Math. Program., Ser. A (2019) 177:85–112
https://doi.org/10.1007/s10107-018-1260-x

FULL LENGTH PAPER

Exact and approximation algorithms for weighted matroid intersection

Chien-Chung Huang[1] · **Naonori Kakimura**[2] · **Naoyuki Kamiyama**[3]

Received: 9 December 2016 / Accepted: 12 March 2018 / Published online: 20 March 2018

Abstract In this paper, we propose new exact and approximation algorithms for the weighted matroid intersection problem. Our exact algorithm is faster than previous algorithms when the largest weight is relatively small. Our approximation algorithm delivers a $(1-\epsilon)$-approximate solution with a running time significantly faster than most known exact algorithms. The core of our algorithms is a decomposition technique: we decompose an instance of the weighted matroid intersection problem into a set of instances of the unweighted matroid intersection problem. The computational advantage of this approach is that we can make use of fast unweighted matroid intersection algorithms as a black box for designing algorithms. More precisely, we show that we can solve the weighted matroid intersection problem via solving W instances of the unweighted matroid intersection problem, where W is the largest given weight, assuming that all given weights are integral. Furthermore, we can find a $(1-\epsilon)$-approximate solution via solving $O(\epsilon^{-1} \log r)$ instances of the unweighted matroid intersection problem, where r is the smaller rank of the two given matroids. Our algorithms make

A preliminary version of this paper appears in the proceeding of ACM-SIAM Symposium on Discrete Algorithms (SODA), 2016.

✉ Chien-Chung Huang
cchuang@di.ens.fr

Naonori Kakimura
kakimura@math.keio.ac.jp

Naoyuki Kamiyama
kamiyama@imi.kyushu-u.ac.jp

[1] CNRS, École Normale Supérieure, PSL, Paris, France

[2] Department of Mathematics, Keio University, Yokohama, Japan

[3] Institute of Mathematics for Industry, Kyushu University, Fukuoka, Japan

use of the weight-splitting approach of Frank (J Algorithms 2(4):328–336, 1981) and the geometric scaling scheme of Duan and Pettie (J ACM 61(1):1, 2014). Our algorithms are simple and flexible: they can be adapted to special cases of the weighted matroid intersection problem, using specialized unweighted matroid intersection algorithms. In addition, we give a further application of our decomposition technique: we solve efficiently the rank-maximal matroid intersection problem, a problem motivated by matching problems under preferences.

Keywords Weighted matroid intersection · Exact algorithms · Approximation algorithms

Mathematics Subject Classification 90C27 Combinatorial Optimization

1 Introduction

In the classical *weighted matroid intersection problem*, we are given two matroids $\mathbf{M}_1 = (S, \mathcal{I}_1)$, $\mathbf{M}_2 = (S, \mathcal{I}_2)$ and a weight function $w\colon S \to \mathbb{Z}_{\geq 0}$, where $\mathbb{Z}_{\geq 0}$ is the set of non-negative integers. Then, the goal is to find a maximum-weight common independent set I of $\mathbf{M}_1$ and $\mathbf{M}_2$, i.e., $I \in \mathcal{I}_1 \cap \mathcal{I}_2$ with $\sum_{e \in I} w(e)$ being maximized. This problem was introduced by Edmonds [13,15] and solved by Edmonds [13,15] and others [1,30,36,37] in 1970s. This problem is a common generalization of various combinatorial optimization problems such as bipartite matchings, packing spanning trees, and arborescences in a directed graph. In addition, it has many applications, e.g., in electric circuit theory [40,45], rigidity theory [45], and network coding [10]. The fact that two matroids capture the underlying common structures behind a large class of polynomially solvable problems has been impressive and motivated substantial follow-up research (see, e.g., [18,47]). Techniques and theorems developed surrounding this problem have become canon in contemporary combinatorial optimization literature.

Since 1970s, quite a few algorithms have been proposed for matroid intersection problems, e.g., [2,9,16,19,49], with better running time and/or simpler proofs. See Table 1 for a summary. Throughout the paper, n is the size of the ground set S, r is the smallest rank of the two given matroids, and W is the largest given weight. The oracle to check the independence of a given set has the running time of τ.

1.1 Our contribution

We propose both exact and approximation algorithms for the weighted matroid intersection problem. Our exact algorithm is faster than known algorithms when the largest given weight W is relatively small. Our approximation algorithm delivers a $(1 - \epsilon)$-approximate solution for every fixed $\epsilon > 0$ in times substantially faster than known exact algorithms in most cases. Our algorithms and their analysis are surprisingly simple. Moreover, these algorithms can be specialized for particular classes of matroids.

The core of our algorithms is a decomposition technique. We show that a given instance of the weighted matroid intersection problem can be decomposed into a set of unweighted versions of the same problem. To be precise, we can solve the

Table 1 Matroid intersection algorithms for general matroids

Algorithm	Weight	Time complexity
Aigner–Dowling [1]	Unweighted	$O(\tau nr^2)$
Cunningham [9] and Gabow–Xu [24]	Unweighted	$O(\tau nr^{1.5})$
Lawler [36,37] and Iri–Tomizawa [30]	Weighted	$O(\tau nr^2)$
Frank [16]	Weighted	$O(\tau n^2 r)$
Brezovec–Cornuéjols–Glover [2]	Weighted	$O(\tau nr^2)$
Fujishige–Zhang [19], Shigeno–Iwata [49] and Gabow–Xu [24]	Weighted	$O(\tau n^2 \sqrt{r} \log rW)$
Lee-Sidford-Wong [38]	Weighted	$O(\tau n^2 \log nW)$
Chekuri–Quanrud [3] ($(1-\epsilon)$-approximation)	Weighted	$O(\tau nr\epsilon^{-2} \log^2 \epsilon^{-1})$
This paper	Weighted	$O(\tau Wnr^{1.5})$
This paper ($(1-\epsilon)$-approximation)	Weighted	$O(\tau\epsilon^{-1} nr^{1.5} \log r)$

See also [13,15,42]. The complexity is measured only by the number of independence oracle calls. In case the original algorithms (Fujishige–Zhang and Gabow–Xu) use (co-)circuit oracles, each call of such oracles is replaced by n independence calls in the table

weighted problem exactly by solving W unweighted ones. Furthermore, we can solve the weighted problem $(1-\epsilon)$-approximately by solving $O(\epsilon^{-1} \log r)$ unweighted ones.

Our decomposition technique not only establishes a hitherto unclear connection between the weighted and unweighted problems, but also leads to computational advantages: the known unweighted matroid intersection algorithms are significantly faster than their weighted counterparts. Thus, we can make use of the former to design faster algorithms. It may be expected that in the future, there will be even more efficient unweighted matroid intersection algorithms, and that would imply our algorithms will become faster as well.

We summarize the complexity of our exact algorithms below. For comparison of our algorithms with previous results, see Tables 1, 2 and 3.

General matroids Given two general matroids, using the unweighted matroid intersection algorithm of Cunningham [9], we can solve the weighted matroid intersection problem in $O(\tau Wnr^{1.5})$ time. This algorithm is faster than all known algorithms when $W = o(\min\{\sqrt{r}, \frac{n \log r}{r}\})$ and $r = O(\sqrt{n})$. A slightly different analysis shows that the same algorithm has the complexity[1] of $O(\tau(\sum_{e\in S} w(e))r^{1.5})$.

Graphic matroids Given two graphic matroids, using the unweighted graphic matroid intersection algorithm proposed by Gabow and Xu [23], we can solve the weighted matroid intersection problem in $O(W\sqrt{r}n \log r)$ time. This is faster than the current fastest algorithm when $W = o(\log^2 r)$. If the graph is relatively dense, that is, $n = \Omega(r^{1.5} \log r)$, then we can use the algorithm of Gabow and Stallman [22] to solve the problem in $O(W\sqrt{r}n)$ time.

[1] This complexity is superior to the previous one only when the given weights are very "unbalanced."

Table 2 Matroid intersection algorithms for graphic matroids

Algorithm	Weight	Time complexity
Gabow–Stallman [22]	Unweighted	$O(\sqrt{r}n)$ if $n = \Omega(r^{3/2} \log r)$
	Unweighted	$O(rn^{2/3} \log^{1/3} r)$ if $n = \Omega(r \log r)$ $\&n = O(r^{3/2} \log r)$
	Unweighted	$O(r^{4/3} n^{1/3} \log^{2/3} r)$ if $n = O(r \log r)$
Gabow–Xu [23]	Unweighted	$O(\sqrt{r}n \log r)$
Gabow–Xu [23]	Weighted	$O(\sqrt{r}n \log^2 r \log(rW))$
This paper	Weighted	$O(W\sqrt{r}n \log r)$
$(1-\epsilon)$-approximation	Weighted	$O(\epsilon^{-1}\sqrt{r}n \log^2 r)$

Table 3 Linear matroid intersection algorithms

Algorithm	Weight	Time complexity
Cunningham [9]	Unweighted	$O(nr^2 \log r)$
Gabow–Xu [24]	Unweighted	$O(nr^{\frac{5-\omega}{4-\omega}} \log r)$
Harvey [27]	Unweighted	$O(nr^{\omega-1})$
Cheung et al. [4]	Unweighted	$O(nr \log r_* + nr_*^{\omega-1})$
Gabow–Xu [24]	Weighted	$O(nr^{\frac{7-\omega}{5-\omega}} \log^{\frac{\omega-1}{5-\omega}} r \log nW)$
Harvey [26]	Weighted	$\tilde{O}(W^{1+\epsilon} nr^{\omega-1})$
This paper	Weighted	$O(nr \log r_* + Wnr_*^{\omega-1})$
$(1-\epsilon)$-approximation	Weighted	$O(nr \log r_* + \epsilon^{-1} nr_*^{\omega-1} \log r_*)$

Here the $\tilde{O}$ notation hides a polynomial of $\log n$ in the complexity

Linear matroids Given two linear matroids (in the form of two r-by-n matrices), using the unweighted linear matroid intersection algorithm of [4], we can solve the weighted matroid intersection problem in $O(nr \log r_* + Wnr_*^{\omega-1})$ time, where ω is the exponent of the matrix multiplication time and $r_* \le r$ is the maximum size of a common independent set. This is faster than all known algorithms when $W = o(r^{\frac{\omega^2-7\omega+12}{5-\omega}})$ (if $\omega \approx 2.37$ [8,25,52], it is when $W = o(r^{0.41})$).

In the graphic matroid intersection problem, two graphs $G_1 = (V_1, E)$ and $G_2 = (V_2, E)$ with the same edge set E are given. An intersection of the two matroids means a subset of edges $E' \subseteq E$ so that E' induces a forest in both G_1 and G_2. In the linear matroid intersection problem, two r-by-n matrices M_1 and M_2 are given. An intersection of the two matroids means a subset of columns so that they are linearly independent in both M_1 and M_2. The graphic and linear matroid intersection problems arise in various branches in engineering. For example, the intersection of graphic matroids has applications in determining the order of complexity of an electrical network [29] and the unique solvability of open networks [44]; the intersection of linear matroids has applications in the analysis of systems of linear differential equations [40,41].

A recent trend in research is to design fast approximation algorithms for fundamental optimization problems, even if they are in **P**. Examples include maximum weight matching [11], shortest paths [51], and maximum flow [6,34,39,48]. Using the algorithms of [4,9,23], our decomposition technique delivers a $(1-\epsilon)$-approximate solution in

1. $O(\tau\epsilon^{-1}nr^{1.5}\log r)$ time with two general matroids,
2. $O(\epsilon^{-1}\sqrt{r}n\log^2 r)$ time with two graphic matroids,
3. $O(nr\log r_* + \epsilon^{-1}nr_*^{\omega-1}\log r_*)$ time with two linear matroids.

Our approximation algorithms are significantly faster than most exact algorithms. Prior to our results, there is only a simple greedy 1/2-approximation algorithm [32,35] dated in 1970s. It should be noted that, by scaling weights to small integers, i.e., rounding W to $O(\frac{r}{\epsilon})$ (cf. Lemma 5), exact algorithms deliver a $(1-\epsilon)$-approximate solution (this is used in [5] for the linear matroid parity). Ours improve on such simple scaling significantly. We note that for general matroids, very recently, Chekuri and Quanrud [3] improved on our results: they obtain a $(1-\epsilon)$-approximate solution in $O(\tau nr\epsilon^{-2}\log^2\epsilon^{-1})$ time.

1.2 Our technique

The idea of reducing a weighted optimization problem into unweighted ones has been successfully applied in the context of maximum-weight matching in bipartite graphs [33] and in general graphs [28,43]. Roughly speaking, these matching algorithms proceed iteratively: in each round, in a subgraph, a maximum-cardinality matching and its optimal dual are computed; the latter is then used to update the edge weights to construct the next subgraph. The optimality of the final solution is shown via the complementary slackness condition.

The difficulty of extending such approaches for matching to the weighted matroid intersection problem lies in the fact that, in the latter problem, the dual variables are harder to reason with and to control. Instead we make use of Frank's weight-splitting approach [16,17]. Frank [16,17] shows that the dual variables used in a primal-dual scheme can be replaced by a much simpler weight-splitting $w = w_1 + w_2$ of the element weights. The complementary slackness condition for optimality can also be replaced by weight-optimality in w_1 and w_2. Our main insight is that the split weights w_1 and w_2 can also be used to re-define two new matroids for subsequent operations. This is analogous to using the dual optimal solution to update the edge weights in the maximum-weight matching [28,33,43].

Our approximation algorithms use the above basic ideas and a scaling technique of [11] for approximating maximum-weight matching. In particular, as in [11], the amount of adjustments done to the weights w_1 and w_2 decreases geometrically in each phase.

1.3 Application: rank-maximal matroid intersection

We consider a variation of the weighted matroid intersection problem, called the *rank-maximal matroid intersection problem*. Suppose that instead of a weight function w, a rank function $\lambda\colon S \to \{1, 2, \ldots, R\}$ is given, where R is some positive integer. The goal is to find a common independent set so that it has the maximum number of elements e with rank 1, and subject to that, it has the maximum number of elements e with rank 2 and so on. The problem is a generalization of the rank-maximal matching problem, introduced by [31] in the context of matching problems with preference lists.

As done in [31], we can reduce this problem to the weighted matroid intersection problem by assigning huge weights, say $\Omega(n^{R-i})$, to elements of rank i. However, such an approach would be inefficient in time and space. We show how to modify our exact algorithm to decompose the problem into R unweighted matroid intersection problems. In particular, we solve the rank-maximal matroid intersection problem using $O(Rnr^{1.5})$ independence oracle calls. Moreover, if the given two matroids are graphic or linear, the running times are reduced to $O(R\sqrt{r}n \log r)$ and $O(Rnr^{\omega-1})$, respectively.

1.4 Outline

The rest of the paper is organized as follows. In Sect. 2, we give definitions and basic properties of matroids. Our exact and approximation algorithms are presented in Sects. 3 and 4, respectively. Implementation details about finding a maximum-cardinality common independent set are described in Sect. 5. The result of the rank-maximal matroid intersection problem is in Sect. 6. The relation of our results with previous work is discussed in Sect. 7.

2 Preliminaries

2.1 Matroids

A *matroid* is a pair $\mathbf{M} = (S, \mathcal{I})$ of a finite set S and a family $\mathcal{I}$ of subsets of S satisfying the following three conditions.

(I0) $\mathcal{I} \neq \emptyset$.
(I1) If $I \subseteq J$ and $J \in \mathcal{I}$, then $I \in \mathcal{I}$.
(I2) If $I, J \in \mathcal{I}$ and $|I| < |J|$, then there is $e \in J \setminus I$ such that $I + e \in \mathcal{I}$.[2]

A set in $\mathcal{I}$ is said to be *independent*, and a maximal independent set is called a *base*. In addition, a minimal non-independent subset C of S is called a *circuit*. A circuit of size one is a *loop*. Throughout the article, we assume that the given matroids have no loops.

Let $\mathbf{M} = (S, \mathcal{I})$ be a matroid and X a subset of S. The *restriction* of $\mathbf{M}$ to X is defined by $\mathbf{M}|X = (X, \mathcal{I}|X)$ with $\mathcal{I}|X = \{I \in \mathcal{I} \mid I \subseteq X\}$. The *contraction* of $\mathbf{M}$ with respect to X is defined as $\mathbf{M}/X = (S \setminus X, \mathcal{I}/X)$ with $\mathcal{I}/X = \{I \subseteq S \setminus$

[2] We use the shorthand $I + e$ and $I - e$ to stand for $I \cup \{e\}$ and $I \setminus \{e\}$, respectively.

$X \mid I \cup B \in \mathcal{I}$ for some base B of $\mathbf{M}|X\}$. The *direct sum* of matroids $\mathbf{M}_1 = (S_1, \mathcal{I}_1)$ and $\mathbf{M}_2 = (S_2, \mathcal{I}_2)$, denoted by $\mathbf{M}_1 \oplus \mathbf{M}_2$, is defined to be $(S_1 \cup S_2, \mathcal{I}')$, where $\mathcal{I}' = \{I_1 \cup I_2 \mid I_1 \in \mathcal{I}_1, I_2 \in \mathcal{I}_2\}$.

Given a matroid $\mathbf{M} = (S, \mathcal{I})$ and a weight function $w \colon S \to \mathbb{Z}_{\geq 0}$, a set $I \in \mathcal{I}$ is said to be *w-maximum*, if its weight $\sum_{e \in I} w(e)$ is maximum among all independent sets in $\mathcal{I}$. A base is called a *w-maximum base*, if its weight is maximum among all bases. Using the family of w-maximum bases of $\mathbf{M} = (S, \mathcal{I})$, one can define a new matroid $\mathbf{M}^w = (S, \mathcal{I}^w)$, where

$$\mathcal{I}^w = \{I \mid I \subseteq B \text{ for some } w\text{-maximum base } B \text{ of } \mathbf{M}\}.$$

It is well known that $\mathbf{M}^w$ is a matroid (see e.g., [14]).

The following lemma states some important properties of such a derived matroid $\mathbf{M}^w$. As these properties are well-known (e.g. see [7]), we omit the proof.

Lemma 1 *Assume that we are given a matroid $\mathbf{M} = (S, \mathcal{I})$ and a weight function $w \colon S \to \{0, 1, \ldots, W\}$. We define $Z(t) = \{e \in S \mid w(e) \geq t\}$ for each integer $t \geq 0$.*

(i) $\mathbf{M}^w = \bigoplus_{t=0}^{W} (\mathbf{M}|Z(t))/Z(t+1)$.
(ii) A set $I \in \mathcal{I}$ is w-maximum if and only if $I \cap Z(t)$ is a base of $\mathbf{M}|Z(t)$ for every $t = 1, 2, \ldots, W$.
(iii) Suppose that a set $I \in \mathcal{I}$ satisfies the condition that $I \cap Z(t)$ is a base in $\mathbf{M}|Z(t)$ for every integer t with $(\min_{e \in S} w(e)) + 1 \leq t \leq W$, and $I + e_0$, where $e_0 \in S \setminus I$, contains a circuit C' of $\mathbf{M}^w$. Then, every element in C' has weight equal to $w(e_0)$. Furthermore, there exists a circuit $C \supseteq C'$ in $I + e$ with respect to $\mathbf{M}$, and each element in $C \setminus C'$ has weight greater than $w(e_0)$.

2.2 Matroid intersection

Suppose that we are given a pair of matroids $\mathbf{M}_1 = (S, \mathcal{I}_1)$ and $\mathbf{M}_2 = (S, \mathcal{I}_2)$ on the same ground set S. A subset I of S is called a *common independent set* if I is in $\mathcal{I}_1 \cap \mathcal{I}_2$. The goal of the *matroid intersection problem* is to find a maximum-cardinality common independent set. Given $\mathbf{M}_1$ and $\mathbf{M}_2$, the *auxiliary graph* is a directed graph $G_{\mathbf{M}_1, \mathbf{M}_2}(I) = (S, E_1 \cup E_2)$, where

$$E_1 = \{ef \mid I + e \notin \mathcal{I}_1,\ I + e - f \in \mathcal{I}_1\},$$
$$E_2 = \{fe \mid I + e \notin \mathcal{I}_2,\ I + e - f \in \mathcal{I}_2\}.$$

In the auxiliary graph $G_{\mathbf{M}_1, \mathbf{M}_2}(I)$, we also define

$$X_1 = \{e \in S \setminus I \mid I + e \in \mathcal{I}_1\},$$
$$X_2 = \{e \in S \setminus I \mid I + e \in \mathcal{I}_2\}.$$

In the auxiliary graph, a directed path from X_2 to X_1 is an *augmenting path*. Let P be a shortest augmenting path. Define $I \Delta P = (I \setminus P) \cup (P \setminus I)$. It is known (e.g. [7]) that $I \Delta P$ is another common independent set, whose size is one larger than I. If there is no augmenting path in the auxiliary graph, then I is already a maximum-cardinality common independent set. Thus, we can find a maximum-cardinality common independent set in a polynomial number of oracle calls; starting with a common independent set $I \in \mathcal{I}_1 \cap \mathcal{I}_2$ (I can be $\emptyset$), we repeatedly augment the current common independent set I to a larger one by finding a shortest augmenting path in $G_{\mathbf{M}_1,\mathbf{M}_2}(I)$. The algorithm constructs an auxiliary graph in each iteration, which takes $O(nr)$ independence oracle calls. Since the number of augmentations is at most r, it runs in $O(nr^2\tau)$ time.

Cunningham [9] improves the running time to $O(nr^{1.5}\tau)$ by finding a maximal number of disjoint augmenting paths in each iteration. For graphic matroids, we can obtain augmentation-type algorithms running in $O(\sqrt{r}n \log r)$ time [23], and $O(\sqrt{r}n)$ time if $n = \Omega(r^{1.5} \log r)$ [22].

Given two matroids $\mathbf{M}_\ell = (S, \mathcal{I}_\ell)$ ($\ell = 1, 2$) and a weight function $w \colon S \to \mathbb{Z}_{\geq 0}$, the *weighted matroid intersection problem* is to find a common independent set with maximum weight. A pair of functions $w_\ell \colon S \to \mathbb{Z}_{\geq 0}$ for $\ell = 1, 2$ is a *weight-splitting of* w if $w(e) = w_1(e) + w_2(e)$ for every $e \in S$. Frank gave two different proofs [16,17] to the following min-max theorem. Note that our result (Theorem 2) gives an alternative proof of Theorem 1, as our algorithm does not rely on Theorem 1.

Theorem 1 *Let $\mathbf{M}_1 = (S, \mathcal{I}_1)$ and $\mathbf{M}_2 = (S, \mathcal{I}_2)$ be two matroids and $w \colon S \to \mathbb{Z}_{\geq 0}$ a weight function. Then the maximum weight of a common independent set is equal to*

$$\min_{w_1, w_2 : weight-splitting} b_1(w_1) + b_2(w_2),$$

where $b_\ell(w_\ell)$ denotes the weight of the w_ℓ-maximum independent set of $\mathbf{M}_\ell$ for $\ell =$ 1, 2.

3 Exact algorithm

In this section, we present an exact algorithm for the weighted matroid intersection problem. Let $W = \max_{e \in S} w(e)$.

Our algorithm runs in W rounds. For ease of presentation, our algorithm starts from Round W and down to Round 1. In Round i, the subset $S' \subseteq S$ of elements e with $w(e) \geq i$ is the ground set of the two matroids.

We maintain a pair of weight functions w_1 and w_2 as a weight splitting of the original weight w. We define a new pair of matroids $\mathbf{M}'_1$ and $\mathbf{M}'_2$ as the restrictions of $\mathbf{M}_1^{w_1}$ and $\mathbf{M}_2^{w_2}$ to S'. In each round, the algorithm finds a maximum-cardinality common independent set I between $\mathbf{M}'_1$ and $\mathbf{M}'_2$ using I', where I' is the common independent set found in the previous round. As we will show in Sect. 5, the augmentation-type algorithm described in Sect. 2.2 can be used to obtain I with the additional property called *near-optimality* (see Definition 1). At the end of the round, we update w_1, w_2 based on the auxiliary graph $G_{\mathbf{M}'_1,\mathbf{M}'_2}(I)$. Below we first present the algorithm and then elaborate on the details.

Algorithm 1: Exact algorithm
Input: two matroids $\mathbf{M}_1 = (S, \mathcal{I}_1)$ and $\mathbf{M}_2 = (S, \mathcal{I}_2)$, a weight function $w \colon S \to \mathbb{Z}_{\geq 0}$, and $W = \max_{e \in S} w(e)$.
Output: $I \in \mathcal{I}_1 \cap \mathcal{I}_2$ where I is a maximum-weight common independent set of $\mathbf{M}_1$ and $\mathbf{M}_2$.
Step 1. Set $i := W$, $w_1 := 0$, $w_2 := w$, and $I' := \emptyset$.
Step 2. While $i > 0$ do the following steps.
(2-1) Set $S' := \{e \in S \mid w_2(e) \geq i\}$.
(2-2) Set $\mathbf{M}'_\ell = (S', \mathcal{I}'_\ell)$ to be $\mathbf{M}_\ell^{w_\ell} | S'$ for $\ell = 1, 2$.
(2-3) Unweighted_Matroid_Intersection (I')

Construct I so that

(i) I is a maximum-cardinality common independent set of $\mathbf{M}'_1$ and $\mathbf{M}'_2$, and
(ii) I is (w_1, w_2)-near-optimal in S'.

(2-4) Update_Weight
(2-4-1) Let $T \subseteq S'$ be the set of elements reachable from X_2 in $G_{\mathbf{M}'_1, \mathbf{M}'_2}(I)$.
(2-4-2) For each $e \in T$, let $w_1(e) := w_1(e) + 1$, $w_2(e) := w_2(e) - 1$.
(2-5) Set $i := i - 1$ and $I' := I$.
Step 3. Return I.

Note that in Step (2–3), Unweighted_Matroid_Intersection takes I', which is the common independent set computed in the previous round, to construct I. The implementation details (depending on the type of given matroids) will be given in Sect. 5. Roughly speaking, we will show that if I' is already (w_1, w_2)-near-optimal in S', then we can compute I, based on I', so that I becomes the maximum cardinality common independent set of $\mathbf{M}'_1$ and $\mathbf{M}'_2$ while remaining (w_1, w_2)-near-optimal.

3.1 Analysis

The final goal of our algorithm is to find a common independent set that is w_1-maximum in $\mathbf{M}_1$ and w_2-maximum in $\mathbf{M}_2$, which would imply that I is w-maximum if $w = w_1 + w_2$. For each integer t, let

$$Z_1(t) = \{e \in S \mid w_1(e) \geq t\},$$
$$Z_2(t) = \{e \in S \mid w_2(e) \geq t\}.$$

Lemma 1(ii) implies that I being w_1-maximum in $\mathbf{M}_1$ and w_2-maximum in $\mathbf{M}_2$ is equivalent to

1. $I \cap Z_1(t)$ is a base of $\mathbf{M}_1 | Z_1(t)$ for every integer $t \geq 1$, and
2. $I \cap Z_2(t)$ is a base of $\mathbf{M}_2 | Z_2(t)$ for every integer $t \geq 1$.

Such a common independent set I of $\mathbf{M}_1, \mathbf{M}_2$ is called (w_1, w_2)-*optimal*.

We relax the above condition as follows. We here define $Z'_\ell(t) = Z_\ell(t) \cap S'$ for each subset $S' \subseteq S$ and $\ell = 1, 2$.

Definition 1 A common independent set I of $\mathbf{M}_1$ and $\mathbf{M}_2$ is (w_1, w_2)-near-optimal *in a subset* $S' \subseteq S$ if

1. $I \cap Z'_1(t)$ is a base of $\mathbf{M}_1|Z'_1(t)$ for every integer $t \geq 1$, and
2. $I \cap Z'_2(t)$ is a base of $\mathbf{M}_2|Z'_2(t)$ for every integer $t \geq \alpha + 1$, where $\alpha = \min_{e \in S'} w_2(e)$.

Note that if $\alpha = 0$ and $S' = S$, then a (w_1, w_2)-near-optimal common independent set in S' is (w_1, w_2)-optimal.

In what follows, we will prove that, during the execution of our algorithm, the current set I is always (w_1, w_2)-near-optimal in S'. To prove this, we analyze the two procedures Unweighted_Matroid_Intersection and Update_Weight used in Steps (2–3) and (2–4).

In Unweighted_Matroid_Intersection of Step (2–3), if we only want a maximum-cardinality common independent set I of $\mathbf{M}'_1$ and $\mathbf{M}'_2$, the step is trivial. The difficulty is how to guarantee that I is also (w_1, w_2)-near-optimal in S' *without* resorting to weighted matroid intersection. The details are deferred to Sect. 5. We use a lemma to summarize the outcome of Step (2–3). Recall that we denote $\mathbf{M}'_\ell = \mathbf{M}_\ell^{w_\ell}|S'$ for $\ell = 1, 2$.

Lemma 2 *Suppose that I' is (w_1, w_2)-near-optimal in a subset S'. Then we can construct another common independent set I, using known unweighted matroid intersection algorithms, that is simultaneously* (i) *a maximum-cardinality common independent set of $\mathbf{M}'_1$ and $\mathbf{M}'_2$, and* (ii) *(w_1, w_2)-near-optimal in S'.*

We next prove that, if the maximum-cardinality common independent set I of $\mathbf{M}'_1$ and $\mathbf{M}'_2$ is (w_1, w_2)-near-optimal in S', then we can modify w_1 and w_2 at Step (2–4) so that I is still (w_1, w_2)-near-optimal in S'.

Lemma 3 *Suppose that all weights of w_1 and w_2 are nonnegative integers, and there are some integers p_1 and p_2 such that $w_1(e) \leq p_1$ and $w_2(e) \geq p_2$ for every $e \in S'$. In addition, suppose that I is* (i) *a maximum-cardinality common independent set of $\mathbf{M}'_1$ and $\mathbf{M}'_2$, and* (ii) *(w_1, w_2)-near-optimal in S'. Then, after the procedure* Update_Weight, *we have*

(1) $I \cap Z'_1(t)$ is a base of $\mathbf{M}_1|Z'_1(t)$ for every integer t with $1 \leq t \leq p_1 + 1$, and
(2) $I \cap Z'_2(t)$ is a base of $\mathbf{M}_2|Z'_2(t)$ for every integer $t \geq p_2$.

It should be noted that Lemma 3 implies that after Step (2–4), I is still (w_1, w_2)-near-optimal in S', since then $\max_{e \in S'} w_1(e) \leq p_1 + 1$ and $\min_{e \in S'} w_2(e) \geq p_2 - 1$.

Proof We only prove (1), since (2) follows symmetrically. To avoid confusion, let $\tilde{Z}'_1(t)$ denote the set $Z'_1(t)$ *after* the weights w_1 and w_2 are updated. Observe that, for every integer t with $1 \leq t \leq p_1 + 1$,

$$\tilde{Z}'_1(t) = Z'_1(t) \cup ((Z'_1(t-1) \setminus Z'_1(t)) \cap T),$$

where we note that $Z'_1(p_1 + 1) = \emptyset$ and $Z'_1(0) = S'$.

As $I \cap Z'_1(t)$ is a base of $\mathbf{M}_1|Z'_1(t)$, we argue that given an element $e \in ((Z'_1(t-1) \setminus Z'_1(t)) \cap T) \setminus I$:

(∗) $I + e \notin \mathcal{I}_1|S'$, and

$(**)$ the circuit of $I + e$ in $\mathbf{M}_1|S'$ is contained in $\tilde{Z}'_1(t)$.

This will establish that $I \cap \tilde{Z}'_1(t)$ is a base of $\mathbf{M}_1|\tilde{Z}'_1(t)$ for every $t = 1, 2, \ldots, p_1 + 1$.

To see $(*)$, observe that in $G_{\mathbf{M}'_1,\mathbf{M}'_2}(I)$, e is not part of X_1. Otherwise, there would be an augmenting path, contradicting to the assumption that I is a maximum-cardinality common independent set in $\mathbf{M}'_1$ and $\mathbf{M}'_2$. Thus, $I + e$ contains a circuit C' in $\mathbf{M}'_1$. Furthermore, by Lemma 1(iii) applied to $\mathbf{M}_1|S'$ (as the assumption is that $I \cap Z'_1(t)$ is a base of $\mathbf{M}_1|Z'_1(t)$ for every $t = 1, 2, \ldots, p_1$), $I + e$ also has a circuit $C \supseteq C'$ in $\mathbf{M}_1|S'$. Thus, $(*)$ is proved.

To see $(**)$, consider an element e' in $C' - e$. Then, e' is contained in $Z'_1(t-1) \setminus Z'_1(t)$ by Lemma 1(iii). Since $e' \in C'$, in $G_{\mathbf{M}'_1,\mathbf{M}'_2}(I)$, there is an arc from e to e'. Thus, e' is part of T. This implies that C' is a subset of $(Z'_1(t-1) \setminus Z'_1(t)) \cap T$, which in turn, by Lemma 1(iii), implies that the circuit $C \supseteq C'$ in $I + e$ with respect to $\mathbf{M}_1|S'$ is a subset of $Z'_1(t) \cup ((Z'_1(t-1) \setminus Z'_1(t)) \cap T) = \tilde{Z}'_1(t)$. The proof of $(**)$ follows. □

By induction on i with Lemmas 2 and 3, we show that the current set I is always (w_1, w_2)-near-optimal.

Lemma 4 *In Round i with $1 \le i \le W$, the following holds.*

(1) $w = w_1 + w_2$,
(2) *after Step* (2-4)*, $I \cap Z'_1(t)$ is a base of $\mathbf{M}_1|Z'_1(t)$ for every integer t with $1 \le t \le W - i + 1$, and*
(3) *after Step* (2-4)*, $I \cap Z'_2(t)$ is a base of $\mathbf{M}_2|Z'_2(t)$ for every integer t with $i \le t \le W$.*

Proof (1) can be easily seen. We prove (2) and (3) by induction on i.

For the base case of $i = W$, since $Z'_1(1) = \emptyset$ and $Z'_2(W+1) = \emptyset$ hold, $I' = \emptyset$ is (w_1, w_2)-near-optimal in S', and thus Lemma 2 implies that we can obtain a maximum-cardinality common independent set I of $\mathbf{M}'_1$ and $\mathbf{M}'_2$ satisfying the condition that $I \cap Z'_1(1)$ is a base of $\mathbf{M}_1|Z'_1(1)$ and $I \cap Z'_2(W+1)$ is a base of $\mathbf{M}_2|Z'_2(W+1)$. Now applying Lemma 3 (with $p_1 = 0$ and $p_2 = W$), we have that $I \cap Z'_1(1)$ is a base of $\mathbf{M}_1|Z'_1(1)$ and $I \cap Z'_2(W)$ is a base of $\mathbf{M}_2|Z'_2(W)$.

For the induction step $i < W$, let I' be the common independent set obtained in Round $i + 1$. By induction hypothesis, $I' \cap Z'_1(t)$ is a base of $\mathbf{M}_1|Z'_1(t)$ for every integer t with $1 \le t \le W - i$ and $I' \cap Z'_2(t)$ is a base of $\mathbf{M}_2|Z'_2(t)$ for every integer t with $i+1 \le t \le W$. Notice that when Round i begins, only elements e with $w_1(e) = 0$ and $w_2(e) = i$ are added to S'. Hence the two conditions remain true after Step (2–1).

By these facts, as $w_2(e) \ge i$ for $e \in S'$, Step (2–3) can be correctly applied by Lemma 2, and we obtain the new independent set I satisfying the two conditions stated in Step (2–3). The proof now follows by applying Lemma 3 (with $p_1 = W - i$ and $p_2 = i$). □

Theorem 2 *The common independent set I returned by Algorithm 1 is a maximum-weight common independent set of $\mathbf{M}_1$ and $\mathbf{M}_2$.*

Proof By Lemma 4, after the last round when $i = 1$, as $S' = S$, $I \cap Z_1(t)$ is a base of $\mathbf{M}_1|Z_1(t)$ for every $t = 1, 2, \ldots, W$, and $I \cap Z_2(t)$ is a base of $\mathbf{M}_2|Z_2(t)$ for every

$t = 1, 2, \ldots, W$. Thus, it follows from Lemma 1(ii) that I is w_ℓ-maximum in $\mathbf{M}_\ell$ for every $\ell = 1, 2$. Then, for every common independent set J, we have

$$w(J) = w_1(J) + w_2(J) \leq w_1(I) + w_2(I) = w(I).$$

Thus, I is a maximum-weight common independent set. This completes the proof. □

The algorithm clearly runs in $O(W(T_\mathrm{u} + T_\mathrm{d}))$ time, where T_u and T_d are the running times for executing Unweighted_Matroid_Intersection and Update_Weight, respectively. Note that T_u and T_d depend on the representation of the given matroids. Their complexities are discussed in Sect. 5.

4 Approximation algorithm

In this section, we will design a $(1 - \epsilon)$-approximation algorithm for the weighted matroid intersection. Let W be the maximum weight. First of all, we show that we can round weights to small integers, and bound W from above.

Lemma 5 *We can reduce a given instance of the weighted matroid intersection problem to one with integral weights whose maximum weight is at most $2r_*/\epsilon$, where $r_* \leq r$ is the maximum size of a common independent set.*

Proof Set $\eta = \epsilon W/2r_*$, and define $w'(e) = \lfloor w(e)/\eta \rfloor$ for each $e \in S$. Then, a $(1-\epsilon/2)$-approximate solution I' for the weight w' is a $(1-\epsilon)$-approximate solution for the weight w. Indeed, since $w(e) - \eta \leq \eta w'(e) \leq w(e)$ for every $e \in S$, we have

$$\begin{aligned}
w(I') &\geq \eta w'(I') \\
&\geq \eta(1-\epsilon/2)w'(I'_{\mathrm{opt}}) \qquad (I'_{\mathrm{opt}} \text{is an optimal solution for} w') \\
&\geq \eta(1-\epsilon/2)w'(I_{\mathrm{opt}}) \qquad (I_{\mathrm{opt}} \text{is an optimal solution for} w) \\
&\geq (1-\epsilon/2)(w(I_{\mathrm{opt}}) - \eta|I_{\mathrm{opt}}|) \\
&\geq (1-\epsilon/2)(w(I_{\mathrm{opt}}) - \eta r_*) \\
&= (1-\epsilon/2)(w(I_{\mathrm{opt}}) - \epsilon W/2) \\
&\geq (1-\epsilon)w(I_{\mathrm{opt}}),
\end{aligned}$$

where the last inequality follows because we assume that the given matroids have no loop, so the element e with $w(e) = W$ is a common independent set, thus, $w(I_{\mathrm{opt}}) \geq W$. □

During the algorithm, the weight w is split so that $w \approx w_1 + w_2$; furthermore, we will guarantee that all weights of w_1 and w_2 are nonnegative multiples of some integer $\delta > 0$, where δ may change in different phases of the algorithm. At the end, we find a common independent set that is w_1-maximum in $\mathbf{M}_1$ and w_2-maximum in $\mathbf{M}_2$, which would imply that I is a $(1-\epsilon)$-approximate solution if $w \leq w_1 + w_2 \leq (1+\epsilon)w$.

For simplicity, we assume that the bound W and ϵ are both powers of 2. Then, our algorithm runs in $1 + \log_2 \epsilon W$ phases. In every phase, we apply a number (roughly

$O(\epsilon^{-1})$) of Unweighted_Matroid_Intersection and Update_Weight operations. Note that $\log_2 \epsilon W = O(\log r)$ by Lemma 5.

Let $\delta_0 = \epsilon W$. For each integer i with $1 \le i \le \log_2 \epsilon W$, define $\delta_i = \delta_0/2^i$. The term δ_i will be the amount of change in the weights w_1 and w_2 during Phase i every time Update_Weight is invoked. For each $e \in S$ and each integer i with $0 \le i \le \log_2 \epsilon W$, define $w^i(e)$ to be the truncated weight of element e in Phase i, i.e., $w^i(e) = \lfloor w(e)/\delta_i \rfloor \delta_i$. Notice that $w^{i+1}(e) = w^i(e)$ or $w^{i+1}(e) = w^i(e) + \delta_{i+1}$. The algorithm, presented below, returns a $(\frac{1}{1+4\epsilon})$-approximate solution.

Algorithm 2: Approximation algorithm

Input: two matroids $\mathbf{M}_1 = (S, \mathcal{I}_1)$ and $\mathbf{M}_2 = (S, \mathcal{I}_2)$, a weight function $w\colon S \to \mathbb{Z}_{\ge 0}$, and $W = \max_{e \in S} w(e)$.

Output: $I \in \mathcal{I}_1 \cap \mathcal{I}_2$ where $w(I) \ge \frac{w(I^{opt})}{1+4\epsilon}$.

Step 1. Set $i := 0$, $w_1 := 0$, $w_2 := w^0$, $I' := \emptyset$, and $h := W$.

Step 2. Applying Algorithm 1:

While $i \le \log_2 \epsilon W$, do the following steps.

(2-0) Set $L := \frac{W}{2^{i+1}}$ if $i < \log_2 \epsilon W$, and $L := 1$ if $i = \log_2 \epsilon W$.

(2-1) While $h \ge L$, do the following steps.

(2-1-1) Set $S' := \{e \in S \mid w_2(e) \ge h\}$.

(2-1-2) Set $\mathbf{M}'_\ell = (S', \mathcal{I}'_\ell)$ to be $\mathbf{M}_\ell^{w_\ell}|S'$ for each $\ell = 1, 2$.

(2-1-3) Unweighted_Matroid_Intersection

Construct I using I' so that

(i) I is a maximum-cardinality common independent set of $\mathbf{M}'_1$ and $\mathbf{M}'_2$, and

(ii) I is (w_1, w_2)-near-optimal in S'.

(2-1-4) Update_Weight

(i) Let $T \subseteq S'$ be the set of elements reachable from X_2 in $G_{\mathbf{M}'_1, \mathbf{M}'_2}(I)$.

(ii) For each $e \in T$, let $w_1(e) := w_1(e) + \delta_i$, $w_2(e) := w_2(e) - \delta_i$.

(2-1-5) Set $h := h - \delta_i$ and $I' := I$.

(2-2) Weight Adjustment:

If $i < \log_2 \epsilon W$, do the following.

(2-2-1) $\forall e \in I'$, let $w_2(e) = w_2(e) + \delta_{i+1}$.

(2-2-2) $\forall e \in S \setminus I'$ where $w^{i+1}(e) = w^i(e) + \delta_{i+1}$, let $w_2(e) = w_2(e) + \delta_{i+1}$.

(2-2-3) Set $h := h + \delta_{i+1}$.

(2-3) Set $i := i + 1$.

Step 3. Return I.

The outer loop Step 2 corresponds to a phase. We use a counter h to keep track of the progress of the algorithm. Initially $h = W$. In Phase i, the weights are always kept as nonnegative multiples of δ_i. In Step (2–1), the two matroids $\mathbf{M}'_1$ and $\mathbf{M}'_2$ are defined on the common ground set $S' = \{e \in S \mid w_2(e) \ge h\}$, and the two procedures Unweighted_Matroid_Intersection and Update_Weight are invoked as was done in the exact algorithm (Algorithm 1) in Sect. 3. The counter h is decreased by the amount of δ_i each time after Update_Weight is invoked in Step (2–1).

Each time h is halved, we make ready to move to the next phase, except in the last phase: in Phase $\log_2 \epsilon W$, we stop when h goes down to 1. The reason that we adjust the w_2-weights at Step (2–2) is that we want to ensure that in the beginning of the next phase, the weights w_1 and w_2 still approximate the next weight w^{i+1} (see Lemma 9). In particular, we increase the w_2-weights of all elements in the current common independent set I'. This is to make sure that I' is still w_2-maximum in the beginning of the next phase (with respect to the newly-defined set S' in Step (2–1)).

4.1 Analysis

We first observe the number of iterations in the algorithm.

Lemma 6 (1) *During Phase i with $0 \le i \le \log_2 \epsilon W$, w_1 and w_2 are nonnegative multiples of δ_i, except in Step (2–2).*

(2) *Step (2–1) is executed at most $\frac{\epsilon^{-1}}{2}$ times in Phase i with $0 \le i < \log_2 \epsilon W$. In the last phase, Step (2–1) is executed $\epsilon^{-1} + 1$ times.*

(3) *The total number of iterations in Step (2–1) is $O(\epsilon^{-1} \log r)$.*

Proof (1) can be easily verified. For (2), observe that in Phase 0, Step (2–1) is executed

$$\frac{W - W/2}{\delta_0} = \frac{\epsilon^{-1}}{2}$$

times. For Phase $i \ge 1$, in the beginning of that phase, $h = \frac{W}{2^i} - \delta_i$. Hence, if $i < \log_2 \epsilon W$, Step (2–1) is executed

$$\frac{(W/2^i - \delta_i) - W/2^{i+1}}{\delta_i} \le \frac{\epsilon^{-1}}{2}$$

times, and if $i = \log_2 \epsilon W$, Step (2–1) is executed

$$\frac{W/2^i - \delta_i}{\delta_i} \le \epsilon^{-1}$$

times. (3) now immediately follows from (2). □

We say an element $e \in S$ *joins* in Phase j if in Phase j, element e becomes a part of the ground set S' in Step (2–1–1) the first time.

Lemma 7 *Suppose that an element $e \in S$ joins in Phase j for some integer j with $j < \log_2 \epsilon W$. Then the following holds.*

(1) $w^j(e) \ge \frac{W}{2^{j+1}} = \frac{\delta_j}{2\epsilon}$.

(2) *In every phase $i \ge j$, $w^i(e) \le w_1(e) + w_2(e) \le w^i(e) + 2\delta_j$.*

(3) *If $e \in S$ joins in the last phase $j = \log_2 \epsilon W$, then $w_1(e) + w_2(e) = w^j(e)$.*

Proof Notice that immediately before e joins in Phase j, we have $w_1(e) + w_2(e) = w^j(e)$. This follows from the observation that unless e is part of I when Step (2–2–1) is executed, the weight splitting $w_1(e)$ and $w_2(e)$ is *exact* with respect $w^{j'}(e)$ for $j' \le j$. (3) follows easily from this observation. In the case that $j < \log_2 \epsilon W$, we have that $w^j(e) \ge w_2(e) \ge \frac{W}{2^{j+1}}$. Thus (1) is proved.

(2) follows from the fact that the difference between the sum of $w_1(e)$ and $w_2(e)$ and the truncated weight $w^{j'}(e)$ grows larger only when Step (2–2–1) is executed in Phase $j' \ge j$ and e is part of the common independent set I in that step. Hence it holds that

$$\begin{aligned} w^i(e) \le w_1(e) + w_2(e) &\le w^i(e) + \sum_{s=j}^{i} \delta_s \\ &\le w^i(e) + 2\delta_j. \end{aligned}$$

This completes the proof. □

Since all weights of w_1, w_2 are nonnegative multiples of δ_i and we modify w_1 and w_2 by δ_i at Update_Weight, we have the following lemma, which can be obtained similarly to Lemma 3 by dividing all the values by δ_i.

Lemma 8 *Suppose that all weights of w_1 and w_2 are nonnegative multiples of δ, and there are some integers p_1 and p_2 such that $w_1(e) \le p_1$ and $w_2(e) \ge p_2$ for every $e \in S'$. In addition, suppose that I is* (i) *a maximum-cardinality common independent set of $\mathbf{M}'_1$ and $\mathbf{M}'_2$, and* (ii) (w_1, w_2)*-near-optimal in S'. Then after the procedure* Update_Weight, *we have*

(1) *$I \cap Z'_1(t)$ is a base of $\mathbf{M}_1|Z'_1(t)$ for every integer t with $1 \le t \le p_1 + \delta$, and*
(2) *$I \cap Z'_2(t)$ is a base of $\mathbf{M}_2|Z'_2(t)$ for every integer $t \ge p_2$.*

Note that the lemma implies that the current independent set I is still (w_1, w_2)-near-optimal in S' after Step (2–1–4).

We finally see that Weight_Adjustment maintains I' (w_1, w_2)-near-optimal in S'.

Lemma 9 *In Phase i, after Step* (2–1) *terminates, we have the following.*

(1) *$I' \cap Z'_1(t)$ is a base of $\mathbf{M}_1|Z'_1(t)$ for every integer $t \ge 1$.*
(2) *$I' \cap Z'_2(t)$ is a base of $\mathbf{M}_2|Z'_2(t)$ for every integer $t \ge h + \delta_i$.*

Proof We first prove the following claim.

Claim In each phase, if (1) and (2) hold before the first iteration of Step (2–1) starts, we have (1) and (2) after the final iteration of Step (2–1) terminates.

Proof We prove the claim by induction on the number of times Step (2–1) is invoked. For the base case, we have (1) and (2) in the beginning by the assumption.

Suppose that we have (1) and (2) for the previous set I' at the beginning of the current iteration in Step (2–1). At Step (2–1–1), some elements may be added into S'.

However, all such elements have $w_1(e) = 0$ and $w_2(e) = h$. Thus, I' still satisfies (1) and (2), and thus it is (w_1, w_2)-near-optimal in S' since $w_2(e) \geq h$ for every $e \in S'$. By Lemma 2, Step (2–1–3) can be correctly implemented, and we obtain a maximum-cardinality common independent set I of $\mathbf{M}'_1$ and $\mathbf{M}'_2$ that is (w_1, w_2)-near-optimal in S'. After Step (2–1–4), by Lemma 8 (by setting $\delta = \delta_i$, $p_1 = \max_{e \in S'} w_1(e)$, and $p_2 = h$), we have that I satisfies (1) and $I \cap Z'_2(t)$ is a base of $\mathbf{M}_2|Z'_2(t)$ for any integer $t \geq h$. Since h is decreased by δ_i in Step (2–1–5), we have (1) and (2) at the end of the current iteration. This proves the claim. □

We prove the lemma by induction on the number of phases. For the base case, as in the beginning of the algorithm, $h = W$ and $I' = \emptyset$, the set I' is (w_1, w_2)-near-optimal in S'. This means that we have (1) and (2) for I', and hence Claim 4.1 implies that we have (1) and (2) after the iterations of Step (2–1) terminates in Phase 0.

For the induction step, suppose that currently the algorithm is in Phase i, and that (1) and (2) are satisfied after Step (2–1) are done. We argue that after the weight adjustment done in Step (2–2), I' still satisfies (1) and (2).

To avoid confusion, let $\tilde{Z}_\ell(t)$ ($\ell = 1, 2$) denote the sets *after* w_2-weights are modified in Steps (2–2–1) and (2–2–2), and let $\tilde{h}$ be the value of h after Step (2–2–3), i.e., $\tilde{h} = h + \delta_{i+1}$.

By Lemma 6(1), all w_1 and w_2 weights are multiples of δ_i in Phase i before Step (2–2). Therefore, after Step (2–1), the fact that I' satisfies (2) implies

($\star$) $I' \cap Z'_2(t)$ is a base of $\mathbf{M}_2|Z'_2(t)$ for every integer $t \geq h + \delta_{i+1}$.

To see this, note that I' satisfying (2) only guarantees this property for $t \geq h + \delta_i$. We can subtract δ_{i+1} further because there is no element with w_2-weight of the form $a\delta_i + \delta_{i+1}$ for some integer $a \geq 0$. Hence the range of t starts from $h + \delta_i - \delta_{i+1} = h + \delta_{i+1}$.

As we increase the w_2-weights of all elements in I' and a subset of elements in $S' \setminus I'$, while leaving the w_1-weights unchanged, we have

(i) $\tilde{Z}_1(t) = Z_1(t)$ for all $t \in \mathbb{Z}_{\geq 0}$.
(ii) $I \cap \tilde{Z}'_1(t)$ is a base of $\mathbf{M}_1|\tilde{Z}'_1(t)$ for every integer $t \geq 1$.
(iii) $I \cap \tilde{Z}'_2(t)$ is a base of $\mathbf{M}_2|\tilde{Z}'_2(t)$ for every integer $t \geq \tilde{h} + \delta_{i+1}$.

(i) and (ii) are easy to see, since w_1-weights are unchanged and (1) holds before Step (2–2). For (iii), consider any integer $t \geq \tilde{h} + \delta_{i+1} = h + 2\delta_{i+1}$. We have that $I' \cap \tilde{Z}'_2(t) = I' \cap Z'_2(t - \delta_{i+1})$, where the latter is a base of $\mathbf{M}_2|Z'_2(t - \delta_{i+1})$ by ($\star$). As $\tilde{Z}'_2(t) \subseteq Z'_2(t - \delta_{i+1})$, we infer that $I' \cap \tilde{Z}'_2(t)$ is still a base of $\mathbf{M}_2|\tilde{Z}'_2(t)$.

Therefore, at the beginning of Phase $i + 1$, we have (1) and (2), and hence the proof follows from Claim 4.1. This completes the proof. □

Lemma 10 *The common independent set I returned by Algorithm 2 is a maximum-weight common independent set, with respect to the final weight function $w_1 + w_2$.*

Proof After the last time Step (2–1–5) is executed, by Lemma 9 and the fact that $S' = S$, $I \cap Z_1(t)$ is a base of $\mathbf{M}_1|Z_1(t)$ for every integer $t \geq 1$, and $I \cap Z_2(t)$ is a base of $\mathbf{M}_2|Z_2(t)$ for every integer $t \geq \delta_{\log_2 \epsilon W}$. Since $\delta_{\log_2 \epsilon W} = 1$, it follows from

Lemma 1(ii) that I is w_1-maximum in $\mathbf{M}_1$ and w_2-maximum in $\mathbf{M}_2$. Therefore, for every common independent set J, we have

$$w_1(J) + w_2(J) \le w_1(I) + w_2(I).$$

The proof follows. □

Theorem 3 *Let I be the common independent set returned by Algorithm 2. Then I is a $(1-4\epsilon)$ approximation.*

Proof For every $e \in S$, if it joins in Phase $j < \log_2 \epsilon W$, then by Lemma 7(2),

$$\begin{aligned} w^{\log_2 \epsilon W}(e) &\le w_1(e) + w_2(e) \le w^{\log_2 \epsilon W}(e) + 2\delta_j \\ &\le (1+4\epsilon) w^{\log_2 \epsilon W}(e), \end{aligned}$$

where the last inequality holds since $\delta_j \le 2\epsilon w^j(e) \le 2\epsilon w^{\log_2 \epsilon W}(e)$ by Lemma 7(1). If $j = \log_2 \epsilon W$, then $w^{\log_2 \epsilon W}(e) = w_1(e) + w_2(e)$ by Lemma 7(3). Since $w^{\log_2 \epsilon W}(e) = w(e)$, we conclude that, for each $e \in S$,

$$w(e) \le w_1(e) + w_2(e) \le (1+4\epsilon) w(e).$$

Thus, letting I_{opt} be the maximum-weight common independent set, Lemma 10 implies

$$\begin{aligned} w(I_{\text{opt}}) &\le w_1(I_{\text{opt}}) + w_2(I_{\text{opt}}) \le w_1(I) + w_2(I) \\ &\le (1+4\epsilon) w(I). \end{aligned}$$

The proof follows. □

5 Implementation of unweighted matroid intersection

In this section, we discuss how to implement the procedure Unweighted_Matroid_Intersection and the actual complexities of our algorithms for various weighted matroid intersection problems.

Let $\mathbf{M}_1$ and $\mathbf{M}_2$ be two matroids, and w_1 and w_2 be weights. Suppose that a given common independent set I' of $\mathbf{M}'_1$ and $\mathbf{M}'_2$ is (w_1, w_2)-near-optimal in a subset $S' \subseteq S$ (recall that $\mathbf{M}'_\ell = \mathbf{M}_\ell^{w_\ell} | S'$ for $\ell = 1, 2$). We explain in the following sections how to find a maximum-cardinality common independent set I between $\mathbf{M}'_1$ and $\mathbf{M}'_2$ that is (w_1, w_2)-near-optimal in S'.

5.1 General matroids

In [9], Cunningham shows how to find a maximum-cardinality common independent set, using $O(nr^{1.5})$ independence oracle calls. This is done by repeatedly finding an augmenting path in the auxiliary graph, as described in Sect. 2.2. We argue that if we apply his algorithm to $\mathbf{M}_1'$ and $\mathbf{M}_2'$ with I' as the initial common independent set, each new independent set resulting from augmentation will satisfy the same property as I'.

Lemma 11 *Suppose that I' is (w_1, w_2)-near-optimal in S', and let P be the shortest path from X_2 to X_1 in $G_{\mathbf{M}_1',\mathbf{M}_2'}(I')$. Then, the set $I = I' \Delta P$ is also (w_1, w_2)-near-optimal.*

Proof By Lemma 1(iii), in $G_{\mathbf{M}_1',\mathbf{M}_2'}(I')$, an element $e \in (Z_1(t) \setminus Z_1(t+1)) \setminus I'$ has outgoing arcs to only other elements in $Z_1(t) \setminus Z_1(t+1)$ for every integer $t \geq 1$. Similarly, an element $e \in (Z_2(t) \setminus Z_2(t+1)) \cap I'$ has only outgoing arcs towards other elements in $(Z_2(t) \setminus Z_2(t+1)) \setminus I'$ for every integer $t \geq p+1$, where $p = \min_{e \in S'} w_2(e)$.

These two facts imply that along the augmenting path P in $G_{\mathbf{M}_1',\mathbf{M}_2'}(I')$, the number of elements in $(Z_1(t) \setminus Z_1(t+1)) \setminus I'$ is the same as the number of elements in $(Z_1(t) \setminus Z_1(t+1)) \cap I'$ for every integer $t \geq 1$. Similarly, the number of elements in $(Z_2(t) \setminus Z_2(t+1)) \cap I'$ is the same as that in $(Z_2(t) \setminus Z_2(t+1)) \setminus I'$ for every integer $t \geq p+1$. Thus, $|I \cap Z_1(t)| = |I' \cap Z_1(t)|$ for every integer $t \geq 1$, and $|I \cap Z_2(t)| = |I' \cap Z_2(t)|$ for every integer $t \geq p+1$. The proof follows. □

Thus, the maximum-cardinality common independent set of $\mathbf{M}_1' = (S', \mathcal{I}_1')$ and $\mathbf{M}_2' = (S', \mathcal{I}_2')$ obtained by Cunningham's algorithm is (w_1, w_2)-near-optimal if so is the initial set. To apply Cunningham's algorithm [9] to $\mathbf{M}_1'$ and $\mathbf{M}_2'$, we need an independence oracle for $\mathbf{M}_1'$ and $\mathbf{M}_2'$ to find an augmenting path. More specifically, for $\ell = 1, 2$, we need to test whether $I' + e \in \mathcal{I}_\ell'$ and whether $I' + e - f \in \mathcal{I}_\ell'$ for a given independent set I', and given elements $e \in S' \setminus I'$ and $f \in I'$. This can be implemented by an independence oracle for $\mathbf{M}_1$ and $\mathbf{M}_2$ as follows. It follows from Lemma 1(iii) that if $I' + e \notin \mathcal{I}_\ell'$, then $I' + e - f \in \mathcal{I}_\ell'$ if and only if $I' + e - f \in \mathcal{I}_\ell$ and $w_\ell(e) = w_\ell(f)$. In addition, $I' + e \in \mathcal{I}_1'$ if and only if $I' + e \in \mathcal{I}_1$ and $w_1(e) = 0$, and $I' + e \in \mathcal{I}_2'$ if and only if $I' + e \in \mathcal{I}_2$ and $w_2'(e) = \min_{e \in S'} w_2(e)$. Thus Unweighted_Matroid_Intersection can be implemented in $O(nr^{1.5})$ independence oracle calls for $\mathbf{M}_1$ and $\mathbf{M}_2$.

We can perform Update_Weight in $O(nr)$ independence oracle calls. Therefore, we have the following theorem for two general matroids.

Theorem 4 *For two general matroids, we can solve the weighted matroid intersection problem exactly in $O(\tau W n r^{1.5})$ time, and $(1-\epsilon)$-approximately in $O(\tau \epsilon^{-1} n r^{1.5} \log r)$ time, where τ is the running time to check the independence of a set in the given matroids.*

For the exact algorithm, a slight sharpening in the running time is possible. Observe that in Round i, Cunningham's algorithm takes $O(\tau |S'| r^{1.5})$ time, where $S' = \{e \in S \mid w(e) \geq i\}$. Since

$$\sum_{i=1}^{W} |\{e \in S \mid w(e) \geq i\}| = \sum_{e \in S} w(e),$$

the total running time is $O(\tau(\sum_{e \in S} w(e))r^{1.5})$. This is superior to the previous one only when the given weights are very "unbalanced."

5.2 Graphic matroids

Suppose that $\mathbf{M}_1$ and $\mathbf{M}_2$ are graphic matroids. That is, $\mathbf{M}_\ell = (S, \mathcal{I}_\ell)$ ($\ell = 1, 2$) is represented by a graph $G_\ell = (V_\ell, S)$ so that $\mathcal{I}_\ell$ is the family of edge subsets in S that are forests in G_ℓ. Note that the number of edges in G_ℓ is $n = |S|$, and the number of vertices is $O(r)$, since we may assume that there is no isolated vertex. Gabow and Xu [23] designed an algorithm that runs in $O(\sqrt{r}n \log r)$ time for the unweighted graphic matroid intersection. Their algorithm is an augmentation-type algorithm, that means it repeatedly finds an augmenting path in the auxiliary graph.

It is well known that, if $\mathbf{M}_\ell$ is graphic, then so is $\mathbf{M}'_\ell = \mathbf{M}_\ell^{w_\ell}|S'$ for a subset S' and $\ell = 1, 2$. Indeed, for a subset $X \subseteq S$, the restriction of G_ℓ to X (the subgraph induced by an edge subset X), denoted by $G_\ell|X$, represents $\mathbf{M}_\ell|X$. Moreover, the graph obtained from G_ℓ by contracting X, denoted by G_ℓ/X, represents $\mathbf{M}_\ell/X$. Then, by Lemma 1(i), $\mathbf{M}'_\ell = \mathbf{M}_\ell^{w_\ell}|S'$ has a graph representation $G'_\ell|S'$, where G'_ℓ is in the form of

$$G'_\ell = \bigoplus_{t=0}^{W} (G_\ell|Z_\ell(t))/Z_\ell(t+1),$$

i.e., G'_ℓ is the disjoint union of graphs $(G_\ell|Z_\ell(t))/Z_\ell(t+1)$ obtained by restriction and contraction. Note that the numbers of vertices and edges in G' are $O(r)$ and n, respectively.

We apply Gabow and Xu's algorithm [23] for the unweighted problem to $\mathbf{M}'_1$ and $\mathbf{M}'_2$ with I' as the initial common independent set. Since I' is (w_1, w_2)-near-optimal, it follows from Lemma 11 that the obtained maximum-cardinality common independent set is (w_1, w_2)-near-optimal in S'. Thus the running time of Unweighted_Matroid_Intersection is $O(\sqrt{r}n \log r)$. Since the reachable set T in the procedure Update_Weight can be found in the end of Gabow and Xu's algorithm, we can perform Update_Weight in linear time. Therefore, we have the following.

Theorem 5 *For two graphic matroids, we can solve the weighted matroid intersection problem exactly in $O(W\sqrt{r}n \log r)$ time, and $(1 - \epsilon)$-approximately in $O(\epsilon^{-1}\sqrt{r}n \log^2 r)$ time.*

5.3 Linear matroids

In the case that $\mathbf{M}_1$ and $\mathbf{M}_2$ are linear, we can use a faster algorithm by Harvey [27] instead of the augmentation-type algorithm. His algorithm is an algebraic one for finding a common base of two linear matroids. We reduce our instance to the problem of finding a common base, that corresponds to a (w_1, w_2)-near-optimal maximum-cardinality common independent set.

We first describe basic properties of a linear matroid $\mathbf{M} = (S, \mathcal{I})$ of rank r. We assume that $\mathbf{M}$ is represented by an $r \times n$ matrix A whose column set is S and row set is denoted by R. We denote by $A[I, J]$ the submatrix consisting of row set I and column set J. For a set X, we denote the complement by $\overline{X}$.

It is known that the restriction and contraction of the linear matroid $\mathbf{M}$ are both linear. Indeed, for a subset $X \subseteq S$, $\mathbf{M}|X$ has the matrix representation $A|X = A[R, X]$. Moreover, taking a nonsingular submatrix of maximum size in $A[R, X]$, denoted by $A[Y, Z]$, we have the matrix representation A/X of the contraction $\mathbf{M}/X$ in the form of

$$A/X = A[\overline{Y}, \overline{X}] - A[\overline{Y}, Z]A[Y, Z]^{-1}A[Y, \overline{X}].$$

The row set of A/X is $\overline{Y} = R \setminus Y$. See e.g., [26] for more details. The direct sum of linear matroids $\mathbf{M}_1$ and $\mathbf{M}_2$ is also linear, whose matrix representation is the block diagonal matrix arranging the two matrices for $\mathbf{M}_1$ and $\mathbf{M}_2$ on the diagonal.

Suppose that we are given a weight function $w\colon S \to \{0, 1, \ldots, W\}$. Then, by Lemma 1(i), $\mathbf{M}^w$ is also linear, and its matrix representation A^w is in the form of

$$A^w = \bigoplus_{t=0}^{W} (A|Z(t))/Z(t+1), \tag{1}$$

where we recall $Z(t) = \{e \in S \mid w(e) \geq t\}$ for $t = 0, \ldots, W + 1$. The size of A^w is the same as A; the ground set of $\mathbf{M}^w$ is S, and the row set of A^w is also R. We denote by $Y(t)$ the set of the nonzero rows in $A^w[R, Z(t)]$ for $t = 0, \ldots, W$. Thus A^w is a block-diagonal matrix whose blocks are $A^w[Y(t) \setminus Y(t+1), Z(t) \setminus Z(t+1)]$ for $t = 0, \ldots, W$, where $Y(W + 1) = \emptyset$. Note that A^w can be computed in $O(nr^{\omega-1})$ time, since this can be obtained by Gaussian elimination (see [26]).

We now go back to the weighted matroid intersection. For $\ell = 1, 2$, let $\mathbf{M}_\ell$ be a linear matroid of rank r_ℓ on S, whose matrix representation is given by an $r_\ell \times n$ matrix A_ℓ with the same field. We also denote by R_ℓ the row set of A_ℓ for $\ell = 1, 2$. Then the following proposition is known in [27].

Proposition 1 *Two linear matroids $\mathbf{M}_1$ and $\mathbf{M}_2$ have a common base if and only if the matrix $N = -A_1 D^{-1} A_2^\top$ is nonsingular, where D is a diagonal matrix of order n such that the set of the diagonal entries is algebraically independent.*

Note that N can be computed in $O(nr^{\omega-1})$ time (see [27]).

Let us consider the procedure Unweighted_Matroid_Intersection. Given a weight-splitting w_1 and w_2 of w, recall $Z_\ell(t) = \{e \in S \mid w_\ell(e) \geq t\}$ for $t = 0, \ldots, W + 1$ and $\ell = 1, 2$. Let $Y_\ell(t)$ be the set of the nonzero rows in $A_\ell^{w_\ell}[R_\ell, Z_\ell(t)]$ for $t = 0, \ldots, W$. For a subset S', let $Z'_\ell(t) = Z_\ell(t) \cap S'$.

Lemma 12 *For two linear matroids, suppose that I' is (w_1, w_2)-near-optimal in a subset S'. Then we can construct a common independent set I, in $O(nr^{\omega-1})$ time, that is simultaneously (i) a maximum-cardinality common independent set of $\mathbf{M}_1'$ and $\mathbf{M}_2'$, and (ii) (w_1, w_2)-near-optimal in S'.*

Proof We denote $A_\ell' = A_\ell^{w_\ell}[R_\ell, S']$, which is a matrix representation of $\mathbf{M}_\ell'|S'$, for $\ell = 1, 2$. We first show the following claim on (w_1, w_2)-near-optimality.

Claim A set J is (w_1, w_2)-near-optimal in a subset S' if and only if there exists $U_\ell \subseteq R_\ell$ ($\ell = 1, 2$) with $Y_1(1) \subseteq U_1$ and $Y_2(p+1) \subseteq U_2$, where $p = \min_{e \in S'} w_2(e)$, such that J is a common base of $A_1'[U_1, S']$ and $A_2'[U_2, S']$.

Proof Suppose J is (w_1, w_2)-near-optimal in S'. Then it follows from (1) that $J \cap Z_1'(t)$ is a base of $\mathbf{M}_1|Z_1'(t)$ for every integer $t \geq 1$ if and only if each submatrix $A_1'[Y_1(t), J \cap Z_1'(t)]$ is nonsingular for every integer $t \geq 1$. Since $A_1'[Y_1(1), J \cap Z_1'(1)]$ is nonsingular, we can take $U_1 \subseteq R_1$ with $|U_1| = |J|$ such that $A_1'[U_1, J]$ is nonsingular and $Y_1(1) \subseteq U_1$. Similarly, there exists $U_2 \subseteq R_2$ with $|U_2| = |J|$ such that $A_2'[U_2, J]$ is nonsingular and $Y_2(p+1) \subseteq U_2$. Thus J is a common base of $A_1'[U_1, S']$ and $A_2'[U_2, S']$.

Conversely, suppose that we have row subsets U_1 and U_2 satisfying the conditions. Since $A_1'[R_1 \setminus Y_1(1), Z_1'(1)]$ is a zero matrix, the base J has a nonsingular submatrix $A_1'[Y_1(1), J \cap Z_1'(1)]$. Since the submatrix is block-diagonal, this is equivalent to that $J \cap Z_1'(t)$ is a base of $\mathbf{M}_1'|S'$ for every $t \geq 1$. The case for A_2' is analogous. □

By the assumption that I' is (w_1, w_2)-near-optimal in S', there exist U_1 and U_2 such that $Y_1(1) \subseteq U_1$, $Y_2(p+1) \subseteq U_2$, and $A_1'[U_1, S']$ and $A_2'[U_2, S']$ have a common base. Among such U_1 and U_2, we take U_1^* and U_2^* with maximum size. We can find a common base I for $A_1'[U_1^*, S']$ and $A_2'[U_2^*, S']$ by Harvey's algorithm in $O(nr^{\omega-1})$ time [27]. Since I satisfies the conditions of the above claim, I is (w_1, w_2)-near-optimal with maximum size in S'. Thus I is a desired set.

It remains to show that we can find such maximum U_1^* and U_2^* in $O(nr^{\omega-1})$ time. Construct $N = -A_1' D^{-1} {A'}_2^\top$ in $O(nr^{\omega-1})$ time, which has the row set R_1 and column set R_2. By Proposition 1, for $U_1 \subseteq R_1$ and $U_2 \subseteq R_2$, both $A_1'[U_1, S']$ and $A_2'[U_2, S']$ have a common base if and only if $N[U_1, U_2]$ is nonsingular, which follows from the fact that $N[U_1, U_2] = -A_1'[U_1, S'] D^{-1} ({A'}_2[U_2, S'])^\top$. Therefore, it suffices to find $U_1^* \subseteq R_1$ and $U_2^* \subseteq R_2$ with maximum size such that $Y_1(1) \subseteq U_1^*$ and $Y_2(p+1) \subseteq U_2^*$ and $N[U_1^*, U_2^*]$ is nonsingular. This can be done in $O(r^\omega)$ time, since the rank of N is at most r. □

Since Update_Weight can be performed in $O(nr^{\omega-1})$ time, we can solve the weighted matroid intersection exactly in $O(Wnr^{\omega-1})$ time and approximately in $O(\epsilon^{-1} nr^{\omega-1} \log r)$ time.

Furthermore, using a preprocessing technique by Cheung et al. [4], we can improve the computational time. Given a positive integer k, their algorithm reduces an $r \times n$ matrix A to an $O(k) \times n$ matrix A' such that, if a column set in A' of size at most k is independent, then it is independent in A with high probability. This can be done in $O(nr)$ time.

We simply use this algorithm where k is set to be the maximum size $r_* \leq r$ of a common independent set of $\mathbf{M}_1$ and $\mathbf{M}_2$. The size r_* can be computed in

$O(nr \log r_* + nr_*^{\omega-1})$ time [4]. After we obtain two $O(r_*) \times n$ matrices by their method, apply our algorithm to obtain a maximum-weight common independent set. This takes $O(Wnr_*^{\omega-1})$ time for an exact algorithm and $O(\epsilon^{-1}nr_*^{\omega-1} \log r_*)$ time for an approximation algorithm.

Therefore, we have the following theorem.

Theorem 6 *For two linear matroids, we can solve the weighted matroid intersection exactly in $O(nr \log r_* + Wnr_*^{\omega-1})$ time and $(1-\epsilon)$-approximately in $O(nr \log r_* + \epsilon^{-1}nr_*^{\omega-1} \log r_*)$ time, where r_* is the size of a common independent set.*

It should be noted that our algorithm is simple in the sense that it involves only a constant matrix and does not need to manipulate a univariate-polynomial matrix [50], unlike the algorithms in [26,46].

6 Rank-maximal matroid intersection

In this section, we deal with the rank-maximal matroid intersection problem. As mentioned in the introduction, this problem can be reduced to a weighted matroid intersection problem whose weight w is drawn from $\{1, n, n^2, \ldots, n^{R-1}\}$. More generally, we consider the case where the weight w is drawn from a geometric series $\{1, u, u^2, \ldots, u^{R-1}\}$, where $u \geq 2$. Let d_k be the difference of two consecutive weights, i.e., $d_k = u^k - u^{k-1}$ for $k = R, R-1, \ldots, 1$. For convenience, we also define $d_0 = 1$.

Our algorithm for the geometric-series weight case is described as follows, where the only difference from our exact algorithm (Algorithm 1 in Sect. 3) is in the dual update Step (2–4): we update w_1 and w_2 with large weight d_k.

Algorithm 3: Exact algorithm for geometric-series weights

Input: two matroids $\mathbf{M}_1 = (S, \mathcal{I}_1)$ and $\mathbf{M}_2 = (S, \mathcal{I}_2)$, a weight function $w\colon S \to \{1, u, u^2, \cdots, u^{R-1}\}$, where $u \geq 2$.

Output: $I \in \mathcal{I}_1 \cap \mathcal{I}_2$ where I is a maximum-weight common independent set of $\mathbf{M}_1$ and $\mathbf{M}_2$.

Step 1. Set $k := R-1$, $w_1 := 0$, $w_2 := w$, and $I' := \emptyset$.

Step 2. While $k > 0$ do the following steps.

- **(2-1)** Set $S' := \{e \in S \mid w_2(e) \geq u^k\}$.
- **(2-2)** Set $\mathbf{M}'_\ell := \mathbf{M}_\ell^{w_\ell} | S'$ for each $\ell = 1, 2$.
- **(2-3)** Unweighted_Matroid_Intersection (I')
  - Construct I so that
  - (i) I is a maximum-cardinality common independent set of $\mathbf{M}'_1$ and $\mathbf{M}'_2$, and
  - (ii) I is (w_1, w_2)-near-optimal in S'.
- **(2-4)** Update_Weight
  - **(2-4-1)** Let $T \subseteq S'$ be the set of elements reachable from X_2 in $G_{\mathbf{M}'_1, \mathbf{M}'_2}(I)$.
  - **(2-4-2)** For each $e \in T$, let $w_1(e) := w_1(e) + d_k$, $w_2(e) := w_2(e) - d_k$.
- **(2-5)** Set $k := k - 1$ and $I' := I$.

Step 3. Return I.

In the following, we prove the correctness of Algorithm 3 by showing that it would have the same outcome as if we had run Algorithm 1 in Sect. 3 instead. For the purpose, we first need a technical lemma, whose proof exploits the property that each weight is at least double the previous weight in the given geometric series of weights w.

Lemma 13 *At Step* (2–1) *of Round* k ($k = R-1, R-2, \ldots, 1$)*, we have the following for* e *and* f *in* S'.

(i) *If* $w_1(e) \neq w_1(f)$*, then* $|w_1(e) - w_1(f)| \geq d_{k+1}$.
(ii) *If* $w_2(e) \neq w_2(f)$*, then* $|w_2(e) - w_2(f)| \geq d_{k+1}$.

Proof We prove (i) by induction on k. When $k = R-1$, all elements have w_1-weights 0. So (i) holds trivially. For the induction step with $k < R-1$, if $w_1(e) \neq w_1(f)$, then at least one of them, say e, is part of S' in the last round (i.e., Round $k+1$). To avoid confusion, the set S' in the last round is denoted by $\overline{S}'$. Also w_1 and w_2 at Step (2–1) of the last round are denoted by $\overline{w}_1$ and $\overline{w}_2$, respectively. Consider two possibilities.

Case 1 Suppose that $f \in \overline{S}'$. By induction hypothesis, either $|\overline{w}_1(e) - \overline{w}_1(f)| \geq d_{k+2}$, or $\overline{w}_1(e) = \overline{w}_1(f)$. In the former case, the difference between the w_1-weights of e and f is changed by at most d_{k+1} in the last round. Therefore, we have

$$|w_1(e) - w_1(f)| \geq d_{k+2} - d_{k+1} \geq d_{k+1},$$

where the last inequality holds because $u \geq 2$. In the latter case, either $w_1(e) = w_1(f)$ (if both $\overline{w}_1(e)$ and $\overline{w}_1(f)$ are updated or unchanged in the last round), or $|w_1(e) - w_1(f)| = d_{k+1}$ (if exactly one of them is updated).

Case 2 Suppose that $f \notin \overline{S}'$. Then $w_1(f) = 0$. If $w_1(e)$ has not been updated so far, then $w_1(e) = 0$. Otherwise, since $w_1(e)$ is increased at Round s for some $s \geq k+1$, we have $w_1(e) \geq d_s \geq d_{k+1}$. The induction step is completed.

(ii) can be proved symmetrically. □

To avoid confusion, let i be the index of the rounds when we apply Algorithm 1 in Sect. 3, and J^i be the independent set obtained in Round i. Let k be the index of the rounds when we apply Algorithm 3, and I^k be the independent set obtained at Round k of Algorithm 3.

Lemma 14 *Define* $i_k = u^k$ *for* $k = R-1, R-2, \ldots, 1$*. For* $k = R-1, R-2, \ldots, 1$ *and* $\ell = 1, 2$*, the weights* w_ℓ *at Round* i_k *of Algorithm 1 are the same as the weights* w_ℓ *at Round* k *of Algorithm 3. Thus, for* $k = R-1, R-2, \ldots, 1$*, the auxiliary graph in Round* k *of Algorithm 3 coincides with one in Round* i_k *of Algorithm 1.*

Proof We prove by induction in k. When $k = R-1$ and $i_k = u^{R-1}$, the lemma holds easily. For the induction step when $k < R-1$, we argue that the update of the w_1- and w_2-weights done in Round $k+1$ of Algorithm 3 are the same as the accumulated updates of the w_1- and w_2-weights done in Algorithm 1 from Round i_{k+1} down to Round $i_k + 1$. To be more precise, in Round $k+1$ in Algorithm 3, all elements in $S' \cap T$ have their w_2-weights decrease by the amount of d_{k+1} and their w_1-weights increase by the same amount. We show that from Round i_{k+1} down to Round $i_k + 1$ in Algorithm 1, the same set of elements have their w_2- and w_1-weights updated (and each round by the amount of one). This would prove the induction step.

For simplicity, we often denote $i_{k+1}-t$ with (t) for $t=0,1,\ldots,d_{k+1}-1$. Let $G^{(t)}$ be the auxiliary graph at Round $i_{k+1}-t$ in Algorithm 1, and $T^{(t)}$ be the reachable set in $G^{(t)}$ found in Step (2–4) of Round $i_{k+1}-t$. We will show the following properties for Algorithm 1.

(1) The ground set $S^{(t)}$ is the same as $S^{(0)}$.
(2) The reachable set $T^{(t)}$ is the same as $T^{(0)}$.
(3) The independent set $J^{(t)}$ is the same as $J^{(0)}$.

To see (1), observe that, in Algorithm 1, all elements e not in $S^{(0)}$ have $w_2^{(0)}(e) \le u^{k+1}-1$. Since $w_2^{(0)}(e)$ is equal to $w(e)$, we see $w_2^{(0)}(e) \le u^k$ for $e \notin S^{(0)}$ by the definition of w. Hence e is not contained in $S^{(t)}$, as $w_2^{(t)}(e) = w_2^{(0)}(e) \le u^k < u^{k+1}-t$ for $0 \le t \le d_{k+1}-1$. Therefore, $S^{(0)} = S^{(t)}$.

To see (2), we first show $T^{(t)} \supseteq T^{(0)}$. Note that, by Lemma 1(iii), for an arc ef in $G^{(0)}$, their w_1-weights $w_1^{(0)}(e)$ and $w_1^{(0)}(f)$ must be the same if $e \notin J^{(0)}$ and $f \in J^{(0)}$, and their w_2-weights $w_2^{(0)}(e)$ and $w_2^{(0)}(f)$ must be the same if $e \in J^{(0)}$ and $f \notin J^{(0)}$. Then, if both e and f are in $T^{(0)}$ or neither of them is in $T^{(0)}$, their w_2-weights (respectively w_1-weights) remain the same in the subsequent rounds, and hence the arc ef appears in $G^{(t)}$. Thus $T^{(t)} \supseteq T^{(0)}$.

To prove $T^{(t)} \subseteq T^{(0)}$, it suffices to show that, in the auxiliary graph $G^{(t)}$, there exists no new arc from an element e in $T^{(0)}$ to an element f not in $T^{(0)}$. Note that, by the definition of $T^{(0)}$, there exists no arc from e to f in $G^{(0)}$.

First suppose that $e \notin J^{(0)}$ and $f \in J^{(0)}$. Then $w_1^{(0)}(e) < w_1^{(0)}(f)$ holds if the arc ef appeared in $G^{(t)}$. It follows from the induction hypothesis of the lemma that $G^{(0)}$ coincides with $G_{\mathbf{M}_1',\mathbf{M}_2'}(I^{k+1})$ at Round $k+1$ of Algorithm 3. This implies by Lemma 13 that $w_1^{(0)}(f) - w_1^{(0)}(e) \ge d_{k+2}$. Hence, for any $t=0,1,\ldots,d_{k+1}-1$, it holds that

$$\begin{aligned} w_1^{(t)}(f) - w_1^{(t)}(e) &\ge w_1^{(0)}(f) - (w_1^{(0)}(e) + t) \\ &\ge d_{k+2} - d_{k+1} + 1 > 0, \end{aligned}$$

where the last inequality follows from $u \ge 2$. Therefore, we always have $w_1^{(t)}(f) > w_1^{(t)}(e)$ for $0 \le t \le d_{k+1}-1$, and thus the arc ef never appears in $G^{(t)}$. Similarly, if $e \in J^{(0)}$ and $f \notin J^{(0)}$, then $w_2^{(0)}(e) - w_2^{(0)}(f) \ge d_{k+2}$ by Lemma 13. Hence we have $w_2^{(t)}(e) - w_2^{(t)}(f) > 0$ for $t=0,1,\ldots,d_{k+1}-1$. Therefore, (2) follows.

Finally, (3) follows from the fact that in each round, there is no augmentation happening, as $G^{(0)}$ has no augmenting path and by (1) and (2) neither does $G^{(t)}$. This completes the proof. □

Theorem 7 *The independent set I returned by Algorithm 3 is an optimal solution.*

Therefore, using Cunningham's algorithm for the unweighted matroid intersection problem as a subroutine, we have the following theorem.

Theorem 8 *The rank-maximal matroid intersection problem can be solved using $O(Rnr^{1.5})$ independence oracle calls.*

We note that even though the actual weights used in Algorithm 3 can be exponentially large, there is indeed no need to store them explicitly (otherwise, we incur the extra cost in time and space). We discuss in Sect. 6.1 the implementations details.

In addition, if the given two matroids are both graphic or both linear, then we can solve the problem fast.

Theorem 9 *The rank-maximal graphic matroid intersection problem can be solved in $O(R\sqrt{r}n\log r)$ time.*

Theorem 10 *The rank-maximal linear matroid intersection problem can be solved in $O(Rnr^{\omega-1})$ time.*

6.1 Implementation of rank-maximal matroid intersection

We discuss how to avoid storing the actual numerical values of the weights when implementing Algorithm 3. As discussed in Sect. 3, we need to draw an arc in the auxiliary graph properly and to do this, we just need to know that given any two elements e and f in S', whether $w_\ell(e)$ is larger than, equivalent to, or smaller than $w_\ell(f)$ for $\ell = 1, 2$. There is no need to know the actual values.

It is easy to see that, at Round k, we have $w_1(e) = 0 + \sum_{t=k+1}^{R-1} d_t \cdot \mathbf{1}_{e\in T^t}$ and $w_2(e) = w(e) - \sum_{t=k+1}^{R-1} d_t \cdot \mathbf{1}_{e\in T^t}$, where T^t is the reachable set found in Step (2-4) of Round t. Using this fact, Lemma 13, and the definition of d_k, we have

Lemma 15 *At Step (2–1) of Round k ($k = R-1, R-2, \ldots, 1$), we have the following for e and f in S'.*

(i) *If $w_1(e) > w_1(f)$, then, in Round $h = k-1, k-2, \ldots, 1$, $w_1(e) > w_1(f)$.*
(ii) *If $w_2(e) > w_2(f)$, then, in Round $h = k-1, k-2, \ldots, 1$, $w_2(e) > w_2(f)$.*

Below we only discuss how to compare the w_1-weight of the elements, since the w_2-weight can be handled symmetrically.

In Round k, we can divide the elements according to their weights w_1. There can be only $O(n)$ such groups, $g^k(1), g^k(2), \ldots$, ordered by their increasing weights. Inside each group $g^k(i)$, in the next round (Round $k-1$), the elements can be split into two subgroups, if a strict subset of the elements in $g^k(i)$ belongs to the reachable set T^k in Step (2–4). However, by Lemma 15, in the next round, we know that elements belonging to these two subgroups of $g^k(i)$ will still have weights w_1 smaller than the elements from the subgroups derived from $g^k(i+1), g^k(i+2), \ldots$, and larger than the elements from the subgroups derived from $g^k(i-1), g^k(i-2), \ldots$. Finally, for the elements e newly-added in Round $k-1$, we have $w_1(e) = 0$. These elements can be either added into the existing group with the smallest weight, or we can create a new group for them (and such a group necessarily has the smallest weight). The maintenance of such data structure as described can be easily done in $O(n)$ time in each round of the algorithm.

7 Relation to other algorithms

For both the weight matching and the weighted matroid intersection problem, when the largest weight is not overly large, faster algorithms have been designed to leverage this fact. The more well-known technique is that of scaling [12,19–21,24,49]. In each phase, the weights are rounded to speed up the computation; the weight rounding becomes more fine-grained in each successive round. In contrast, in [28,33,43] and our present work, there is no rounding of weights. In each phase, only a subset of the edges (the ground set) is considered. Such subsets enlarge over the phases. We mentioned in passing that the recent result of Lee–Sidford–Wong [38] is a far department from the previous two general techniques, where a novel cutting-plane method is employed.

Since the weighted bipartite matching problem is a special case of the weighted matroid intersection problem, we can apply our exact algorithm to the special case: in fact our algorithm would behave similarly to the one by Kao et al. [33] with the same running time, though the data structures used are different.

It may be worthwhile contrasting our exact algorithm with the algorithm of Frank [16] for two general matroids. He uses the weights w_1 and w_2 to "suppress" some edges in the original auxiliary graph. It can be shown that the modified auxiliary graph in his algorithm would be identical to the auxiliary graph of our matroids defined in each round. He augments the current independent set I repeatedly in the modified auxiliary graph, preserving the condition that I is a maximum-weight common independent set with size $|I|$. On the other hand, our algorithm only maintains the relaxed optimality condition, and dramatically augments I with the aid of unweighted matroid intersection algorithms.

We mentioned earlier that Chekuri and Quanrud [3] have further improved the running time of our approximation algorithm for general matroids. Their speeding-up is achieved by a more sophisticated weight adjustment. In particular, in Step 2 of Algorithm 2, instead of finding a maximum-cardinality common independent set as we have done (this takes $O(nr^{1.5}\tau)$ time), they only compute a common independent set whose size $(1-\epsilon)$-approximates the former (thus they only need $\tilde{O}(nr\epsilon^{-1}\tau)$ time).

References

1. Aigner, M., Dowling, T.A.: Matching theory for combinatorial geometries. Trans. Am. Math. Soc. **158**(1), 231–245 (1971)
2. Brezovec, C., Cornuéjols, G., Glover, F.: Two algorithms for weighted matroid intersection. Math. Program. **36**(1), 39–53 (1986)
3. Chekuri, C., Quanrud, K.: Fast approximations for matroid intersection. In: Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA (2016)
4. Cheung, H.Y., Kwok, T.C., Lau, L.C.: Fast matrix rank algorithms and applications. J. ACM **60**(5), 31 (2013)
5. Cheung, H.Y., Lau, L.C., Leung, K.M.: Algebraic algorithms for linear matroid parity problems. ACM Trans. Algorithms **10**(3), 10 (2014)
6. Christiano, P., Kelner, J.A., Madry, A., Spielman, D.A., Teng, S.: Electrical flows, Laplacian systems, and faster approximation of maximum flow in undirected graphs. In: Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC, pp. 273–282 (2011)
7. Cook, W.J., Cunningham, W.H., Pulleyblank, W.R., Schrijver, A.: Combinatorial Optimization. Wiley, Hoboken (1997)

8. Coppersmith, D., Winograd, S.: Matrix multiplication via arithmetic progressions. J. Symb. Comput. **9**(3), 251–280 (1990)
9. Cunningham, W.H.: Improved bounds for matroid partition and intersection algorithms. SIAM J. Comput. **15**(4), 948–957 (1986)
10. Dougherty, R., Freiling, C., Zeger, K.: Network coding and matroid theory. Proc. IEEE **99**(3), 388–405 (2011)
11. Duan, R., Pettie, S.: Linear-time approximation for maximum weight matching. J. ACM **61**(1), 1 (2014)
12. Duan, R., Pettie, S., Su, H.-H.: Scaling algorithms for weighted matching in general graphs. In: Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 781–800 (2017)
13. Edmonds, J.: Submodular functions, matroids, and certain polyhedra. In: Guy, R., Hanani, H., Sauer, N., Schönheim, J. (eds.) Combinatorial Structures and Their Applications, pp. 69–87. Springer, London (1970)
14. Edmonds, J.: Matroids and the greedy algorithm. Math. Program. **1**(1), 127–136 (1971)
15. Edmonds, J.: Matroid intersection. Ann. Discret. Math. **4**, 39–49 (1979)
16. Frank, A.: A weighted matroid intersection algorithm. J. Algorithms **2**(4), 328–336 (1981)
17. Frank, A.: A quick proof for the matroid intersection weight-splitting theorem. Technical Report QP-2008-03, Egerváry Research Group on Combinatorial Optimization (2008)
18. Fujishige, S.: Submodular Functions and Optimization, 2nd edn. Elsevier, Amsterdam (2005)
19. Fujishige, S., Zhang, X.: An efficient cost scaling algorithm for the independent assignment problem. J. Oper. Res. Soc. Jpn. **38**(1), 124–136 (1995)
20. Gabow, H., Tarjan, R.: Faster scaling algorithms for network problems. SIAM J. Comput. **18**(5), 1013–1036 (1989)
21. Gabow, H., Tarjan, R.: Faster scaling algorithms for general graph-matching problems. J. ACM **38**(4), 815–853 (1991)
22. Gabow, H.N., Stallmann, M.F.M.: Efficient algorithms for graphic matroid intersection and parity (extended abstract). In: Proceedings of the 12th Colloquium on Automata, Languages and Programming, ICALP, pp. 210–220 (1985)
23. Gabow, H.N., Xu, Y.: Efficient algorithms for independent assignments on graphic and linear matroids. In: Proceedings of the 30th Annual Symposium on Foundations of Computer Science, FOCS, pp. 106–111 (1989)
24. Gabow, H.N., Xu, Y.: Efficient theoretic and practical algorithms for linear matroid intersection problems. J. Comput. Syst. Sci. **53**(1), 129–147 (1996)
25. Gall, F.L.: Powers of tensors and fast matrix multiplication. In: Proceedings of the International Symposium on Symbolic and Algebraic Computation, ISSAC, pp. 296–303 (2014)
26. Harvey, N.J.A.: An algebraic algorithm for weighted linear matroid intersection. In: Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, pp. 444–453 (2007)
27. Harvey, N.J.A.: Algebraic algorithms for matching and matroid problems. SIAM J. Comput. **39**(2), 679–702 (2009)
28. Huang, C.-C., Kavitha, T.: New algorithms for maximum weight matching and a decomposition theorem. Math. Oper. Res. **42**(2), 411–426 (2017)
29. Iri, M.: Applications of matroid theory. In: Mathematical Programming—The State of the Art, pp. 158–201 (1983)
30. Iri, M., Tomizawa, N.: An algorithm for finding an optimal "independent assignment". J. Oper. Res. Soc. Jpn. **19**(1), 32–57 (1976)
31. Irving, R.W., Kavitha, T., Mehlhorn, K., Michail, D., Paluch, K.E.: Rank-maximal matchings. ACM Trans. Algorithms **2**(4), 602–610 (2006)
32. Jenkyns, T.A.: The efficacy of the "greedy" algorithm. In: Proceedings of the 7th Southeastern International Conference on Combinatorics, Graph Theory, and Computing, pp. 341–350 (1976)
33. Kao, M.-Y., Lam, T.W., Sung, W.-K., Ting, H.-F.: A decomposition theorem for maximum weight bipartite matchings. SIAM J. Comput. **31**(1), 18–26 (2001)
34. Kelner, J.A., Lee, Y.T., Orecchia, L., Sidford, A.: An almost-linear-time algorithm for approximate max flow in undirected graphs, and its multicommodity generalizations. In: Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, pp. 217–226 (2014)
35. Korte, B., Hausmann, D.: An analysis of the greedy heuristic for independence systems. In: Alspach, P.H.B., Miller, D. (eds.) Algorithmic Aspects of Combinatorics, Volume 2 of Annals of Discrete Mathematics, pp. 65–74. Elsevier, Amsterdam (1978)

36. Lawler, E.L.: Optimal matroid intersections. In: Guy, R., Hanani, H., Sauer, N., Schönheim, J. (eds.) Combinatorial Structures and Their Applications, pp. 233–234. Gordon and Breach, London (1970)
37. Lawler, E.L.: Matroid intersection algorithms. Math. Program. **9**(1), 31–56 (1975)
38. Lee, Y., Sidford, A., Wong , S.C.-W.: A faster cutting plane method and its implications for combinatorial and convex optimization. In: Proceedings of the 56th Annual Symposium on Foundations of Computer Science, FOCS (2015)
39. Lee, Y.T., Rao, S., Srivastava, N.: A new approach to computing maximum flows using electrical flows. In: Proceedings of the 45th Symposium on Theory of Computing Conference, STOC, pp. 755–764 (2013)
40. Murota, K.: Matrices and Matroids for Systems Analysis, 2nd edn. Springer, Berlin (2000)
41. Murota, K., Iri, M., Nakamura, M.: Combinatorial canonical form of layered mixed matrices and its applications to block-triangularization of systems of linear/nonlinear equations. SIAM J. Algebraic Discret. Methods **8**, 123–149 (1987)
42. Pap, G.: A matroid intersection algorithm. Technical Report TR-2008-10, Egerváry Research Group on Combinatorial Optimization (2008)
43. Pettie, S.: A simple reduction from maximum weight matching to maximum cardinality matching. Inf. Process. Lett. **112**(23), 893–898 (2012)
44. Recski, A.: Terminal solvability and n-port interconnection problem. In: IEEE International Symposium on Circuits and Systems, pp. 988–991 (1979)
45. Recski, A.: Matroid Theory and its Applications in Electric Network Theory and in Statics. Springer, Berlin (1989)
46. Sankowski, P.: Maximum weight bipartite matching in matrix multiplication time. Theor. Comput. Sci. **410**(44), 4480–4488 (2009)
47. Schrijver, A.: Combinatorial Optimization: Polyhedra and Efficiency. Springer, Berlin (2003)
48. Sherman, J.: Nearly maximum flows in nearly linear time. In: Proceedings of the 54th Annual Symposium on Foundations of Computer Science, FOCS (2013)
49. Shigeno, M., Iwata, S.: A dual approximation approach to weighted matroid intersection. Oper. Res. Lett. **18**(3), 153–156 (1995)
50. Storjohann, A.: High-order lifting and integrality certification. J. Symb. Comput. **36**, 613–648 (2003)
51. Thorup, M., Zwick, U.: Approximate distance oracles. J. ACM **52**(1), 1–24 (2005)
52. Williams, V.V.: Multiplying matrices faster than Coppersmith–Winograd. In: Proceedings of the 44th Annual ACM Symposium on Theory of Computing, STOC, pp. 887–898 (2012)