An Algebraic Algorithm for Weighted Linear Matroid Intersection

Nicholas J. A. Harvey*
Massachusetts Institute of Technology
nickh@mit.edu

Abstract

We present a new algebraic algorithm for the classical problem of weighted matroid intersection. This problem generalizes numerous well-known problems, such as bipartite matching, network flow, etc. Our algorithm has running time $\tilde{O}(nr^{\omega-1}W^{1+\epsilon})$ for linear matroids with n elements and rank r, where ω is the matrix multiplication exponent, and W denotes the maximum weight of any element. This algorithm is the fastest known when W is small. Our approach builds on the recent work of Sankowski (2006) for Weighted Bipartite Matching and Harvey (2006) for Unweighted Linear Matroid Intersection.

1 Introduction

The matroid intersection problem — finding a maximum cardinality (or maximum weight) independent set in two given matroids — is a classical optimization problem initially studied in the early 1970s by several authors [1, 5, 7, 21]. This work led to significant developments concerning integral polyhedra [28], submodular functions [10], electrical networks [17], convex analysis [25], etc. The matroid intersection problem generalizes numerous well-known problems, such as weighted bipartite matching, min-cost network flow, packing spanning trees, etc. Efficient algorithms are known for solving matroid intersection, and these algorithms have found applications in areas such as approximation algorithms [3, 13, 16], mixed matrix theory [24], and network coding [15].

The efficiency of matroid algorithms is often measured relative to an oracle which answers queries about the matroids. The weighted matroid intersection problem can be solved in strongly polynomial time, even in an oracle model. The most efficient strongly polynomial algorithm is due to Brezovec et al. [2]; it uses $O(nr)$ oracle calls and has running time $\tilde{O}(nr^2)$, where n is the size of the ground set, and the matroids have rank r. (These terms are defined in Section 2.) For an important class of matroids, known as linear matroids, one can devise more efficient algorithms that work directly on a matrix rather than via an oracle. Gabow and Xu [12] devised an algorithm for linear matroids which uses fast matrix multiplication and obtains a running time of $\tilde{O}(nr^{1.77}\log W)$. The exponent 1.77 is due to a parameter-balancing step in their analysis, and one is tempted to conjecture that it could be improved. For example, if $n = r$, their bound is $\tilde{O}(n^{2.77}\log W)$, although $O(n^{\omega}\log W)$ seems to be a more natural bound. Throughout this paper, $\omega < 2.38$ denotes a value such that two $n \times n$ matrices can be multiplied in time $O(n^{\omega})$.

For the *unweighted* linear matroid intersection problem, somewhat faster algorithms are known. Gabow and Xu show that the unweighted version of their algorithm has runtime $O(nr^{1.62})$. The present author recently obtained [14] an algorithm[1] that has runtime $\tilde{O}(nr^{\omega-1}) = \tilde{O}(nr^{1.38})$. The latter algorithm uses randomized algebraic techniques that have been fruitful in other recent work [22, 26, 27]. Until recently, it was not known how to apply these algebraic techniques efficiently to weighted problems. The difficulty can be understood by considering two classical methods for weighted combinatorial optimization problems.

- *Primal-dual methods.* These typically intersperse primal updates with dual updates. In contrast, algebraic methods tend to be efficient when many primal updates can be performed in a large batch. Thus primal-dual and algebraic methods seem largely incompatible.
- *Primal augmentation methods.* These methods typically guarantee optimality of the primal solution while carefully increasing its size (e.g., using shortest augmenting paths). It is difficult[2] to augment the primal in large batches while maintaining optimality, and thus it is difficult to apply algebraic techniques.

*Supported by a Natural Sciences and Engineering Research Council of Canada PGS Scholarship, by NSF contract CCF-0515221 and by ONR grant N00014-05-1-0148. This work was performed while visiting the University of Waterloo and Microsoft Research.

[1] Our unweighted algorithm [14] requires an addition technical assumption: the two given matroids must be represented over the same field. This assumption cannot be satisfied for all matroids, but it is satisfied for the large classes of matroids that typically arise in applications.

[2] The notion of 1-optimality has been useful for this task [11].

Traditionally, the most successful way to use algebraic methods for weighted optimization problems is to represent the weights using polynomials. This method has been used in classical papers, e.g., the work of Karp et al. [19] and Mulmuley et al. [23] on parallel matching algorithms. Working with large polynomials is very efficient in a parallel setting because the polynomial manipulations are easily parallelized. It is significantly more challenging to manipulate polynomials efficiently in a sequential setting. Naively extending an algorithm to work with univariate polynomials instead of numbers typically decreases its performance by a factor linear in the polynomials' degree. This increase in running time is usually prohibitive.

Fortunately, a powerful algorithmic tool was recently developed for computing with matrices whose entries are univariate polynomials. This tool, due to Storjohann [29], allows one to compute the determinant of an $n \times n$ matrix A whose entries are univariate polynomials of degree W in time $\tilde{O}(n^\omega W^{1+\epsilon})$. In fact, Storjohann shows that the same running time suffices to solve a linear system $Ax = b$ for x. (We will use this more general algorithm crucially in Section 6.2.)

While Storjohann's tool is certainly powerful, it is not immediately clear how it can be used in algebraic algorithms for combinatorial optimization problems. For example, the recent unweighted algorithms [22, 14] must compute a matrix inverse (which is more difficult than solving a linear system), and furthermore must sequentially apply numerous updates to this inverse matrix. So it is not obvious whether Storjohann's tool can be applied in this setting.

1.1 Sankowski's Approach Sankowski [27] recently showed that Storjohann's tool can indeed be used to solve the weighted bipartite matching problem efficiently. His method relies on some nice insights concerning the dual problem (weighted vertex cover). We now briefly summarize his method.

(S1): Given an optimum dual solution, one can compute an optimum primal solution using an *un*weighted matching algorithm in $O(n^\omega)$ time [22].

(S2): Finding an optimum dual amounts to solving a sequence of shortest path computations.

(S3): Each shortest path computation can be performed by solving a weighted bipartite matching problem on a certain *perturbed graph*, which is slightly different than the original graph. Here, one need not construct an optimum solution to the matching problems, only the optimum weight.

(S4): Storjohann's determinant algorithm easily allows one to compute the *weight* of an optimum solution. In fact, Storjohann's algorithm can simultaneously compute the optimum weight for numerous perturbed graphs.

Using these ideas, Sankowski shows that $\tilde{O}(n^\omega W^{1+\epsilon})$ time suffices to compute a maximum weight bipartite matching on a graph with n vertices.

1.2 Our Results While Sankowski's approach is quite clever, it is unfortunately difficult to apply to other problems. For example, it is still unknown whether the approach can be applied to non-bipartite matching, a natural extension of bipartite matching.

This paper extends Sankowski's approach in a different direction: we consider the matroid intersection problem, which is another natural extension of bipartite matching. We give a new algorithm which solves weighted matroid intersection in time $\tilde{O}(nr^{\omega-1} W^{1+\epsilon})$, where n is the size of the ground set, r is the maximum rank of the two matroids, the weights are non-negative integers, and W is their maximum value. The matroids must be linear and represented over the same field. Our result gives the first improvement of the $\tilde{O}(nr^{1.77} \log W)$ bound of Gabow and Xu mentioned above, assuming that W is small, say $W = O(r^{0.38})$.

Our algorithm is based on Sankowski's method mentioned in Section 1.1. Applying this method to the matroid intersection problem is non-trivial, and there are several technical obstacles that we must overcome.

(S1): Given an optimum dual solution, how can one construct an optimum primal solution? This step is more difficult for matroid intersection than for bipartite matching because the duality framework for matroid intersection is much more complicated. We resolve this difficulty using some classical matroid lemmas, and give an efficient algorithm based on fast matrix multiplication techniques.

(S3): How can one construct an optimum dual solution using perturbed instances? This step is more difficult for matroid intersection than for bipartite matching because when one augments or modifies a matroid intersection solution, one must take care to ensure than independence is not violated. We resolve this difficulty with a mixture of new and classical technical lemmas.

(Sparsity): The algebraic approach for matroid intersection described in [14] involves computations on certain sparse matrices. However, Storjohann's tool assumes that matrices are dense, so it is unclear whether this tool can yield the desired level of efficiency. We resolve this difficulty first by reformulating the matroid intersection problem as an "independent assignment problem", then by employing some linear algebra tricks which enable the use of Storjohann's tool.

2 Preliminaries

If S is a set, $S + t$ denotes $S \cup \{t\}$. If M is a matrix, a submatrix containing rows S and columns T is denoted $M[S, T]$. A submatrix containing all rows is denoted $M[*, T]$. The i^{th} row (column) of M is denoted $M_{i,*}$ ($M_{*,i}$). An entry of M is denoted $M_{i,j}$. If x is a vector, its i^{th} component is denoted x_i or $x(i)$. If w is a vector indexed by S and $T \subseteq S$ then $w(T) := \sum_{s \in T} w(s)$.

2.1 Matroids A *matroid* is a combinatorial object defined on a ground set S. There are several important ancillary objects relating to matroids, any one of which suffices to characterize matroids. Below we list those objects that play a role in this paper, and we use "base families" to define matroids. For further details, see Schrijver [28] or Murota [24].

Base family. This non-empty family $\mathcal{B} \subseteq 2^S$ satisfies the axiom:

> Let $B_1, B_2 \in \mathcal{B}$. For each $x \in B_1 \setminus B_2$, there exists $y \in B_2 \setminus B_1$ such that $B_1 - x + y \in \mathcal{B}$.

A matroid can be defined as a pair $\mathbf{M} = (S, \mathcal{B})$, where $\mathcal{B}$ is a base family over S. A member of $\mathcal{B}$ is called a *base*. It follows from the axiom above that all bases are equicardinal. This cardinality is called the *rank of the matroid* $\mathbf{M}$.

Independent set family. This family $\mathcal{I} \subseteq 2^S$ is defined as $\mathcal{I} = \{\, I : I \subseteq B \text{ for some } B \in \mathcal{B} \,\}$. A member of $\mathcal{I}$ is called an *independent set*. Any subset of an independent set is clearly also independent, and a maximum-cardinality independent set is clearly a base.

Rank function. This function, $\text{rank} : 2^S \to \mathbb{N}$, is defined $\text{rank}(T) = \max_{I \in \mathcal{I},\, I \subseteq T} |I|$. A maximizer of this expression is called a *base for* T in $\mathbf{M}$. A set I is independent iff $\text{rank}(I) = |I|$.

Since all of the objects listed above can be used to characterize matroids, we sometimes write $\mathbf{M} = (S, \mathcal{I})$, or $\mathbf{M} = (S, \mathcal{I}, \mathcal{B})$, etc. To emphasize the matroid associated to one of these objects, we often write $\mathcal{B}_{\mathbf{M}}$, $\text{rank}_{\mathbf{M}}$, etc.

A *linear representation over* $\mathbb{F}$ of a matroid $\mathbf{M} = (S, \mathcal{I})$ is a matrix Q over $\mathbb{F}$ with columns indexed by S, satisfying the condition that $Q[*, I]$ has full column-rank iff $I \in \mathcal{I}$. There do exist matroids which do not have a linear representation over any field. However, many interesting matroids can be represented over some field; such matroids are called *linear matroids*.

Given a matroid $\mathbf{M} = (S, \mathcal{I}, \mathcal{B})$, there are several interesting ways to construct new matroids.

Restriction. The matroid $\mathbf{M}|T$ has ground set T and independent sets $\{\, I : I \in \mathcal{I}_{\mathbf{M}} \text{ and } I \subseteq T \,\}$.

Direct Sum. Let $\mathbf{M}_1 = (S_1, \mathcal{I}_1)$ and $\mathbf{M}_2 = (S_2, \mathcal{I}_2)$ be two matroids where $S_1 \cap S_2 = \emptyset$. Their direct sum, denoted $\mathbf{M}_1 \oplus \mathbf{M}_2$, has ground set $S_1 \cup S_2$ and independent sets

$$\mathcal{I}_{\mathbf{M}_1 \oplus \mathbf{M}_2} = \{\, I_1 \cup I_2 : I_1 \in \mathcal{I}_1 \;\&\; I_2 \in \mathcal{I}_2 \,\}.$$

Coloop addition. For any $t \notin S$, we define a new matroid $\mathbf{M}' = (S + t, \mathcal{B}')$ such that $\mathcal{B}' = \{\, B + t : B \in \mathcal{B} \,\}$. The element t is called a *coloop* in $\mathbf{M}'$. More generally, an element is a coloop of a matroid if it is contained in every base.

Parallel addition. For any $s \in S$ and $t \notin S$, we define a new matroid $\mathbf{M}' = (S + t, \mathcal{I}')$ in which t and s are *parallel*. That is, $\mathcal{I} \subseteq \mathcal{I}'$ and whenever $I \in \mathcal{I}$ and $I \ni s$, we also have $I - s + t \in \mathcal{I}'$. Every $I \in \mathcal{I}'$ contains at most one of s and t.

Contraction. The contracted matroid $\mathbf{M}/T$ has ground set $S \setminus T$. To define this matroid, first fix a base B_T of T in $\mathbf{M}$. (So $\text{rank}_{\mathbf{M}}(T) = \text{rank}_{\mathbf{M}}(B_T)$.) The base family of $\mathbf{M}/T$ is defined as: $B \in \mathcal{B}_{\mathbf{M}/T}$ iff $B \cup B_T \in \mathcal{B}_{\mathbf{M}}$. The rank function of $\mathbf{M}/T$ satisfies: $\text{rank}_{\mathbf{M}/T}(X) = \text{rank}_{\mathbf{M}}(X \cup T) - \text{rank}_{\mathbf{M}}(T)$.

The class of linear matroids is closed under all of these operations.

2.2 Matroid Intersection Suppose we are given two matroids $\mathbf{M}_1 = (S, \mathcal{B}_{\mathbf{M}_1})$ and $\mathbf{M}_2 = (S, \mathcal{B}_{\mathbf{M}_2})$ which have the same rank r, and a vector $w \in \mathbb{N}^S$ specifying weights on the ground set. A set $B \subseteq S$ is called a *common base* if $B \in \mathcal{B}_{\mathbf{M}_1} \cap \mathcal{B}_{\mathbf{M}_2}$. A *common independent set* is a set $I \in \mathcal{I}_{\mathbf{M}_1} \cap \mathcal{I}_{\mathbf{M}_2}$. A common independent set I is called *extreme* if it has maximum weight amongst all common independent sets of cardinality $|I|$.

The *matroid intersection problem*[3] is to find an extreme common base B. In the context of the matroid intersection problem, it is convenient to use the shorthand $\mathcal{B}_1$ instead of $\mathcal{B}_{\mathbf{M}_1}$, and rank_1 instead of $\text{rank}_{\mathbf{M}_1}$, etc.

2.3 Duality We now consider the intersection problem for two matroids $\mathbf{M}_1$ and $\mathbf{M}_2$ which have a common base. As before, let w be a weighting of the ground set. There are several duality notions for the matroid intersection problem. A *weight splitting* [8] is

[3] The problem is usually defined as finding a common independent set rather than a common base, although the two problems are equivalent. The distinction is similar to maximum matching vs. perfect matching. For the purposes of this paper, the formulation in terms of bases is more convenient.

a pair of vectors (w^1, w^2) such that $w = w^1 + w^2$. An *optimum weight splitting* is one satisfying the following property: B is an extreme common base iff, for both $i \in \{1, 2\}$, B is a maximum weight base for $\mathbf{M}_i$ with respect to weight vector w^i. An optimum weight splitting (w^1, w^2) necessarily exists; furthermore, if w is integer, then an optimum weight splitting exists with both w^1 and w^2 integer.

3 Algorithm Overview

A high-level overview of our algorithm is given below. Each step is explained further in a later section of the paper. The steps are best understood as a sequence of reductions, so the subsequent sections explain them in reverse order.

Input: Two linear matroids $\mathbf{M}_1$ and $\mathbf{M}_2$ and a weight vector w. The matroids are represented as $r \times n$ matrices Q_1 and Q_2 over the same field $\mathbb{F}$.

Step A: Define a certain family of perturbed matroid intersection instances. Compute the maximum weight of a common base for all perturbed instances in $\tilde{O}(nr^{\omega-1}W^{1+\epsilon})$ time. (Section 6)

Step B: Using the maximum weights for these perturbed instances, compute an optimum weight splitting (w^1, w^2) for the original problem. (Section 5)

Step C: Given an optimum weight splitting, compute an optimum primal solution (i.e., an extreme common base B) in $\tilde{O}(nr^{\omega-1})$ time. (Section 4)

4 Step C: Optimum Primal from Optimum Weight Splitting

Suppose that we are given an optimum weight splitting (w^1, w^2). We wish to find an extreme common base for $\mathbf{M}_1$ and $\mathbf{M}_2$. From the definition of an optimum weight splitting, we know that it suffices to find a set $B \subseteq S$ such that

- B is a common base for $\mathbf{M}_1$ with maximum weight with respect to w^1, and
- B is a common base for $\mathbf{M}_2$ with maximum weight with respect to w^2.

How can we find such a set B? This task is made quite easy by the following useful lemma.

Lemma 1. *Let $\mathbf{M} = (S, \mathcal{B})$ be a matroid and let $w \in \mathbb{R}^S$ be a weighting of S. Define $\mathcal{B}^w$ to be the set of maximum weight bases, i.e., $\mathcal{B}^w := \arg\max_{B \in \mathcal{B}} w(B)$. Then $\mathbf{M}^w := (S, \mathcal{B}^w)$ is also a matroid.*

Proof. Edmonds [6, p130] or Cook et al. [4, p287]. □

So to find our desired set B, it suffices to find *any* common base of $\mathbf{M}_1^{w^1}$ and $\mathbf{M}_2^{w^2}$. This is an unweighted matroid intersection problem, which can be solved in $O(nr^{\omega-1})$ time [14]. Now the question remains: how is $\mathbf{M}^w$ constructed, and can this be done efficiently?

The length of w is $n = |S|$, but suppose that it has only k distinct weights $w_1 > \cdots > w_k$. Define a nested sequence of sets $\emptyset = W_0 \subset W_1 \subset \cdots \subset W_k = S$ as follows: $W_i := \{ s \in S : w(s) \geq w_i \}$. Define $\mathbf{M}_{W_i} := (\mathbf{M}/W_{i-1})|W_i$. It can be shown that $\mathbf{M}^w$ is precisely the direct sum $\bigoplus_{i=1}^k \mathbf{M}_{W_i}$. As remarked earlier, the class of linear matroids is closed under contraction, truncation, and direct sums. Thus $\mathbf{M}^w$ is actually linear.

The remainder of this section shows that a linear representation of $\mathbf{M}^w$ can be efficiently computed. We begin with some facts concerning representations.

Fact 2. *Let $\mathbf{M}$ be a matroid and let Q be a representation of $\mathbf{M}$. Any matrix obtained from Q by elementary row operations is also a representation of $\mathbf{M}$.*

Fact 3. *Let $\mathbf{M} = (S, \mathcal{I})$ be a matroid and let Q be a representation of $\mathbf{M}$. Let $T \subseteq S$ be arbitrary. Then $Q[*, T]$ is a representation of $\mathbf{M} \,|\, T$.*

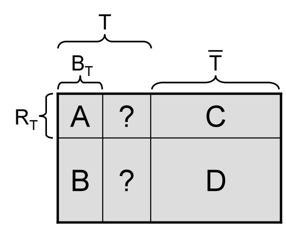
Fact 4. *Let $\mathbf{M} = (S, \mathcal{I})$ and $\mathbf{M}' = (S', \mathcal{I}')$ be matroids on disjoint ground sets. Let Q and Q' respectively be representations of $\mathbf{M}$ and $\mathbf{M}'$ over the same field. Then $\begin{pmatrix} Q & 0 \\ 0 & Q' \end{pmatrix}$ is a representation of $\mathbf{M} \oplus \mathbf{M}'$.*

Lemma 5. *Let $\mathbf{M} = (S, \mathcal{I})$ be a matroid represented by a matrix Q with row-set R and column-set S. Let $T \subseteq S$ be arbitrary, and let $Q[R_T, B_T]$ be a maximum-rank square submatrix of $Q[*, T]$. Then*

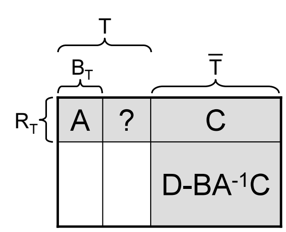
$$\hat{Q} := Q[\overline{R_T}, \overline{T}] - Q[\overline{R_T}, B_T] \cdot Q[R_T, B_T]^{-1} \cdot Q[R_T, \overline{T}]$$

is a representation of $\mathbf{M}/T$.

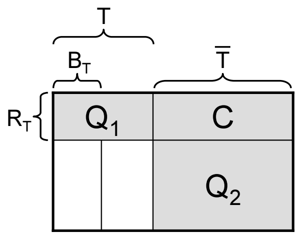
Let us restate this lemma more visually. Initially, the matrix Q has the form



Now suppose that we use the submatrix A to eliminate the entries beneath it. We obtain the matrix

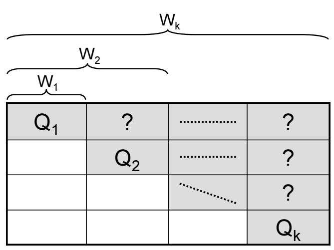


The lemma asserts that $D - BA^{-1}C$ is a representation of $\mathbf{M}/T$. Stated differently, suppose that we perform Gaussian elimination on Q, where the columns in T are used for pivoting before the columns in $\overline{T}$. The resulting matrix has the form

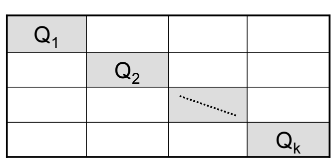


But by Fact 2 and Lemma 5, Q_1 and Q_2 are representations for $M|T$ and M/T respectively.

Now consider our earlier problem in which we have a matroid $\mathbf{M} = (S, \mathcal{I})$ and a nested sequence of sets $W_1 \subset \cdots W_k = S$. We have a linear representation Q for $\mathbf{M}$ and we wish to compute a representation of $\mathbf{M}^w$. Suppose we perform Gaussian elimination on Q, ensuring that we pivot on columns in W_i before any columns in $W_j \setminus W_i$ if $j > i$. The resulting matrix has the form



An inductive argument shows that Q_i represents $(\mathbf{M}/W_{i-1})|W_i = \mathbf{M}_{W_i}$ for all $1 \leq i \leq k$. Moreover, these matrices $Q_1, \ldots, Q_k$ can all be constructed in time $O(nr^{\omega-1})$, since this is the time required by Gaussian elimination. Finally, let Q^w be the matrix



By Fact 4, this is a representation of $\bigoplus_{i=1}^{k} \mathbf{M}_{W_i}$, which is precisely $\mathbf{M}^w$. Thus we have shown that computing Q^w requires only $O(nr^{\omega-1})$ time.

5 Step B: Optimum Weight Splitting from Perturbed Instances

In this section, we describe a family of "perturbed instances" for which the weights of their respective optimum solutions yield an optimum weight splitting for the original instance.

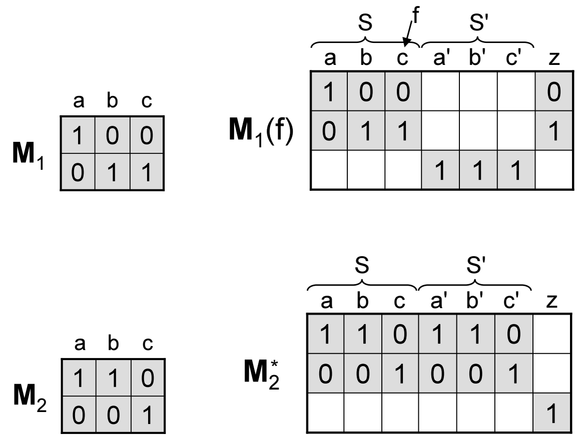


Figure 1: An example of the perturbed matroids constructed in Section 5. Linear representations of $\mathbf{M}_1$ and $\mathbf{M}_2$ are shown, together with representations of a matroid $\mathbf{M}_1(f)$ and $\mathbf{M}_2^*$. Note that $\{a, c\}$ is a common base of $\mathbf{M}_1$ and $\mathbf{M}_2$, and that $\{a, c', z\}$ is a common base of $\mathbf{M}_1(c)$ and $\mathbf{M}_2^*$,

There are n perturbed versions of $\mathbf{M}_1$, denoted $\{ \mathbf{M}_1(f) : f \in S \}$, and there is one perturbed version of $\mathbf{M}_2$, denoted $\mathbf{M}_2^*$. All perturbed matroids have the ground set $S \cup S' + z$, where S' is a copy of S and z is a single new element. The original weight vector w on the ground set S is extended to a weight vector on $S \cup S' + z$ by giving all new elements weight zero.

The matroid $\mathbf{M}_1(f)$ has the base family

$$\begin{aligned} &\{ B + s' : \forall B \in \mathcal{B}_{\mathbf{M}_1} \text{ and } \forall s' \in S' \} \;\; \cup \\ &\{ B - f + z + s' : \forall s' \in S' \text{ and } \forall B \text{ s.t. } f \in B \in \mathcal{B}_{\mathbf{M}_1} \}. \end{aligned}$$

In other words, a base of $\mathbf{M}_1(f)$ is simply a base of $\mathbf{M}_1$ which must additionally contain exactly one element of S', and which can contain either f or z but not both. This is indeed a matroid since it can be constructed by the coloop addition and parallel addition operations mentioned in Section 2.

The matroid $\mathbf{M}_2^*$ has bases defined as follows. Given any base B of $\mathbf{M}_2$, we must add the element z, and we may replace any $s \in B$ with its primed-counterpart s'. In other words, the element z is a new coloop, and each element s' is chosen to be parallel to s.

For convenience, let $w(\mathbf{M}_1 \cap \mathbf{M}_2)$ now denote the maximum weight of a common base in $\mathbf{M}_1$ and $\mathbf{M}_2$ under weight function w. The key property of the perturbed matroids lies in the following theorem.

Theorem 6. *Define a vector $w^1 \in \mathbb{Z}^S$ as follows: $w^1(f) = w(\mathbf{M}_1 \cap \mathbf{M}_2) - w(\mathbf{M}_1(f) \cap \mathbf{M}_2^*)$. Define $w^2 = w - w^1$. Then (w^1, w^2) constitute an optimum weight splitting for $\mathbf{M}_1$ and $\mathbf{M}_2$ with weight vector w.*

6 Step A: Solving the Perturbed Instances

The preceding section showed that we can construct an optimum dual solution by computing the values $w(\mathbf{M}_1 \cap \mathbf{M}_2)$ and $w(\mathbf{M}_1(f) \cap \mathbf{M}_2^*)$ for each $f \in S$. The purpose of this section is to show that these computations can be done efficiently. We will do so by viewing the matroid intersection problem in the context of a somewhat more general problem, known as the *independent assignment problem* [18].

An instance of the independent assignment problem is a tuple $G = (S_1 \cup S_2, E, w, \mathbf{M}_1, \mathbf{M}_2)$ where $(S_1 \cup S_2, E)$ is a bipartite graph with $E \subseteq S_1 \times S_2$; $w \in \mathbb{N}^E$ is a weight vector on the edges; and $\mathbf{M}_1 = (S_1, \mathcal{B}_1)$ and $\mathbf{M}_2 = (S_2, \mathcal{B}_2)$ are matroids. For $M \subseteq E$, let $\partial_1 M$ denote the subset of S_1 covered by M, and define $\partial_2 M$ similarly. A *basic matching* is one in which $\partial_1 M \in \mathcal{B}_1$ and $\partial_2 \in \mathcal{B}_2$. The objective is to find a basic matching such that the weight $w(M)$ is maximized. The maximum weight of a basic matching is denoted $w(G)$.

Note that any weighted matroid intersection problem can be viewed as an instance of the independent assignment problem. (Just ensure that each vertex of G has degree 1.)

Given $G = (S_1 \cup S_2, E, w, \mathbf{M}_1, \mathbf{M}_2)$ and an edge (u, v) in G, the *contracted problem* $G/(u, v)$ is obtained by deleting the vertices u and v (and their incident edges), truncating w appropriately, contracting u in $\mathbf{M}_1$ and contracting v in $\mathbf{M}_2$.

6.1 The Independent Assignment Problem G for Perturbed Instances

We now define a specific instance G of the independent assignment problem which captures the intersection problem for matroids $\mathbf{M}_1(f)$ and $\mathbf{M}_2^*$. Let S_1 and S_2 be disjoint copies of S, i.e., each $s \in S$ appears as $s_1 \in S_1$ and $s_2 \in S_2$. The underlying bipartite graph has left-vertices $S_1 + r$ and right-vertices $S_2 + z$. The edges of this graph are as follows.

- For each $s \in S$, add edge (s_1, s_2) with weight w_s.
- For each $s \in S$, add edge (r, s_2) with weight zero.
- For each $s \in S$, add edge (s_1, z) with weight zero.

The matroid associated with $S_1 + r$ is $\mathbf{M}_1^r$, which is obtained by adding r as a new coloop to $\mathbf{M}_1$. Similarly, the matroid associated with $S_2 + z$ is $\mathbf{M}_2^z$, obtained by adding z as a new coloop to $\mathbf{M}_2$.

The motivation for constructing this instance G is demonstrated by the following lemma — if we contract an edge (f_1, z) and then compute the optimum weight of a basic matching in the resulting instance, this determines $w\big(\mathbf{M}_1(f) \cap \mathbf{M}_2^*\big)$.

LEMMA 7. $w\big(G/(f_1, z)\big) = w\big(\mathbf{M}_1(f) \cap \mathbf{M}_2^*\big)$.

6.2 Connection to Linear Algebra

In this section, we show how to efficiently compute the values $w\big(G/(f_1, z)\big)$ for each $f \in S$. The key idea is to use a matrix introduced in previous work [14] that relates to the unweighted independent assignment problem, which is also known as the bipartite matroid matching problem.

Let n now denote the number of vertices on each side of G, and let r now denote the rank of the associated matroids. That is, we now have $n = |S| + 1$ and $r = \operatorname{rank} \mathbf{M}_1 + 1$. For each edge (i, j) in G, associate an indeterminate $t_{i,j}$. Define an $n \times n$ matrix T where $T_{i,j} = t_{i,j}$ if (i, j) is an edge, and $T_{i,j}$ is zero otherwise. Let Q_1 be an $r \times n$ matrix whose columns represent $\mathbf{M}_1$ over a field $\mathbb{F}$. Also, let Q_2 be a $n \times r$ matrix whose rows represent $\mathbf{M}_2$ over $\mathbb{F}$. Define Z to be the following matrix:

$$Z = \begin{pmatrix} & Q_1 & \\ Q_2 & & I \\ & I & T \end{pmatrix},$$

where the submatrices I are the identity matrix of size $n \times n$. It is known that Z is non-singular iff G has a basic matching [14]. Now define $N = Q_1 T Q_2$ and $\tilde{Q} = Q_2 N^{-1} Q_1$. It also follows from [14] that $\tilde{Q}_{v,u} \neq 0$ iff there exists a basic matching containing the edge (u, v).

These results can be generalized to include weights by the standard polynomial technique. Let q be a new indeterminate. We redefine T so that $T_{i,j} = t_{i,j} \cdot q^{w(i,j)}$ if (i, j) is an edge. The determinant of Z is now a polynomial in q, and the maximum degree of q in this polynomial is the maximum weight of a basic matching. Furthermore, assuming that $\tilde{Q}_{v,u}$ is non-zero, $w\big(G/(u, v)\big)$ is precisely the maximum degree of q in the polynomial $(\det Z) \cdot \tilde{Q}_{v,u}$. This fact is the key to the algorithm: the desired polynomials $w\big(G/(f_1, z)\big)$ can all be computed from the z^{th} row of $\tilde{Q}$.

6.3 Computing $\tilde{Q}_{z,*}$ The first step of our computation is to compute $\det Z$. Standard manipulations show that $\det Z = \det N$, so we turn our attention to computing N efficiently. We show that this can be done by a randomized algorithm in $\tilde{O}(nr^{\omega-1} W^{1+\epsilon})$ time. By standard arguments, substituting random values for the indeterminates $t_{i,j}$ does not affect the result of our computation, with high probability. The resulting matrices only involve the indeterminate q. We now claim that computing TQ_2 requires only $O(nr)$ time. This follows because T has only $O(n)$ non-zero entries (since G has only $O(n)$ edges). Also, the entries of T and Q_2 are all monomials, so all polynomial manipulations are trivial during this step.

Next, consider computing $N = Q_1(TQ_2)$. We claim that this requires only time $\tilde{O}(nr^{\omega-1}W)$. Suppose initially that Q_1 and TQ_2 have numeric entries. The matrix Q_1 has size $r \times n$ and the product TQ_2 has size $n \times r$. Such matrices can be multiplied in time $O(\frac{n}{r} r^{\omega}) = O(nr^{\omega-1})$, by working with $r \times r$ blocks. Since the entries of Q_1 and TQ_2 are actually polynomials of degree at most W, each arithmetic operation during matrix multiplication actually takes time $\tilde{O}(W)$ (e.g., using the FFT). Therefore $\tilde{O}(nr^{\omega-1}W)$ time suffices to compute N. The size of N is $r \times r$, so Storjohann's subroutine can compute $\det N = \det Z$ in time $\tilde{O}(r^{\omega}W^{1+\epsilon})$. The highest degree of q in $\det Z$ is the value of $w(\mathbf{M}_1 \cap \mathbf{M}_2)$. As we recall, this value is also needed by the algorithm (cf. Theorem 6).

The next step of our computation is to compute the z^{th} row of $\tilde{Q}$. This row can be written as $\big((Q_2)_{z,*}\, N^{-1}\big) Q_1$ The vector $v = (Q_2)_{z,*}\, N^{-1}$ can be computed in $\tilde{O}(nr^{\omega-1}W^{1+\epsilon})$ time using Storjohann's algorithm for solving linear systems. Note that the entries of v are rational functions for which the numerators and denominators are polynomials in q of degree at most rW. At this point, we can eliminate the denominators: we eventually need to multiply the entries of $\tilde{Q}_{z,*}$ by $\det Z$, and doing so now allows us to work with polynomials rather than rational functions in the subsequent steps. The time for this computation is only $\tilde{O}(nrW)$ since each entry of v is a polynomial of degree at most rW (as is $\det Z$), and v has n entries.

Let the resulting vector be denoted $v' = \det Z \cdot v$. It remains to compute $v' \cdot Q_1$. Recall that v' is a row vector of length r and that Q_1 has size $r \times n$. Computing the product $v' \cdot Q_1$ naively requires time $O(nr^2W)$, since the entries of v' have could have degree $2rW$.

The following simple trick enables the use of fast matrix multiplication to compute $v' \cdot Q_1$ more quickly. Let us write the i^{th} entry of v' more explicitly as $v'_i = \sum_{j=0}^{2rW} \alpha_{j,i} q^j$. Let A be the $(2rW+1) \times r$ matrix defined by $A_{j,i} = \alpha_{j,i}$, and let $\vec{q}$ be the row vector of length $(2rW+1)$ defined by $\vec{q}_j = q^j$. Then $v' = \vec{q} \cdot A$. Now we observe that the product $A \cdot Q_1$ may be computed in $O(nr^{\omega-1}W)$ time, working with $r \times r$ blocks. Multiplying by $\vec{q}$ is clearly just a trivial syntactic transformation. Thus, in total, $\tilde{O}(nr^{\omega-1}W^{1+\epsilon})$ time suffices to compute $\vec{q} \cdot A \cdot Q_1 = v' \cdot Q_1 = (\det Z) \cdot \tilde{Q}_{z,*}$. By our preceding discussion, the maximum degree of $(\det Z) \cdot \tilde{Q}_{z,f}$ is $w\big(G/(f_1, z)\big)$, so we have shown that these values can all be efficiently computed.

Acknowledgements

I would like to thank Joseph Cheriyan and Jim Geelen for instigating the work on this project, and for contributing several important ideas. I also thank Mohit Singh for generously helping to work through some of the more complicated technical steps.

References

[1] M. Aigner and T. A. Dowling. Matching theory for combinatorial geometries. *Transactions of the American Mathematical Society*, 158(1):231–245, July 1971.

[2] C. Brezovec, G. Cornuéjols, and F. Glover. Two algorithms for weighted matroid intersection. *Mathematical Programming*, 36:39–53, 1986.

[3] P. Chalasani and R. Motwani. Approximating capacitated routing and delivery problems. *SIAM Journal on Computing*, 26(6):2133–2149, 1999.

[4] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization.* Wiley, 1997.

[5] J. Edmonds. Submodular functions, matroids, and certain polyhedra. In R. Guy, H. Hanani, N. Sauer, and J. Schönheim, editors, *Combinatorial Structures and Their Applications*, pages 69–87. Gordon and Breach, 1970.

[6] J. Edmonds. Matroids and the greedy algorithm. *Mathematical Programming*, 1:127–136, 1971.

[7] J. Edmonds. Matroid intersection. In P. L. Hammer, E. L. Johnson, and B. H. Korte, editors, *Discrete Optimization I*, volume 4 of *Annals of Discrete Mathematics*, pages 39–49. North-Holland, 1979.

[8] A. Frank. A weighted matroid intersection algorithm. *Journal of Algorithms*, 2(4):328–336, 1981.

[9] S. Fujishige. A primal approach to the independent assignment problem. *Journal of the Operations Research Society of Japan*, (20):1–15, 1977.

[10] S. Fujishige. *Submodular Functions and Optimization*, volume 58 of *Annals of Discrete Mathematics.* Elsevier, second edition, 2005.

[11] H. N. Gabow and R. E. Tarjan. Faster scaling algorithms for network problems. *SIAM Journal on Computing*, 18(5):1013–1036, Oct. 1989.

[12] H. N. Gabow and Y. Xu. Efficient theoretic and practical algorithms for linear matroid intersection problems. *Journal of Computer and System Sciences*, 53(1):129–147, 1996.

[13] M. X. Goemans. Minimum bounded degree spanning trees. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2006.

[14] N. J. A. Harvey. Algebraic structures and algorithms for matching and matroid problems. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2006.

[15] N. J. A. Harvey, D. R. Karger, and K. Murota. Deterministic network coding by matrix completion. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 05)*, pages 489–498, 2005.

[16] R. Hassin and A. Levin. An efficient polynomial time approximation scheme for the constrained minimum spanning tree problem using matroid intersection. *SIAM Journal on Computing*, 33(2):261–268, 2004.

[17] M. Iri. Applications of matroid theory. In A. Bachem, M. Grötschel, and B. Korte, editors, *Mathematical Programming: The State of the Art*, pages 158–201. 1983.

[18] M. Iri and N. Tomizawa. An algorithm for finding an optimal "independent assignment". *Journal of the Operations Research Society of Japan*, 19:32–57, 1976.

[19] R. M. Karp, E. Upfal, and A. Wigderson. Constructing a perfect matching is in random NC. *Combinatorica*, 6(1):35–48, 1986.

[20] S. Krogdahl. A combinatorial proof for a weighted matroid intersection algorithm. Technical Report Computer Science Report 17, Institute of Mathematical and Physical Sciences, University of Tromsø, 1976.

[21] E. L. Lawler. Matroid intersection algorithms. *Mathematical Programming*, 9:31–56, 1975.

[22] M. Mucha and P. Sankowski. Maximum matchings via Gaussian elimination. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 248–255, 2004.

[23] K. Mulmuley, U. V. Vazirani, and V. V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987.

[24] K. Murota. *Matrices and Matroids for Systems Analysis.* Springer-Verlag, 2000.

[25] K. Murota. *Discrete Convex Analysis.* SIAM, 2003.

[26] P. Sankowski. Processor efficient parallel matching. In *Proceedings of the 17th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, pages 165–170, 2005.

[27] P. Sankowski. Weighted bipartite matching in matrix multiplication time. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP)*, 2006.

[28] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency.* Springer-Verlag, 2003.

[29] A. Storjohann. High-order lifting and integrality certification. *Journal of Symbolic Computation*, 36:613–648, 2003.

A Proof of Lemma 5

Proof. Without loss of generality, we may assume that the rank of $\mathbf{M}$ is $|R|$. It follows from the definition that B_T is a base for T in $\mathbf{M}$. Thus,

$$\begin{aligned}
&B \in \mathcal{B}_{\mathbf{M}/T} \\
&\quad \iff B \cup B_T \in \mathcal{B}_{\mathbf{M}} \\
&\quad \iff Q[*, B \cup B_T] \text{ is non-singular} \\
&\quad \iff \text{the Schur complement of } Q[R_T, B_T] \\
&\qquad\quad \text{in } Q[*, B \cup B_T] \text{ is non-singular} \\
&\qquad\quad \text{(since } Q[R_T, B_T] \text{ is itself non-singular)}
\end{aligned}$$

This Schur complement is precisely the matrix

$$Q[\overline{R_T}, B] - Q[\overline{R_T}, B_T] \cdot Q[R_T, B_T]^{-1} \cdot Q[R_T, B],$$

which can be more conveniently written $\hat{Q}[*, B]$. Thus we have shown that $B \in \mathcal{B}_{\mathbf{M}/T}$ iff $\hat{Q}[*, B]$ is non-singular. Thus $\hat{Q}$ is a representation of $\mathbf{M}/T$. □

B Proof of Theorem 6

Theorem 6 follows directly from the two lemmas given in this section. Before starting the proofs, we need some preliminary definitions and facts.

B.1 Preliminaries

DEFINITION. Let $\mathbf{M} = (S, \mathcal{I})$ be a matroid and let $I \in \mathcal{I}$. The *augmentation graph* $D_{\mathbf{M}}(I) = (S, A)$ is a directed graph defined as follows. For arbitrary $u \in I$ and $v \notin I$, the graph has arcs:

$$(u, v) \in A \qquad (\text{if } I + v \notin \mathcal{I} \text{ and } I - u + v \in \mathcal{I}).$$

DEFINITION. Let $\mathbf{M}_1 = (S, \mathcal{I}_1)$ and $\mathbf{M}_2 = (S, \mathcal{I}_2)$ be matroids, let w be a weight function on S, and let $I \in \mathcal{I}_1 \cap \mathcal{I}_2$. The *auxiliary graph* $D_{\mathbf{M}_1,\mathbf{M}_2}(I) = (S, A)$ is a directed graph whose arcs are formed by the union of $D_{\mathbf{M}_1}(I)$ and the reverse of $D_{\mathbf{M}_2}(I)$. In other words, for arbitrary $u \in I$ and $v \notin I$, the graph has arcs:

$$\begin{aligned}
(u, v) \in A &\qquad (\text{if } I + v \notin \mathcal{I}_1 \text{ and } I - u + v \in \mathcal{I}_1) \\
(v, u) \in A &\qquad (\text{if } I + v \notin \mathcal{I}_2 \text{ and } I - u + v \in \mathcal{I}_2).
\end{aligned}$$

The auxiliary graph has lengths on its vertices. A vertex $u \in S$ has length $l(u) = w(u)$ if $u \in I$ and $l(u) = -w(u)$ otherwise. The length of a path is the sum of the lengths of the vertices that it traverses, excluding the initial vertex.

DEFINITION. In this paper, we will also use a slight modification of the auxiliary graph. Define $D^r_{\mathbf{M}_1,\mathbf{M}_2}(I)$ by adding to $D_{\mathbf{M}_1,\mathbf{M}_2}(I)$ a vertex r with arcs (r, e) for each $e \in I$.

DEFINITION. A *potential function* is a function $p : S \to \mathbb{R}$ satisfying $l(v) \geq p(v) - p(u)$ for each arc (u, v) in the auxiliary graph.

The following theorem combines several results due to Krogdahl [20], Fujishige [9], and Brezovec et al. [2].

THEOREM 8.

- *If there exists a potential function for the auxiliary graph $D_{\mathbf{M}_1,\mathbf{M}_2}(I)$, then it has no negative-length cycles.*
- *If $D_{\mathbf{M}_1,\mathbf{M}_2}(I)$ has no negative-length cycles then I is extreme.*
- *Conversely, suppose that I is an extreme common independent set for $\mathbf{M}_1$ and $\mathbf{M}_2$.*
  - *Then neither $D_{\mathbf{M}_1,\mathbf{M}_2}(I)$ nor $D^r_{\mathbf{M}_1,\mathbf{M}_2}(I)$ has any negative-length cycles.*
  - *Let $p(f)$ be the shortest-path distance from r to f in $D^r_{\mathbf{M}_1,\mathbf{M}_2}(I)$. Then p is a potential function.*
  - *Define $w^1(f)$ to be $p(f)$ if $f \in I$ and $w(f) + p(f)$ otherwise. Define $w^2 = w - w^1$. Then (w^1, w^2) form an optimum weight splitting.*

B.2 The Lemmas In the subsequent lemmas, we will use the following notation. Let I be a maximum-weight common base for $\mathbf{M}_1$ and $\mathbf{M}_2$. Let $I(f)$ be a maximum-weight common base for $\mathbf{M}_1(f)$ and $\mathbf{M}_2^*$. Note that $|I(f)| = |I|+1$. For the reader's convenience, we will illustrate the arguments by continuing the example of Figure 1 in Figure 2.

LEMMA 9. *For each $f \in S$, $w^1(f) \le w(\mathbf{M}_1 \cap \mathbf{M}_2) - w(\mathbf{M}_1(f) \cap \mathbf{M}_2^*)$.*

Proof. Consider augmenting $\mathbf{M}_1(f)$ and $\mathbf{M}_2^*$ by adding a new element t which is parallel in $\mathbf{M}_1(f)$ to all elements in S' and parallel in $\mathbf{M}_2^*$ to z. The weight of t is 0. Call the resulting matroids $\tilde{\mathbf{M}}_1(f)$ and $\tilde{\mathbf{M}}_2^*$. Note that $I(f)$ and $I+t$ are both common bases for $\tilde{\mathbf{M}}_1(f)$ and $\tilde{\mathbf{M}}_2^*$. See Figure 2 (a)-(b).

By a lemma of Brualdi [28, Corollary 39.12a], $D_{\tilde{\mathbf{M}}_1(f)}(I+t)$ has a perfect matching M_1 on the vertices in the symmetric difference $(I+t) \triangle I(f)$. Furthermore, if we reverse the directions of the arcs in M_1, it becomes a perfect matching in $D_{\tilde{\mathbf{M}}_1(f)}(I(f))$. (This useful property is not mentioned in [28], but the proof directly shows that this property holds.) See Figure 2 (c). Also, there is a perfect matching M_2 on $(I+t) \triangle I(f)$ in $D_{\tilde{\mathbf{M}}_2^*}(I(f))$ whose reversal is a perfect matching in $D_{\tilde{\mathbf{M}}_2^*}(I+t)$. See Figure 2 (d). The union $M_1 \cup M_2$ forms a collection of vertex-disjoint directed cycles in $D_{\tilde{\mathbf{M}}_1(f),\tilde{\mathbf{M}}_2^*}(I+t)$. See Figure 2 (e). Furthermore, the reversal of $M_1 \cup M_2$ is a collection of cycles in $D_{\tilde{\mathbf{M}}_1(f),\tilde{\mathbf{M}}_2^*}(I(f))$. Note that the length $l(M_1 \cup M_2) = l(I+t) - l(I(f)) = w(\mathbf{M}_1 \cap \mathbf{M}_2) - w(\mathbf{M}_1(f) \cap \mathbf{M}_2^*)$.

Consider any cycle C in $M_1 \cup M_2$ which avoids the element t. We claim that the length $l(C)$ is zero. The only inbound arcs to S' in $D_{\tilde{\mathbf{M}}_1(f),\tilde{\mathbf{M}}_2^*}(I+t)$ are from t and the only outbound arc from z is to t. Therefore C is disjoint from $S'+z$, and hence is also a cycle in $D_{\mathbf{M}_1,\mathbf{M}_2}(I)$. Since I is extreme, Theorem 8 implies C cannot have negative length. Now consider the reversal of C. By our preceding discussion, it is a cycle in $D_{\tilde{\mathbf{M}}_1(f),\tilde{\mathbf{M}}_2^*}(I(f))$, although its length becomes negated. Since $I(f)$ is extreme, this resulting cycle cannot have negative length, and hence C cannot have positive length. Therefore we have proven the claim that $l(C) = 0$.

So consider the unique cycle which traverses element t. (Such a cycle must exist since $t \in I+t$ but $t \notin I(f)$. Uniqueness follows since the cycles in $M_1 \cup M_2$ are vertex disjoint.) Removing element t, we obtain a path in $D^r_{\mathbf{M}_1(f),\mathbf{M}_2^*}(I)$ which begins at an element in S' and ends at element z. See Figure 2 (e). Furthermore, this path is disjoint from $S' \cup z$ except for its starting and ending vertices. We now adjust this path to obtain

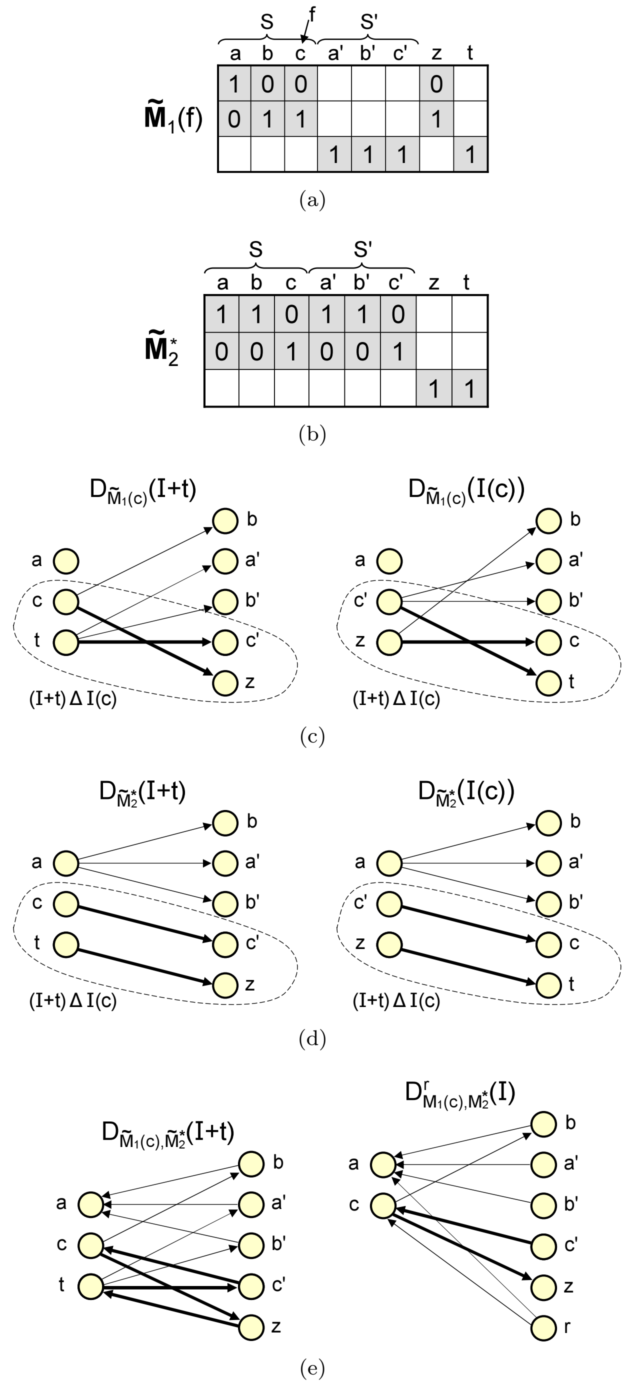


Figure 2: Examples for Lemma 9. (a)-(b) We will use the bases $I = \{a, c\} \in \mathcal{B}_{\mathbf{M}_1} \cap \mathcal{B}_{\mathbf{M}_2}$ and $I(c) = \{a, c', z\} \in \mathcal{B}_{\mathbf{M}_1(c)} \cap \mathcal{B}_{\mathbf{M}_2^*}$. (c) Augmentation graphs. In $D_{\tilde{\mathbf{M}}_1(c)}(I+t)$, the bold edges are M_1. In $D_{\tilde{\mathbf{M}}_1(c)}(I(c))$, the bold edges are the reversal of M_1. (d) More augmentation graphs. In $D_{\tilde{\mathbf{M}}_1(c)}(I+t)$, the bold edges are the reversal of M_2. In $D_{\tilde{\mathbf{M}}_1(c)}(I(c))$, the bold edges are M_2. (e) $M_1 \cup M_2$ forms a cycle. Removing element t, it forms a path in $D^r_{\mathbf{M}_1(f),\mathbf{M}_2^*}(I)$.

one starting at r and ending at f.

First, consider the starting vertex. The second vertex on the path belongs to I by definition of the auxiliary graph. Since r has outbound arcs to all vertices in I, we may simply replace the initial vertex with r. This does not affect the length since the initial vertex of a path does not contribute to its length. The path length is $w(\mathbf{M}_1 \cap \mathbf{M}_2) - w(\mathbf{M}_1(f) \cap \mathbf{M}_2^*)$, since element $l(t) = 0$ and, as argued above, all cycles avoiding t have length 0.

Next, consider the ending vertex, namely z, and the penultimate vertex, which we call e. We necessarily have $e \in I$. There are two cases.

Case 1: $f \in I$. We claim that the only arc inbound to z is (f, z). To see this, note that the existence of an arc (g, z) where $f \neq g \in I$ implies that $I - g + z \in \mathcal{I}_{\mathbf{M}_1(f)}$. But f and z are parallel and $f \in I$, which is a contradiction. So the claim is proven, and therefore $e = f$. We simply delete the last vertex (namely z) and obtain a path ending at f; this does not affect the length since $l(z) = 0$. Thus the shortest-path distance from r to f is at most $w(\mathbf{M}_1 \cap \mathbf{M}_2) - w(\mathbf{M}_1(f) \cap \mathbf{M}_2^*)$.

Case 2: $f \notin I$. Note that (e, z) is an arc by definition of e, and that f and z are parallel in $\mathbf{M}_1(f)$, implying that (e, f) is an arc. We remove z from the path and replace it with f. This decreases the length of the path by $w(f)$. Therefore the shortest-path distance from r to f is at most $w(\mathbf{M}_1 \cap \mathbf{M}_2) - w(\mathbf{M}_1(f) \cap \mathbf{M}_2^*) - w(f)$. In other words, the shortest-path distance from r to f, plus $w(f)$, is at most $w(\mathbf{M}_1 \cap \mathbf{M}_2) - w(\mathbf{M}_1(f) \cap \mathbf{M}_2^*)$.

Having analyzed these shortest path distances, we now define the optimum weight splitting vector w^1. Following Theorem 8, $w^1(f)$ is the shortest path distance from r to f if $f \in I$. If $f \notin I$ then $w^1(f)$ is the shortest path distance from r to f plus $w(f)$. In both cases, this quantity is at most $w(\mathbf{M}_1 \cap \mathbf{M}_2) - w(\mathbf{M}_1(f) \cap \mathbf{M}_2^*)$, so the lemma is proven. □

LEMMA 10. *For each $f \in S$, $w^1(f) \geq w(\mathbf{M}_1 \cap \mathbf{M}_2) - w(\mathbf{M}_1(f) \cap \mathbf{M}_2^*)$.*

Proof. Deferred to the full version of the paper. □

C Proof of Lemma 7

Proof. "$\leq$" direction: Suppose that M is a basic matching for $G/(f_1, z)$. Let (z, t_2) be the unique arc incident with z in M. Define

$$B := \{\, s : (s_1, s_2) \in M \,\} + t' + z.$$

(Here, t' is the member of S' corresponding to $t_2 \in S_2$.) Note that $w(B) = w(M)$. We now show that $B \in \mathcal{B}_{\mathbf{M}_1(f)} \cap \mathcal{B}_{\mathbf{M}_2^*}$, establishing the desired inequality.

$$\begin{aligned}
&\partial_1 M \in \mathcal{B}_{\mathbf{M}_1^r/f_1} \\
&\quad\implies \partial_1 M - r + f \in \mathcal{B}_{\mathbf{M}_1} \\
&\quad\implies \partial_1 M - r + z + t' \in \mathcal{B}_{\mathbf{M}_1(f)} \\
&\qquad\quad \text{(since f and z are parallel in $\mathbf{M}_1(f)$)} \\
&\quad\implies B \in \mathcal{B}_{\mathbf{M}_1(f)}
\end{aligned}$$

$$\begin{aligned}
&\partial_2 M \in \mathcal{B}_{\mathbf{M}_2^z/z} \\
&\quad\implies \partial_2 M \in \mathcal{B}_{\mathbf{M}_2} \\
&\quad\implies \partial_2 M - t + t' + z \in \mathcal{B}_{\mathbf{M}_2^*} \\
&\qquad\quad \text{(since t and t' are parallel in $\mathbf{M}_2^*$)} \\
&\quad\implies B \in \mathcal{B}_{\mathbf{M}_2^*}
\end{aligned}$$

"$\geq$" direction: Suppose that B is a common base for $\mathbf{M}_1(f) \cap \mathbf{M}_2^*$. Note that $|B \cap S'| = 1$, by definition of $\mathbf{M}_1(f)$. Let t' be the unique element in $B \cap S'$. Define

$$M := \{\, (s_1, s_2) : s \in B \cap S \,\} + (r, t_2).$$

(Here, t_2 is the member of S_2 corresponding to $t' \in S'$.) Note that $w(M) = w(B)$. We now show that M is a basic matching for $G/(f_1, z)$, establishing the desired inequality.

$$\begin{aligned}
&B \in \mathcal{B}_{\mathbf{M}_1(f)} \\
&\quad\implies B - z + f \in \mathcal{B}_{\mathbf{M}_1(f)} \\
&\quad\implies B - z + f - t' \in \mathcal{B}_{\mathbf{M}_1} \\
&\quad\implies B - z - t' \in \mathcal{B}_{\mathbf{M}_1/f} \\
&\quad\implies \partial_1 M \in \mathcal{B}_{\mathbf{M}_1^r/f_1}
\end{aligned}$$

$$\begin{aligned}
&B \in \mathcal{B}_{\mathbf{M}_2^*} \\
&\quad\implies B - t' + t - z \in \mathcal{B}_{\mathbf{M}_2} \\
&\quad\implies \partial_2 M \in \mathcal{B}_{\mathbf{M}_2^z/z}
\end{aligned}$$

□