'_i \in H}\, \theta_B(i)$ and recall that: (a) $\theta_B(i) \cap B = \gamma(i)$, for $i \in E \backslash B$, $\theta_B(i) = \{i\}$ for $i \in B$ (by definition); (b) $\sigma(U) \backslash U \subseteq E \backslash B$ (see Property 2.4); and (c) $\sigma(U)$ is saturated (see Property 2.3).

Observing that $N(H) = \{v''_j : j \in \gamma(i), v'_i \in H, i \notin B\} \cup \{v''_j : v'_j \in H, j \in B\}$ and using (a) above we obtain $|N(H)| = |B \cap U|$. Moreover we have $|B \cap U| = |B \cap \sigma(U)| = r(U)$ (the first equality descends from (b), whereas the second one descends from (c)): thus $r(U) = |N(H)|$. Using the hypothesis that S is an independent set we know that $|S \cap U| \leq r(U)$ from which $|S(H) \cap U| \leq r(U)$, where $S(H) = \{i \in E : v'_i \in H\}$. But $S(H) \subseteq U$ (since $i \in \theta_B(i)$ $\forall i \in S$), hence $|S(H)| \leq r(U)$ and $|H| \leq |N(H)|$ holds. □

The proof of the above theorem suggests a simple method to check if a subset of E obtained by adding an element e to an independent set S is still independent. Given $S \subseteq E$ let G_S be the bipartite graph associated with S, defined as in the proof of the above Theorem 3.1. Since S is independent we know that G_S has a matching $\mathcal{M}$ with $|\mathcal{M}| = |V_1| = |S|$. To prove that $S \cup \{e\}$ is independent we have to show that $G_{S\cup\{e\}}$ has a matching of cardinality $|S| + 1$. Graph $G_{S\cup\{e\}}$ is obtained from G_S by adding to the vertex set V_1 the single vertex, say v'_e, associated with element e, and adding the corresponding edges to L. The matching $\mathcal{M}$ of G_S can be transformed into a matching $\mathcal{M}'$ with $|\mathcal{M}'| = |\mathcal{M}| + 1$ if and only if there is an augmenting path of $G_{S\cup\{e\}}$ emanating from v'_e.

The above observation leads to an improved implementation of the greedy algorithm for M_B: we sort the elements of E by non-increasing weights and start with an empty solution $S = \emptyset$. We iteratively try to add an element at a time to the current partial solution S. At each iteration we consider the bipartite graph $G_{S\cup\{e\}}$ obtained from G_S by adding the vertex v'_e and the edges corresponding to the current element e. Then we look for an augmenting path emanating from the exposed vertex v'_e. If the augmenting path exists we add e to the current partial solution S and we update the current partial matching $\mathcal{M}$ and the corresponding mapping. If otherwise the augmenting path does not exist we disregard e, removing vertex v'_e and the edges incident to it in $G_{S\cup\{e\}}$.

Note that the construction of all the graphs used by the above algorithm can be done in $O(m\varphi)$ (since for each element we have to determine its fundamental circuit), whereas the search for an augmenting path requires at most $O(n^2)$ computing time (since the two vertex sets have at most n vertices each). It follows that this implementation of the greedy algorithm runs in $O(mn^2 + m\varphi)$ time.

3.1 An efficient implementation

The complexity of the greedy algorithm described above can be improved as follows. First observe that, at each iteration, the search for an augmenting path may succeed or not. Since at most n successful augmentations are performed, the global number of operations due to such augmentations is $O(n^3)$. Now consider an iteration in which the augmenting path does not exist. We will show that either the computation of the possible alternating tree requires $O(n)$ time, or we can reduce the number of vertices of V_2 . It follows that all the unsuccessful iterations require $O(mn + n^3)$ time thus improving our previous bound and yielding an overall $O(mn + n^3 + m\varphi)$ algorithm.

The following property holds.

Property 3.1 *At any unsuccessful iteration of the algorithm consider the associated independent set S, the corresponding matching $\mathcal{M}$ (with $|\mathcal{M}| = |S|$), and the element $e \notin S$ such that there is no augmenting path in $G_{S\cup\{e\}}$ emanating from v'_e. Let $V_2(e) = \{v''_j \in V_2 : j \in \gamma(e)\}$ and $R(e) \subseteq V_2$ be the set of vertices reachable from $V_2(e)$ by means of alternating paths. Then all edges of $\mathcal{M}$ with a vertex in $R(e)$ do not belong to any augmenting path emanating from a vertex associated with an element in $E \backslash S$, in any subsequent iteration of the algorithm.*

Proof. If no augmenting path emanating from v'_e exists, then all the vertices of $V_2(e)$ are matched in $\mathcal{M}$ and there is no even alternating path starting from a vertex of $V_2(e)$ and ending with an exposed vertex of V_2 (note that the same property holds for paths starting from a vertex of $R(e)$). It immediately follows that no augmenting path starting from another vertex v'_f, corresponding to an element $f \in E \backslash S \cup \{e\}$, can use a vertex of $R(e)$, otherwise an augmenting path would exist also for v'_e. □

¿From the above Property 3.1 we have that when the current element e cannot be added to the partial solution S, then all vertices in $R(e)$ can be removed from the graph. Let us consider an algorithm which, at any unsuccessful iteration, does not add vertex e to the graph and deletes the vertices of $R(e)$. The computational effort of these unsuccessful iterations depends on the existence of elements in $V_2(e)$. If this set is empty the number of operations performed is $O(|\gamma(e)|) \leq O(n)$ and this kind of iterations may occur at most $O(m)$ times thus yielding an $O(mn)$ running time. If otherwise $V_2(e) \neq \emptyset$ we perform a search of an augmenting path and a deletion of vertices. In this case each iteration requires $O(n^2)$ operations, but it occurs at most n times since each iteration removes at least one vertex of V_2. Therefore the global computational effort for all the unsuccessful iterations is $O(mn + n^3 + m\varphi)$, which determines the global complexity of the algorithm.

Let us temporarily return to the previous version of the algorithm which does not reduce the vertex set. The relations among the sets $R(e)$ defined above and the saturated closed sets are exploited in the following property which will be used in the next Section 3.2.

Property 3.2 *At any unsuccessful iteration let $R(e)$ be defined as in Property 3.1. The set $B'' = B''(e) = \{j \in E : v''_j \in V_2 \cap R(e)\}$ is the skeleton of a closed set $\theta = \sigma(B'')$, which is saturated by the target base B, for matroid M, and by the base B_G obtained with the greedy algorithm, for matroid M_B.*

Proof. Consider the current matching $\mathcal{M}$ and let us define $B' = B'(e) = \{i \in E : (v'_i, v''_j) \in \mathcal{M}, v''_j \in R(e)\}$ and note that it is independent for M_B by construction, since it is a subset of the current solution obtained by the greedy. Observe that, by definition, $|B'| = |B''|$ and B'' is the skeleton of the set θ, which is saturated for M.

We now prove that θ is saturated also for M_B. Let r_B denote the rank function of the base-matroid M_B. We have just shown that $B' \subset \theta$, hence $r_B(B') \leq r_B(\theta)$, but $\theta \in \Theta_B$, so from Definition 2.3 we have $r_B(\theta) \leq r(\theta)$. Now observe that the skeleton of a closed set saturated for M is independent for M_B, hence $r_B(\theta) = r(\theta)$. Further note that due to the independence of B' in M_B it is $r_B(B') = |B'|$, so we obtain

$$|B'| = r_B(B') \leq r_B(\theta) = r(\theta) = |B''|$$

Recalling $|B'| = |B''|$ we conclude that $|B'| = r_B(\theta)$ and θ is also saturated by B_G, for matroid M_B, B' being the skeleton of θ for M_B. □

3.2 Linear Programming formulation

In this section we propose a linear programming model for optimizing a linear function on a base-matroid. Considering the greedy algorithm for the base-matroid presented in the previous section, the solution of the primal problem can be easily obtained. An efficient method to obtain the dual solution is less trivial and will be the main concern of this section.

Given a matroid $M = (E, \mathcal{F})$ with rank function r and weighting function c, it is well known that the corresponding optimization problem is equivalent to the following continuous linear programming problem (see e.g. [11]).

$$(P) \quad \max\{cx : \sum_{e\in\theta} x_e \leq r(\theta) \ \forall\ \theta \in \Theta,\ x \in \mathbb{R}_+^m\}$$

The polytope vertices of problem P belong to $\{0,1\}^n$, hence each variable x_e takes value 1 if the element e is selected, and value zero otherwise. The dual of P is:

$$(D) \quad \min\{ry : \sum_{\theta: e\in\theta} y_\theta \geq c_e \ \forall e \in E, y \in \mathbb{R}_+^{|\Theta|}\}$$

and the complementary slackness conditions of pair P-D are:

$$\begin{aligned} (\sum_{e\in\theta} x_e - r(\theta))y_\theta &= 0, \quad \theta \in \Theta && (6)\\ (\sum_{\theta:e\in\theta} y_\theta - c_e)x_e &= 0, \quad e \in E && (7) \end{aligned}$$

The base-matroid optimization problem associated with the target base B is

$$(BMP) \ \max\{cx' : x' \in PB \cap \{0,1\}^m\}$$

where

$$PB = \{x' \in \mathbb{R}_+^m : \sum_{e\in\theta} x'_e \leq r(\theta) \ \forall\ \theta \in \Theta_B\}$$

The continuous relaxation of this problem is

$$(CBMP) \ \max\{cx' : x' \in PB,\ x'_e \geq 0 \text{ for } e \in B,\ 0 \leq x'_e \leq 1 \text{ for } e \in E\backslash B\}$$

Note that the unit upper bounds must be explicitly given only for variables associated with elements in $E\backslash B$, indeed for each $e \in B$ the singleton $\{e\}$ belongs to Θ_B and the corresponding rank constraint (in PB) reads $x'_e \leq 1$.

In the following we will prove that similarly to the case of the classical matroid problem, PB is defined on an integral polytope, hence $CBMP$ is a valid formulation for the base-matroid optimization problem. Consider the dual of $CBMP$

$$\begin{aligned} (DCBMP) \quad \min\ & ry' + \mathbf{1}\mu && (8)\\ \sum_{\theta\in\Theta_B:e\in\theta} y'_\theta &\geq c_e,\ e \in B && (9)\\ \sum_{\theta\in\Theta_B:e\in\theta} y'_\theta + \mu_e &\geq c_e,\ e \in E\backslash B && (10)\\ y' &\in \mathbb{R}_+^{|\Theta_B|} && (11)\\ \mu &\in \mathbb{R}_+^m && (12) \end{aligned}$$

and the complementary slackness conditions of $CBMP$-$DCBMP$

$$\begin{aligned} x'_e(\sum_{\theta\in\Theta_B:e\in\theta} y'_\theta - c_e) &= 0, \quad e \in B && (13)\\ x'_e(\sum_{\theta\in\Theta_B:e\in\theta} y'_\theta + \mu_e - c_e) &= 0, \quad e \in E\backslash B && (14)\\ y'_\theta(\sum_{e\in\theta} x'_e - r(\theta)) &= 0, \quad \theta \in \Theta_B && (15)\\ \mu_e(x'_e - 1) &= 0, \quad e \in E\backslash B && (16) \end{aligned}$$

The optimal solution to $DCBMP$ can be obtained with a procedure similar to that used to compute the optimal solution of problem D (the dual of the generic matroid problem, see [11]), but giving zero value to each y'_θ with $\theta \notin \Theta_B$, and assigning suitable values to the μ variables.

Theorem 3.2 *Let $B_G = \{e_1, \ldots, e_n\}$ be the solution to problem BMP obtained through the greedy algorithm with the ordering $c_{e_1} \geq c_{e_2} \geq \cdots \geq c_{e_n}$, let $S_h = \{e_1, \ldots, e_h\}$ for $h = 1, \ldots, n$, and $\theta_h = argmax\{|\theta| : \theta \in \Theta_B, \theta \subseteq \sigma_B(S_h), e_h \in \theta, \theta \text{ is saturated for } M_B\}$ ($\theta_h = \emptyset$ if no such θ exists), where σ_B denotes the closure operator for matroid M_B. Moreover let $\Gamma = \{\theta_h \neq \emptyset, h = 1, \ldots, n\}$ and $\pi(h) = \min\{k > h : e_h \in \theta_k\}$ for $h = 0, \ldots, n-1$. An optimal solution to $DCBMP$ can be computed through the following procedure:*

```
step 1: set  $y' = 0$;  $\mu = 0$;
step 2: set  $y'_{\theta_n} = c_{e_n}$;
        for  $h = n-1, \ldots, 1$,  if  $\theta_h \in \Gamma$  then set  $y'_{\theta_h} = c_{e_h} - c_{e_{\pi(h)}}$,  otherwise set $\mu_{e_h} = c_{e_h} - c_{e_{\pi(h)}}$
```

Proof. The dual values (y', μ) computed through steps 1-2 are clearly non-negative. To prove the theorem we show that given the primal solution with $x'_e = 1$ iff $e \in B_G$, which is feasible for BMP and $CBMP$, the optimality conditions hold for the pair $(x', (y', \mu))$.

Consider the generic element $e_h \in B_G$ and let us define θ^* as the set $\theta_i \in \Gamma$ with the largest index $i < h$, if any, $\theta^* = \emptyset$, otherwise. One can see that $\theta^* \subseteq \theta_h$, hence we show how to obtain θ_h starting from θ^*. Two cases may occur:

(a) if $e_h \in B_G \cap B$, then θ_h is certainly non-empty, since at least the closed set $\{e_h\} \subseteq \sigma_B(S_h)$ is saturated for M and M_B. If e_h does not induce other elements to enter into $\sigma_B(S_h)$, then $\theta_h = \theta^* \cup \{e_h\}$. If otherwise e_h induces elements to enter $\sigma_B(S_h)$, then there exists $\theta \in \Theta_B$, with $e_h \in \theta$ and $|\theta \cap S_h| = r(\theta)$ (i.e. θ is saturated for M and M_B). In this case the set $\theta^* \cup \theta$ is the largest set saturated for both matroids (see Property 2.3) and $\theta_h = \theta^* \cup \theta$.

(b) If $e_h \in B_G \backslash B$, then the addition of e_h to S_{h-1} may induce or not elements different from e_h to enter $\sigma_B(S_h)$. In the first case the same reasoning of case (a) applies and $\theta_h = \theta^* \cup \theta$, in the second case $\theta_h = \emptyset$.

¿From the above reasoning one can see that the collections of sets $\{\sigma_B(S_i)\}$ and Γ are laminar. Further observe that: (i) $\theta_n = E$ ($\in \Gamma$); (ii) θ_h is certainly non-empty if $e_h \in B_G \cap B$, hence μ_{e_h} may have a positive value only if $e_h \in B_G \backslash B$.

¿From (i) above and the fact that Γ is laminar it immediately follows that given any $e \in E$ there is at least a set of Γ containing it. Let k be the smallest index such that $e \in \theta_k$, then

$$\sum_{\theta \in \Theta_B : e \in \theta} y'_\theta = \sum_{i=k}^{n} y'_{\theta_i} = c_{e_k} \tag{17}$$

If $e \notin B_G$ then $c_e \leq c_{e_k}$, otherwise the greedy algorithm would have selected e instead of e_k. Using (17) and the assignment of values to y' and μ one can see that that (9) and (10) are satisfied with the '$\geq$' sign for all $e \in E \backslash B_G$. For each $e_h \in B_G$, if $e_h \in B$, then $e_h \in \theta_h$ (see (ii) above) and $k = h$, so (9) is satisfied with the '$=$' sign. If otherwise $e_h \notin B$, then two cases may occur: (a) $\theta_h \neq \emptyset$, then $e_h \in \theta_h$ and $k = h$; (b) $\theta_h = \emptyset$, so $k = \pi(h)$ and $\mu_{e_h} = c_{e_h} - c_{e_k}$. In both cases (10) is satisfied with the '$=$' sign.

The above reasoning also proves that the terms in parenthesis in (13) and (14) have value zero when $e \in B_G$. On the other side $x'_e = 0$ for each $e \notin B_G$, hence (13) and (14) hold. The variable y'_θ may be assigned a positive value only when $\theta \in \Gamma$, i.e. θ is saturated by B and B_G. It follows that $y'_\theta > 0$ only if $\sum_{e \in \theta} x'_e = r(\theta)$, hence (15) hold. Finally μ_e is assigned a positive value only if

$e \in B_G \backslash B$ (see again (ii) above) and also the last conditions (16) hold. □

The above theorem proves that the system $\sum_{e \in \theta} x'_e \leq r(\theta)$ for $\theta \in \Theta_B, x' \in \mathbb{R}^m_+$ is totally dual integral, hence PB is an integral polytope and:

Theorem 3.3 *CBMP is a valid formulation for the base-matroid optimization problem.*

Let us discuss the computational complexity of finding the saturated sets $\theta_h \in \Gamma$ defined in Theorem 3.2 and necessary to compute the values of the non-zero dual variables. We propose an implementation in which each θ_h is not completely defined in a single step, rather it is constructed during the execution of the greedy algorithm by adding one element at a time to an initially empty set.

We use a version of the greedy algorithm which terminates only when all the m elements of E have been considered. We start by defining the sets $\eta_1 = \eta_2 = \ldots = \eta_n = \emptyset$. For each element e we determine the smallest index h such that $e \in \theta_h$ (see below for details), and we add e to η_h. At the end of the algorithm we compute $\theta_h = \cup_{i=1}^{h} \eta_i$, for each $h = 1, \ldots, n$ such that $\eta_h \neq \emptyset$. The correctness of this procedure immediately descends from the laminarity of Γ and from the fact that each element has been considered and added to the η set corresponding to the smallest set of Γ that contains it.

The key aspect of the procedure is the computation of the correct index h. For sake of simplicity we first introduce a method based on a (not efficient) implementation of the greedy which adds to the graph also the vertices corresponding to elements not inserted in B_G and does not delete the $R(e)$ sets form the bipartite graph used to perform the tests of independence (see Property 3.2). Then we show how to improve this unefficient greedy with an implementation based on a labeling technique.

Consider a generic iteration of the greedy and let e be the element currently examined. Three cases may occur:

1. $e \in B \cap B_G$: let $e_l = e$ (i.e. e is the l-th element added to the current partial solution) then the required index is $h = l$ (see the proof of Theorem 3.2).

2. $e \notin B_G$: recall that we have computed $R(e)$ without finding any augmenting path. From Property 3.2 we know that set $\theta = \sigma(B''(e))$ is saturated by base B for M and by the current partial base for M_B. Let h be the smallest index such that $e \in \theta_h, \theta_h \in \Gamma$. Observe that: (i) set $\theta \subseteq \sigma_B(S_h)$, since it is saturated by $\{e_1, \ldots, e_h\}$; and (ii) $\theta \subseteq \theta_h$, otherwise $\theta \cup \theta_h$ is saturated for both matroids and has larger cardinality than θ_h: a contradiction. Further observe that due to the fact that $e \in \sigma_B(S_h)$ and $e \notin \sigma_B(S_i)$, for $i < h$, then e can enter into a set of Γ only together with e_h. It follows that $e_h \in \theta$, so during the execution of the algorithm we can compute the value of index h by scanning set $B'(e)$ $(= \theta \cap \{e_1, \ldots, e_h\})$ and identifying the last element of B_G inserted into it.

3. $e \in B_G \backslash B$: in this case we are not guaranteed to identify efficiently the required index for all elements, so we postpone the insertion of e in the suitable η set at the end of the greedy. More precisely when all the m elements have been examined we consider, in turn, each element $e \in B_G \backslash B$. We temporary create a copy, say $\tilde{e}$, of e and we compute $R(\tilde{e})$. The required index is found as in case 2, by considering set $\theta = \sigma(B''(\tilde{e}))$.

We now show how to implement the above procedure without computing explicitly all the sets $\sigma(B''(e))$. We know that if a vertex is reached by an alternating path, at an unsuccessful iteration, then in the next iterations it can not belong to any augmenting path (see Proposition 3.2). In

Section 3.1 we have already shown that deleting these vertices we can reduce the computational complexity of the greedy, however for computing the dual values we should entirely scan each alternating tree to compute the index of the η set in which the current element has to be inserted (cases 2 and 3 above). Instead of re-scanning a tree we can maintain a trace of the previously examined trees by using the following simple labeling technique. When an unsuccessful iteration occurs we associate at each vertex $v''_j \in V_2(e)$ a label storing the index of the last element inserted in B_G and related with one of the vertices of the subtree rooted at v''_j. In the next iterations if we reach vertex v''_j, we can stop the search for this branch of the alternating tree since the whole information needed to compute the dual value is stored in the label. The smallest index h such that $e \in \theta_h$ is identified by considering all the elements associated to vertices explicitly reached and the labels of the leaves. Using this trick we can return to the original implementation which, at each unsuccessful iteration, does not add the vertex associated with the current element $e \notin B_G$ to the graph, and deletes all the vertices in the $R(e)$ set. The additional computational effort required to compute the dual values is $O(mn)$ for identifying the correct η sets during the execution of the greedy, plus $O(n^3)$ for the computations due to the elements in $B_G \backslash B$, so the following theorem holds.

Property 3.3 *The dual values defined by Theorem 3.2 can be determined during the execution of the greedy algorithm, without increasing its computational complexity.*

Example

Let us consider the graphic matroid depicted in Figure 2, and let us be given the target base $B = \{a, b, c, d, e, j, l, m, n\}$ (thick edges). The weights associated with the edges are reported in the following table, sorted by nonincreasing value (breaking ties by the lexicographic order of the names):

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
element	**n**	g	q	f	p	**b**	**a**	r	**d**	i	**c**	**j**	**l**	h	**m**	**e**
cost	10	9	8	7	7	6	5	5	4	4	3	3	3	2	2	1

By applying the greedy algorithm of Section 3 we can augment the solution until we examine edge r. At that point the partial solution is $\{n, g, q, f, p, b, a\}$, the currently defined η sets are $\eta_1 = \{n\}$, $\eta_2 = \eta_3 = \eta_4 = \eta_5 = \emptyset$, $\eta_6 = \{b\}$ and $\eta_7 = \{a\}$. The nonempty η sets correspond to elements in $B \cap B_G$ (case 1). The elements $\{g, q, f, p\}$ will be inserted in the suitable η set at the end of the greedy (case 3). The bipartite graph used to determine the independence of element r is reported in Figure 3.a (the thick edges give the current base-mapping). The alternating tree starting from vertex r' is given in Figure 4.a (in square brackets we report the label associated at each vertex, whereas in parenthesis we report the labels that will be associated at each vertex after the computation of the tree). No augmenting path exists and set $\{a, b, r\}$ is dependent for the base-matroid. Examining set $B'(r) = \{a, b\}$ we find that element a is the last one added, so $\eta_h = \eta_7 = \{a, r\}$. In the next iteration we add d to B_G, we set $\eta_8 = \{d\}$ and we update the base-mapping (see Figure 3.b). Then we examine element i, we find the alternating tree of Figure 4.b and we add i to η_8. Then we add c, j, l to η_7, η_5 and η_5, respectively. Element h enter in the solution with matching $[h', m'']$, element m is added to η_9 and element e is added to η_8. We have thus obtained the η sets: $\eta_1 = \{n\}$, $\eta_2 = \eta_3 = \eta_4 = \emptyset$, $\eta_5 = \{j, l\}$, $\eta_6 = \{b\}$, $\eta_7 = \{a, c, r\}$, $\eta_8 = \{d, e, i\}$, $\eta_9 = \{m\}$. The base mapping is reported in Figure 3.b. The optimal solution of the base-matroid is thus: $\{g, q, f, n, p, b, a, d, h\}$; it should be observed that the solution is not feasible for the graphic matroid as it contains a cycle ($\{q, n, p\}$).

We now consider the elements in $B_G \backslash B$. We first duplicate g' obtaining $\tilde{g}'$ and we compute the corresponding alternating tree, see Figure 4.c: element g is added to η_8. The next elements $\{q, f, p, h\}$ are added to η_5, η_7, η_5 and η_9 respectively. The final η and θ sets are:

$$
\begin{array}{lllllll}
\eta_1 & = & \{n\} & \qquad & \theta_1 & = & \{n\} \\
\eta_5 & = & \{j, l, p, q\} & & \theta_5 & = & \{j, l, n, p, q\} \\
\eta_6 & = & \{b\} & & \theta_6 & = & \{b, j, l, n, p, q\} \\
\eta_7 & = & \{a, c, f, r\} & & \theta_7 & = & \{a, b, c, f, j, l, n, p, q, r\} \\
\eta_8 & = & \{d, e, g, i\} & & \theta_8 & = & \{a, b, c, d, e, f, g, i, j, l, n, p, q, r\} \\
\eta_9 & = & \{h, m\} & & \theta_9 & = & E
\end{array}
$$

The primal and dual solution are summarized in the following table.

	1	2	3	4	5	6	7	8	9
element	n	g	q	f	p	b	a	d	h
cost	10	9	8	7	7	6	5	4	2
variable	y_{θ_1}	μ_g	μ_q	μ_f	y_{θ_5}	y_{θ_6}	y_{θ_7}	y_{θ_8}	y_{θ_9}
value	3	5	1	2	1	1	1	2	2

4 Inverse matroid problem

The *inverse matroid problem* can be stated as follows. Given a matroid $M = (E, \mathcal{F})$, and a *target base* B of M (not necessarily optimal) find the *perturbation parameters* δ_e to be added to the weighting coefficients c_e, for each $e \in E$, such that B is optimal for the matroid problem defined by the new weights $w_e = c_e + \delta_e$, and a function of the values δ_e is minimized. In this paper we focus on the objective function given by the sum of the absolute values of δ_e, that is $\sum_{e \in E} |\delta_e|$.

Since B must be optimal for the weightening w, one can prove (see, e.g. [7]) that in the optimal solution of the inverse matroid problem $w_e \geq c_e$ for each $e \in B$ and $w_e \leq c_e$ for each $e \in E \backslash B$. Therefore, the inverse matroid problem is equivalent to finding the vector d which minimizes $\sum_{e \in E} d_e$ and such that B is an optimal base for the matroid problem with weights

$$
w_e = \begin{cases} c_e + d_e & e \in B \\ c_e - d_e & e \in E \backslash B \end{cases} \tag{18}
$$

The target base B is optimal with respect to the new weights (18) if there exists a dual feasible vector y which satisfies the complementary slackness conditions (6)-(7), written with w instead of c. Reminding that the only saturated closed sets, with respect to B, are those of Θ_B, then (6) implies $y_\theta = 0$ for $\theta \notin \Theta_B$ and the inverse matroid problem can be formulated as follows.

$$
(PI) \quad \min \sum_{e \in E} d_e \tag{19}
$$

$$
\sum_{\theta \in \Theta_B : e \in \theta} y_\theta \;=\; c_e + d_e \quad e \in B \tag{20}
$$

$$
\sum_{\theta \in \Theta_B : e \in \theta} y_\theta \;\geq\; c_e - d_e \quad e \in E \backslash B \tag{21}
$$

$$
y \;\in\; \mathbb{R}_+^{|\Theta_B|} \tag{22}
$$

$$
d \;\in\; \mathbb{R}_+^m \tag{23}
$$

Constraint (20) derive from the optimality conditions (7), whereas (21) impose the feasibility of the dual solution y. In order to construct the optimal solution of problem PI let us consider the dual:

$$
\begin{aligned}
(DI) \quad \max \sum_{e\in E} c_e x_e & & & & (24)\\
\sum_{e\in\theta} x_e &\leq 0 & & \theta\in\Theta_B & (25)\\
x_e &\geq -1 & & e\in B & (26)\\
0\leq x_e &\leq 1 & & e\in E\backslash B & (27)
\end{aligned}
$$

Using the transformation

$$
x'_e = \begin{cases} 1+x_e & e\in B\\ x_e & e\in E\backslash B \end{cases} \tag{28}
$$

problem DI can be rewritten as

$$
\begin{aligned}
(DI') \quad -\sum_{e\in B} c_e + \max \sum_{e\in E} c_e x'_e & & & & (29)\\
\sum_{e\in\theta} x'_e &\leq r(\theta) & & \theta\in\Theta_B & (30)\\
x'_e &\geq 0 & & e\in B & (31)\\
0\leq x'_e &\leq 1 & & e\in E\backslash B & (32)
\end{aligned}
$$

Problem DI' is a base-matroid optimization problem (see Section 3.2, problem $CBMP$) hence it can be efficiently solved by means of the greedy algorithm described in the previous section. From this solution we immediately obtain the solution of problem DI and in the next section we show how to construct the solution of the inverse problem PI.

4.1 Determining the optimal perturbations

Let us introduce the complementary slackness condition of PI-DI.

$$
\begin{aligned}
x_e(\sum_{\theta:\theta\in\Theta_B, e\in\theta} y_\theta + d_e - c_e) &= 0 \quad e\in E\backslash B & (33)\\
y_\theta(\sum_{e\in\theta} x_e) &= 0 \quad \theta\in\Theta_B & (34)\\
(x_e+1)d_e &= 0 \quad e\in B & (35)\\
(x_e-1)d_e &= 0 \quad e\in E\backslash B & (36)
\end{aligned}
$$

Given the optimal solution x', obtained through the greedy algorithm of Section 3, the optimal solution of the inverse problem is obtained with the following procedure.

Procedure $InverseMatroid()$

step i. Determine the optimal solution (y,μ) of the dual of problem DI' (see Theorem 3.2)

step ii. Determine the values of x_e through the inverse of (28), that is:

$$
x_e = \begin{cases} x'_e - 1 & e\in B\\ x'_e & e\in E\backslash B \end{cases}
$$

and note that this solution is optimal for problem DI.

step iii. For each $e \in B$ define the value of d_e as follows: if $x'_e = 1$ set $d_e = 0$, otherwise ($x'_e = 0$) set $d_e = \sum_{\theta \in \Theta_B : e \in \theta} y_\theta - c_e$.

step iv. For each $e \in E \backslash B$ define the value of d_e as follows: if $x'_e = 1$ set $d_e = c_e - \sum_{\theta \in \Theta_B : e \in \theta} y_\theta$, otherwise ($x'_e = 0$) set $d_e = 0$.

Theorem 4.1 *The solution d, y determined through the above procedure $InverseMatroid$ is optimal for PI.*

Proof. We have already observed that x defined at step **ii** is a feasible solution for DI. To prove the thesis we show that d, y is a feasible solution for PI, and that x, d, y, satisfy the complementary slackness conditions (33)-(36).

First note that y has nonnegative values (see Theorem 3.2), then consider separately the case $e \in B$ and $e \in E \backslash B$.

Case $e \in B$ (step iii). From (9) we know that $d_e = \sum_{\theta \in \Theta_B : e \in \theta} y_\theta - c_e \geq 0$ satisfying (20). If we set $d_e = 0$, we have $x'_e = 1$ and from (13) condition (20) follows again.

Case $e \in E \backslash B$ (step iv). From (14) we know that when $x'_e = 1$ we have $c_e - \sum_{\theta : e \in \theta} y_\theta \geq 0$ hence d_e is assigned a non-negative value and both (21) and (33) hold. When $x'_e = 0$ (33) trivially holds, whereas form (16) we have $\mu_e = 0$ and (10) implies that (21) holds.

We conclude the proof by observing that the remaining condition (34) directly descends from (15) by applying transformation (28), and that (35) and (36) hold by construction of d. □

Example (continued)

The original base of the graphic matroid of Figure 2 has value 37 and applying procedure $InverseMatroid$ we obtain a dual solution having value 58. Then at step iii we set to zero the perturbation associated with the elements in $B \cap B_G$ and we compute the following values for the elements in $B \backslash B_G$: $d_c = 2$, $d_e = 3$, $d_j = 4$, $d_l = 4$, $d_m = 0$. At step iv the computation of the perturbation of the elements in $B_G \backslash B$ gives $d_f = 2$, $d_g = 5$, $d_h = 0$, $d_p = 0$, $d_q = 1$, whereas the remaining values are set to zero. The optimal solution of the inverse problem has value 21.

4.2 The Inverse Spanning Tree Problem

Finding a spanning tree of maximum (minimum) cost is one of the basic problems in combinatorial optimization, whose inverse version has been solved by Ahuja et. al [3]. It is well known that a spanning tree of a graph is the base of a graphic matroid, therefore the inverse spanning tree problem (ISTP) can be immediately modeled by means of a base-matroid and procedure $InverseMatroid$ of Section 4.1 is an alternative approach for solving ISTP. In this section we briefly compare the two approaches.

Given a graph with n vertices and m edges the basic algorithm of Ahuja et. al [3], solves ISTP in $O(n^3)$ time. Using a cost scaling algorithm ISTP can be solved in $O(n^2 m \log(nC))$ time, where C denotes the largest cost in the data. The key step of the algorithm is the solution of an assignment problem with a special structure.

Our method for the solution of the generic inverse matroid problem starts by solving problem DI' of Section 4 with the greedy algorithm of Section 3, which runs in $O(mn + n^3 + m\varphi)$ time. For a graphic matroid the value φ of the computational complexity of a procedure which determines a fundamental circuit, is bounded by n and m is bounded by n^2, therefore our greedy algorithm takes $O(n^3)$. During the execution of the greedy we compute the dual values with no additional cost (Property 3.3). Since the number of positive dual values is bounded by $O(n)$, steps iii and iv can certainly be performed in $O(mn)$, hence our algorithm runs in $O(n^3)$, as the basic algorithm of Ahuja et. al [3].

5 Conclusions

In this paper we presented the base-matroid defined starting from a matroid and one of its bases. After presenting some general properties, we show that the base-matroid is actually a matroid; we devise a non-trivial efficient greedy algorithm to compute the optimal base of the corresponding base-matroid optimization problem. One of the applications of the base-matroid is in the field of inverse matroid optimization. For this reason we discuss in detail the LP formulation of the base-matroid and we propose an efficient algorithm for computing the primal and dual solution.

Consider a base-matroid defined on the optimal base of the original matroid optimization problem. It can be observed that the optimal solution of the base-matroid optimization is exactly the given base. As a consequence the inverse optimization problem has optimal solution of value zero, that is no perturbation must be introduced.

It is interesting to note that, being a matroid, the definition of the base-matroid can be iterated. That is we can define the base-matroid of a base-matroid and so on. Provided that the bases used to define the sequence of base-matroids are different from the optimal base of the previous base-matroid, the process can be iterated n times. Each time the number of constraints defining the matroid decreases. At the last iteration the only constraint defining the matroid is $|S| \leq n$, that is the uniform matroid.

References

[1] Ahuja, R. and J. Orlin. Inverse Optimization, Part I: Linear Programming and General Problem. *Working Paper, Sloan School SWP 4002*, 1998.

[2] Ahuja R. and J. Orlin. Inverse Optimization, Part II: Network Flow Problems. *Working Paper, Sloan School SWP 4003*, 1998.

[3] Ahuja, R., J. Orlin and P. T. Sokkalingam. Solving Inverse Spanning Tree Problems Through Network Flow Techniques. *Operations Research* 47, 1999, 291-298.

[4] Burton D. and L. Toint. On an Instance of the Inverse Shortest Path problem. *Mathematical Programming* 53, 1992, pp 45-61.

[5] Cai, M. and Y. Li. Inverse Matroid Intersection Problem. *ZOR-Mathematical Methods of Operations Research* 45, 1997, pp 235-243.

[6] Cai, M., Yang X. and Y. Li. Inverse Polymatroid Flow Problem. *Journal of Combinatorial Theory* 3, 1999, pp 125-126.

[7] Frank, A. A Weightd Matroid Intersection Algorithm. *Journal of Algorithms* 2, 1981, pp 328-336.

[8] Garey M.R. and D.S. Johnson, Computer and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco, 1979.

[9] Hu, X. and Z. Liu. A Strongly Polynomial Algorithm for the Inverse Shortest Arborescence Problem. *Discrete Applied Mathematics* 82, 1998, pp 135-154.

[10] Moser, T. J., Shortest paths calculation of seismic rays, *Geophysics* 56, 1991, pp 59-67.

[11] Nemhauser G.L. and L.A. Wolsey, *Integer and Combinatorial Optimization* Wiley, New-York, 1988.

[12] Welsh D. J. A, *Matroid Theory* Academic Press, London 1976.

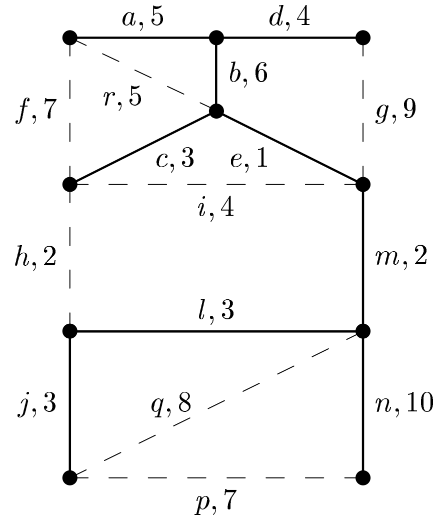

Figure 2: The graph and the target base (thick edges)

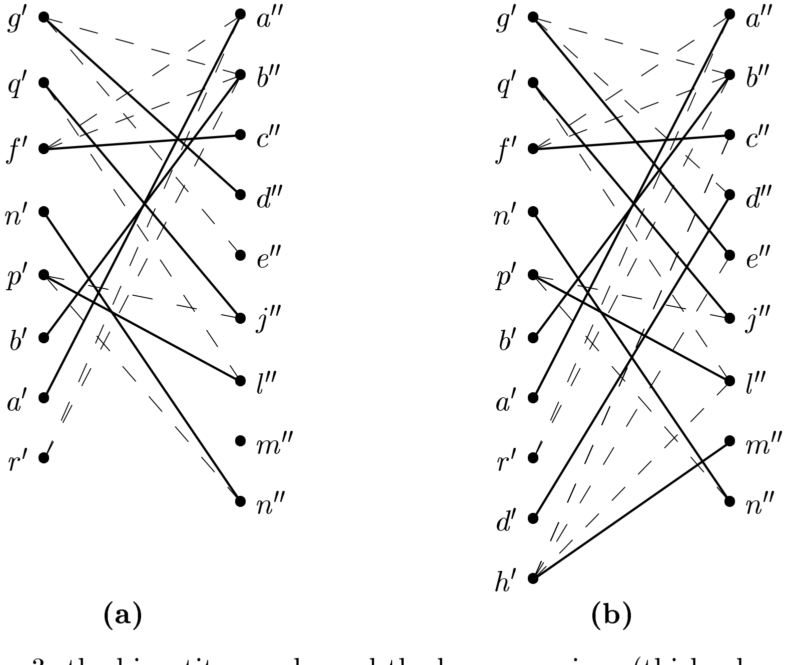

Figure 3: the bipartite graphs and the base meppings (thick edges)

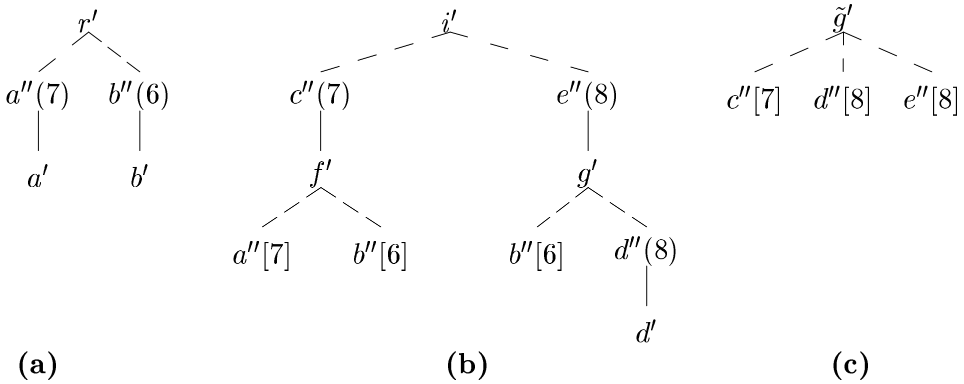

Figure 4: three alternating trees