Electronic Colloquium on Computational Complexity, Report No. 121 (2021)

Matroid Intersection: A pseudo-deterministic parallel reduction from search to weighted-decision

Sumanta Ghosh* Rohit Gurjar†

Abstract

We study the matroid intersection problem from the parallel complexity perspective. Given two matroids over the same ground set, the problem asks to decide whether they have a common base and its search version asks to find a common base, if one exists. Another widely studied variant is the weighted decision version where with the two matroids, we are given small weights on the ground set elements and a target weight W, and the question is to decide whether there is a common base of weight at least W. From the perspective of parallel complexity, the relation between the search and the decision versions is not well understood. We make a significant progress on this question by giving a pseudo-deterministic parallel (NC) algorithm for the search version that uses an oracle access to the weighted decision.

The notion of pseudo-deterministic NC was recently introduced by Goldwasser and Grossman [GG17], which is a relaxation of NC. A pseudo-deterministic NC algorithm for a search problem is a randomized NC algorithm that, for a given input, outputs a fixed solution with high probability. In case the given matroids are linearly representable, our result implies a pseudo-deterministic NC algorithm (without the weighted decision oracle). This resolves an open question posed by Anari and Vazirani [AV20].

1 Introduction

Most often, a search problem can be efficiently solved using an oracle for a closely related decision problem. For example, if you have access to a decision oracle that tells you whether a given graph has a perfect matching, you can efficiently construct a perfect matching in a given graph using the decision oracle. Such search-to-decision reductions usually involve self-reducibility and make a linear number of oracle calls sequentially. However such reductions do not fit into the framework of parallel complexity, where one can make multiple oracle calls in parallel, but wants poly-logarithmic time complexity. For a more detailed discussion on the difference in parallel complexity of search and decision problems, see [KUW88].

Graph matching and related problems like linear matroid intersection and linear matroid matching were one of the first problems to be studied from the parallel complexity perspective [Lov79, BvzGH82]. The decision versions of these problems ask to decide the existence of the respective combinatorial substructures:

- Matching: Does a given graph contain a perfect matching – a set of disjoint edges that cover all the vertices in the graph?
- Linear Matroid Intersection: Given two sets of m vectors each, is there a set of indices $B \subseteq [m]$ that corresponds to a basis set in each of the two sets?

*Department of Computer Science, IIT Bombay. Email: besusumanta@gmail.com.
†Department of Computer Science, IIT Bombay. Email: rohitgurjar0@gmail.com.

ISSN 1433-8092

- Linear Matroid Matching/Parity: Given a set of pairs of vectors, is there a subset of pairs whose union will give a basis for the union of all pairs?

The search versions of these problems ask for constructing the respective combinatorial substructures (if one exists). The matching problem in bipartite graphs is a special case of all the three problems above (see Figure 1). A bipartite graph is a graph whose vertices can be partitioned into two parts such that every edge connects a vertex from one part to one in the other part. Even in the special case of bipartite matching, the questions of the exact parallel complexity of decision and search and whether decision and search are equivalent in a parallel sense still remain unresolved.

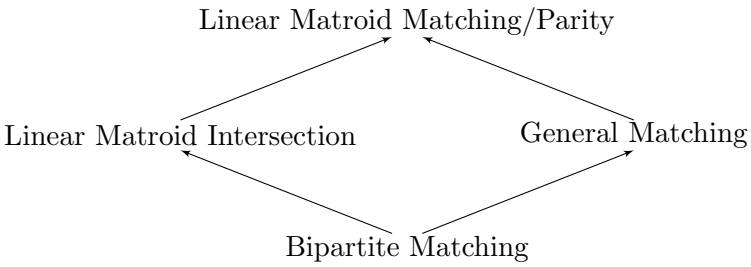

Figure 1: Reductions among the four problems. $A \to B$ represents that problem A reduces to problem B.

The first efficient randomized parallel algorithms for the three decision problems above followed from the results of Lovász [Lov79]. Lovász gave randomized algorithms for these problems by first reducing these decision questions to testing whether the determinant of a certain symbolic matrix is nonzero, as a polynomial. Then he used the fact that the zeroness of a polynomial can be tested efficiently by just evaluating it at a random point [Sch80, Zip79, DL78, Ore22]. Hence, the questions were basically reduced to computing determinant of a randomly generated matrix. Interestingly, there are efficient parallel (NC) algorithms for computing the determinant of a matrix [Ber84, Csa76, BCP84]. An NC algorithm is one which uses polynomially many parallel processors and takes only polylogarithmic time. Thus, the algorithms of Lovász [Lov79] can be viewed as randomized parallel (RNC) algorithms for the three decision problems. However, this did not imply any parallel algorithms for the search versions.

Randomized parallel (RNC) algorithms for the search versions of these problems were obtained some years later [KUW86, MVV87, NSV94]. However, these results did not go via a parallel search-to-decision reduction. Instead, they gave randomized parallel (RNC) reductions from the search version to a variant of the decision problem, namely *weighted decision*. For example, the weighted decision version for perfect matchings asks: given a graph with *small* weights on edges and a target weight W, is there a perfect matching of weight at most W (or at least W). Here the weight of a perfect matching is defined to be the sum of the weights of the edges in the perfect matching. It turns out that Lovász's RNC algorithms can be appropriately modified to solve the weighted decision versions as well, when the given weights are small. The search-to-weighted-decision reductions together with the weighted decision algorithms implied randomized parallel search algorithms for the three problems. We elaborate a bit on the reductions.

Reductions from search to weighted-decision. Karp, Upfal and Wigderson [KUW86] do not explicitly talk about weights, but their reduction is from finding a perfect matching to a subroutine that can be viewed as weighted decision with 0-1 weights on the edges. From the perspective of our current investigation, the result of Mulmuley, Vazirani, and Vazirani [MVV87] is much

more interesting. They showed that using the weighted decision oracle, one can compute a perfect matching with just two rounds of parallel calls to the oracle. The crucial ingredient in their algorithm was the powerful Isolation Lemma which states that if the edges of a graph are assigned random weights from a polynomially bounded range uniformly and independently then with high probability, there is a *unique* minimum weight perfect matching in the graph. Once we have such a weight assignment, we can first find the minimum weight w^* of a perfect matching by calling the weighted decision oracle for each possible target value W in a polynomially bounded range. Then for each edge e in parallel, delete e and ask the oracle if there is a perfect matching of weight at most w^*. The answer will be no if and only if e is a part of the unique minimum weight perfect matching. Thus, in two rounds of polynomially many parallel oracle calls, we can compute the unique minimum weight perfect matching.

The amazing thing about the Isolation Lemma is that it applies to not just the family of perfect matchings in a graph, but to arbitrary families of subsets. Thus, the above described search-to-weighted-decision reduction of [MVV87] can be made to work for any problem that admits a similar self-reducibility property. Narayan, Saran, and Vazirani [NSV94] used the same Isolation Lemma based reduction to give RNC algorithms for the search versions of linear matroid intersection and linear matroid matching.

Derandomization. Since the work of Lovász [Lov79], it has been a big open question to derandomize these results i.e., to find deterministic parallel (NC) algorithms for these problems. While derandomization results have been obtained for the matching problem in many special classes of graphs [DKR10, TV12, DK98, GK87, AHT07], the question remains open even for bipartite graphs. Only recently, there was a significant progress made when a quasi-NC algorithm was obtained for finding a perfect matching in a bipartite graph [FGT16, FGT19]. A quasi-NC algorithm runs in polylogarithmic time but can use quasipolynomially ($2^{\log^{O(1)} n}$) many parallel processors, so this result brought the problem quite close to the class NC. Similar quasi-NC algorithms were later obtained for linear matroid intersection [GT17] and matching in general graphs [ST17] as well.

In the quest of understanding the deterministic parallel complexity of these problems, an interesting question one can ask is whether there is a deterministic parallel (NC) search-to-decision reduction. An easier question would be to ask for an NC reduction from search to weighted-decision, i.e., derandomizing the reductions of [MVV87, KUW86, NSV94] described above. Soon after the quasi-NC result for bipartite matching [FGT16], Goldwasser and Grossman [GG17] started quite an interesting line of enquiry, where they answered the above question positively for bipartite matching. They observed that the quasi-NC algorithm can be modified to give a deterministic parallel (NC) search-to-weighted-decision reduction for bipartite matching. Their main result was, what they call, a pseudo-deterministic NC algorithm for bipartite matching, which followed from this reduction.

Pseudo-determinism. The notion of pseudo-deterministic algorithms was introduced by Gat and Goldwasser [GG11] which is applicable only for search problems. For a given instance of a search problem, a randomized algorithm can possibly give different outputs for different choices of the random seed. Pseudo-deterministic algorithms are randomized algorithms which give a fixed output for a given input with high probability. Note that the earlier described RNC algorithm of [MVV87] for matching is not pseudo-deterministic because for a given graph, it will output different perfect matchings for different possibilities of the randomly chosen weight assignment.

It is not hard to see that if one gives a *deterministic* reduction from a search problem to a decision problem that is known to have a randomized algorithm, then one immediately gets a pseudo-deterministic algorithm for the search problem (see [GGR13, Theorem 2.2]). That is why

the NC search-to-weighted-reduction for bipartite matching [GG17] implied a pseudo-deterministic NC algorithm for bipartite matching, i.e., an RNC algorithm that, for a given graph, outputs the same perfect matching with high probability. One interesting implication of this result is that if one finds an NC algorithm for the weighted-decision of bipartite matching, one will get an NC algorithm for the search version as well.

A natural question arises: can we similarly modify the quasi-NC algorithms for linear matroid intersection [GT17] and matching in general graphs [ST17] into NC search-to-weighted-decision reductions, and thus, get pseudo-deterministic NC algorithms for the search versions? It looks quite possible because one can can extract out an abstract framework from [GG17] for converting these quasi-NC algorithms into pseudo-deterministic NC algorithms. But as we discuss below, a straightforward application of this framework does not work out for linear matroid intersection or matching in general graphs. A key step in [GG17] is to *compute a succinct description* of the set of all (possibly exponentially many) minimum weight perfect matchings in a weighted bipartite graph in NC, given the weighted-decision oracle. However, it is not immediately clear how to solve the analogous question in NC for linear matroid intersection or matching in general graphs. Interestingly, in an earlier work in a different context, Cygan, Gabow, and Sankowski [CGS15] had already solved this question for matching in general graphs. They had designed a procedure based on LP duality to compute a succinct description of the set of all minimum weight perfect matchings, via the weighted-decision oracle. Moreover, as observed in [San18], this procedure can also be parallelized using standard techniques. Armed with this heavy hammer, Anari and Vazirani [AV20] give an NC search-to-weighted-decision reduction, and thus, get a pseudo-deterministic NC algorithm for perfect matching in general graphs. Anari and Vazirani [AV20] put it as an open question to obtain similar results for linear matroid intersection. In this work, we take up this challenge.

Our contributions. In the setting of linear matroid intersection, the analogue of a perfect matching is referred as a *common base* – a set of indices that corresponds to a basis in both the sets of vectors. For the weighted version, it is well understood how to succinctly describe the set of minimum or maximum weight common bases, i.e., the minimizing/maximizing face of the common base polytope; see e.g., [Sch03, Chapter 41]. Any face of the common base polytope is characterized by its tight sets. Suppose that M_1 and M_2 are two matroids over the same ground set E. Then, a subset S of E is called a tight set for a maximizing face (of the common base polytope), if for some matroid M_i the following holds: for every maximum weight common base B, the set $S \cap B$ spans the set S. Note that the number of tight sets of a maximizing face can be exponentially large. However, they are known to have succinct representations. We give a randomized NC algorithm to compute a succinct and unique representation for the tights sets of a maximizing face.

Theorem 1.1. *[Informal version of Theorem 7.5] There exists a randomized NC algorithm to compute a succinct and* unique *description for the tight sets of a maximizing face of the common base polytope, given the weighted-decision oracle.*

For a maximizing face of the common base polytope, all the tight sets for some matroid M_i forms a lattice family, and our description for tight sets is motivated by the succinct representation of lattice families based on the partial order of its prime subsets (also known as irreducible subsets). We construct a digraph in bottom-up fashion, using bases from the maximizing face of the common base polytope, such that it contains the necessary information regarding the tight sets of maximizing face. From this digraph we shall be able to compute the succinct description. Here, we would like to mention that the succinct representation of lattice families using the partial order of its prime subsets is well known and has been used in multiple previous algorithms [Sch03, Chapter 49],

[ILG87, EMSV12, BEL+16]. However, all these applications do not fall in the category of parallel computation.

Note that the uniqueness of the description is important because then this RNC algorithm is by default pseudo-deterministic, as there is only one possible output. Once we have designed this heavy hammer, it is relatively easier to combine the procedure of [GT17] with the abstract framework provided by [GG17] and obtain a pseudo-deterministic NC search-to-weighted-decision reduction. This leads to our first main result.

Theorem 1.2. *The search version of the linear matroid intersection problem has a pseudo-deterministic NC algorithm.*

General Matroid Intersection. Our main technical contributions are applicable to not just linear matroid intersection but also to matroid intersection. In the general matroid intersection problem, instead of two sets of vectors, we are given two matroids on the same ground set and the goal is to find a set of elements that forms a base in each of the two matroids (see Section 4 for definitions). In this problem, the matroids are not given explicitly but only via a independence or rank oracle. Thus, it does not makes sense to talk about NC or RNC algorithms for this problem. One can however consider a parallel oracle model where we can make polynomially many queries to the oracle in parallel (see [KUW88]). To the best of our knowledge, there is no such parallel algorithm known for the decision or the search version of matroid intersection, even with sub-linear number of rounds of parallel oracle calls. This makes the question all the more interesting whether decision and search are equivalent in a parallel sense.

Interestingly, the search-to-weighted-decision reduction of [NSV94] applies to general matroid intersection as well and can be said to be in RNC. Our results make a significant progress on this question by giving a pseudo-deterministic NC reduction from search to weighted decision. Formally, we can show the following.

Theorem 1.3. *There is a pseudo-deterministic NC algorithm for finding a common base of two matroids M_1 and M_2 on the same ground set E, provided that the algorithm has an oracle access to the following decision question: given two matroids with polynomially bound (in $|E|$) weights on the ground set elements and a target weight W, is there a common base of weight at least W? Furthermore, the oracle calls need to be made only for the following pairs of matroids: $\langle M_1, M_2\rangle$, $\langle M_1, M_1\rangle$, and $\langle M_2, M_2\rangle$.*

Note that in the above theorem, as there is no explicit input, the ground set size is taken as the input size.

Discussion. There are many natural open questions that are highlighted by our work. The big question is whether there is an NC algorithm for linear matroid intersection. Going to the more general setting, is there some kind of parallel algorithm for matroid intersection? Another question which can generate some new ideas is whether there is an NC reduction from search to decision for linear matroid intersection. For general matroid intersection, it would be interesting to find a parallel search to decision reduction even with the use of randomization.

The third question mentioned in the beginning, that is, linear matroid matching is completely open, in the sense that not even a quasi-NC algorithm is known for it. Given the wide applicability of the Isolation Lemma, the randomized parallel search-to-weighted-decision reduction of Mulmuley, Vazirani, and Vazirani [MVV87] would work for any combinatorial problem with an appropriate self-reducibility property, including NP-hard problems like maximum independent set. An intriguing meta-question is – what is the most general setting where we can find deterministic or pseudo-deterministic parallel search-to-weighted-decision reductions.

2 Previous works

We start by briefly describing the techniques of previous works [NSV94, GT17, GG17] that will be helpful in both comprehending as well as describing our work. Wherever these works talk about a minimization problem, we will describe it in terms of maximization, just for convenience. We will be using the following notations for the two versions of the matroid intersection problem.

- search-MI: Given two matroids on a common ground set, compute a common base.
- weighted-decision-MI: Given two matroids on the same ground set, polynomially bounded weights on the ground set elements, and a target weight W, is there a common base of weight at least W?

See Section 4 for basic terminology in matroid theory. Whenever we are in a setting where the matroids are not given explicitly, we will consider the ground set size as the input size.

The result of Narayanan, Saran, and Vazirani [NSV94] can be interpreted as an RNC reduction from search-MI to weighted-decision-MI. The first step of this reduction is to assign weights to the ground set elements, randomly and independently from a small range. Then from the Isolation Lemma [MVV87], one can say that there is a unique maximum weight common base of the two matroids, with high probability. Here, the *weight of a common base* is defined to be sum of the weights of the elements in the common base. We can first find the maximum weight w^* of a common base by calling the weighted-decision-MI oracle for each possible target value W in a small range. Then for each ground set element e in parallel, increase its weight by one and find out the new maximum weight. The maximum weight increases if and only if e is a part of the unique maximum weight common base. This way we can find the unique maximum weight common base.

Note that the uniqueness property is crucial for this construction and that is the only place where randomness is needed. And this construction is not pseudo-deterministic because for different choices of random weights, we will get a different maximum weight common base. There has been several efforts to *deterministically* construct a weight assignment in NC that *isolates* a common base, i.e., ensures unique maximum weight common base, but this goal has not been achieved till now. A recent work [GT17] came quite close to this goal and constructed an isolating weight assignment in quasi-NC. This work generalizes the ideas used to do the same for bipartite matching in [FGT16]. We build on their ideas to construct an isolating weight assignment in pseudo-deterministic NC. We first give a brief description of their result.

Isolating a common base in quasi-NC. Suppose that $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$ are two matroids over the same ground set E where $\mathcal{B}_1$ and $\mathcal{B}_2$ are the family of bases of M_1 and M_2, respectively. Let $m = |E|$ and r_1 and r_2 be the rank functions of the matroids. The main idea of [GT17] is to isolate a common base in $\log m$ rounds, where in each round they significantly reduce the set of maximum weight common bases, and finally bring it down to just one maximum weight common base. In each of these rounds, they deterministically propose a set of $\text{poly}(m)$ weight assignments, one of which will do the desired reduction in the set of maximum weight common bases. In a round, they have no way of figuring out which one out of these $\text{poly}(m)$ weight assignments will do the job. So, they have to try all $\text{poly}(m)^{\log m}$ combinations of these weight assignments. Moreover, for any particular combination, they have to combine the $\log m$ weight assignments on different scales, which means their weights become as large as $\text{poly}(m)^{\log m}$. Due to these two factors, their construction is in quasi-NC and not in NC.

To measure the progress in each round, they need a succinct way to describe the current set of maximum weight common bases. The most convenient way to understand the set of maximum

weight common bases is through the *common base polytope*. The common base polytope $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ is a polytope formed by taking convex hull of the 0-1 indicator vectors of the sets in $\mathcal{B}_1 \cap \mathcal{B}_2$. For any weight assignment $\mathbf{w} \in \mathbb{R}^E$, the weight of a common base B is defined as a linear function, and thus, one can obtain the maximum weight common bases by maximizing the function $\sum_{e \in E} \mathbf{w}_e \mathbf{x}_e$ over $P(\mathcal{B}_1 \cap \mathcal{B}_2)$. In particular, the set of maximum weight common bases will always be the set of corners of a face of $P(\mathcal{B}_1 \cap \mathcal{B}_2)$.

Edmonds [Edm70] gave a nice description of $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ using the rank functions r_1 and r_2. He showed that a point $\mathbf{x} \in \mathbb{R}^E$ is in $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ if and only if it satisfies the following constraints:

$$\mathbf{x}_e \geq 0 \quad \forall e \in E, \tag{1}$$

$$\mathbf{x}(S) = \sum_{e \in S} \mathbf{x}_e \leq r_i(S) \quad \forall S \subset E,\ i = 1, 2, \tag{2}$$

$$\mathbf{x}(E) = \sum_{e \in E} \mathbf{x}_e = r_1(E) = r_2(E). \tag{3}$$

The construction in [GT17] crucially uses the description of the common base polytope $P(\mathcal{B}_1 \cap \mathcal{B}_2)$. In terms of the polytope, their construction of the weight assignment is such that in each round, the maximum weight face of $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ gets significantly smaller and after $\log m$ rounds, the maximum weight face is simply a corner point. The key notions they introduced to measure the improvement in each iteration are *cycles* with respect to a face and their *circulations* with respect to a weight assignment.

Suppose that F is a face of $P(\mathcal{B}_1 \cap \mathcal{B}_2)$. A subset S of E is called a *tight set* of M_i with respect to F if the corresponding inequality in (2) is tight for F i.e, for all $\mathbf{x} \in F$, $\mathbf{x}(S) = r_i(S)$. Then [GT17] showed that for every face F, we have two partitions of E, denoted by $\mathsf{partition}_1[F]$ and $\mathsf{partition}_2[F]$, such that every tight set of M_i with respect to F is a union of the sets from $\mathsf{partition}_i[F]$. The partitions of E naturally induce a bipartite graph, denoted by $\mathcal{G}[F]$, with the left vertex set $\mathsf{partition}_1[F]$, the right vertex set $\mathsf{partition}_2[F]$ and the edge set E: the edge corresponding to an element $e \in E$ is incident on the vertex corresponding to a set $v \in \mathsf{partition}_i[F]$ if and only if $e \in v$. A sequence of distinct elements $(e_1, \ldots, e_k)$ from E is called a *cycle* with respect to F if it forms a cycle in the graph $\mathcal{G}[F]$.

Let $\mathcal{C}_F$ denotes the set of cycles with respect to a face F of $P(\mathcal{B}_1 \cap \mathcal{B}_2)$. Then [GT17] showed that for face F, if $\mathcal{C}_F = \emptyset$ then F is a corner point of the polytope $P(\mathcal{B}_1 \cap \mathcal{B}_2)$. Their idea was to keep eliminating cycles via appropriate modification of the weight assignment and get smaller and smaller maximizing face of $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ to eventually reach a corner point. For a weight assignment $\mathbf{w}$ on E, define the *circulation* for a (even length) cycle as the absolute value of the difference of weights in the two sets of alternating edges. Let C be a cycle, say with respect to $F = P(\mathcal{B}_1 \cap \mathcal{B}_2)$, and let $\mathbf{w}$ be a weight assignment such that the circulation of C is non-zero w.r.t. $\mathbf{w}$. Then they showed that the cycle C does not appear in the maximizing face with respect to $\mathbf{w}$. Now if the weight assignment $\mathbf{w}$ gives non-zero circulation to *all* the cycles in $P(\mathcal{B}_1 \cap \mathcal{B}_2)$, then all the cycles in the maximizing face F will be eliminated, i.e. $\mathcal{C}_F = \emptyset$, and F will be a corner. However, with polynomially bounded weights, one cannot expect to give nonzero circulation to all the cycles at once, since the number of cycles can be exponentially large.

One of the key ideas in [GT17, FGT16] was to eliminate the cycles in rounds. In each round, they double the length of the eliminated cycles and reach to face of a smaller dimension. Thus, in $\log m$ rounds, one can eliminate all the cycles and reach a corner point of $P(\mathcal{B}_1 \cap \mathcal{B}_2)$. They used the fact that if in a graph all the cycles have length greater than 2^i, then there are at most m^4 many cycles of length at most 2^{i+1} [Sub95]. This implies that, at each iteration, we have to give nonzero circulation to at most m^4 many cycles. Using a hashing technique (for example see [FGT16, Lemma

2.3]), one can design a family of $O(m^6)$ many weight assignments with weights bounded by $O(m^6)$ such that for any set of m^4 many cycles, one of the weight assignments gives nonzero circulation to each of m^4 many cycles. Now, as described earlier, we consider all possible combinations of weight assignments from different rounds to get a family of $\text{poly}(m^{\log m})$ many weight assignments with weights bounded by $\text{poly}(m^{\log m})$ such that for any two matroids on a ground set of size m, at least one weight assignment isolates a common base.

In this paper, we give a *pseudo-deterministic NC* reduction from search-MI to weighted-decision-MI. This line of work was started by Goldwasser and Grossman [GG17]. One can extract an abstract framework from [GG17] with the following two steps to get a pseudo-deterministic NC search-to-weighted-decision reduction: 1) Like [FGT16, GT17], an iterative approach of designing an isolating weight assignment family, 2) Succinct representation of the maximum weight faces of the underlying polytope with an RNC algorithm to compute it, assuming the oracle access to the weighted decision. For example, a face of the bipartite matching polytope is completely described by the set edges that participate in some perfect matching in that face, and [GG17] gives an NC algorithm to compute it using the respective weighted decision oracle.

The faces of the perfect matching polytope for general graphs are more complicated than their bipartite counterpart. Here, any face is described by a maximal laminar family of tight odd cuts. The work of [CGS12, San18] give an NC procedure, with the oracle access to the weight decision problem, to compute a maximal laminar family of tight odd cuts. This result supplies the second ingredient of the [GG17] framework, which helped [AV20] give an NC reduction from search to weighted decision for general perfect matching.

Our reduction also follows the abstract framework of [GG17]. We use the iterative approach developed by [GT17]. On top of that, we need an RNC algorithm (using the oracle access to weighted-decision-MI) to compute a succinct representation for a maximum weight face of the common base polytope $P(\mathcal{B}_1 \cap \mathcal{B}_2)$. However, none of the previous ideas help to answer this question, and we need something completely new.

3 Proof techniques

In this section, we briefly describe the proof ideas of our results. Our proofs strongly rely on some structural properties of lattice families over finite sets. Therefore, we briefly discuss the necessary notations and facts about lattice families. For a finite set E, a family of subsets $\mathcal{L}$ of E is called a *lattice family* over E if it is closed under set union and intersection and for every element $a \in E$ there exists a set in $\mathcal{L}$ containing a. For every element $a \in E$ there exists a *unique* smallest set in $\mathcal{L}$ containing a. Such sets are called as *prime sets* of $\mathcal{L}$. All the sets in a lattice family can be written as a union of its prime sets. Every lattice family $\mathcal{L}$ over E induces a *unique partition* $\mathcal{P}$ of E such that every set in $\mathcal{L}$ is a disjoint union of sets in $\mathcal{P}$. Moreover, the sets in $\mathcal{P}$ can be written as a sequence $(S_1, \ldots, S_\ell)$ with the following property: for all $k \in [\ell]$, $\cup_{j=1}^{k} S_j$ is in $\mathcal{L}$. A family $\mathcal{L}' \subseteq \mathcal{L}$ is called a *sublattice* of $\mathcal{L}$, if $\mathcal{L}'$ is also a lattice family over E. The partition $\mathcal{P}$ is a refinement of the partition $\mathcal{P}'$ induced by $\mathcal{L}'$, that is for all $S \in \mathcal{P}'$, the sets in $\mathcal{P}$ having a nonempty intersection with S form a partition of S. See Section 4.2 for the proofs of some of these properties .

3.1 Proof Idea of Theorem 1.1

We discuss a succinct representation for the maximum weight face of the common base polytope and an RNC algorithm to compute it. First, we define some notations. Supppose that $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$ are two matroids with $\mathcal{B}_1$ and $\mathcal{B}_2$ as their family of the bases and r_1 and r_2 as the rank functions, respectively. Let $m = |E|$. Let $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ be the common base polytope of M_1

and M_2 defined by the equations (1), (2), (3), and F be a face of $P(\mathcal{B}_1 \cap \mathcal{B}_2)$. Then a subset S of E is called a *tight set* for M_i (with respect to F) if for all $\mathbf{x} \in F$, $\mathbf{x}(S) = r_i(S)$. For all $i \in [2]$, let $\mathsf{tight\text{-}sets}_i[F]$ denote the family of all tight sets for M_i with respect to the face F. Edmonds [Edm70] showed that for all $i \in [2]$, $\mathsf{tight\text{-}sets}_i[F]$ forms a lattice family over E.

Suppose that $\mathbf{w}$ is a weight assignment on E. Let $F_{\mathbf{w}}$ be the maximizing face of the common base polytope $P(\mathcal{B}_1 \cap \mathcal{B}_2)$, with respect to $\mathbf{w}$. The face $F_{\mathbf{w}}$ can be uniquely represented by $\mathsf{tight\text{-}sets}_1[F_{\mathbf{w}}]$ and $\mathsf{tight\text{-}sets}_2[F_{\mathbf{w}}]$. However, we can not compute them explicitly with our limited computational resources since the size of each family can be exponentially large. On the other hand, since $\mathsf{tight\text{-}sets}_i[F_{\mathbf{w}}]$ is a lattice family over E, each $\mathsf{tight\text{-}sets}_i[F_{\mathbf{w}}]$ has a succinct representation using partial order defined on its prime sets. More specifically, one can define a pre-order $\preceq_i$ (that is, reflexive and transitive) on E as follows: for all $a, b \in E$, $a \preceq_i b$ if and only if in $\mathsf{tight\text{-}sets}_i[F_{\mathbf{w}}]$, the prime set containing b is a subset of the prime set containing a. The pre-order $\preceq_i$ gives a succinct representation of $\mathsf{tight\text{-}sets}_i[F_{\mathbf{w}}]$, that is for every $S \subseteq E$, S is in $\mathsf{tight\text{-}sets}_i[F_{\mathbf{w}}]$ if and only if S is transitively closed under $\preceq_i$. Such succinct representation for lattice familes is well known (see [Sch03, Chapter 49] [1]). For any $a \in E$, the transitive closure of a in $\preceq_i$ is same as the prime set in $\mathsf{tight\text{-}sets}_i[F_{\mathbf{w}}]$ containing a. Also, the collection of all maximal subsets of E which are symmetric under $\preceq_i$ is same as the partition E induced by $\mathsf{tight\text{-}sets}_i[F_{\mathbf{w}}]$. If one consider the digraph representaion of $\prec_i$, (that is (a, b) is an edge if and only if $a \preceq_i b$) then in $\mathsf{tight\text{-}sets}_i[F_{\mathbf{w}}]$, the prime set containing a is same the set of vertices reachable from a in the digraph and the partition of E induced by $\mathsf{tight\text{-}sets}_i[F_{\mathbf{w}}]$ is same as the set of strongly connected components. Thus, the prime sets of $\mathsf{tight\text{-}sets}_i[F_{\mathbf{w}}]$ contain all the information regarding it. In our context, we compute the following succinct objects related to $F_{\mathbf{w}}$: $\mathsf{prime\text{-}sets}_i[F_{\mathbf{w}}]$ and $\mathsf{partition}_i[F_{\mathbf{w}}]$ for all $i \in [2]$, where $\mathsf{prime\text{-}sets}_i[F]$ be the set of all primes sets of the lattice family $\mathsf{tight\text{-}sets}_i[F_{\mathbf{w}}]$ and $\mathsf{partition}_i[F_{\mathbf{w}}]$ denote the partition of E induced by $\mathsf{tight\text{-}sets}_i[F]$. Recall from the description of [GT17] (or, Definition 4.12) that the cycles of the bipartite graph induced by $\mathsf{partition}_1[F_{\mathbf{w}}]$ and $\mathsf{partition}_2[F_{\mathbf{w}}]$ define the cycles with respect to the face $F_{\mathbf{w}}$. And, the tight constraints coming from sets in $\mathsf{prime\text{-}sets}_i[F_{\mathbf{w}}]$ serve as a basis for all the tight constraints from $\mathsf{tight\text{-}sets}_i[F_{\mathbf{w}}]$. Here, we would like to mention that basis forming families of tight sets are well studied (see [Sch03]). However, to best of our knowledge, no efficient parallel algorithm is known to compute them. Also, the succinct representation of lattices using the partial order of its prime sets has been widely used to design algorithms for different optimization problems. For example, computing optimal stable matching [ILG87], problems in computational geometry [EMSV12, BEL$^+$16], submodular function minimization [Sch03, Chapter 49]. However, the context of these applications are very different from parallel computation.

With the above two objects, we also need the following characterization: for all $i \in [2]$, there exists a function $N_i^{F_{\mathbf{w}}}$ from $\mathsf{partition}_i[F_{\mathbf{w}}]$ to $\mathbb{Z}_{\geq 0}$ such that a base $B \in \mathcal{B}_1 \cap \mathcal{B}_2$ is in the face $F_{\mathbf{w}}$ if and only if for all $i \in [2]$ and $S \in \mathsf{partition}_i[F_{\mathbf{w}}]$, we have $|S \cap B| = N_i^{F_{\mathbf{w}}}(S)$. Here, we would like to mention that both the notions of partition and the function $N_i^{F_{\mathbf{w}}}$ and the criteria we just mentioned were already introduced in [GT17], but were a bit weaker in the following ways: Our criteria is an exact characterization, however, they showed it for one direction. Our partition has an additional "chain property" ensured by the structural properties of the lattice families. All these additional points will be useful in our proofs. For details see Lemma 4.11 in Section 4.5 and [GT17, Section 3.2].

Now we briefly discuss about our RNC algorithm to compute $\mathsf{prime\text{-}sets}_i[F_{\mathbf{w}}]$ and $\mathsf{partition}_i[F_{\mathbf{w}}]$ for all $i \in [2]$. One important point is that our algorithm is equipped with the oracle access to

[1] Our definition of $\preceq_i$ is exactly opposite to the definition used [Sch03, Chapter 49], that is according to their definition, $a \preceq_i b$ if and only if the prime set containing a is a subset of the prime set containing b.

weighted-decision-MI. Our idea is the following: We first compute a random vertex, equivalently a random base, B in the face $F_{\mathbf{w}}$. The base B can be computed in RNC using the oracle access to weighted-decision-MI (see Lemma 5.3). Then iteratively construct a chain of subsets of bases from $F_{\mathbf{w}}$

$$\{B\} = \mathcal{B}_0 \subseteq \mathcal{B}_1 \subseteq \cdots \subseteq \mathcal{B}_\ell$$

such that the minimal face containing $\mathcal{B}_\ell$ is same as $F_{\mathbf{w}}$ and $\ell = \lceil \log m \rceil$. Next we briefly discuss how to construct the set $\mathcal{B}_j$ from $\mathcal{B}_{j-1}$ and compute $\mathsf{prime\text{-}sets}_i[F_{\mathbf{w}}]$ and $\mathsf{partition}_i[F_{\mathbf{w}}]$ from the set of common bases $\mathcal{B}_\ell$.

For all $j \in \{0, \ldots, \ell\}$, let F_j denotes the minimal face containing $\mathcal{B}_j$. For all $j \in [\ell]$, the set $\mathcal{B}_j$ contains the elements in $\mathcal{B}_{j-1}$ with the following extra elements: For all $i \in [2]$, $A \in \mathsf{partition}_i[F_{j-1}]$, we add a common base $B_{ij}^{(A)}$ (if it exists) from the face $F_{\mathbf{w}}$ with the property

$$|A \cap B_{ij}^{(A)}| \neq N_i^{F_{j-1}}(A). \tag{4}$$

We know that for all $i \in [2]$ $A \in \mathsf{partition}_i[F_{j-1}]$, every base in F_{j-1} contains exactly $N_i^{F_{j-1}}(A)$ many elements from A. However, our property on $B_{ij}^{(A)}$ says that we want a base from $F_{\mathbf{w}}$ which violates that condition, and if exists, we can compute such a base in RNC using the oracle access to weighted-decision-MI (see Lemma 5.4). Next, we discuss how to compute $\mathsf{partition}_i[F_j]$ in NC. Note that, after computing $\mathsf{partition}_i[F_j]$, $N_i^{F_j}$ can be computed in NC by computing $|B \cap A|$, for some $B \in \mathcal{B}_j$, in parallel for all $A \in \mathsf{partition}_i[F_j]$.

The set families $\mathsf{tight\text{-}sets}_i[F_j]$ for all $i \in [2]$ form lattice families over E, and given B_j, we are interested to compute $\mathsf{prime\text{-}sets}_i[F_j]$ and $\mathsf{partition}_i[F_j]$ in NC. As we mentioned earlier, every lattice family has a digraph representation based on the partial order on primes sets of lattice family. Given this digraph representation of $\mathsf{tight\text{-}sets}_i[F_j]$, one can compute $\mathsf{prime\text{-}sets}_i[F_j]$ and $\mathsf{partition}_i[F_j]$ in NC. However, given $\mathcal{B}_j$, it is not clear how to construct the digraph representation of the lattice family $\mathsf{tight\text{-}sets}_i[F_j]$ in NC. We show that, instead of this digraph, it would sufficient for us if we work with a subgraph $G_i[\mathcal{B}_j]$ defined as follows: the vertex set is same as the ground set E and for all $a, b \in E$, (a, b) is an edge of $G_i[\mathcal{B}_j]$ if and only if

there exists a base $B \in \mathcal{B}_j$ such that $b \in B$ and $(B \setminus \{b\}) \cup \{a\}$ is also a base of M_i.

More specifically, we prove that for every $a \in E$ the prime set in $\mathsf{tight\text{-}sets}_i[F_j]$ containing a is same as the set of vertices reachable from a in $G_i[F_j]$ and $\mathsf{partition}_i[F_j]$ is same as the set of strongly connected components in $G_i[\mathcal{B}_j]$. Using this characterization, we can compute $\mathsf{prime\text{-}sets}_i[F_j]$ and $\mathsf{partition}_i[F_j]$ in NC, given the graph $G_i[\mathcal{B}_j]$. For more details see Section 6. Also, using the weighted decision oracle we can compute $G_i[\mathcal{B}_j]$ in NC (Lemma 7.4). Thus, given $\mathcal{B}_j$, $\mathsf{prime\text{-}sets}_i[F_j]$ and $\mathsf{partition}_i[F_j]$ are computable in NC. Here, we would like to mention that constructing directed graphs using base exchange property is a well known technique in matroid literature and has been used in various contexts. For example, one can see the augmenting path based algorithm for matroid intersection in [Sch03, Section 41.2], and some other context in [Sch03, Section 40.3]. The definition of $G_i[\mathcal{B}_j]$ is very close to the definition used in the second example.

At very high level, this part of our algorithm is doing exactly the opposite of the idea used to construct isolating weight assignment family in [FGT16, GT17, ST17]. They start from a face of the polytope and iteratively move to the subfaces of smaller dimensions until a corner point is reached. On the other hand, we are starting from a corner point of the face and iteratively reaching bigger faces until we cover the whole face.

Now we give a very brief overview of the correctness of our algorithm. For all $j \in \{0, 1, \ldots, \ell\}$,

since F_j is a subface of $F_{\mathbf{w}}$, $\mathsf{tight\text{-}sets}_i[F_{\mathbf{w}}]$ is a sublattice of $\mathsf{tight\text{-}sets}_i[F_j]$ for all $i \in [2]$. Therefore $\mathsf{partition}_i[F_j]$ is a refinement of $\mathsf{partition}_i[F_{\mathbf{w}}]$, that is for all $S \in \mathsf{partition}_i[F_{\mathbf{w}}]$, the sets in $\mathsf{partition}_i[F_j]$ having nonempty intersection with S create a partition of S. Let $\mathcal{W}_{ij}^{(S)}$ denote the family of sets in $\mathsf{partition}_i[F_j]$ which have nonempty intersection with $S \in \mathsf{partition}_i[F_{\mathbf{w}}]$. As we move from $(j-1)$th iteration to jth iteration, our algorithm satisfies the following property: either the size of the smallest sets in $\mathcal{W}_{ij}^{(S)}$ satisfying the equation 4 becomes double, or if no such set exists in $\mathcal{W}_{ij}^{(S)}$, it becomes equal to $\{S\}$. Thus, after ℓth iteration, $\mathsf{partition}_i[F_\ell]$ becomes equal to $\mathsf{partition}_i[F_{\mathbf{w}}]$ for all $i \in [2]$. This leads us to prove that $F_\ell = F_{\mathbf{w}}$. Therefore, $\mathsf{prime\text{-}sets}_i[F_\ell]$ is also same as $\mathsf{prime\text{-}sets}_i[F_{\mathbf{w}}]$. For details see Section 7.

3.2 Proof idea of Theorem 1.3

In this section, we give a proof overview of Theorem 1.3, which states that there is a pseudo-deterministic NC algorithm for the matroid intersection search problem that uses the weighted-decision oracle. Since the weighted-decision for *linear* matroid intersection can be solved in RNC (see Lemma 4.19), we get a pseudo-deterministic NC algorithm for the search version of linear matroid intersection, that is, Theorem 1.2.

Suppose that $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$ are two matroids with $\mathcal{B}_1$ and $\mathcal{B}_2$ as the family of bases, respectively. Let $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ be the common base polytope of M_1 and M_2. Let $\mathbf{w}_0$ be a weight assignment defined as $\mathbf{w}_0(a) = 1$ for all $a \in E$. Then the maximizing face of $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ with respect to $\mathbf{w}_0$ is the polytope itself. Let $m = |E|$ and $\ell = \lceil \log m \rceil$. Now our idea is the following: We start from the weight assignment $\mathbf{w}_0$ and inductively construct a sequence of weight assignments

$$\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_\ell$$

such that for all $j \in \{0, 1, \ldots, \ell\}$, the weights in $\mathbf{w}_j$ are bounded by $O(m)$ and the length of the shortest cycle with respect to the face F_j is greater than 2^j where F_j denotes the maximizing face with respect to $\mathbf{w}_j$. Therefore, the face F_ℓ does not have any cycle, and from [GT17], it has a unique base. Now using the oracle access to $\mathsf{weighted\text{-}decision\text{-}MI}$ the base in F_ℓ can be computed in NC (see Lemma 5.3). Next we discuss how to construct $\mathbf{w}_j$ from $\mathbf{w}_{j-1}$.

For all $j \in \{0, 1, \ldots, \ell\}$, let $\mathcal{C}_{F_j}$ denotes the set of all cycles with respect to the face F_j. From the induction hypothesis, for some j, all the cycles in $\mathcal{C}_{F_j}$ have length greater than 2^j. Then from [GT17], there are at most m^4 many cycles of length at most 2^{j+1}. Let $\mathcal{W}$ be a polynomially large family of weight assignments with polynomially bounded weights such that one of the weight assignments in $\mathcal{W}$ gives nonzero circulation to all the cycles in $\mathcal{C}_{F_j}$ of length at most 2^{j+1}. There are well known NC constructions of such a family $\mathcal{W}$ (see e.g., [FGT16, Lemma 2.3]). For each $\mathbf{w} \in \mathcal{W}$ we do the following in parallel: combine $\mathbf{w}_j$ and $\mathbf{w}$ in decreasing order of precedence. Let $\mathbf{w}'$ be the combined weight and $F_{\mathbf{w}'}$ is the maximizing face with respect to it. Now using our RNC algorithm discussed in the previous section, compute $\mathsf{prime\text{-}sets}_i[F_{\mathbf{w}'}]$ and $\mathsf{partition}_i[F_{\mathbf{w}'}]$ for all $i \in [2]$. Now, construct the bipartite graph $\mathcal{G}[F_{\mathbf{w}'}]$ from $\mathsf{partition}_1[F_{\mathbf{w}'}]$ and $\mathsf{partition}_2[F_{\mathbf{w}'}]$ as defined in the description of [GT17]. The length of the shortest cycles in $\mathcal{G}[F_{\mathbf{w}'}]$ can be computed in NC. Thus, in NC, we can compute the lexicographically smallest weight assignment $\mathbf{w} \in \mathcal{W}$ such that the length of the shortest cycles in $\mathcal{G}[F_{\mathbf{w}'}]$ is greater than 2^{j+1}.

Next we show how to compute $\mathbf{w}_{j+1}$ from $\mathbf{w}'$ such that weights in $\mathbf{w}_{j+1}$ are bounded by $O(m)$. Define $\mathbf{w}_{j+1}$ as the following:

$$\mathbf{w}_{j+1} = \sum_{i=1}^{2} \sum_{S \in \mathsf{prime\text{-}sets}_i[F_{\mathbf{w}'}]} \mathbf{1}_S,$$

where $\mathbf{1}_S \in \mathbb{R}^E$ denotes the indicator vector for the set S. From the defnition, it is clear that weights are bounded by $2m$, and can be computed in NC from $\mathsf{prime\text{-}sets}_1[F_{\mathbf{w}'}]$ and $\mathsf{prime\text{-}sets}_2[F_{\mathbf{w}'}]$. Using the description of $P(\mathcal{B}_1 \cap \mathcal{B}_2)$, we can show that every point $\mathbf{x}$ in the maximizing face F_{j+1} must satisfy $\mathbf{x}(S) = r_i(S)$ for all $i \in [2]$, $S \in \mathsf{prime\text{-}sets}_i[F_{\mathbf{w}'}]$. This implies that $\mathsf{prime\text{-}sets}_i[F_{\mathbf{w}'}]$ is a subset of $\mathsf{tight\text{-}sets}_i[F_{j+1}]$. Thus $\mathsf{tight\text{-}sets}_i[F_{\mathbf{w}'}]$ is a subset of $\mathsf{tight\text{-}sets}_i[F_{j+1}]$ since all the sets in a lattice family can be written as a union of its prime sets. This helps us to show that $F_{\mathbf{w}'}$ is same as F_{j+1}. Also, one can verify that each step of our algorithm as has a unique answer, therefore it is pseudo-deterministic. For details see Section 8.

Organization of this paper In Section 4, we discuss all the preliminaries and necessary notations. Section 5 describes various problems which can be solved efficiently using weighted-decision-MI oracle. In Section 6, we describe and prove our graph theoretic characterization of tight sets. In Section 7, we give our RNC algorithm for computing a succinct representation of a maximizing face. Finally, in Section 8, we describe our algorithm for search-MI based on weighted-decision-MI oracle.

4 Preliminaries and Notations

We use $\mathbb{R}$, $\mathbb{Z}_{\geq 0}$ to denote the set of real numbers and the set of non-negative integers, respectively. For any positive integer n, $[n]$ denotes the set of integers $\{1, 2, \ldots, n\}$. For a set E and a vector in $\mathbf{u} \in \mathbb{R}^E$, we use $\mathbf{u}_a$ for an $a \in E$ to denote the coordinate of $\mathbf{u}$ indexed by a. For any subset S of E, $\mathbf{u}(S) = \sum_{a \in S} \mathbf{u}_a$. By $\mathbf{1}$ we denote the vector whose all coordinates are 1. For any subset $S \subseteq E$, its *indicator vector* $\mathbf{1}_S \in \mathbb{R}^E$ is defined as follows: for all $a \in E$, the coordinate indexed by a is 1 if $a \in S$, otherwise it is zero. When S is a singleton set, that is $S = \{a\}$ for some $a \in E$, we use $\mathbf{1}_a$ to denote $\mathbf{1}_{\{a\}}$. For a set E, $\mathcal{P}(E)$ denotes the set of all subsets of E. For a set E, a *weight assignment* $\mathbf{w}$ on E is a function from E to $\mathbb{Z}_{\geq 0}$. It can also be represented as a vector in $\mathbb{Z}^E_{\geq 0}$. For a set E, a family of subsets $\mathcal{P}$ of E is called a *partition* of E, if the sets in $\mathcal{P}$ are mutually disjoint and for every element $a \in E$ there exists a set in $\mathcal{P}$ containing a. For a subset A of a set E, $\bar{A}$ denotes the *complement* set of A.

4.1 Isolation Lemma

Suppose that E is a finite set, and $\mathcal{F} \subseteq \mathcal{P}(E)$ is a family of subsets. A weight assignment $\mathbf{w} : E \to \mathbb{Z}_{\geq 0}$ is called *isolating* for $\mathcal{F}$ if there is a unique maximum weight subset in $\mathcal{F}$. The *Isolation Lemma* by Mulmuley, Vazirani and Vazirani [MVV87] states that for any of family subsets $\mathcal{F}$ of E, a weight assignment with small random weights is isolating for $\mathcal{F}$ with high probability. Formally, it says the following.

Lemma 4.1 (Isolation Lemma [MVV87])**.** *Let $\mathcal{F}$ be a family of subsets over a finite set E. Let $|E| = m$, $0 < \epsilon < 1$ and $\ell = \lceil \frac{2m}{\epsilon} \rceil$. Let $\mathbf{w} : E \to \mathbb{Z}_{\geq 0}$ be a weight assignment where the weights are chosen uniformly and independently from $\{0, \ldots, \ell\}$. Then, there is a unique maximum weight set in $\mathcal{F}$ with probability $\geq 1 - \epsilon$.*

For proof one can see [KS01, Lemma 4]. Although the proof shows the uniqueness of minimum weight subset, it is also applicable to show the uniqueness of maximum weight subset. The most powerful thing about the Isolation Lemma is that it is independent on the choice of the family and its size. It is a widely used tool in theoretical computer science [MVV87, NSV94, Wig94, KS01, AM08].

4.2 Some structural results about Lattice families

Suppose that E is a finite set. A family of subsets $\mathcal{L}$ (of E) is called a *lattice family* over E if it is closed under intersection and union, that is for all $A, B \in \mathcal{L}$ both $A \cap B$ and $A \cup B$ are in $\mathcal{L}$, and for all $a \in E$ there exists a set in $\mathcal{L}$ containing a. For every element $a \in E$, there exists a *unique* smallest set in $\mathcal{L}$ containing a. Otherwise, the intersection of the multiple smallest sets containing a gives us a smaller size set in $\mathcal{L}$ containing a. We call such a set as *prime set* of the lattice family. They can be also defined as the sets in $\mathcal{L}$ which cannot be written as a union of two smaller sets. In the literature, prime sets are referred as *join-irreducible* (or, *join-prime*) elements of $\mathcal{L}$. A subset $\mathcal{L}'$ of $\mathcal{L}$ is called a *sublattice* of $\mathcal{L}$, if $\mathcal{L}'$ is also a lattice family over E. Here, we describe some structural properties about lattice families which will be useful for us. For more detailed study on lattice families one can see [Sta11, Chapter 3].

Lemma 4.2. *Let $\mathcal{L}$ be a lattice family over a finite set E. Then, every set in $\mathcal{L}$ can be written as a union of prime sets of $\mathcal{L}$.*

The proof directly follows from the definition of prime sets. Next we show the existence of a partition of E such that every set in $\mathcal{L}$ is a disjoint union of sets from the partition.

Lemma 4.3. *Let $\mathcal{L}$ be a lattice family over a finite set E. Then there exists a unique partition $\mathcal{P}$ of E such that every set in $\mathcal{L}$ is a disjoint union of sets from $\mathcal{P}$. Furthermore, the sets in the partition $\mathcal{P}$ can be written in a sequence $(S_1, \ldots, S_\ell)$ such that for all $i \in [\ell]$, the set $\cup_{j=1}^{i} S_i$ is in $\mathcal{L}$.*

Proof. Let $\mathcal{I}$ be the prime sets of $\mathcal{L}$. Let θ be a mapping defined over $\mathcal{I}$ as follows: for all $I \in \mathcal{I}$,

$$\theta(I) := I \setminus (\cup_{I' \in \mathcal{I}: I' \subset I} I').$$

Let $\mathcal{P} = \{\theta(I) \mid I \in \mathcal{I}\}$. Next, we show that $\mathcal{P}$ is a partition of E.

Let a be an element of E. Let I_a be the smallest set in $\mathcal{I}$ containing a. Then $\theta(I_a)$ contains a. Therefore, for every element $a \in E$, there exists a set in $\mathcal{P}$ containing a. Let I_1, I_2 be two distinct elements in $\mathcal{I}$ such that $\theta(I_1)$ and $\theta(I_2)$ are not disjoint, and let $a \in \theta(I_1) \cap \theta(I_2)$. Then, both I_1 and I_2 are the smallest sets in L containing a, which is a contradiction. Thus, $\mathcal{P}$ is a partition of E.

For any $I \in \mathcal{I}$, let $C(I)$ be the subset of $\mathcal{I}$ defined as follows: for all $I' \in \mathcal{I}$, I' is in $C(I)$ if and only if $I' \subseteq I$. The set $C(I)$ can also be seen as Birkhoff's representation for the set I. For details about Birkhoff's representation see [Bir37], or [Sta11, Chapter 3]. We show that

$$I = \cup_{I' \in C(I)} \theta(I').$$

Since for all $I' \in C(I)$, $\theta(I') \subseteq I' \subseteq I$, the set $\cup_{I' \in C(I)} \theta(I')$ is a subset of I. Let a be an element in I, and I_a be the smallest set in $C(I)$ containing a. It is not hard to see I_a is also the smallest set in $\mathcal{I}$ containing a. Therefore, $\theta(I_a)$ contains a. Hence, $I \subseteq \cup_{I' \in C(I)} \theta(I')$. This implies that every set in $\mathcal{I}$ is a disjoint union of sets from $\mathcal{P}$. Now applying Lemma 4.2, we get that every set in $\mathcal{L}$ is a disjoint union of sets from $\mathcal{P}$.

Let $\mathcal{I} = \{I_1, \ldots, I_\ell\}$ such that

$$|I_1| \leq |I_2| \leq \cdots \leq |I_\ell|.$$

Then for all $i \in [\ell]$, $C(I_i) \subseteq \{I_1, \ldots, I_i\}$. Now from the previous paragraph, we can claim that

$$\cup_{j=1}^{i} I_j = \cup_{j=1}^{i} \theta(I_j).$$

Since $\cup_{j=1}^{i} I_j$ is in $\mathcal{L}$, $\cup_{j=1}^{i} \theta(I_j)$ is also in $\mathcal{L}$. Let $S_i = \theta(I_i)$ for all $i \in [\ell]$. Then, $(S_1, \ldots, S_\ell)$ is our desired sequence of the sets in $\mathcal{P}$.

To show uniqueness, assume that there exists another partition $\mathcal{T} = \{T_1, \ldots, T_s\}$ (of E) such that every set in $\mathcal{L}$ is a disjoint union of sets from $\mathcal{T}$, and the elements of $\mathcal{T}$ can be written as a sequence

$$(T_1, \ldots, T_s)$$

such that for all $k \in [s]$, $\cup_{j=1}^{k} T_j$ is a set from $\mathcal{L}$. We show that for all $k \in [s]$, $T_i \in \mathcal{P}$. For all $k \in [s]$, let B_k be the Birkhoff's representation for the set $\cup_{j=1}^{k} T_j$, that is B_k be the subset of $\mathcal{I}$ defined as follows: for all $I \in \mathcal{I}$, I is in B_k if and only if $I \subseteq \cup_{j=1}^{k} T_j$.

First, we show that $B_k \setminus B_{k-1}$ is singleton. It is not hard to see that $B_k \setminus B_{k-1}$ is nonempty. Let I be the smallest set from $B_k \setminus B_{k-1}$. Let $B'_k = B_{k-1} \cup \{I\}$. Then $U = \cup_{I' \in B'_k} I'$ is an element in $\mathcal{L}$ such that

$$\cup_{j=1}^{k-1} T_j \subset U \subseteq \cup_{j=1}^{k} T_j.$$

This implies that $T_k \cap U$ is nonempty. Therefore, $T_k \subset U$. Hence, $U = \cup_{j=1}^{k} T_j$. One can observe that like B_k, the set B'_k also satisfies the following property: if $I' \in B'_k$ then every set I'' in $\mathcal{I}$ with $I'' \subset I'$ is also in B'_k. Thus, using Birkhhoff's representation theorem (also known as the fundamental theorem for finite distributive lattices) [Bir37], we get that $B_k = B'_k$. Now,

$$\begin{aligned} \cup_{j=1}^{k} T_j &= (\cup_{I' \in B_{k-1}} I') \cup I \\ &= (\cup_{I' \in B_{k-1}} \theta(I')) \cup \theta(I). \end{aligned}$$

Since

$$\cup_{j=1}^{k-1} T_j = \cup_{I' \in B_{k-1}} I' = \cup_{I' \in B_{k-1}} \theta(I'),$$

$T_k = \theta(I)$. Therefore, $T_k \in \mathcal{P}$. For $k = 1$, we assume both B_{k-1} and $\cup_{j=1}^{k-1} T_j$ are empty sets.

□

In the following lemma, we describe some structural properties of a sublattice.

Lemma 4.4. *Let $\mathcal{L}$ be a lattice family over a finite set E, and $\mathcal{L}'$ be a sublattice of $\mathcal{L}$. For all $a \in E$, let E_a and E'_a be the smallest sets containing a in $\mathcal{L}$ and $\mathcal{L}'$, respectively. Let $\mathcal{P}$ and $\mathcal{P}'$ be the partitions of E, as mentioned in Lemma 4.3, corresponding to $\mathcal{L}$ and $\mathcal{L}'$, respectively. For all $A \in \mathcal{P}'$, let $\mathcal{W}_A$ be the family containing all the sets from $\mathcal{P}$ which have nonempty intersection with A. Then,*

1. *for all $a \in E$, $E_a \subseteq E'_a$.*
2. *for all $A \in \mathcal{P}'$, $\mathcal{W}_A$ is a partition of A.*

Proof. Since $\mathcal{L}'$ is a sublattice of $\mathcal{L}$, E'_a is also a set in $\mathcal{L}$ containing a. Therefore, E_a is subset of E'_a. Otherwise, $E_a \cap E'_a$ is a smaller set in $\mathcal{L}$ containing a.

From Lemma 4.3, the elements in $\mathcal{P}'$ can be written as a sequence

$$(S_1, \ldots, S_\ell)$$

such that for all $k \in [\ell]$, $\cup_{j=1}^{k} S_j$ is in $\mathcal{L}'$. Since $\mathcal{L}'$ is a sublattice of $\mathcal{L}$, for all $k \in [\ell]$, $\cup_{j=1}^{k} S_j$ is also in $\mathcal{L}$. Let B be a set from $\mathcal{W}_{S_k}$ for some $k \in [\ell]$, and a be an element in $S_k \cap B$. Now we show that $B \subseteq S_k$. Since $\cup_{j=1}^{k} S_j$ is a set in $\mathcal{L}$ containing a, B is a subset of $\cup_{j=1}^{k} S_j$. On the other hand, $\cup_{j=1}^{k-1} S_j$ is a set in $\mathcal{L}$ not containing a. Thus, for all $j \in [k-1]$, $S_j \cap B = \emptyset$. Therefore, B is a subset of S_k. Also, all the sets in $\mathcal{W}_{S_k}$ are mutually disjoint and for every $a \in S_k$ there exists a set in $\mathcal{W}_{S_k}$ containing a. Therefore, $\mathcal{W}_{S_k}$ is a partition of S_k for all $k \in [\ell]$. □

4.3 Matroid Intersection Polytope

Matroid is a well studied object in both mathematics and computer science with its origin around mid 1930s. There is a huge literature on matroid theory. For example, one can see some excellent text books like [Oxl06, Sch03]. Here we mention some basic definitions and known results about matroid which will be useful for us.

An ordered pair $M = (E, \mathcal{I})$ is called *matroid* if E is a finite set and $\mathcal{I}$ is a family of subsets of E satisfying the following properties:

1. *Closure under subsets*: For every $I \in \mathcal{I}$, all its subsets are also in $\mathcal{I}$.
2. *Augmentation property:* For every $I, J \in \mathcal{I}$ with $|J| > |I|$, there exists an element a in $J \setminus I$ such that $I \cup \{a\}$ is also in $\mathcal{I}$.

The set E is called the *ground set*, and the sets in $\mathcal{I}$ are called the *independent sets*. A set in $\mathcal{I}$ is called a *base* if it is an inclusion-wise maximal set in $\mathcal{I}$. Note that by Augmentation property, every base of M has the same cardinality. Let $\mathcal{B}$ denote the collection of all bases. We use m throughout the paper to denote the cardinality of E.

A well known example of matroid is the *linear matroids*. A linear matroid is defined by a martix M in $\mathbb{R}^{k \times m}$ such that the columns are indexed by the elements of E with $|E| = m$. Let $\mathcal{I}$ be the collection of those subsets I of E such that the columns indexed by I are linearly independent. Then one can show that $(E, \mathcal{I})$ satisfies the axioms of matroid. An easy class of matroid is the *uniform matroids*. They are defined by a finite set E and a positive integer k, and the independent sets are the subsets of E with size $\leq k$. Another popular example of matroid is the *graphic matroids*. They are defined by an undirected graph $G = (V, E)$, where the ground set is the set of edges E, and the independent sets are subset of edges which form a forest. It is not hard to see that forests satisfy the matroid axioms. For more examples, see [Sch03, Chapter 39].

Matroid rank. Suppose that $M = (E, \mathcal{I})$ is a matoid. For any subset S of E, the *rank* of S is defined as

$$\max\{|I| \mid I \subseteq S \text{ and } I \in \mathcal{I}\}.$$

It is analogous to the notion of rank in linear algebra. The rank of a matroid is defined as the rank of its groud set, or equivalently, the size of its bases. The *rank function* of M, denoted by r, is a function from $\mathcal{P}(E)$ to $\mathbb{Z}_{\geq 0}$ where for all $S \in \mathcal{P}(E)$, $r(S)$ is defined as the rank of the set S. The rank function r satisfies the *submodularity property*, that is, for every $S, T \in \mathcal{P}(E)$,

$$r(S \cup T) + r(S \cap T) \leq r(S) + r(T).$$

For proof see [Sch03, Theorem 39.8].

Matroid Intersection. Suppose that $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$ are two matroids over the same ground set E. Let $\mathcal{B}_1$ and $\mathcal{B}_2$ be the set of bases, respectively. In matroid intersection, we are interested in studying the family of common independent sets which is $\mathcal{I}_1 \cap \mathcal{I}_2$, and the family of common bases which is $\mathcal{B}_1 \cap \mathcal{B}_2$. Note that the ordered pair $(E, \mathcal{I}_1 \cap \mathcal{I}_2)$ may not be a matroid anymore. There are many interesting computational problems which can formulated in the language of intersection of two matroids. For example,

- As mentioned in the introduction, given two set of vectors finding a common basis of them can be formulated as finding a common base of two linear matroids.

- Finding a perfect matching in a bipartite graph can also be seen as finding a common base of two linear matroids. For details one can see [GT17, Section 2.2]

For more examples see [GT17, Section 4.4].

Matroid Polytope. A *polytope* is a convex hull of finitely many points from $\mathbb{R}^m$. Every polytope P can be described as a intersection of halfspaces, that is $P = \{\mathbf{x} \in \mathbb{R}^m \mid A\mathbf{x} \leq \mathbf{b}\}$ where A is a martix in $\mathbb{R}^{k\times m}$ and $\mathbf{b}$ is a vector in $\mathbb{R}^k$. A set of points F in P is called *face* if and only if there exists a vector $\mathbf{c} \in \mathbb{R}^m$ such that F is the set of points in P attaining $\max\{\langle \mathbf{c}, \mathbf{x}\rangle \mid \mathbf{x} \in P\}$. Alternatively, a face F is a set of points in P if and only if F is non-empty and $F = \{\mathbf{x} \in P \mid A'\mathbf{x} = \mathbf{b}'\}$ for some subsystem $A'\mathbf{x} \leq \mathbf{b}'$ of $A\mathbf{x} \leq \mathbf{b}$. A equation $\langle \mathbf{a}, \mathbf{x}\rangle \leq b$ in the system $A\mathbf{x} \leq \mathbf{b}$ is called a *tight constraint* with respect to the face F if for every point $\mathbf{x} \in F$, $\langle \mathbf{a}, \mathbf{x}\rangle = b$. A face can be uniquely defined by its tight constraints.

Suppose that E is a finite set. Then for every family of subsets $\mathcal{F}$ (of E), a polytope $P(\mathcal{F})$ can be defined by taking the convex hull of $\{\mathbf{1}_S \in \mathbb{R}^E \mid S \in \mathcal{F}\}$. For a matroid $M = (E, \mathcal{I})$, its *matroid polytope* is defined as $P(\mathcal{I})$, that is, the convex hull of the independent sets of M. Edmonds [Edm70] gave a nice description of the matroid polytope using its rank function.

Theorem 4.5 (Matroid Polytope [Edm70])**.** *Let $M = (E, \mathcal{I})$ be a matroid with r be the rank function. Then for every $\mathbf{x} \in \mathbb{R}^E$, $\mathbf{x} \in P(\mathcal{I})$ if and only if it satisfies the following inequalities:*

$$\begin{aligned} \mathbf{x}_e &\geq 0 \quad \forall e \in E, && (5)\\ \mathbf{x}(S) &\leq r(S) \quad \forall S \subseteq E. && (6) \end{aligned}$$

For proof one can see [Sch03, Section 40.2]. Let $\mathcal{B}$ be the set of bases of M. Then the *matroid base polytope*, defined as $P(\mathcal{B})$, is clearly a face of of the matroid polytope. It is the set points in $P(\mathcal{I})$ which satisfy

$$\mathbf{x}(E) = r(E). \tag{7}$$

Matroid Intersection Polytope. Given two matroids $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$, the *matroid intersection polytope*, defined as $P(\mathcal{I}_1 \cap \mathcal{I}_2)$, has also a nice description.

Theorem 4.6 (Matroid Intersection Polytope[Edm70])**.** *Let $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$ be two matroids with r_1 and r_2 be the rank functions, respectively. Then*

$$P(\mathcal{I}_1 \cap \mathcal{I}_2) = P(\mathcal{I}_1) \cap P(\mathcal{I}_2).$$

That is, for every point $\mathbf{x} \in \mathbb{R}^E$, $\mathbf{x}$ is in $P(\mathcal{I}_1 \cap \mathcal{I}_2)$ if and only if it satisfies the following inequalities:

$$\begin{aligned} \mathbf{x}_e &\geq 0 \quad \forall e \in E, && (8)\\ \mathbf{x}(S) &\leq r_i(S) \quad \forall S \subseteq E,\ i = 1, 2. && (9) \end{aligned}$$

For proof one can see [Sch03, Section 41.4]. Let $\mathcal{B}_1$ and $\mathcal{B}_2$ be the set of bases, respectively. Then the common base polytope $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ can be defined as the set of points in $P(\mathcal{I}_1 \cap \mathcal{I}_2)$ which satisfy

$$\mathbf{x}(E) = r_i(E),\ i = 1, 2. \tag{10}$$

This implies that $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ is same as $P(\mathcal{B}_1) \cap P(\mathcal{B}_2)$. Also, for any face F of $P(\mathcal{B}_1 \cap \mathcal{B}_2)$, $F = F_1 \cap F_2$ where F_i is the face of $P(\mathcal{B}_i)$ defined by the tight constraints of F which correspond to the matroid M_i.

4.4 Combining multiple weight assignments to a single one

Suppose that $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$ are two matroids with the family of bases $\mathcal{B}_1$ and $\mathcal{B}_2$, respectively. Let $\mathbf{w}_0, \ldots, \mathbf{w}_\ell$ be a sequence of weight assignments on E, and $F_0, \ldots, F_\ell$ be a sequence of faces of the common base polytope $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ with the following properties:

1. F_0 be the set of points in $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ maximizing the weight assignment $\mathbf{w}_0$.
2. For all $i \in [\ell]$, F_i is the set of points in F_{i-1} maximizing the weight assignment $\mathbf{w}_i$.

We combine the weight assignments $\mathbf{w}_0, \ldots, \mathbf{w}_\ell$ in decreasing precedence. It is a standard trick used in many othe previous works, for example [FGT16, GT17, ST17]. Let N be a positive integer larger than the weight of any point in $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ with respect to any of these weight assignments. For $i = 0, 1, \ldots, \ell$, define a weight assignment

$$W_i = \mathbf{w}_0 N^i + \cdots + \mathbf{w}_i N^0.$$

Now we show that

Lemma 4.7. *The face F_ℓ is the set of points in $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ maximizing the weight assignment W_ℓ.*

Proof. We use induction to prove the lemma. We show that for all $i = 0, \ldots, \ell$, F_i is the set of maximizing points in $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ with respect to the weight assignment W_i. It is clearly true for $i = 0$. From the induction hypothesis, assume that from some $i \in [\ell]$, F_{i-1} is same as the set of maximizing points in $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ with respect to the weight assignment W_{i-1}. Then F_{i-1} is also same as the set of maximizing points in $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ with respect to the weight assignment NW_{i-1}. Since the construction of W_i promises that NW_{i-1} always dominates $\mathbf{w}_i$, the set of maximizing points in $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ with respect to $W_i = NW_{i-1} + \mathbf{w}_i$ is a subset of F_{i-1}. Moreover, it exactly contains the maximizing points in F_{i-1} with respect to $\mathbf{w}_i$, that is same as the points in F_i. □

4.5 Face of the Matroid Intersection Polytope

Suppose that $M = (E, \mathcal{I})$ is a matroid with r is the rank function and $\mathcal{B}$ is the set of bases. Let $P(\mathcal{B})$ be the matroid base polytope, and A be a set of points from $P(\mathcal{B})$. For a subset S of E, S is called a *tight set* with respect to A if for all $\mathbf{x} \in A$, $\mathbf{x}(S) = r(S)$. We use $\mathsf{tight\text{-}sets}[A]$ to denote the family of all tight sets with respect to A. Edmonds [Edm70] showed the following structural property for all the tight sets of a point in $P(\mathcal{B})$.

Lemma 4.8 ([Edm70])**.** *For any point $\mathbf{x} \in P(\mathcal{B})$ and any sets $S, T \subseteq E$, if $\mathbf{x}(S) = r(S)$ and $\mathbf{x}(T) = r(T)$ then*

$$\mathbf{x}(S \cup T) = r(S \cup T) \text{ and } \mathbf{x}(S \cap T) = r(S \cap T).$$

Proof. From the lemma hypothesis,

$$\begin{aligned} r(S) + r(T) = \mathbf{x}(S) + \mathbf{x}(T) &= \mathbf{x}(S \cup T) + \mathbf{x}(S \cap T) \\ &\leq r(S \cup T) + r(S \cap T) \\ &\leq r(S) + r(T). \end{aligned}$$

The first inequality follows from the inequalities of the matroid polytope (Equation 5). The second inequality comes from the submodularity property of the rank function. Therefore, to make all the inequalities to equalities, $\mathbf{x}(S \cup T)$ and $\mathbf{x}(S \cap T)$ must be equal to $r(S \cup T)$ and $r(S \cap T)$, respectively. □

Using the above lemma, we get the following structural property of the family $\mathsf{tight\text{-}sets}[A]$.

Corollary 4.9. *Let $M = (E, \mathcal{I})$ be a matroid with r be the rank function and $\mathcal{B}$ be the set of bases. Let A be a set of points from $P(\mathcal{B})$. Then $\mathsf{tight\text{-}sets}[A]$ is a lattice family over A. Furthermore, for any subset A' of A, $\mathsf{tight\text{-}sets}[A]$ is a sublattice of $\mathsf{tight\text{-}sets}[A']$.*

Proof. Let $\mathbf{x}$ be a point in A. Then, from Lemma 4.8, the family of subsets $\mathsf{tight\text{-}sets}[\{\mathbf{x}\}]$ (over E) is closed under intersection and union. Since

$$\mathsf{tight\text{-}sets}[A] = \cap_{\mathbf{x}\in A}\mathsf{tight\text{-}sets}[\{\mathbf{x}\}],$$

$\mathsf{tight\text{-}sets}[A]$ is also closed under intersection and union. Also, for every $\mathbf{x} \in E$ there exists a set in $\mathsf{tight\text{-}sets}[A]$ containing $\mathbf{x}$ since $\mathbf{x}(E) = r(E)$ for all $\mathbf{x} \in A$. Therefore $\mathsf{tight\text{-}sets}[A]$ is a lattice family over E. Since A' is a subset of A, $\cap_{\mathbf{x}\in A}\mathsf{tight\text{-}sets}[\{\mathbf{x}\}]$ is also a subset of $\cap_{\mathbf{x}\in A'}\mathsf{tight\text{-}sets}[\{\mathbf{x}\}]$. Therefore $\mathsf{tight\text{-}sets}[A]$ is a sublattice of $\mathsf{tight\text{-}sets}[A']$. □

The above corollary promise us that $\mathsf{tight\text{-}sets}[A]$ is a lattice family over E. We use $\mathsf{prime\text{-}sets}[A]$ to denote the family of prime sets of $\mathsf{tight\text{-}sets}[A]$. According to Lemma 4.3, the lattice family $\mathsf{tight\text{-}sets}[A]$ induces a unique partition of E. We denote it by $\mathsf{partition}[A]$.

Suppose that $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$ are two matroids with r_1 and r_2 are the rank functions and $\mathcal{B}_1$ and $\mathcal{B}_2$ are the set of bases, respectively. Let A be a set of points in $P(\mathcal{B}_1 \cap \mathcal{B}_2)$. For all $i \in [2]$, $\mathsf{tight\text{-}sets}_i[A]$ denotes the family of tight sets with respect to A and correspond to the matroid M_i, that is a subset S of E is in $\mathsf{tight\text{-}sets}_i[A]$ if and only if $\mathbf{x}(S) = r_i(S)$ for all $\mathbf{x} \in A$. Like Corollary 4.9, we get the following structural property of $\mathsf{tight\text{-}sets}_i[A]$.

Corollary 4.10. *Let $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$ be two matroids with the family of bases $\mathcal{B}_1$ and $\mathcal{B}_2$ and the rank functions r_1 and r_2, respectively. Let A be a set of points from $P(\mathcal{B}_1 \cap \mathcal{B}_2)$. Then for all $i \in [2]$, $\mathsf{tight\text{-}sets}_i[A]$ is a lattice family over E. Furthermore, for any subset A' of A, $\mathsf{tight\text{-}sets}_i[A]$ is a sublattice of $\mathsf{tight\text{-}sets}_i[A']$ for all $i \in [2]$.*

Now we describe a structural property of the faces of $P(\mathcal{B}_1 \cap \mathcal{B}_2)$. A slightly weaker version of this result was shown in [GT17, Lemma 3.4].

Lemma 4.11. *Let $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$ be two matroids with the family of bases $\mathcal{B}_1$ and $\mathcal{B}_2$ and the rank functions r_1 and r_2, respectively. Let F be a face of the polytope $P(\mathcal{B}_1 \cap \mathcal{B}_2)$. For all $i \in [2]$, let $\mathsf{partition}_i[F]$ denotes the partition of E, as promised by Lemma 4.3, corresponding to the lattice family $\mathsf{tight\text{-}sets}_i[F]$. Then for every $i \in [2]$ there exists a function N_i^F from $\mathsf{partition}_i[F]$ to $\mathbb{Z}_{\geq 0}$ with the following properties:*

1. *Every element in $\mathsf{tight\text{-}sets}_i[F]$ is a disjoint union of sets from $\mathsf{partition}_i[F]$.*
2. *If for some $a \in E$, $\mathbf{x}_a = 0$ for all $\mathbf{x} \in F$, then for all $i \in [2]$ there exists a singleton set S in $\mathsf{partition}_i[F]$ containing a and $N_i^F(S) = 0$.*
3. *For every base $B \in \mathcal{B}_1 \cap \mathcal{B}_2$, B is in the face F if and only if for all $i \in [2]$ $S \in \mathsf{partition}_i[F]$, $|B \cap S| = N_i^F(S)$.*

The differences between the above lemma and [GT17, Lemma 3.4] are the following: The point 3 in the conclusion of the above lemma is an exact characterization for a common base to be in F. On the other hand, [GT17, Lemma 3.4] showed only one direction. Our definition of partitions have the additional "chain property" as promised by Lemma 4.3. These additional properties will be used in our proofs. Also, our proof is different from them. We prove it using the structural properties of lattice families shown in Section 4.2 which may have independent interest.

Proof. From Lemma 4.3, we know that for all $i \in [2]$, the elements of $\mathsf{partition}_i[F]$ can be written as a sequence

$$(S_{i1}, \ldots, S_{i\ell_i})$$

such that for all $k \in [\ell_i]$, $\cup_{j=1}^{k} S_{ij}$ is an element in $\mathsf{tight\text{-}sets}_i[F]$. Now, for all $i \in [2]$, the function N_i^F is defined as follows: for all $k \in [\ell_i]$,

$$N_i^F(S_{ik}) := r_i\left(\cup_{j=1}^{k} S_{ij}\right) - r_i\left(\cup_{j=1}^{k-1} S_{ij}\right).$$

From the definition of the rank function, it is not hard to see that for all $k \in [\ell_i]$, $N_i^F(S_{ik})$ is a non-negative integer. Lemma 4.3 also ensures that every element in $\mathsf{tight\text{-}sets}_i[F]$ is a disjoint union of sets from $\mathsf{partition}_i[F]$. Next, we show the other two properties mentioned in the lemma.

Let $a \in E$ such that for all $\mathbf{x} \in F$, $\mathbf{x}_a = 0$. Let $a \in S_{ik}$ for some $i \in [2]$ and $k \in [\ell_i]$. Now we show that S_{ik} is a singleton set. For the sake of contradiction, assume that S_{ik} is not a singleton set, and b be an element from $S_{ik} \setminus \{a\}$. Let E_b the smallest set in $\mathsf{tight\text{-}sets}_i[F]$ containing b. Then $a \in E_b$ since every set in $\mathsf{tight\text{-}sets}_i[F]$ is a disjoint union of sets from $\mathsf{partition}_i[F]$. Let $E'_b = E_b \setminus \{a\}$. From the definition of the rank function,

$$r_i(E_b) \geq r_i(E'_b).$$

On the other hand, since $E_b \in \mathsf{tight\text{-}sets}_i[F]$, for all $\mathbf{x} \in F$,

$$\begin{aligned}\mathbf{x}(E_b) = r_i(E_b) &= \mathbf{x}_a + \mathbf{x}(E'_b)\\ &= \mathbf{x}(E'_b) \text{ since } \mathbf{x}_a = 0\\ &\leq r_i(E'_b).\end{aligned}$$

This implies that $r_i(E_b) = r_i(E'_b)$, thus for all $\mathbf{x} \in F$, $\mathbf{x}(E'_b) = r_i(E'_b)$. Therefore, $E_b \cap E'_b$ is a smaller set in $\mathsf{tight\text{-}sets}_i[F]$ containing b. This is a contradiction. Hence, S_{ik} is a singleton set containing a. Now

$$\begin{aligned}N_i^F(S_{ik}) &= r_i\left(\cup_{j=1}^{k} S_{ij}\right) - r_i\left(\cup_{j=1}^{k-1} S_{ij}\right)\\ &= \mathbf{x}\left(\cup_{j=1}^{k} S_{ij}\right) - \mathbf{x}\left(\cup_{j=1}^{k-1} S_{ij}\right) \text{ for any } \mathbf{x} \in F\\ &= 0 \text{ since for all } \mathbf{x} \in F,\ \mathbf{x}_a = 0.\end{aligned}$$

Let B be a common base (of M_1 and M_2) in the face F. Let $i \in [2]$. Then for all $k \in [\ell_i]$,

$$|B \cap \left(\cup_{j=1}^{k} S_{ij}\right)| = r_i\left(\cup_{j=1}^{k} S_{ij}\right).$$

Since all the sets in $\mathsf{partition}_i[F]$ are mutually disjoint,

$$|B \cap S_{ik}| = r_i\left(\cup_{j=1}^{k} S_{ij}\right) - r_i\left(\cup_{j=1}^{k-1} S_{ij}\right) = N_i^F(S_{ik}).$$

For the converse direction, assume that B is a common base of M_1 and M_2 such that for all $i \in [2]$ $k \in [\ell_i]$, $|B \cap S_{ik}| = N_i^F(S)$. Let $i \in [2]$. Let $S \in \mathsf{tight\text{-}sets}_i[F]$. Then there exists a subset $\{k_1, \ldots, k_s\}$ of $[\ell_i]$, with $k_1 < \cdots < k_s$, such that

$$S = S_{ik_1} \cup \cdots \cup S_{ik_s}.$$

Now for all $\mathbf{x} \in F$,

$$\begin{aligned} r_i(S) = \mathbf{x}(S) &= \sum_{r=1}^{s} \mathbf{x}(S_{ik_r}) \\ &= \sum_{r=1}^{s} \mathbf{x}\left(\cup_{j=1}^{k_r} S_{ij}\right) - \mathbf{x}\left(\cup_{j=1}^{k_r-1} S_{ij}\right) \\ &= \sum_{r=1}^{s} r_i\left(\cup_{j=1}^{k_r} S_{ij}\right) - r_i\left(\cup_{j=1}^{k_r-1} S_{ij}\right) \\ &= \sum_{r=1}^{s} N_i^F(S_{ij}), \end{aligned}$$

where $N_i^F(S_{i0})$ is assumed to be zero. Hence, $|B \cap S| = r_i(S)$. This implies that B satisfies all the tight constraints of the face F. Therefore, B is in the face F. □

4.6 Some frequently used notations

Suppose that $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$ are two matroids with r_1 and r_2 are the rank functions and $\mathcal{B}_1$ and $\mathcal{B}_2$ are the set of bases, respectively. Now we mention some notations which will be frequently used in our work. Some of the them already defined. However, it would be helpful to keep our important notations in a single place. For a subset $A \subseteq P(\mathcal{B}_1 \cap \mathcal{B}_2)$ and $i \in [2]$,

- $\mathsf{tight\text{-}sets}_i[A]$ denotes the family of all tight sets with respect to A, that is a subset S of E is in $\mathsf{tight\text{-}sets}_i[A]$ if and only if $\mathbf{x}(S) = r_i(S)$ for all $\mathbf{x} \in A$. Applying Corollary 4.10, we know that $\mathsf{tight\text{-}sets}_i[A]$ forms a lattice family over E.

- $\mathsf{prime\text{-}sets}_i[A]$ denotes the family of prime sets of the lattice family $\mathsf{tight\text{-}sets}_i[A]$, that is $\mathsf{prime\text{-}sets}_i[A] = \{E_i[a] \mid a \in E\}$ where $E_i[a]$ is the smallest set in $\mathsf{tight\text{-}sets}_i[A]$ containing a. Using Lemma 4.2, every set in $\mathsf{tight\text{-}sets}_i[A]$ can be expressed as a union of sets from $\mathsf{prime\text{-}sets}_i[A]$.

- As promised by Lemma 4.3, $\mathsf{partition}_i[A]$ denotes the partition of E induced by the lattice family $\mathsf{tight\text{-}sets}_i[F]$. It is already defined in Lemma 4.11.

- When A is a face of $P(\mathcal{B}_1 \cap \mathcal{B}_2)$, N_i^A denotes the function, as promised by Lemma 4.11, from $\mathsf{partition}_i[A]$ to $\mathbb{Z}_{\geq 0}$.

- $\mathcal{G}[A]$ denotes the bipartite graph with $\mathsf{partition}_1[A]$ is the left part of the vertex set, $\mathsf{partition}_2[A]$ is the right part of the vertex set and E is the set of edges. For an edge $a \in E$, S_1 and S_2 are its end points if and only if S_1 and S_2 are the sets in $\mathsf{partition}_1[A]$ and $\mathsf{partition}_2[A]$, respectively, containing a.

4.7 Cycles in Matroid Intersection

Suppose that $\mathcal{B}_1$ and $\mathcal{B}_2$ are the set of bases for matroids M_1 and M_2, respectively. In this section, we define the notion of cycle for matroid intersection and its properties which we will need. Most of definitions and results are borrowed from [GT17, Section 3.3].

Definition 4.12 (Cycle)**.** *Let F be a face in the polytope $P(\mathcal{B}_1 \cap \mathcal{B}_2)$. Then, a sequence $(e_1, \ldots, e_{2r})$ of distinct elements from E is called a* cycle *with respect to the face F, if the consecutive pairs are alternately in a set from $\mathsf{partition}_1[F]$ and a set from $\mathsf{partition}_2[F]$. That is, for all $i \in [r]$*

$$e_{2i-1}, e_{2i} \in S, \text{ for some } S \in \mathsf{partition}_1[F]$$
$$e_{2i}, e_{2i+1} \in T, \text{ for some } T \in \mathsf{partition}_2[F],$$

where $e_{2i+1} = e_1$. Alternatively, a sequence $(e_1, \ldots, e_{2r})$ of distinct elements from E is called a cycle *with respect to the face F, if it forms a cycle in the bipartite graph $\mathcal{G}[F]$.*

To motivate the above the definition of cycle, one can observe that it becomes same as the cycle in a bipartite graph when we reduce the bipartite graph matching problem to linear matroid intersection problem. Also, one can observe that if for an element $e \in E$, $\mathbf{x}_e = 0$ for all $\mathbf{x} \in F$, it does not participate in any cycle since from Lemma 4.11 it appears as a singleton set in both $\mathsf{partition}_1[F]$ and $\mathsf{partition}_2[F]$. Using the notion of cycle, [GT17] shows a nice characterization for the faces containing a single vertex, or equivalently, a corner point of the polytope $P(\mathcal{B}_1 \cap \mathcal{B}_2)$. Such faces contain a unique common base of M_1 and M_2 since the corner points of $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ are exactly same as the set of common bases $\mathcal{B}_1 \cap \mathcal{B}_2$. For a face F, let $\mathcal{C}_F$ denotes the set of all cycles with respect to the face F.

Lemma 4.13. *For a face F of the polytope $P(\mathcal{B}_1 \cap \mathcal{B}_2)$, if $\mathcal{C}_F = \emptyset$, then F contains a unique common base of M_1 and M_2.*

The proof comes from the observation that if a face F contains at least two bases, then it must contain a cycle. For detailed proof see [GT17, Lemma 3.6 and Corollary 3.7]. On the other hand, using the notion of circulation, they give an approach to eliminate the cycles from a face so that we can reach a corner point of $P(\mathcal{B}_1 \cap \mathcal{B}_2)$.

Definition 4.14 (Circulation)**.** *For a weight assignment $\mathbf{w} : E \to \mathbb{Z}_{\geq 0}$, the* circulation $c_{\mathbf{w}}(C)$ *for a cycle $C = (e_1, \ldots, e_r)$ is defined as*

$$c_{\mathbf{w}}(C) = |\mathbf{w}(e_1) - \mathbf{w}(e_2) + \cdots - \mathbf{w}(e_r)|.$$

Lemma 4.15. *Let $\mathbf{w} : E \to \mathbb{Z}_{\geq 0}$ be a weight assignment, and $F_{\mathbf{w}}$ be the set of points in $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ maximizing the weight assignment $\mathbf{w}$. Then for any cycle C with respect to $F_{\mathbf{w}}$, $c_{\mathbf{w}}(C) = 0$.*

The converse of the above lemma says that if $c_{\mathbf{w}}(C) \neq 0$, then C does not appear in $\mathcal{C}_{F_{\mathbf{w}}}$. To prove this lemma, take a relatively interior point $\mathbf{a}$ in the face $F_{\mathbf{w}}$. For example, consider $\mathbf{a}$ as the average of all the corner points of $F_{\mathbf{w}}$. Let $\delta^C = \sum_{i=1}^{r} (-1)^i \mathbf{1}_{e_i}$. Then we move to a new point from $\mathbf{b} = \mathbf{a} + \epsilon \delta^C$ for some sufficiently small constant $\epsilon > 0$ such that we remain inside the face $F_{\mathbf{w}}$. Now, without loss of generality, if we assume $\langle \mathbf{w}, \delta^C \rangle > 0$, then $\mathbf{b}$ becomes a point in $F_{\mathbf{w}}$ with larger weight. This gives a contradiction. For more detailed proof see [GT17, Lemma 3.10]. In addition with the above results, we also need the following lemma.

Lemma 4.16. *Let F be a face of the polytope $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ such that $\mathcal{C}_F$ has no cycle of length r. Then one can construct a set of $O(m^6)$ many weight assignments with weights bounded by $O(m^6)$ in NC such that one of the weight assignment will give nonzero circulation to all the cycles in $\mathcal{C}_F$ of length at most $2r$.*

The proof can be divided into two parts. The first part shows that the number of cycles in $\mathcal{C}_F$ of length at most $2r$ is upper bounded by m^4. For details see [GT17, Lemma 3.11]. The other half gives

the construction of a family of weight assignments such that for any set of m^4 cycles, one of the weight assignments gives nonzero circulation to each of the m^4 cycles. Designing such a family of weight assignments is a standard trick used to give distinct weights to a small family of sets. It appears in many other results in various other forms [FKS84, CRS95, KS01, AGKS15, FGT16, ST17]. For a detailed proof one can see [FGT16, Lemma 2.3].

4.8 An RNC-algorithm for Linear Matroid Intersection

Suppose that $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$ are two linear matroids given by two matrices U and V. Let $m = |E|$. Without loss of generality, we can assume both the matrices are $n \times m$ and have full row rank. Then the decision version of linear matroid intersection problem asks to decide whether M_1 and M_2 have a common base. The search version of this problem is to compute a common base of M_1 and M_2. Narayanan, Saran and Vazirani [NSV94] gives a randomized NC algorithm for both these problems. They reduce the decision version of the linear matroid intersection problem to deciding nonzeroness of the determinant of a symbolic matrix.

Lemma 4.17. *Let X be an $m \times m$ diagonal matrix with variables on the diagonal, that is $X_{a,a} = x_a$ for all $a \in E$. Let D be a symbolic matrix defined as UXV^T. Then M_1 and M_2 have a common base if and only if* $\det(D) \neq 0$.

Proof. Applying Cauchy-Binet formula [Zen93],

$$\det(D) = \sum_{B \subseteq E \,:\, |B| = n} \det(U_B) \det(V_B) \prod_{a \in E} x_a, \tag{11}$$

where U_B and V_B are submatrices of U and V, respectively, with columns indexed by B. Thus, for a subset B of E with the cardinality n, the coefficient of $\prod_{a \in E} x_a$ is nonzero if and only if B is a base of both M_1 and M_2. Therefore, M_1 and M_2 have a common base if and only if $\det(D)$ is nonzero. □

From Lemma 4.1, we know that a random weight assignment $\mathbf{w}$ with polynomially bounded weights is isolating for any family of subsets with high probability. Now replace each variable x_a by $x^{\mathbf{w}_a}$ in Equation 11, and compute $\det(D)(x)$. Then it will remain nonzero with high probability, if both M_1 and M_2 have a common base. Moreover, the determinant of matrix with entries are low degree univariate polynomials can be computed in NC [BCP84]. Therefore, we have an RNC algorithm to decide whether two linear matroids have a common base. One can also compute the isolated base in NC. For all $a \in E$, in parallel, delete the columns indexed by a from both U and V and re-compute $\det(D)(x)$. If the maximum degree term changes, then a is in the isolated base. Thus,

Theorem 4.18 ([NSV94])**.** *Linear matroid intersection is in RNC.*

For our work, we need to compute the weight of a maximum common base, and the technique in [NSV94] also gives an RNC algorithm for that problem.

Lemma 4.19. *Let $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$ are two linear matroids given by two $n \times m$ full rank matrices U and V, respectively. Let $\mathbf{w} : E \to \mathbb{Z}_{\geq 0}$ be a weight assignment. Then the weight of a maximum weight common base can be computed in randomized NC. Furthermore, for any positive integer c, the success probability of the algorithm can be made $1 - \frac{1}{m^c}$.*

Proof. Let $\ell = 2m^{c+1}$. Let $\mathbf{w}'$ be a weight assignment such that the weights are picked uniformly and independently from $\{0, \ldots, \ell\}$. Using the techniques mentioned in Section 4.4, combine the weight assignments $\mathbf{w}$ and $\mathbf{w}'$ with decreasing order in precedence. Let $\mathbf{w}^*$ be the combined weight. Then Lemma 4.1 and 4.7 ensures that with respect to $\mathbf{w}^*$ we have a unique maximum weight common base B^* with probability $1 - \frac{1}{m^c}$. Thus, using the above mentioned technique we can compute B^* in RNC. Moreover, the construction of $\mathbf{w}^*$ promises that B^* is a maximum weight common base of M_1 and M_2 with respect to $\mathbf{w}$. Therefore compute $\mathbf{w}(B^*)$. □

5 The Weighted Decision Oracle

We assume that our algorithm is equipped with the oracle access of the following decision problem.

Definition 5.1 (weighted-decision-MI)**.** *Given two matroids $M_1 = (E, \mathcal{I}_1)$, $M_2 = (E, \mathcal{I}_2)$ with small weights on E and a target weight W, is there exists a common base of M_1 and M_2 with weight at least W?*

Suppose that $\mathcal{B}_1$ and $\mathcal{B}_2$ are the set of bases of M_1 and M_2, respectively. Then, given the oracle access to the above problem with polynomially bounded weights $\mathbf{w} \in \mathbb{Z}^E_{\geq 0}$, using the binary search one can compute the function

$$\max_{\mathbf{x} \in P(\mathcal{B}_1 \cap \mathcal{B}_2)} \langle \mathbf{w}, \mathbf{x} \rangle$$

in NC. Hence, we can say that

Lemma 5.2. *Let $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$ be two matroids with $\mathcal{B}_1$ and $\mathcal{B}_2$ be the family of bases, respectively. Then, given M_1 and M_2 with polynomially bounded weights $\mathbf{w} \in \mathbb{Z}^E_{\geq 0}$, the function*

$$\max_{\mathbf{x} \in P(\mathcal{B}_1 \cap \mathcal{B}_2)} \langle \mathbf{w}, \mathbf{x} \rangle$$

can be computed in NC, provided that the algorithm has an oracle access to weighted-decision-MI*.*

In the following lemma, we show that how to compute a maximum weight base in RNC using the oracle access to weighted-decision-MI.

Lemma 5.3. *Let $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$ be two matroids with the family of bases $\mathcal{B}_1$ and $\mathcal{B}_2$, respectively. Let $\mathbf{w}$ be a weight assignment on E with polynomially bounded weights. Then, given M_1 M_2 and $\mathbf{w}$ as inputs, a base in $\mathcal{B}_1 \cap \mathcal{B}_2$ maximizing the weight assignment $\mathbf{w}$ can be computed in randomized NC, provided that the algorithm has an oracle access to* weighted-decision-MI*. Furthermore, for every positive integer c, the success probability of the algorithm can be made $\geq 1 - \frac{1}{m^c}$, where $m = |E|$.*

Proof. Let $F_{\mathbf{w}}$ be the maximizing face of $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ with respect to $\mathbf{w}$. Let $\ell = 2m^{c+1}$, and $\mathbf{w}'$ be a weight assignment where weights are picked uniformly and independently from $\{0, 1, \ldots, \ell\}$. Then Lemma 4.1 ensures that with probability $\geq 1 - \frac{1}{m^c}$, there exists a unique maximizing base in $F_{\mathbf{w}}$ with respect to $\mathbf{w}'$. As discussed in Section 4.4, combine the weight assignments $\mathbf{w}$ and $\mathbf{w}'$ with decreasing precedence. Let W be the combined weight. Then the weights in W are also polynomially bounded. From Lemma 4.7, the maximizing face F_W (with respect to W) contains a unique base from $F_{\mathbf{w}}$.

Let B^* be the base in F_W. Now we describe how to compute the elements of B^* in NC. Let w^* be the weight of B^* with respect to W. From Lemma 5.2, with the help of the oracle access, w^*

can be computed in NC. Now for every $a \in E$, consider a weight assignment W_a defined as follows: for all $b \in E$,

$$W_a(b) = \begin{cases} W(b)+1 & \text{if } b = a \\ W(b) & \text{otherwise.} \end{cases}$$

Then an element $a \in E$ is in B^* if and only if the maximum weight base with respect to W_a has weight w^*+1. For all $a \in E$, in parallel, we can compute the weight of the maximum weight base with respect to W_a and decide whether a is in B. Lemma 5.2 ensures that this step can be done in NC. This completes our proof. □

Our next lemma can be seen as a stronger version of the previous one. Given the oracle access to weighted-decision-MI, it shows how to compute, in RNC, a maximum weight base with some additional constraint.

Lemma 5.4. *Let $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$ be two matroids with $\mathcal{B}_1$ and $\mathcal{B}_2$ be the family of bases, respectively. Let $\mathbf{w}$ be a weight assignment on E with polynomially bounded weights. Let A be a subset of E, and n be a non-negative integer. Then, given $\langle M_1, M_2, \mathbf{w}, A, n\rangle$ as input, if exists, a base B in $\mathcal{B}_1 \cap \mathcal{B}_2$ maximizing the weight assignment $\mathbf{w}$ with $|B \cap A| \neq n$ can be computed in randomized NC, provided that the algorithm has an oracle access to* weighted-decision-MI. *Furthermore, for every positive integer c, the success probability of the algorithm can be made $\geq 1 - \frac{1}{m^c}$, where $m = |E|$.*

Proof. Let $F_{\mathbf{w}}$ be the maximizing face of $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ with respect to $\mathbf{w}$, and $\mathcal{B}_{\mathbf{w}}$ be the set of common bases present in $F_{\mathbf{w}}$. Then, it is not hard to see that $\mathcal{B}_{\mathbf{w}}$ contains a base B with $|B \cap A| \neq n$ if and only if

$$\text{either } n \neq \max_{B \in \mathcal{B}_{\mathbf{w}}} |B \cap A|, \text{ or } n \neq \min_{B \in \mathcal{B}_{\mathbf{w}}} |B \cap A|.$$

Let F_A and $F_{\bar{A}}$ be the subfaces of $F_{\mathbf{w}}$ maximizing the weight assignments $\mathbf{1}_A$ and $\mathbf{1}_{\bar{A}}$, respectively. Let B be a base in $F_{\mathbf{w}}$. Then one can observe that B is also a base in F_A if and only if

$$|B \cap A| = \max_{B' \in \mathcal{B}_{\mathbf{w}}} |B' \cap A|.$$

Similarly, B is a base in $F_{\bar{A}}$ if and only

$$|B \cap A| = \min_{B' \in \mathcal{B}_{\mathbf{w}}} |B' \cap A|.$$

Let $\mathbf{w}^A$ be the combined weight of $\mathbf{w}$ and $\mathbf{1}_A$ such that $\mathbf{w}$ is given higher precedence over $\mathbf{1}_A$. Similarly, let $\mathbf{w}^{\bar{A}}$ be the combined weight of $\mathbf{w}$ and $\mathbf{1}_{\bar{A}}$ such that $\mathbf{w}$ is given higher precedence over $\mathbf{1}_{\bar{A}}$. Then from Lemma 4.7, F_A and $F_{\bar{A}}$ are the faces of $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ maximizing the weight assignments $\mathbf{w}^A$ and $\mathbf{w}^{\bar{A}}$, respectively. Using Lemma 5.3, we can compute two bases B_A and $B_{\bar{A}}$ from F_A and $F_{\bar{A}}$, respectively, in RNC with success probability $\geq 1 - \frac{1}{m^c}$. Now if $|B_A \cap A| \neq n$, output B_A. Otherwise, check whether $|B_{\bar{A}} \cap A| = n$. If it is not equal to n, output $B_{\bar{A}}$. Otherwise, there is no base B in $F_{\mathbf{w}}$ with $|B \cap A| \neq n$. □

6 Graph theoretic characterization of tight sets

Suppose that $M = (E, \mathcal{I})$ is a matroid with r is the rank function and $\mathcal{B}$ is the set of bases. Let $\mathcal{B}' \subseteq \mathcal{B}$ be a set of bases of M. We give a graph theoretic characterization for tight-sets$[\mathcal{B}']$, the tight sets with respect to $\mathcal{B}'$. For any set of bases $\mathcal{B}'$, we define a *directed* graph, *denoted by* $G[\mathcal{B}']$, as follows: The vertex set is same as the ground set E, and for any $a, b \in E$, (a, b) is an edge of

$G[\mathcal{B}']$ if and only if

there exists a base $B \in \mathcal{B}'$ such that $b \in B$ and $(B \setminus \{b\}) \cup \{a\}$ is also a base of M.

The property used to define an ordered pair (a, b) being an edge of the graph $G[\mathcal{B}']$ is known as *base exchange property.*

Lemma 6.1. *Let $M = (E, \mathcal{I})$ be a matroid with r be the rank function and $\mathcal{B}$ be the set of bases. Let $\mathcal{B}' \subseteq \mathcal{B}$ be a set of bases for M. Then for every subset $S \subseteq E$, S is in $\mathsf{tight\text{-}sets}[\mathcal{B}']$ if and only if there exists no directed edge from S to $E \setminus S$ in the graph $G[\mathcal{B}']$.*

Remark. Constructing directed graphs using base exchange property is a well known technique in matroid literature and has been used in various contexts. For example, one can see the augmenting path based algorithm for matroid intersection in [Sch03, Section 41.2], and some other context in [Sch03, Section 40.3]. The definition of $G[\mathcal{B}']$ is very close to the definition used in the second example. Also, the relation between the subset of vertices with no outgoing edges and the tight sets, in some weaker forms, is also used in those examples. The main difference with the previous ones is that the above lemma gives an exact characterization of the tight sets.

Proof. Let S be a set in $\mathsf{tight\text{-}sets}[\mathcal{B}']$. We show that there exists no directed edge from S to $E \setminus S$ in $G[\mathcal{B}']$. For the sake of contradiction, assume that there exists an $a \in S$ and a $b \in E \setminus S$ such that (a, b) is an edge of the graph $G[\mathcal{B}']$. Then, there exists a base $B \in \mathcal{B}'$ containing b such that $B' = (B \setminus \{b\}) \cup \{a\}$ is also a base of M. Since $S \in \mathsf{tight\text{-}sets}[\mathcal{B}']$, $S \cap B$ is an independent set of M with size $r(S)$. Therefore,

$$|S \cap B| = \max\{|I| \mid I \in \mathcal{I} \text{ and } I \subseteq S\}.$$

On the other hand, $S \cap B' \subseteq S$ is also an independent set of M with size greater than $|S \cap B|$, which is a contradiction. Therefore, there exists no directed edge from S to $E \setminus S$ in $G[\mathcal{B}']$.

For the converse direction, assume that $S \notin \mathsf{tight\text{-}sets}[\mathcal{B}']$. Then there exists a base $B \in \mathcal{B}'$ such that

$$|B \cap S| < r(S).$$

Let B_S denotes the set $B \cap S$, and $A \subseteq S$ be an independent set of M with $|A| = r(S)$. Then,

$$|B_S| < |A| = r(S).$$

Since both B_S and A are independent sets (of the matroid M) with $|A| > |B_S|$, there exists an element $a \in A \setminus B_S$ such that $B'_S = B_S \cup \{a\}$ is also an independent set of M. Now applying Augmentation Property (of matroid) repeatedly, we can extend the independent set B'_S to a base B' (of M) such that

$$\exists\, b \in B \setminus B_S,\ B' = (B \setminus \{b\}) \cup \{a\}.$$

This implies that (a, b) is an edge in the graph $G[\mathcal{B}']$. Since $a \in S$ and $b \in B \setminus B_S \subseteq E \setminus S$, there exists a directed edge from S to $E \setminus S$ in $G[\mathcal{B}']$. □

For a subset $\mathcal{B}' \subseteq \mathcal{B}$, let $F_{\mathcal{B}'}$ be the minimal face of $P(\mathcal{B})$ containing $\mathcal{B}'$. The set family $\mathsf{tight\text{-}sets}[F_{\mathcal{B}'}]$ forms a lattice family over E, and given the set $\mathcal{B}'$ we are interested to compute $\mathsf{prime\text{-}sets}[F_{\mathcal{B}'}]$ and $\mathsf{partition}[F_{\mathcal{B}'}]$ in NC. It is well known that every lattice family $\mathcal{L}$ over a finite set E has a succinct digraph representation $G_{\mathcal{L}}$ as follows: the vertex set is same as E and for all $a, b \in E$, (a, b) is an edge of $G_{\mathcal{L}}$ if and only if the prime set containing b is a subset of the prime set

containing a. It is not hard to show that for every $a \in E$, the set of vertices reachable from a in $G_{\mathcal{L}}$ is the prime set in $\mathcal{L}$ containing a. Also, the partition of E induced by $\mathcal{L}$ is the set of strongly connected components of $G_{\mathcal{L}}$. For more on the digraph representation of lattice families see [Sch03, Chapter 49] [2]. The digraph representation of lattice families has been widely used in designing (sequential) algorithms [ILG87, EMSV12, BEL$^+$16].

As we mentioned, given $\mathcal{B}'$, we want to compute $\mathsf{prime\text{-}sets}[F_{\mathcal{B}'}]$ and $\mathsf{partition}[F_{\mathcal{B}'}]$ in NC. However, given $\mathcal{B}'$, it is not clear how to construct the digraph representation of the lattice family $\mathsf{tight\text{-}sets}[F_{\mathcal{B}'}]$ in NC. In the next lemma, we show that instead of $G_{\mathsf{tight\text{-}sets}[F_{\mathcal{B}'}]}$, it is sufficient to work with the graph $G[\mathcal{B}']$. More specifically, we show that for every $a \in E$ the prime set in $\mathsf{tight\text{-}sets}[F_{\mathcal{B}'}]$ containing a is same as the set of vertices reachable from a in $G[F_{\mathcal{B}'}]$ and $\mathsf{partition}[F_{\mathcal{B}'}]$ is same as the set of strongly connected components in $G[\mathcal{B}']$. Also, using the weighted decision oracle we can compute $G[\mathcal{B}']$ in NC (Lemma 7.4). Therefore, given $\mathcal{B}'$, the following lemma also gives us a way to compute $\mathsf{prime\text{-}sets}[F_{\mathcal{B}'}]$ and $\mathsf{partition}[F_{\mathcal{B}'}]$ in NC.

Lemma 6.2. *Let $M = (E, \mathcal{I})$ be a matroid with r be the rank function and $\mathcal{B}$ be the set of bases. Let $\mathcal{B}'$ be a subset of $\mathcal{B}$, and $F_{\mathcal{B}'}$ be the minimal face of $P(\mathcal{B})$ containing $\mathcal{B}'$. For all $a \in E$, let $E[a]$ be the set of vertices reachable from a in the graph $G[\mathcal{B}']$. Then*

1. *$\mathsf{prime\text{-}sets}[F_{\mathcal{B}'}] = \{E[a] \mid a \in E\}$.*
2. *the set of all strongly connected components in $G[\mathcal{B}']$ is same as $\mathsf{partition}[F_{\mathcal{B}'}]$.*

Proof. First, we show that 1) $\mathsf{prime\text{-}sets}[\mathcal{B}'] = \{E[a] \mid a \in E\}$, and 2) the set of strongly connected components in $G[\mathcal{B}']$ is same as $\mathsf{partition}[\mathcal{B}']$. Later, we prove that $\mathsf{tight\text{-}sets}[\mathcal{B}']$ is same as $\mathsf{tight\text{-}sets}[F_{\mathcal{B}'}]$. Hence, $\mathsf{prime\text{-}sets}[\mathcal{B}'] = \mathsf{prime\text{-}sets}[F_{\mathcal{B}'}]$ and $\mathsf{partition}[\mathcal{B}'] = \mathsf{partition}[F_{\mathcal{B}'}]$ which will complete our proof.

We prove that $\mathsf{prime\text{-}sets}[\mathcal{B}'] = \{E[a] \mid a \in E\}$. The proof has two steps. In the first step, we show that for all $a \in E$, $E[a]$ is in $\mathsf{tight\text{-}sets}[\mathcal{B}']$. Later, we show that $E[a]$ is the smallest set in $\mathsf{tight\text{-}sets}[\mathcal{B}']$ containing a, which will prove our claim. For the sake of contradiction, assume that $E[a]$ is not in $\mathsf{tight\text{-}sets}[\mathcal{B}']$. Then, from Lemma 6.1, there exists an edge (b, c) in $G[\mathcal{B}']$ such that $b \in E[a]$ and $c \in E \setminus E[a]$. This implies that c is also reachable from a. This is a contradiction since $c \in E \setminus E[a]$. Hence, $E[a] \in \mathsf{tight\text{-}sets}[\mathcal{B}']$. Now, we show that $E[a]$ is the smallest set in $\mathsf{tight\text{-}sets}[\mathcal{B}']$ containing a. Let E_a be the smallest set in $\mathsf{tight\text{-}sets}[\mathcal{B}']$. Then, E_a is a subset of $E[a]$. Otherwise, $E_a \cap E[a]$ is a much smaller set containing a in $\mathsf{tight\text{-}sets}[\mathcal{B}']$ since $\mathsf{tight\text{-}sets}[\mathcal{B}']$ is a lattice family over E (Corollary 4.9). Since $E_a \in \mathsf{tight\text{-}sets}[\mathcal{B}']$, from Lemma 6.1, there exists no directed edge from E_a to $E \setminus E_a$ in the graph $G[\mathcal{B}']$. Hence, no vertex in $E \setminus E_a$ is reachable from a in $G[\mathcal{B}']$. This implies that $E[a]$ is a subset of E_a. Therefore, $E[a] = E_a$.

Now we show that the set of strongly connected components in $G[\mathcal{B}']$ is same as $\mathsf{partition}[\mathcal{B}']$. Using Lemma 4.3, the elements of $\mathsf{partition}[\mathcal{B}']$ can be written in a sequence

$$(S_1, S_2, \ldots, S_\ell)$$

such that for all $k \in [\ell]$, $\cup_{j=1}^{k} S_j$ is a set in $\mathsf{tight\text{-}sets}[\mathcal{B}']$. Let a be an element from S_k for some $k \in [\ell]$. Let S_L be the union of all the sets on the left of S_k , that is $S_L = \cup_{j=1}^{k-1} S_j$. Let S_R be the union of all the sets on right of S_k, that is $S_R = \cup_{j=k+1}^{\ell} S_j$. Let b be an element in E. Now we divide our proof into the following cases.

[2] In [Sch03, Chapter], the digraph representation is described in terms of a pre-order $\preceq$ on E defined as follows: $a \preceq b$ if and only if (b, a) is an edge in $G_{\mathcal{L}}$.

1. **When** $b \in S_L$**:** Since $S_L \in \mathsf{tight\text{-}sets}[\mathcal{B}']$, from Lemma 6.1, there is no directed edge from S_L to $E \setminus S_L$. Hence, there is no directed path from b to a in $G[\mathcal{B}']$.

2. **When** $b \in S_R$**:** Let $S_C = S_L \cup S_k$. Since $S_C \in \mathsf{tight\text{-}sets}[\mathcal{B}']$, like the previous case, we can show that there exists no directed path from a to b in $G[\mathcal{B}']$.

3. **When** $b \in S_k$**:** Since $E[a] \in \mathsf{tight\text{-}sets}[\mathcal{B}']$, from Lemma 4.3, $E[a]$ is a union of some sets from $\mathsf{partition}[\mathcal{B}']$. Also, the union must contain the set S_k since the sets in $\mathsf{partition}[\mathcal{B}']$ are disjoint. Hence, $b \in E[a]$. This implies that there exists a directed path from a to b in $G[\mathcal{B}']$. Similarly, we can show that there exists a directed path from b to a in $G[\mathcal{B}']$.

Hence, for any two vertices a, b in the graph $G[\mathcal{B}']$, there are directed paths both from a to b and from b to a if and only if both a and b belong to the same set in $\mathsf{partition}[\mathcal{B}']$. Therefore, the set of strongly connected components in $G[\mathcal{B}']$ is same as $\mathsf{partition}[\mathcal{B}']$.

Now we show that $\mathsf{tight\text{-}sets}[\mathcal{B}']$ is same as $\mathsf{tight\text{-}sets}[F_{\mathcal{B}'}]$. Since $\mathcal{B}'$ is a subset of $F_{\mathcal{B}'}$, $\mathsf{tight\text{-}sets}[F_{\mathcal{B}'}]$ is a subset of $\mathsf{tight\text{-}sets}[\mathcal{B}']$. Let F be the face of $P(\mathcal{B})$ defined as follows: for all $\mathbf{x} \in \mathbb{R}^E$, $\mathbf{x}$ is in F if and only if for all $S \in \mathsf{tight\text{-}sets}[\mathcal{B}']$, $\mathbf{x}(S) = r(S)$. Then $\mathcal{B}'$ is a subset of F, and thus, $\mathsf{tight\text{-}sets}[\mathcal{B}']$ is same as $\mathsf{tight\text{-}sets}[F]$. On the other hand, F is a subset of $F_{\mathcal{B}'}$ since $F_{\mathcal{B}'}$ is the minimal face containing $\mathcal{B}'$. Therefore $\mathsf{tight\text{-}sets}[F]$ is a subset of $\mathsf{tight\text{-}sets}[F_{\mathcal{B}'}]$. Thus $\mathsf{tight\text{-}sets}[\mathcal{B}'] = \mathsf{tight\text{-}sets}[F_{\mathcal{B}'}]$. This implies that $\mathsf{prime\text{-}sets}[\mathcal{B}'] = \mathsf{prime\text{-}sets}[F_{\mathcal{B}'}]$ and $\mathsf{partition}[\mathcal{B}'] = \mathsf{partition}[F_{\mathcal{B}'}]$, which completes our proof. □

Now we extend our previous two lemmas for the matroid intersection. It will be crucially used in our work. First we extend the definition of our graph, used in the previous lemmas, for the matroid intersection. Suppose that $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$ are the two matroids with $\mathcal{B}_1$ and $\mathcal{B}_2$ are the family of bases, respectively. Let $\mathcal{B} \subseteq \mathcal{B}_1 \cap \mathcal{B}_2$ be a set of common bases of M_1 and M_2. Then for every $i \in [2]$ we define a directed graph, *denoted by* $G_i[\mathcal{B}]$, as follows: the vertex set is same as the ground set E, and for $a, b \in E$, (a, b) is an edge of $G_i[\mathcal{B}]$ if and only if

there exists a base $B \in \mathcal{B}$ *such that* $b \in B$ *and* $(B \setminus \{b\}) \cup \{a\}$ *is also a base of* M_i.

For matroid intersection we show the following lemma.

Lemma 6.3. *Let* $M_1 = (E, \mathcal{I}_1)$ *and* $M_2 = (E, \mathcal{I}_2)$ *be two matroids with* $\mathcal{B}_1$ *and* $\mathcal{B}_2$ *be the family of bases, and* r_1 *and* r_2 *be the rank functions, respectively. Let* $\mathcal{B} \subseteq \mathcal{B}_1 \cap \mathcal{B}_2$ *be a set of common bases of* M_1 *and* M_2*. Let* $F_{\mathcal{B}}$ *be the minimal face of* $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ *containing* $\mathcal{B}$*. For all* $i \in [2]$ *and* $a \in E$*, let* $E_i[a]$ *be the set of vertices reachable from* a *in the graph* $G_i[\mathcal{B}]$*. Then for all* $i \in [2]$*,*

1. *a subset* S *of* E *is in* $\mathsf{tight\text{-}sets}_i[\mathcal{B}]$ *if and only if there is no directed edge from* S *to* $E \setminus S$ *in the graph* $G_i[\mathcal{B}]$*.*

2. $\mathsf{prime\text{-}sets}_i[F_{\mathcal{B}}] = \{E_i[a] \mid a \in E\}$*.*

3. *the set of all strongly connected components in* $G_i[\mathcal{B}]$ *is same as* $\mathsf{partition}_i[F_{\mathcal{B}}]$*.*

Proof. From the equations 8, 9 and 10, one can show that the face $F_{\mathcal{B}}$ can be defined as $F_1 \cap F_2$ where for $i \in [2]$, F_i is the minimal face of $P(\mathcal{B}_i)$ containing $\mathcal{B}$. Now applying Lemma 6.1 and 6.2, we get the above one. □

7 An RNC-algorithm to compute the max-weight face

Suppose that $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$ are two matroids with r_1 and r_2 are the rank functions and $\mathcal{B}_1$ and $\mathcal{B}_2$ are the set of bases, respectively. Let $m = |E|$. Let $\mathbf{w}$ be a weight assignment on E with polynomially large weights, and $F_{\mathbf{w}}$ is the maximizing face of $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ with respect to $\mathbf{w}$. In this section, we describe a randomized NC algorithm to compute $\mathsf{prime\text{-}sets}_i[F_{\mathbf{w}}]$ and $\mathsf{partition}_i[F_{\mathbf{w}}]$ for all $i \in [2]$, provided that the algorithm has an oracle access to $\mathsf{weighted\text{-}decision\text{-}MI}$.

Our idea is the following: We start from a random vertex B in $F_{\mathbf{w}}$ (or, equivalently, a random common base in $F_{\mathbf{w}}$), and inductively construct a chain of subset of bases from $F_{\mathbf{w}}$

$$\{B\} = \mathcal{B}_0 \subseteq \mathcal{B}_1 \subseteq \cdots \subseteq \mathcal{B}_\ell$$

such that the minimal face containing $\mathcal{B}_\ell$ is same as $F_{\mathbf{w}}$ and $\ell = \lceil \log m \rceil$. Then, using Lemma 6.3, for all $i \in [2]$ we can compute $\mathsf{prime\text{-}sets}_i[F_{\mathbf{w}}]$ and $\mathsf{partition}_i[F_{\mathbf{w}}]$ from the graph $G_i[\mathcal{B}_\ell]$ in NC. Now we briefly discuss how to construct the set $\mathcal{B}_j$ for all $j \in [\ell]$. For all $j \in \{0, \ldots, \ell\}$, let F_j denotes the minimal face containing $\mathcal{B}_j$. For all $j \in [\ell]$, the set $\mathcal{B}_j$ contains the elements in $\mathcal{B}_{j-1}$ with the following extra elements: For all $i \in [2]$ $A \in \mathsf{partition}_i[F_{j-1}]$, if exists, we add a base $B_{ij}^{(A)}$ from the face $F_{\mathbf{w}}$ with the property

$$|A \cap B_{ij}^{(A)}| \neq N_i^{F_{j-1}}(A).$$

From Lemma 4.11, we know that for all $i \in [2]$ $A \in \mathsf{partition}_i[F_{j-1}]$, every base in F_{j-1} contains exactly $N_i^{F_{j-1}}(A)$ many elements from A. However, our property on $B_{ij}^{(A)}$ says that we want a base from $F_{\mathbf{w}}$ which violates that condition, and Lemma 5.4 ensures that, if exists, we can compute such a base in randomized NC using the oracle access to $\mathsf{weighted\text{-}decision\text{-}MI}$. In Algorithm 1, we describe all the steps to compute $\mathsf{prime\text{-}sets}_i[F_{\mathbf{w}}]$ and $\mathsf{partition}_i[F_{\mathbf{w}}]$ for all $i \in [2]$.

7.1 Correctness

Now we discuss about the correctness of Algorithm 1. First, we describe some notations which will be used in our proofs. Let $\ell = \lceil \log m \rceil$. For all $j \in \{0, 1, \ldots, \ell\}$ $i \in [2]$ $S \in \mathsf{partition}_i[F_{\mathbf{w}}]$,

- $\mathcal{W}_{ij}^{(S)}$ denotes the set
$$\{A \in \mathsf{partition}_i[F_j] \mid A \cap S \neq \emptyset\}.$$
Since F_j is a subface of $F_{\mathbf{w}}$, using Corollary 4.10, we get that $\mathsf{tight\text{-}sets}_i[F_{\mathbf{w}}]$ is a sublattice of $\mathsf{tight\text{-}sets}_i[F_j]$. Therefore, Lemma 4.4 ensures that $\mathcal{W}_{ij}^{(S)}$ is a partition of S.

- $\mathcal{I}_{ij}^{(S)}$ denotes the set
$$\left\{A \in \mathcal{W}_{ij}^{(S)} \mid \exists \text{ a base } B \text{ in } F_{\mathbf{w}} \text{ s.t. } |B \cap A| \neq N_i^{F_j}(A)\right\}.$$
From Lemma 4.11, we know that every base B from F_j contains exactly $N_i^{F_j}(A)$ many elements from every $A \in \mathcal{W}_{ij}^{(S)}$. On the other hand, $\mathcal{I}_{ij}^{(S)}$ denotes the following subset of $\mathcal{W}_{ij}^{(S)}$: For all $A \in \mathcal{W}_{ij}^{(S)}$, $A \in \mathcal{I}_{ij}^{(S)}$ if and only if there exists a base from $F_{\mathbf{w}}$ which violates the numerical constraint on A satisfied by the bases in F_j.

In the next lemma, we describe an invariant satisfied by $\mathcal{I}_{ij}^{(S)}$ as the for loop in Algorithm 1 moves from $(j-1)$th iteration to jth iteration. This helps us to argue how quickly the subface F_j becomes equal to the face $F_{\mathbf{w}}$.

Algorithm 1 Computing prime sets and partitions corresponding to a max-weight face

Input: Two matroids $M_1 = (E, \mathcal{I}_1)$, $M_2(E, \mathcal{I}_2)$, and a weight assignment $\mathbf{w} : E \to \mathbb{Z}_{\geq 0}$.
Output: $\mathsf{prime\text{-}sets}_i[F_{\mathbf{w}}]$ and $\mathsf{partition}_i[F_{\mathbf{w}}]$ for all $i \in [2]$, where $F_{\mathbf{w}}$ denotes the max-weight face.
Assumption: Oracle access to $\mathsf{weighted\text{-}decision\text{-}MI}$.

Compute a base B in $F_{\mathbf{w}}$.
$\mathcal{B}_0 \leftarrow \{B\}$.
for all $i \in [2]$ **do in parallel**
    Compute the graph $G_i[\mathcal{B}_0]$.
    Let F_0 be the minimal face containing $\mathcal{B}_0$.
    Compute $\mathsf{prime\text{-}sets}_i[F_0]$, $\mathsf{partition}_i[F_0]$ and $N_i^{F_0}$.
end for
for $j \leftarrow 1$ to $\lceil \log m \rceil$ **do**
    $\mathcal{B}_j \leftarrow \mathcal{B}_{j-1}$.
    for all $i \in [2]$ **do in parallel**
        for all $A \in \mathsf{partition}_i[F_{j-1}]$ **do in parallel**
            If exists, compute a base $B_{ij}^{(A)}$ in $F_{\mathbf{w}}$ such that

$$|A \cap B_{ij}^{(A)}| \neq N_i^{F_{j-1}}(A).$$

            $\mathcal{B}_j \leftarrow \mathcal{B}_j \bigcup \left\{ B_{ij}^{(A)} \right\}$.
        end for
    end for
    for all $i \in [2]$ **do in parallel**
        Let F_j be the minimal face containing $\mathcal{B}_j$.
        Compute $\mathsf{prime\text{-}sets}_i[F_j]$, $\mathsf{partition}_i[F_j]$ and $N_i^{F_j}$ using Lemma 6.3.
    end for
end for
return $\mathsf{prime\text{-}sets}_i[F_\ell]$ and $\mathsf{partition}_i[F_\ell]$ for $i \in [2]$ and $\ell = \lceil \log m \rceil$.

Lemma 7.1 (Invariant)**.** *Let $m = |E|$, and $\ell = \lceil \log m \rceil$. Then for all $j \in [\ell]$ $i \in [2]$ $S \in \mathsf{partition}_i[F_{\mathbf{w}}]$, either*

$$\mathcal{I}_{ij}^{(S)} = \emptyset,$$

or

$$\min_{A \in \mathcal{I}_{ij}^{(S)}} |A| \geq 2 \cdot \min_{A \in \mathcal{I}_{i(j-1)}^{(S)}} |A|.$$

Proof. For some $j \in [\ell]$ $i \in [2]$ $S \in \mathsf{partition}_i[F_{\mathbf{w}}]$, let $\mathcal{I}_{ij}^{(S)}$ be nonempty and A be an element from $\mathcal{I}_{ij}^{(S)}$. First, we show that A is a disjoint union of sets from $\mathcal{W}_{i(j-1)}^{(S)}$. Since F_j is a subface of $F_{\mathbf{w}}$, from Corollary 4.10, $\mathsf{tight\text{-}sets}_i[F_{\mathbf{w}}]$ is a sublattice of $\mathsf{tight\text{-}sets}_i[F_j]$. Therefore, applying Lemma 4.4, we get that $\mathcal{W}_{ij}^{(S)}$ is a partition of S. Hence, A is a subset of S. This implies that $\mathcal{W}_{i(j-1)}^{(S)}$ contains all the sets in $\mathsf{partition}_i[F_{j-1}]$ which have nonempty intersection with $A \in \mathsf{partition}_i[F_j]$. Also, $\mathsf{tight\text{-}sets}_i[F_{j-1}]$ is a sublattice of $\mathsf{tight\text{-}sets}_i[F_j]$ since F_{j-1} is a subface of F_j. Therefore, from Lemma 4.4, the set A can be written as $\cup_{k=1}^{s} A_k$ where each A_k is a set from $\mathcal{W}_{i(j-1)}^{(S)}$. Our goal is to show that at least two sets in $\{A_1, \ldots, A_s\}$ are from $\mathcal{I}_{i(j-1)}^{(S)}$ which will complete our proof.

As a first step, we show that there exists at least a $k \in [s]$ such that A_k is from $\mathcal{I}_{i(j-1)}^{(S)}$. For the sake of contradiction, assume that no such k exists. Then for every base B in the face $F_{\mathbf{w}}$ contains exactly $\sum_{k=1}^{s} N_i^{F_{j-1}}(A_k)$ elements from A. From Lemma 4.11, every base C in F_{j-1} contains exactly $N_i^{F_{j-1}}(A_k)$ many elements from A_k for all $k \in [s]$. On the other hand, F_{j-1} is a subface of F_j. Therefore, C is also a base in F_j. Hence, again applying Lemma 4.11, we get that C contains exactly $N_i^{F_j}(A)$ many elements from A. Thus,

$$N_i^{F_j}(A) = \sum_{k=1}^{s} N_i^{F_{j-1}}(A_k). \tag{12}$$

This implies that every base B in the face $F_{\mathbf{w}}$ also contains exactly $N_i^{F_j}(A)$ elements from A. Therefore A is a set from $\mathcal{W}_{ij}^{(S)} \setminus \mathcal{I}_{ij}^{(S)}$, which is a contradiction. Hence, without loss of generality, assume $A_s \in \mathcal{I}_{i(j-1)}^{(S)}$. Next we show that $\{A_1, \ldots, A_{s-1}\}$ also contains a set from $\mathcal{I}_{i(j-1)}^{(S)}$.

For the sake of contradiction, assume that for all $k \in [s-1]$, A_k is a set from $\mathcal{W}_{i(j-1)}^{(S)} \setminus \mathcal{I}_{i(j-1)}^{(S)}$. Therefore, every base B in $F_{\mathbf{w}}$ contains exactly $N_i^{F_{j-1}}(A_k)$ many elements from A_k for all $k \in [s-1]$. Since A_s is a set from $\mathcal{I}_{i(j-1)}^{(S)}$, the step 12 of Algorithm 1 promises that there exists a base $B_{ij}^{(A_s)}$ in $\mathcal{B}_j$ such that

$$|A_s \cap B_{ij}^{(A_s)}| \neq N_i^{F_{j-1}}(A_s).$$

On the other hand, since $B_{ij}^{(A_s)}$ is a base in $\mathcal{B}_j$,

$$|A \cap B_{ij}^{(A_s)}| = N_i^{F_j}(A) = |A_s \cap B_{ij}^{(A_s)}| + \sum_{k=1}^{s-1} N_i^{F_{j-1}}(A_k). \tag{13}$$

From Equation 12 and 13,

$$|A_s \cap B_{ij}^{(A_s)}| = N_i^{F_{j-1}}(A_s),$$

which is a contradiction. Therefore, there exists a $k \in [s-1]$ such that A_k is in $\mathcal{I}_{i(j-1)}^{(S)}$. This completes our proof. □

Our next lemma tells what happens when in the above lemma $I_{ij}^{(S)}$ becomes empty. It will be helpful to prove the successful termination of Algorithm 1.

Lemma 7.2. *Let $m = |E|$ and $\ell = \lceil \log m \rceil$. For some $j \in \{0, 1, \ldots, \ell\}$ $i \in [2]$ $S \in \mathsf{partition}_i[F_{\mathbf{w}}]$, let $\mathcal{I}_{ij}^{(S)}$ be the empty set. Then $\mathcal{W}_{ij}^{(S)} = \{S\}$, that is $S \in \mathsf{partition}_i[F_j]$.*

Proof. For the sake of contradiction, assume that $\mathcal{W}_{ij}^{(S)} = \{A_1, \ldots, A_s\}$ where $s \geq 2$. Since F_j is a subface of $F_{\mathbf{w}}$, from Corollary 4.10, $\mathsf{tight\text{-}sets}_i[F_{\mathbf{w}}]$ is a sublattice of $\mathsf{tight\text{-}sets}_i[F_j]$. Therefore, from Lemma 4.4, $\{A_1, \ldots, A_s\}$ is a partition of S. From Lemma 4.3, the elements of $\mathsf{partition}_i[F_{\mathbf{w}}]$ can be written as a sequence

$$(S_1, \ldots, S_\ell)$$

such that for all $k \in [\ell]$, $\cup_{r=1}^{k} S_r$ is in $\mathsf{tight\text{-}sets}_i[F_{\mathbf{w}}]$. Let $S = S_k$ for some $k \in [\ell]$, and $S_L = \cup_{r=1}^{k-1} S_r$.

Similarly, the elements of $\mathsf{partition}_i[F_j]$ can also be written as a sequence

$$(T_1, \ldots, T_n) \tag{14}$$

such that for all $k \in [n]$, $\cup_{r=1}^{k} T_r$ is in $\mathsf{tight\text{-}sets}_i[F_j]$. Without loss of generality, assume that among all the sets in $\mathcal{W}_{ij}^{(S)}$, A_1 appears first in the sequence. Let $T_k = A_1$ for some $k \in [n]$, and $T_L = \cup_{r=1}^{k-1} T_r$. Let a be an element in A_1. Let E_a and E'_a be the smallest sets in $\mathsf{tight\text{-}sets}_i[F_{\mathbf{w}}]$ and $\mathsf{tight\text{-}sets}_i[F_j]$, respectively, containing a. Since $S_L \in \mathsf{tight\text{-}sets}_i[F_{\mathbf{w}}]$ and $\mathsf{tight\text{-}sets}_i[F_{\mathbf{w}}]$ is a sublattice of $\mathsf{tight\text{-}sets}_i[F_j]$, S_L is also in $\mathsf{tight\text{-}sets}_i[F_j]$. Our overall goal is to show that $S_L \cup A_1$ is in $\mathsf{tight\text{-}sets}_i[F_{\mathbf{w}}]$, which will give us contradiction. As a first step, we prove that that $S_L \cup A_1$ is an element in $\mathsf{tight\text{-}sets}_i[F_j]$.

To show $S_L \cup A_1$ is in $\mathsf{tight\text{-}sets}_i[F_j]$, first we prove that E'_a is a subset of $S_L \cup A_1$. Since $\mathsf{tight\text{-}sets}_i[F_{\mathbf{w}}]$ is a sublattice of $\mathsf{tight\text{-}sets}_i[F_j]$, from Lemma 4.4, E'_a is a subset of E_a. Also, E_a is a subset of $S_L \cup S$ since both of them are sets in $\mathsf{tight\text{-}sets}_i[F_{\mathbf{w}}]$ containing a and the former is the smallest such set. Therefore,

$$E'_a \subseteq S_L \cup S. \tag{15}$$

From Sequence 14, we know that $T_L \cup A_1$ is in $\mathsf{tight\text{-}sets}_i[F_j]$ containing a. Since E'_a is the smallest set in $\mathsf{tight\text{-}sets}_i[F_j]$ containing a,

$$E'_a \subseteq T_L \cup A_1. \tag{16}$$

From Equation 15 and 16, we get that

$$\begin{aligned} E'_a &\subseteq (S_L \cup S) \cap (T_L \cup A_1) \\ &= (S_L \cap T_L) \cup (S_L \cap A_1) \cup (S \cap T_L) \cup (S \cap A_1) \\ &= (S_L \cap T_L) \cup A_1, \text{ since } S_L \cap A_1 = S \cap T_L = \emptyset \text{ and } A_1 \subseteq S \\ &\subseteq S_L \cup A_1. \end{aligned}$$

Since both S_L and E'_a are in $\mathsf{tight\text{-}sets}_i[F_j]$, $S_L \cup E'_a$ is also in $\mathsf{tight\text{-}sets}_i[F_j]$. The set $S_L \cup E'_a$ is a subset of $S_L \cup A_1$ since E'_a is a subset of $S_L \cup A_1$. From Lemma 4.11, we know that every element in $\mathsf{tight\text{-}sets}_i[F_j]$ can be written as a disjoint union of sets from $\mathsf{partition}_i[F_j]$. Hence, A_1 is a subset of E'_a. Therefore, $S_L \cup A_1$ is a subset of $S_L \cup E'_a$, which implies that $S_L \cup E'_a = S_L \cup A_1$. Hence, $S_L \cup A_1$ is in $\mathsf{tight\text{-}sets}_i[F_j]$.

Now we show that $S_L \cup A_1$ is also in $\mathsf{tight\text{-}sets}_i[F_{\mathbf{w}}]$. Since both S_L and $S_L \cup A_1$ are in

$\mathsf{tight\text{-}sets}_i[F_j]$, for every base B in the face F_j the following holds:

$$\begin{aligned}|B \cap (S_L \cup A_1)| &= r_i(S_L \cup A_1) \\ &= |B \cap S_L| + |B \cap A_1|, \text{ since } S_L \cap A_1 = \emptyset \\ &= r_i(S_L) + N_i^{F_j}(A_1).\end{aligned}$$

On the other hand, S_L is also in $\mathsf{tight\text{-}sets}_i[F_{\mathbf{w}}]$. Hence, for all base C in $F_{\mathbf{w}}$, the cardinality of $C \cap S_L$ is same as $r_i(S_L)$. Since the set $\mathcal{I}_{ij}^{(S)}$ is empty, again for every base C in $F_{\mathbf{w}}$, $|C \cap A_1| = N_i^{F_j}(A_1)$. Therefore for every base C in $F_{\mathbf{w}}$,

$$|C \cap (S_L \cup A_1)| = r_i(S_L) + N_i^{F_j}(A_1) = r_i(S_L \cup A_1),$$

which implies that $S_L \cup A_1$ is also in $\mathsf{tight\text{-}sets}_i[F_{\mathbf{w}}]$. This is a contradiction since from Lemma 4.11, any set in $\mathsf{tight\text{-}sets}_i[F_{\mathbf{w}}]$ containing an element from S must contain the whole set S. Thus, $\mathcal{W}_{ij}^{(S)} = \{S\}$, which completes our proof. □

Lemma 7.3 (Termination)**.** *Let $m = |E|$, and $\ell = \lceil \log m \rceil$. Then $F_\ell = F_{\mathbf{w}}$.*

Proof. Since the set $\mathcal{B}_\ell$ is formed by taking bases from $F_{\mathbf{w}}$, the face F_ℓ is a subface of $F_{\mathbf{w}}$. Lemma 7.1 implies that after ℓth iteration of the for loop, $\mathcal{I}_{i\ell}^{(S)} = \emptyset$ for all $i \in [2]$ $S \in \mathsf{partition}_i[F_{\mathbf{w}}]$. Otherwise, $\mathcal{I}_{i\ell}^{(S)}$ contains a set of size greater than m, which is a contradiction. Now applying Lemma 7.2 we get that $\mathcal{W}_{i\ell}^{(S)} = \{S\}$ for all $i \in [2]$ $S \in \mathsf{partition}_i[F_\ell]$. This implies that $\mathsf{partition}_i[F_\ell]$ is same as $\mathsf{partition}_i[F_{\mathbf{w}}]$. Let B be a base in F_ℓ. Therefore B is also base in $F_{\mathbf{w}}$. Now applying Lemma 4.11, we get that for all $i \in [2]$ $S \in \mathsf{partition}_i[F_\ell]$,

$$N_i^{F_\ell}(S) = |S \cap B| = N_i^{F_{\mathbf{w}}}(S).$$

Again, from Lemma 4.11, we know that a common base B (of M_1 and M_2) is in F_ℓ if and only if for all $i \in [2]$ $S \in \mathsf{partition}_i[F_\ell]$,

$$N_i^{F_\ell}(S) = |B \cap S|,$$

which is also satisfied by every base present in $F_{\mathbf{w}}$. Therefore, $F_{\mathbf{w}}$ is a subset of F_ℓ. Thus, $F_\ell = F_{\mathbf{w}}$. This completes our proof. □

From the above lemma, at the end of ℓth iteration of the for loop, $\mathsf{prime\text{-}sets}_i[F_\ell] = \mathsf{prime\text{-}sets}_i[F_{\mathbf{w}}]$ and $\mathsf{partition}_i[F_\ell] = \mathsf{partition}_i[F_{\mathbf{w}}]$ for all $i \in [2]$. They can be computed using the graph theoretic characterization given in Lemma 6.3.

7.2 Time Complexity and Success Probability

In Algorithm 1, for all $j \in \{0, 1, \ldots, \ell\}$, we use Lemma 6.3 to compute $\mathsf{prime\text{-}sets}_i[F_j]$ and $\mathsf{partition}_i[F_j]$ for all $i \in [2]$. Hence, we need to compute the graph $G_i[\mathcal{B}_j]$ for all $i \in [2]$. Our next lemma describes the time complexity of computing the graph $G_i[\mathcal{B}_j]$.

Lemma 7.4. *Let $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$ be two matroids with the family of bases $\mathcal{B}_1$ and $\mathcal{B}_2$, respectively. Let $\mathcal{B}$ be a polynomially large subset of $\mathcal{B}_1 \cap \mathcal{B}_2$. Then, given M_1, M_2 and $\mathcal{B}$ as inputs, the graph $G_i[\mathcal{B}]$ for all $i \in [2]$ can be computed in NC, provided that the algorithm has an oracle access to* $\mathsf{weighted\text{-}decision\text{-}MI}$.

Proof. Let $i \in [2]$. Let a, b be two elements from E, and B be a base from $\mathcal{B}$. First we show that how to decide whether $B' = (B \setminus \{b\}) \cup \{a\}$ is a base of the matroid M_i. It is not hard to see that B' is a base of M_i if and only if the weight of the maximum weight base in M_i with respect to the weight assignment $\mathbf{1}_{B'}$ is equal to $|B|$. Also, the weight of the maximum weight base with respect to $\mathbf{1}_{B'}$ can be at most $|B|$. Therefore using a single oracle call to weighted-decision-MI we can decide whether B' is a base of M_i. Observe that in the input of the oracle both matroids will be M_i. From the definition of $G_i[\mathcal{B}]$, (a, b) is an edge of it if and only if there exists a base $B \in \mathcal{B}$ containing b such that $B' = (B \setminus \{b\}) \cup \{a\}$ is a base of M_i . Therefore, in parallel, for all $a, b \in E$ and $B \in \mathcal{B}$ we check whether $b \in B$ and $B' = (B \setminus \{b\}) \cup \{a\}$ is base of M_i. Thus, we get all the edges of $G_i[\mathcal{B}]$ in NC. □

Now we show that Algorithm 1 runs in randomized NC. Also, we show that for every positive integer c, the success probability of the algorithm can be made at least $1 - \frac{1}{m^c}$, where $m = |E|$. Let $c_0 = c + 2$, and $\ell = \lceil \log m \rceil$. Then using Lemma 5.3, we can compute the set $\mathcal{B}_0$ in randomized NC with probability at least $1 - \frac{1}{m^{c_0}}$. Let $j \in [\ell]$. At jth iteration, applying Lemma 5.4, we can compute $B_{ij}^{(A)}$ in randomized NC with probability $1 - \frac{1}{m^{c_0}}$ for all $i \in [2]$ and $A \in \mathsf{partition}_i[F_{j-1}]$. The size of the family $\mathsf{partition}_i[F_{j-1}]$ can be at most m. Thus, using union bound we get that the set $\mathcal{B}_j$, assuming $\mathcal{B}_{j-1}$ is successfully computed, can be computed in randomized NC with probability at least $1 - \frac{2m}{m^{c_0}}$. Therefore, using union bound we get that for all $j \in \{0, 1, \ldots, \ell\}$, the set $\mathcal{B}_j$ can be computed with probability at least $1 - \frac{O(\ell m)}{m^{c_0}}$. Next, assuming all $\mathcal{B}_j$s are perfectly computed, we show that the other steps can be done in NC.

Let $j \in \{0, 1, \ldots, \ell\}$. From Lemma 6.3, we know that for all $i \in [2]$,

1. $\mathsf{prime\text{-}sets}_i[F_j] = \{E_i[a] \mid a \in E\}$, where $E_i[a]$ is the set vertices reachable from a in the graph $G_i[\mathcal{B}_j]$, and
2. $\mathsf{partition}_i[F_j]$ is same as the set of strongly connected components in $G_i[\mathcal{B}_j]$.

Using Lemma 7.4, for all $i \in [2]$, the graph $G_i[\mathcal{B}_j]$ can be computed in NC. Given a directed graph G and a vertex a, the set of vertices reachable from a in G can be computed in NC. Also, all the strongly connected components of G can be computed in NC. Therefore, $\mathsf{prime\text{-}sets}_i[F_j]$ and $\mathsf{partition}_i[F_j]$ for all $i \in [2]$ can be computed in NC. The function $N_i^{F_j}$ can be computed in NC by computing $|B \cap A|$, for some $B \in \mathcal{B}_j$, in parallel for all $A \in \mathsf{partition}_i[F_j]$. Thus, Algorithm 1 computes $\mathsf{prime\text{-}sets}_i[F_{\mathbf{w}}]$ and $\mathsf{partition}_i[F_{\mathbf{w}}]$ for all $i \in [2]$ in randomized NC with success probability at least $1 - \frac{1}{m^c}$.

The above discussions can be summarized in the following theorem.

Theorem 7.5. *Let $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$ be two matroids with $\mathcal{B}_1$ and $\mathcal{B}_2$ be the family of bases, respectively. Let $\mathbf{w}$ be a weight assignment on E with polynomially bounded weights, and $F_{\mathbf{w}}$ be the maximizing face of $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ with respect to $\mathbf{w}$. Then, given M_1, M_2 and $\mathbf{w}$ as inputs, Algorithm 1 computes $\mathsf{prime\text{-}sets}_i[F_{\mathbf{w}}]$ and $\mathsf{partition}_i[F_{\mathbf{w}}]$ for all $i \in [2]$ in randomized NC, provided that the algorithm has an oracle access to* weighted-decision-MI. *Furthermore, for all positive integer c, the success probability of the algorithm can be made at least $1 - \frac{1}{m^c}$, where $m = |E|$.*

8 The Oracle based Algorithm

Suppose that $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$ are two matroids with r_1 and r_2 are the rank functions and $\mathcal{B}_1$ and $\mathcal{B}_2$ are the family of bases, respectively. Let $m = |E|$, and $\ell = \lceil \log m \rceil$. We are interested in designing a pseudo-deterministic NC algorithm to compute a common base $B \in \mathcal{B}_1 \cap \mathcal{B}_2$, provided

that the algorithm has an oracle access to weighted-decision-MI. Briefly, our strategy will be the following: We start with an weight assignment $\mathbf{w}_0$ such that the maximizing face of $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ with respect to $\mathbf{w}_0$ is same as the polytope itself. Then, starting from $\mathbf{w}_0$, we design a sequence of weight assignments

$$\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_\ell$$

such that the maximizing face with respect to $\mathbf{w}_\ell$ has a unique base and our algorithm will output it. For all $j \in \{0, 1, \ldots, \ell\}$, F_j will denote the maximizing face of $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ with respect to the weight fuction $\mathbf{w}_j$. The detailed steps of our algorithm are given in Algorithm 2.

Algorithm 2 Pseudo-deterministic NC algorithm for computing a common base of two matroids

Input: Two matroids $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$.
Output: A common base of M_1 and M_2, if exists.
Assumption: Oracle access to weighted-decision-MI.

1: $\mathbf{w}_0 \leftarrow \mathbf{1}$.
2: **for** $j \leftarrow 1$ to $\lceil \log m \rceil$ **do**
3: Compute a family of weight assignments $\mathcal{W}$ as promised by Lemma 4.16.
4: **for all** $\mathbf{w} \in \mathcal{W}$ **do in parallel**
5: As mentioned in Section 4.4, combine $\mathbf{w}_{j-1}$ and $\mathbf{w}$ with descending order in precedence.
6: For a $\mathbf{w} \in \mathcal{W}$, let $\mathbf{w}'$ be the combined weight.
7: Let $F_{\mathbf{w}'}$ be the maximizing face of $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ with respect to $\mathbf{w}'$.
8: For all $i \in [2]$, compute $\mathsf{prime\text{-}sets}_i[F_{\mathbf{w}'}]$ and $\mathsf{partition}_i[F_{\mathbf{w}'}]$ using Algorithm 1.
9: Compute the graph $\mathcal{G}[F_{\mathbf{w}'}]$ and the length of its shortest cycles.
10: **end for**
11: Take some fixed ordering on $\mathcal{W}$, like lexicographic ordering.
12: Take the smallest $\mathbf{w}$ such that the length of the shortest cycle in $\mathcal{G}[F_{\mathbf{w}'}] > 2^j$.
13: Using Lemma 8.1, compute $\mathbf{w}_j$ from $\mathsf{prime\text{-}sets}_1[F_{\mathbf{w}'}]$ and $\mathsf{prime\text{-}sets}_2[F_{\mathbf{w}'}]$ such that

$$F_j = F_{\mathbf{w}'}.$$

14: **end for**
15: Compute the base present in the face $F_{\lceil \log m \rceil}$ and output.

8.1 Correctness

Our next lemma will be crucially used to compute $\mathbf{w}_j$ from $\mathbf{w}'$ at step 13 of Algorithm 2.

Lemma 8.1. *Let $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$ be two matroids with the family of bases $\mathcal{B}_1$ and $\mathcal{B}_2$, respectively. Let F be a face of the polytope $P(\mathcal{B}_1 \cap \mathcal{B}_2)$. Then there exists a weight assignment $\mathbf{w}$ on E with the following properties:*

1. *The face F is same as the set of points in $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ maximizing the weight assignment $\mathbf{w}$.*
2. *Weights are bounded by $O(|E|)$.*
3. *Given $\mathsf{prime\text{-}sets}_1[F]$ and $\mathsf{prime\text{-}sets}_2[F]$, $\mathbf{w}$ can be computed in NC.*

Proof. Consider the weight assignment defined as

$$\mathbf{w} = \sum_{i=1}^{2} \sum_{S \in \textsf{prime-sets}_i[F]} \mathbf{1}_S.$$

Now we show that $\mathbf{w}$ is our desired weight assignment. Let $F_{\mathbf{w}}$ be the maximizing face of $P(\mathcal{B}_1 \cap \mathcal{B}_2)$ with respect to $\mathbf{w}$. From the description of the polytope $P(\mathcal{B}_1 \cap \mathcal{B}_2)$, we know that for all $i \in [2]$ $S \subseteq E$ and $\mathbf{x} \in P(\mathcal{B}_1 \cap \mathcal{B}_2)$, $\langle \mathbf{1}_S, \mathbf{x} \rangle \leq r_i(S)$. Therefore, for all $\mathbf{x} \in P(\mathcal{B}_1 \cap \mathcal{B}_2)$,

$$\langle \mathbf{w}, \mathbf{x} \rangle \leq \sum_{i=1}^{2} \sum_{S \in \textsf{prime-sets}_i[F]} r_i(S).$$

Since F is nonempty and $\langle \mathbf{1}_S, \mathbf{x} \rangle = r_i(S)$ for all $\mathbf{x} \in F$ $i \in [2]$ $S \in \textsf{prime-sets}_i[F]$,

$$\max_{\mathbf{x} \in P(\mathcal{B}_1 \cap \mathcal{B}_2)} \langle \mathbf{w}, \mathbf{x} \rangle = \sum_{i=1}^{2} \sum_{S \in \textsf{prime-sets}_i[F]} r_i(S).$$

This implies that F is a subset of $F_{\mathbf{w}}$ and for all $i \in [2]$, $\textsf{prime-sets}_i[F]$ is a subset of $\textsf{tight-sets}_i[F_{\mathbf{w}}]$. Therefore, from Corollary 4.10, $\textsf{tight-sets}_i[F_{\mathbf{w}}]$ is a sublattice of $\textsf{tight-sets}_i[F]$ for all $i \in [2]$. Also, from Lemma 4.2, $\textsf{tight-sets}_i[F]$ is a subset of $\textsf{tight-sets}_i[F_{\mathbf{w}}]$ for all $i \in [2]$. Therefore, $\textsf{tight-sets}_i[F] = \textsf{tight-sets}_i[F_{\mathbf{w}}]$ for all $i \in [2]$.

Let $a \in E$ such that $\mathbf{x}_a = 0$ for all $\mathbf{x} \in F$. We show that $\mathbf{x}_a = 0$ for all $\mathbf{x} \in F_{\mathbf{w}}$. From Lemma 4.11, a appears as a singleton set in $\textsf{partition}_i[F]$ with $N_i^F(\{a\}) = 0$ for all $i \in [2]$. Since $\textsf{tight-sets}_i[F] = \textsf{tight-sets}_i[F_{\mathbf{w}}]$ for all $i \in [2]$, $\textsf{partition}_i[F]$ is same as $\textsf{partition}_i[F_{\mathbf{w}}]$ for all $i \in [2]$. Therefore a also appears as a singleton set in $\textsf{partition}_i[F_{\mathbf{w}}]$ for all $i \in [2]$. Now if $N_i^{F_{\mathbf{w}}}(\{a\})$ is not equal to zero for some $i \in [2]$, then it must be one. This implies that $\{a\}$ is in $\textsf{tight-sets}_i[F_{\mathbf{w}}]$. Thus it is also in $\textsf{tight-sets}_i[F]$, which contradicts that $N_i^F(\{a\}) = 0$. Therefore, $N_i^{F_{\mathbf{w}}}(\{a\})$ is equal to zero for all $i \in [2]$. This implies that $\mathbf{x}_a = 0$ for all $\mathbf{x} \in F_{\mathbf{w}}$. Thus F is subface of $F_{\mathbf{w}}$, which implies $F = F_{\mathbf{w}}$. The other two properties of $\mathbf{w}$ directly follows from the definition of $\mathbf{w}$. □

Suppose that for all $j \in \{0, 1, \ldots, \ell\}$, $\mathcal{C}_{F_j}$ denotes the set of all the cycles with respect to the face F_j. Our next lemma shows that as we move jth iteration to $(j+1)$th iteration, the length of the smallest cycles in $\mathcal{C}_{F_{j+1}}$ becomes doubled.

Lemma 8.2. *The set $\mathcal{C}_{F_j}$ has no cycle of length $\leq 2^j$ for all $j \in \{0, 1, \ldots, \ell\}$.*

Proof. We prove the lemma using induction. For $j = 0$, it is clearly true since the length of any cycle is greater than one. Now assume that for some $j \in [\ell]$, $\mathcal{C}_{F_{j-1}}$ has no cycle of length $\leq 2^{j-1}$. From Lemma 4.16, $\mathcal{W}$ contains a weight assignment such that it gives nonzero circulation to all the cycles in $\mathcal{C}_{F_{j-1}}$ of length at most 2^j. With respect to the ordering defined on $\mathcal{W}$, let $\mathbf{w}$ be the smallest weight assignment having such property. Let $\mathbf{w}'$ be the combined weight of $\mathbf{w}_{j-1}$ and $\mathbf{w}$ with decreasing precedence. Then from Lemma 4.7, $F_{\mathbf{w}'}$ is the face which contains the maximizing points in F_{j-1} with respect to $\mathbf{w}$. Therefore, from Lemma 4.15, the length of any cycle with respect to the face $F_{\mathbf{w}'}$ is greater than 2^j. Lemma 8.1 promises that $F_{\mathbf{w}'}$ is same as F_j, which completes our proof. □

Now we show that our algorithm is pseudo-deterministic. The step 8 of Algorithm 2 is the only place where randomness is used. This randomness is used by Algorithm 1 to compute $\textsf{prime-sets}_i[F_{\mathbf{w}'}]$

and $\mathsf{partition}_i[F_{\mathbf{w}'}]$ for all $i \in [2]$. For a face F of $P(\mathcal{B}_1 \cap B_2)$ and $i \in [2]$, $\mathsf{prime\text{-}sets}_i[F]$ and $\mathsf{partition}_i[F]$ is unique. Thus, after ℓth iteration, the face F_ℓ also becomes unique with high probability. Also, from Lemma 8.2, $\mathcal{C}_{F_\ell} = \emptyset$ since the length of any cycle can be at most m. Hence, from Lemma 4.13, F_ℓ contains a unique common base of both M_1 and M_2, and applying Lemma 5.3, we can compute it. Thus, our algorithm outputs the same common base of M_1 and M_2 with high probability, and is therefore pseudo-deterministic.

8.2 Time Complexity and Success Probability

Now we show that Algorithm 2 runs in randomized NC. From Lemma 4.16, the family of weight assignments $\mathcal{W}$ can be computed in NC. For all $\mathbf{w} \in \mathcal{W}$, the combined weight assignment $\mathbf{w}'$ (of $\mathbf{w}$ and $\mathbf{w}_j$) is also computable in NC. Computing $\mathsf{prime\text{-}sets}_i[F_{\mathbf{w}'}]$ and $\mathsf{partition}_i[F_{\mathbf{w}'}]$, for all $i \in [2]$, can be done in RNC using Algorithm 1. From the definition (see Section 4.6), it not hard to see that the graph $\mathcal{G}[F_{\mathbf{w}'}]$ is computable in NC. Given a graph G, the length of its shortest cycles can be computed in NC. Hence, the length of the shortest cycles of $\mathcal{G}[F_{\mathbf{w}'}]$ is computable in NC. Using Lemma 8.1, the weight $\mathbf{w}_j$ at step 13 can be computed in NC from $\mathsf{prime\text{-}sets}_1[F_{\mathbf{w}'}]$ and $\mathsf{prime\text{-}sets}_2[F_{\mathbf{w}'}]$. Thus, each iteration of the for loop can be done in RNC. Applying Lemma 5.3, the base in $F_{\mathbf{w}_\ell}$ is computable in NC. Since the for loop runs for $\lceil \log m \rceil$ many rounds, our algorithm runs in RNC.

Now we analysis the success probability of our algorithm. More specifically, every positive integer c, we show that the success probability of the algorithm can be made $1 - \frac{1}{m^c}$. Let $c_0 = c + 7$. Algorithm 1, called at step 8, is the only place where randomness is used. At each call of Algorithm 1, run it with success probability probability $1 - \frac{1}{m^{c_0}}$. From Lemma 4.16, the size of the weight assignment family $\mathcal{W}$ is m^6. Hence the total number of calls to Algorithm 1 is bounded by ℓm^6. Therefore, by the union bound the success probability of all the executions of Algorithm 1 is at least $1 - \frac{\ell m^6}{m^{c_0}}$, which is $\geq 1 - \frac{1}{m^c}$. Since all other steps are deterministic, the success probability of our algorithm is at least $1 - \frac{1}{m^c}$.

From the above discussions, we can conclude that

Theorem 1.3 (restated). *There is a pseudo-deterministic NC algorithm for finding a common base of two matroids M_1 and M_2 on the same ground set E, provided that the algorithm has an oracle access to the following decision question: given two matroids with polynomially bound (in $|E|$) weights on the ground set elements and a target weight W, is there a common base of weight at least W? Furthermore, the oracle calls need to be made only for the following pairs of matroids: $\langle M_1, M_2 \rangle$, $\langle M_1, M_1 \rangle$, and $\langle M_2, M_2 \rangle$.*

From Lemma 4.19, $\mathsf{weighted\text{-}decision\text{-}MI}$ has an RNC algorithm when the input matroids are linear matroids. Hence,

Theorem 1.2 (restated). *The search version of the linear matroid intersection problem has a pseudo-deterministic NC algorithm.*

Acknowledgements

We thank the anonymous reviewers for pointing towards the relevant literature on lattice families and other related topics.

References

[AGKS15] Manindra Agrawal, Rohit Gurjar, Arpita Korwar, and Nitin Saxena. Hitting-sets for ROABP and sum of set-multilinear circuits. *SIAM Journal on Computing*, 44(3):669–697, 2015. 22

[AHT07] Manindra Agrawal, Thanh Minh Hoang, and Thomas Thierauf. The polynomially bounded perfect matching problem is in NC^2. In *24th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 4393 of *Lecture Notes in Computer Science*, pages 489–499. Springer Berlin Heidelberg, 2007. 3

[AM08] Vikraman Arvind and Partha Mukhopadhyay. Derandomizing the isolation lemma and lower bounds for circuit size. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques, 11th International Workshop, APPROX, and 12th International Workshop, RANDOM*, pages 276–289, 2008. 12

[AV20] Nima Anari and Vijay V. Vazirani. Matching is as easy as the decision problem, in the NC model. In Thomas Vidick, editor, *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*, volume 151 of *LIPIcs*, pages 54:1–54:25. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. 1, 4, 8

[BCP84] Allan Borodin, Stephen Cook, and Nicholas Pippenger. Parallel computation for well-endowed rings and space-bounded probabilistic machines. *Information and Control*, 58(1-3):113–136, July 1984. 2, 22

[BEL+16] Kevin Buchin, David Eppstein, Maarten Löffler, Martin Nöllenburg, and Rodrigo I. Silveira. Adjacency-preserving spatial treemaps. *J. Comput. Geom.*, 7(1):100–122, 2016. 5, 9, 26

[Ber84] Stuart J. Berkowitz. On computing the determinant in small parallel time using a small number of processors. *Information Processing Letters*, 18(3):147 – 150, 1984. 2

[Bir37] Garrett Birkhoff. Rings of sets. *Duke Math. J.*, 3(3):443–454, 09 1937. 13, 14

[BvzGH82] Allan Borodin, Joachim von zur Gathen, and John Hopcroft. Fast parallel matrix and GCD computations. *Information and Control*, 52(3):241 – 256, 1982. 1

[CGS12] Marek Cygan, Harold N. Gabow, and Piotr Sankowski. Algorithmic applications of baur-strassen's theorem: Shortest cycles, diameter and matchings. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 531–540. IEEE Computer Society, 2012. 8

[CGS15] Marek Cygan, Harold N. Gabow, and Piotr Sankowski. Algorithmic applications of baur-strassen's theorem: Shortest cycles, diameter, and matchings. *J. ACM*, 62(4), September 2015. 4

[CRS95] Suresh Chari, Pankaj Rohatgi, and Aravind Srinivasan. Randomness-optimal unique element isolation with applications to perfect matching and related problems. *SIAM Journal on Computing*, 24(5):1036–1050, 1995. 22

[Csa76] Laszlo Csanky. Fast parallel matrix inversion algorithms. *SIAM Journal on Computing*, 5(4):618–623, 1976. 2

[DK98] Elias Dahlhaus and Marek Karpinski. Matching and multidimensional matching in chordal and strongly chordal graphs. *Discrete Applied Mathematics*, 84(1–3):79 – 91, 1998. 3

[DKR10] Samir Datta, Raghav Kulkarni, and Sambuddha Roy. Deterministically isolating a perfect matching in bipartite planar graphs. *Theory of Computing Systems*, 47:737–757, 2010. 3

[DL78] Richard A. Demillo and Richard J. Lipton. A probabilistic remark on algebraic program testing. *Information Processing Letters*, 7(4):193 – 195, 1978. 2

[Edm70] Jack Edmonds. Submodular functions, matroids, and certain polyhedra. In *Combinatorial Structures and Their Applications, Gordon and Breach, New York*, pages 69–87, 1970. 7, 9, 16, 17

[EMSV12] David Eppstein, Elena Mumford, Bettina Speckmann, and Kevin Verbeek. Area-universal and constrained rectangular layouts. *SIAM J. Comput.*, 41(3):537–564, 2012. 5, 9, 26

[FGT16] Stephen A. Fenner, Rohit Gurjar, and Thomas Thierauf. Bipartite perfect matching is in quasi-nc. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA*, pages 754–763, 2016. 3, 6, 7, 8, 10, 11, 17, 22

[FGT19] Stephen Fenner, Rohit Gurjar, and Thomas Thierauf. Bipartite perfect matching is in quasi-NC. *SIAM Journal on Computing*, 0(0):STOC16–218–STOC16–235, 2019. 3

[FKS84] Michael L. Fredman, János Komlós, and Endre Szemerédi. Storing a sparse table with $O(1)$ worst case access time. *J. ACM*, 31(3):538–544, June 1984. 22

[GG11] Eran Gat and Shafi Goldwasser. Probabilistic search algorithms with unique answers and their cryptographic applications. *Electronic Colloquium on Computational Complexity (ECCC)*, 18:136, 2011. 3

[GG17] Shafi Goldwasser and Ofer Grossman. Bipartite perfect matching in pseudo-deterministic NC. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPIcs*, pages 87:1–87:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. 1, 3, 4, 5, 6, 8

[GGR13] Oded Goldreich, Shafi Goldwasser, and Dana Ron. On the possibilities and limitations of pseudodeterministic algorithms. In Robert D. Kleinberg, editor, *Innovations in Theoretical Computer Science, ITCS '13, Berkeley, CA, USA, January 9-12, 2013*, pages 127–138. ACM, 2013. 3

[GK87] Dima Grigoriev and Marek Karpinski. The matching problem for bipartite graphs with polynomially bounded permanents is in NC (extended abstract). In *28th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 166–172, 1987. 3

[GT17] Rohit Gurjar and Thomas Thierauf. Linear matroid intersection is in quasi-nc. In *49th Annual ACM Symposium on Theory of Computing*, pages 821–830, 2017. 3, 4, 5, 6, 7, 8, 9, 10, 11, 16, 17, 18, 20, 21

[ILG87] Robert W. Irving, Paul Leather, and Dan Gusfield. An efficient algorithm for the "optimal" stable marriage. *J. ACM*, 34(3):532–543, 1987. 5, 9, 26

[KS01] Adam R. Klivans and Daniel A. Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*, pages 216–223, 2001. 12, 22

[KUW86] Richard M. Karp, Eli Upfal, and Avi Wigderson. Constructing a perfect matching is in random NC. *Combinatorica*, 6(1):35–48, 1986. 2, 3

[KUW88] Richard M. Karp, Eli Upfal, and Avi Wigderson. The complexity of parallel search. *Journal of Computer and System Sciences*, 36(2):225 – 253, 1988. 1, 5

[Lov79] László Lovász. On determinants, matchings, and random algorithms. In *FCT*, volume 79, pages 565–574, 1979. 1, 2, 3

[MVV87] Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7:105–113, 1987. 2, 3, 5, 6, 12

[NSV94] H. Narayanan, Huzur Saran, and Vijay V. Vazirani. Randomized parallel algorithms for matroid union and intersection, with applications to arboresences and edge-disjoint spanning trees. *SIAM J. Comput.*, 23(2):387–397, 1994. 2, 3, 5, 6, 12, 22

[Ore22] Øystein Ore. Über höhere Kongruenzen. *Norsk Mat. Forenings Skrifter Ser. I*, 7(15):27, 1922. 2

[Oxl06] James G. Oxley. *Matroid Theory (Oxford Graduate Texts in Mathematics)*. Oxford University Press, Inc., New York, NY, USA, 2006. 15

[San18] Piotr Sankowski. NC algorithms for weighted planar perfect matching and related problems. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, volume 107 of *LIPIcs*, pages 97:1–97:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. 4, 8

[Sch80] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, October 1980. 2

[Sch03] Alexander Schrijver. *Combinatorial optimization : polyhedra and efficiency. Vol. B. , Matroids, trees, stable sets. chapters 39-69.* Algorithms and combinatorics. Springer-Verlag, Berlin, Heidelberg, New York, N.Y., et al., 2003. 4, 9, 10, 15, 16, 25, 26

[ST17] Ola Svensson and Jakub Tarnawski. The matching problem in general graphs is in quasi-nc. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 696–707, 2017. 3, 4, 10, 17, 22

[Sta11] Richard P. Stanley. *Enumerative Combinatorics*, volume 1 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, 2 edition, 2011. 13

[Sub95] Ashok Subramanian. A polynomial bound on the number of light cycles in an undirected graph. *Information Processing Letters*, 53(4):173 – 176, 1995. 7

[TV12] Raghunath Tewari and N. V. Vinodchandran. Green's theorem and isolation in planar graphs. *Information and Computation*, 215:1–7, 2012. 3

[Wig94] Avi Wigderson. Nl/poly <= +/poly (preliminary version). In *Proceedings of the Ninth Annual Structure in Complexity Theory Conference, Amsterdam, The Netherlands, June 28 - July 1, 1994*, pages 59–62. IEEE Computer Society, 1994. 12

[Zen93] Jiang Zeng. A bijective proof of Muir's identity and the Cauchy-Binet formula. *Linear Algebra and its Applications*, 184:79–82, 1993. 22

[Zip79] Richard Zippel. Probabilistic algorithms for sparse polynomials. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, EUROSAM '79, pages 216–226, 1979. 2

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il