Preprocessing Under Uncertainty: Matroid Intersection

Stefan Fafianie[1], Eva-Maria C. Hols[2], Stefan Kratsch[3], and Vuong Anh Quyen[4]

1 **Department of Computer Science, University of Bonn, Germany**
fafianie@cs.uni-bonn.de
2 **Department of Computer Science, University of Bonn, Germany**
hols@cs.uni-bonn.de
3 **Department of Computer Science, University of Bonn, Germany**
kratsch@cs.uni-bonn.de
4 **Department of Computer Science, University of Bonn, Germany**
vuong@cs.uni-bonn.de

Abstract

We continue the study of preprocessing under uncertainty that was initiated independently by Assadi et al. (FSTTCS 2015) and Fafianie et al. (STACS 2016). Here, we are given an instance of a tractable problem with a large static/known part and a small part that is dynamic/uncertain, and ask if there is an efficient algorithm that computes an instance of size polynomial in the uncertain part of the input, from which we can extract an optimal solution to the original instance for all (usually exponentially many) instantiations of the uncertain part.

In the present work, we focus on the MATROID INTERSECTION problem. Amongst others we present a positive preprocessing result for the important case of finding a largest common independent set in two linear matroids. Motivated by an application for intersecting two gammoids we also revisit MAXIMUM FLOW. There we tighten a lower bound of Assadi et al. and give an alternative positive result for the case of low uncertain capacity that yields a MAXIMUM FLOW instance as output rather than a matrix.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases preprocessing, uncertainty, maximum flow, matroid intersection

Digital Object Identifier 10.4230/LIPIcs.MFCS.2016.35

1 Introduction

Recently, Assadi et al. [2] and, independently, Fafianie et al. [12] initiated a study of problems where part of the input is *dynamic* or *uncertain*. While the introduced concepts are differently named, i.e., *dynamic sketching* [2] and *preprocessing under uncertainty* [12], and are rooted in different areas, i.e., *streaming algorithms* and *parameterized complexity* respectively, the fundamental goal is the same: Given an instance x that is largely static/known and a small specified part, say of k bits, that is dynamic/uncertain. Can we extract from x in polynomial time an instance (or just any string) x' of size polynomial in k such that optimal solutions for x for any instantiation of the k bits of dynamic/uncertain part can also be computed just from x' and the k bits? Since there are 2^k instantiations of k bits this is clearly nontrivial, as we can afford neither the time nor the space to simply precompute and store 2^k solutions, even for polynomial-time solvable problems (which are the focus of this work). Arguably,

this parallel development of essentially the same goal in different contexts speaks to the generality and importance of the question. Let us anyway recall some of the motivation.[1]

We are often faced with inputs that appear over time or that are subject to changes. The areas of *online algorithms* and *streaming algorithms* deal with the extreme case that the entire input is only revealed right before or during the computation, or that the data is continuously changing. What if we already hold "most" of the input and there is only a small amount of data that is uncertain or subject to changes? Can we do better than under the strict settings of online or streaming algorithms? Canonical examples are machine scheduling in a factory with only few irregularly occurring jobs, or routing in a network when there is a small set of links that are frequently congested. Unlike online algorithms, the goal is to obtain optimal solutions rather than approximate ones. To make this possible, we do not insist on committing to (parts of) a solution without knowing the uncertain part and only require to preprocess and shrink the given part to size polynomial in the amount of uncertain information. In other words, we do not desire a solution that is (approximately) robust to uncertainty, but to do as much work as possible before receiving the uncertain part (and then compute a solution). In the introduction of [12] this is sketched for the simple case of computing a shortest s,t-path in a road network when the transit times of some k roads is not known in advance; let us discuss a different straightforward example for illustration.

Consider the Closest Pair problem where we are given a set P of points in the Euclidean plane such that for k points $P' \subseteq P$ it is not known beforehand whether they appear in the final input. For any instantiation of availability of points in P' we know that the closest pair of points is either contained entirely in $P \setminus P'$ or P', or one point is in $P \setminus P'$ while the other is in P'. Accordingly, it suffices to store the closest pair in $P \setminus P'$, the set P', and for each point $p' \in P'$ its closest point in P. Thus, we obtain an equivalent instance with at most $2|P'| + 2$ points such that, after removing any subset of P', the closest pair of points has the same distance as it would have in the input instance.

The example shows that such a preprocessing can be quite simple and fast, and be feasible even for larger amounts of uncertain data. On the other end of the spectrum, there are problem settings where even an arbitrary amount of preprocessing time does not suffice to obtain a polynomially large sketch of the instance. Conveniently, the lower bound proofs do not require any complexity assumptions, but rely on fundamental information-theoretic arguments that are implicit in the well-known lower bound for the membership game: Therein, Alice holds any subset $S \subseteq [n]$ unknown to Bob, whereas Bob holds an integer $i \in [n]$ unknown to Alice, and communication is only allowed from Alice to Bob. How many bits of information does Alice need to send to Bob in order for Bob to be able to answer whether $i \in S$? The answer is that n bits are necessary and (obviously) sufficient. Note that the sent information necessarily works for all $i \in [n]$ that Bob could hold, and hence must represent the entire set S. In other words, the lower bound also gives us an incompressibility argument for n bits of information. Fafianie et al. [12] give lower bounds for small connected vertex cover and linear programming under various forms of uncertainty by showing that the existence of an efficient preprocessing algorithm would lead to a violation of the aforementioned bounds.

Assadi et al. [2] explore graph problems where uncertainty is restricted to a set T of k *terminal* vertices, e.g., their adjacency is uncertain. Their main positive result is on the maximum matching problem. Using the *Tutte matrix* of the input graph with n vertices, they show that storing only a $2k \times 2k$ matrix whose entries are in $\mathbb{Z}_p$ (p is any prime of

[1] In the following we will stick to the terms as used in [12].

magnitude $\Theta(\frac{n}{\delta})$ with $0 < \delta < 1$) and an integer suffices to extract the rank of the Tutte matrix for each instantiation/change to the adjacency of the terminals; this yields the size of a maximum matching of G. This result then gives rise to a *cut-preserving sketch* result where the value of all $(S, T \setminus S)$-cuts in G is preserved for $S \subseteq T$ in a sketch of size $\mathcal{O}(kC^2)$ where C is the total capacity of edges incident on T. They obtain this result by constructing a bipartite graph and creating a dynamic sketch for the maximum matching problem. In addition, they prove a lower bound of $\Omega(C/\log C)$ bits which implies a lower bound of $2^{\Omega(k)}$ bits. Furthermore, they show how to obtain a sketch of size $\mathcal{O}(k^4)$ for s,t-CONNECTIVITY, by again using the dynamic sketch for the maximum matching problem. These results extend to MAXIMUM FLOW as well: it follows that the maximum flow problem has a sketch of size $\mathcal{O}((k+C')^4)$ (in the form of a $8(k+C')^2 \times 8(k+C')^2$) matrix); here C' is the total capacity of edges between terminals, which is arguably an advantage over the dependency on C. They also point out that the maximum flow problem has a $2^{\Omega(k)}$ lower bound on size of dynamic sketches which follows from the lower bound for the cut-preserving sketch. Finally, they give an $\mathcal{O}(k)$ size dynamic sketch for the MINIMUM SPANNING TREE problem.

The result for MINIMUM SPANNING TREE was obtained independently by Fafianie et al. [12], where it is generalized to the problem of finding a minimum weight basis of a matroid when the presence of k ground set elements is uncertain. (The MST problem is the same as finding a minimum weight basis of a *graphic matroid*.) If the matroid is given by a matrix respectively by oracle access then the output is a smaller matroid given by a (smaller) matrix respectively by restricted access to the original oracle (e.g., smaller ground set). These results work also in the more general setting where the *weights* of k edges/elements are not known beforehand. Furthermore, for the BIPARTITE MATCHING problem with k uncertain vertices or edges, Fafianie et al. [12] show how to efficiently reduce to a new graph G' whose maximum matchings, relative to availability of the uncertain vertices, differ from those of the input graph G by a fixed (and known) amount. In other words, the output in all three cases is an equivalent instance of the same problem.

The known results leave different directions for further study. The main direction would be to study other polynomial-time solvable regarding preprocessing under uncertainty respectively dynamic sketching. There were several positive results, but the lower bounds for MAXIMUM FLOW and LINEAR PROGRAMMING show limitations for more general problems. Among other fundamental polynomial-time problems there are certainly MATROID INTERSECTION, LINEAR MATROID PARITY, STABLE MARRIAGE, and problem families such as string matching or scheduling. We note that the same problem may have several interesting variants for making parts of its input uncertain. Due the importance of MAXIMUM FLOW and LINEAR PROGRAMMING also restrictions of the problems, e.g., restricted input graphs or special types of LPs, seem reasonable in order to obtain positive results for them. Another question would be how important it is to have the output be an instance of the same problem. This is arguably beneficial for applications, especially if existing algorithms for the underlying problem can be applied as-is. Likely, lower bounds will be unaffected by this decision since we only know how to get lower bounds for the bit size of the encoding; this is similar to the situation of lower bounds for kernelization in parameterized complexity (cf. [6, 10]).

Our work. We focus mainly on preprocessing under uncertainty for the MATROID INTERSECTION problem (Section 4) and present three positive results, including the important case of intersecting two linear matroids. We also revisit MAXIMUM FLOW (Section 3) where we tighten a lower bound of Assadi et al. [2] and give a positive result that is used as a subroutine for GAMMOID INTERSECTION. We conclude with a brief discussion and point out some open problems (Section 5). Let us discuss the results in some more detail.

Matroid Intersection. In the Matroid Intersection problem we are given as input two matroids over the same ground set E, and the goal is to find a set of maximum cardinality which is independent in both matroids. This is a classical optimization problem studied since the early 1970s, e.g., in [1, 11, 18], and generalizes a wide range of concrete problems such as the bipartite matching problem and the colorful spanning tree problem. There are also applications of matroid intersection outside of combinatorial optimization [9, 20]. For many algorithms the independent sets of a matroid are given by an independence oracle, i.e., a blackbox algorithm which answers whether a given subset of the ground set is independent or not. Another common way is to represent a matroid by a matrix over some field: Matroids which can be represented in this way are called *linear matroids*. One can obtain more efficient algorithms for linear matroid intersection that work directly on a matrix rather than via an oracle; e.g., Gabow and Xu [13] make use of fast matrix multiplication. It is also possible to provide matroids implicitly, e.g., by the underlying graphs (*gammoids, graphic matroids*). Finding a maximum common independent set of two matroids is solvable in polynomial time [11] but finding a maximum common independent set of three matroids is NP-hard; e.g. the DIRECTED HAMILTON PATH problem can be formulated as the intersection of three matroids [26]. We study the LINEAR MATROID INTERSECTION problem in the setting that the presence of k elements in the ground set is uncertain. Solving all possible 2^k instantiations in polynomial time is impossible; but we will show how to construct a small encoding from which we can compute the size of a maximum common independent set for all instantiations.

To get the result for LINEAR MATROID INTERSECTION we use a result of Harvey [15], which determines the size of a maximum common independent set by computing the rank of a matrix Z that contains the matrix representations of the two linear matroids as sub-matrices. We use this matrix Z to compute a $2k \times 2k$ matrix and an integer from which we can compute the size of a maximum common independent set for all 2^k instantiations. The construction of the $2k \times 2k$ matrix uses similar ideas as the construction of the $2k \times 2k$ matrix for the dynamic sketching scheme for the maximum matching problem of Assadi et al. [2] and the compression for the K-set-cycle problem of Wahlström [25]. However, we have to be much more careful during row and column operations because our initial matrix has entire rows and columns whose presence is uncertain in the final instance; the uncertain part in the paper of Assadi et al. [2] is contained in a $k \times k$ sub-matrix of the initial matrix.

Since the output of our preprocessing is not an instance of LINEAR MATROID INTERSECTION, this poses the question of whether special cases of the problem permit a preprocessing whose output is an equivalent instance of the same problem. We prove this for the fairly general case of the intersection of two *gammoids*, which contains several classes of well-studied matroids (e.g., *transversal matroids*) and for the ROOTED ARBORESCENCE problem, where we want to determine the existence of a rooted arborescence in a directed graph with some uncertain arcs; note that the ROOTED ARBORESCENCE problem can be described as the intersection of a *partition matroid* and a *graphic matroid*. For the gammoid intersection problem, we show how to compute two new gammoids over the same ground set of size $\mathcal{O}(|T|^3)$ and an offset value from which we can compute the size of a maximum common independent set for all 2^k instantiations. For the ROOTED ARBORESCENCE problem we compute a graph with $k+1$ vertices from which we can decide for all 2^k instantiations whether the input instance has a rooted arborescence. We complement this by a lower bound of $\binom{k}{\lceil k/2 \rceil}$ bits for the case of k uncertain vertices.

Maximum Flow. The problem of finding a maximum flow is one of the most important problems besides the more general problem of solving linear programs and has been explicitly studied in graph theory and combinatorial optimization. We show that if there is an arc set

F of unit-capacity arcs whose presence is uncertain (equivalently: there is a total of $|F|$ units of uncertain capacity), while capacity of other arcs is arbitrary, then we can efficiently reduce to an equivalent unit-capacity flow network with $\mathcal{O}(|F|^3)$ vertices. If we are only interested in any encoding of small size, then instead of the obvious $\mathcal{O}(|F|^6)$ bits encoding size this can also be represented as a gammoid, using $\mathcal{O}(|F|^3)$ bits, which follows from results from Kratsch and Wahlström [17]. This improves the upper bound of Assadi et al. [2] who give a dynamic sketching scheme with sketch size $\mathcal{O}((k+C')^4)$ for the maximum flow problem (represented as a $\mathcal{O}((k+C')^2) \times \mathcal{O}((k+C')^2)$ matrix); since the size of a sketch is measured by the number of machine words of length $\mathcal{O}(\log(n))$, their sketch needs $\mathcal{O}((k+C')^4 \log(n))$ bits. Recall that C' is the total capacity of all edges between the k terminals; by small modifications to the graph $|F|$ and C' become comparable. We complement this by a lower bound of 2^k bits for the case of k uncertain arcs with large capacity which slightly improves the lower bound of $2^{\Omega(k)}$ of Assadi et al. [2]. Furthermore we show that our lower bound is tight, even when the encoding only preserves the parity of the maximum flows.

Further related work. Generally, apart from the areas of online and streaming algorithms, there are several models of optimization problems on uncertainty, such as *stochastic optimization* or *robust optimization*. We refer interested readers to some papers [3, 4, 5, 7] for more information. Some ideas of our work come from the area of kernelization from parameterized complexity, which is about preprocessing algorithms for NP-hard problems. Some particular results from this field inspired our work, namely a result of Pilipczuk et al. [22] on Steiner trees connecting terminals on the outer face of a plane graph, and a result of Kratsch and Wahlström [16] on cut-covering sets (which is also used in Section 3).

2 Preliminaries

Let $[n]$ denote the set $\{1, 2, \ldots, n\}$. If U is a set, then $\binom{U}{k}$ are all its subsets of size k.

We mostly use graph notation as given by Diestel [8]. For a graph $G = (V, E)$ and set of edges $F \subseteq E$, let $V(F)$ denote the vertices incident with F. For a vertex $v \in V$ we denote by $\delta(v)$ the set of edges that are incident to v; thus $\delta(v) = \{e \in E \mid v \in e\}$. Let $D = (V, A)$ be a directed graph and $v \in V$ be a vertex of D. For a vertex $v \in V$ we denote by $\delta^-(v)$ (resp. $\delta^+(v)$) the set of arcs $(u, v) \in A$ (resp. $(v, u) \in A$) with $u \in V$. Let $N^-(v)$ (resp. $N^+(v)$) denote the in-neighbors (resp. out-neighbors) of v. If F is a subset of V then we use $D - F$ to denote the graph obtained from D by deleting all vertices in F and $D[F]$ to denote the graph induced in D by F. If F is a subset of A then we use $D - F$ to refer the graph obtained from D by removing all edges in F. If f is a flow in D then we use $|f|$ to denote the value of f.

We use standard matroid notation as given by Oxley [21]. A *matroid* is a pair $(E, \mathcal{I})$, where E is a finite set of elements, called *ground set*, and $\mathcal{I}$ is a family of subsets of E which are called *independent sets* such that:

1. $\emptyset \in \mathcal{I}$.
2. If $A \in \mathcal{I}$, then for every subset $B \subseteq A$ we have $B \in \mathcal{I}$.
3. If A and B are two independent sets in $\mathcal{I}$ and $|A| > |B|$, then there is an element $e \in A \setminus B$ such that $B \cup \{e\} \in \mathcal{I}$.

Given a matroid $\mathcal{M} = (E, \mathcal{I})$ and $F \subseteq E$, we denote by $\mathcal{M}/F$ the matroid obtained from $\mathcal{M}$ by contracting F. The *rank function* corresponding to $\mathcal{M}$ is a function $r \colon 2^E \to \mathbb{N}$ which is defined by $r(S) \colon = \max\{|I| : I \subseteq S, I \in \mathcal{I}\}$.

Let us recall some well-known types of matroids. For any matrix A over some field F there is an associated matroid $\mathcal{M}$ on the set of columns with independence defined by linear

independence of the column vectors. We then say that A *represents* $\mathcal{M}$, and representable matroids are also called *linear matroids*. Let $U_1, \ldots, U_m$ be a collection of pairwise disjoint sets and $d_1, \ldots, d_m$ be integers with $0 \leq d_i \leq |U_i|$ for each $i = 1, \ldots, m$. If we set $E = \cup_{i=1}^{m} U_i$ and $\mathcal{I} = \{I \subseteq E \colon |I \cap U_i| \leq d_i \text{ for all } i = 1, \ldots, m\}$ then $(E, \mathcal{I})$ becomes a matroid, and matroids of this form are called *partition matroids*. The family of forests in a graph $G = (V, E)$ forms a matroid on E. Matroids that can be represented in this way are called *graphic matroids*. Let $G = (V, E)$ be a graph and S and T be two subsets of V. In the set T, we define a subset $U \subseteq T$ to be independent if there are $|U|$ vertex-disjoint paths from S onto U in G. Then this constructs a matroid on T, and matroids of this type are called *gammoids*.

3 Maximum flow

In the MAXIMUM FLOW problem we are given a directed graph $G = (V, A)$, capacities $c \colon A \to \mathbb{N}$, and two vertices $s, t \in V$; the task is to find a flow $f \colon A \to \mathbb{N}$ of maximum value. We consider preprocessing for the MAXIMUM FLOW problem for the case that capacity respectively presence of arcs in a set $F \subseteq A$ is not yet known. Results of this type were previously obtained by Assadi et al. [2]. We tighten one of their lower bounds and give a variant for the case of preserving the parity of the maximum flow. Moreover, we obtain a positive result for the case of $|F|$ uncertain arcs of unit capacity, which is a subroutine for our result for GAMMOID INTERSECTION. Crucially, the output of the latter is again an instance of MAXIMUM FLOW; it also implies a slightly improved encoding size in bits when represented by a matrix. In this section all capacities are integers, implying that there always exists an integral maximum flow. We tacitly assume that all considered maximum flows are integral.

▶ **Theorem 1.** *There is no algorithm that, given an instance $G = (V, A)$, $c \colon A \to \mathbb{N}^+$, and vertices $s, t \in V$ of* MAXIMUM FLOW *together with a set $F \subseteq A$, returns an encoding that requires fewer than $2^{|F|}$ bits, from which we can correctly extract the value of a maximum s,t-flow in $G - (F \setminus F')$ for all $F' \subseteq F$.*

It can be checked that the theorem is tight for the family of graphs used for the lower bound construction. The point is that the relevant information about each graph consists only of $2^{|F|}$ bits, and all flow values can be computed once the graph is known. In general, the lower bound should not be seen as the question of outright storing the $2^{|F|}$ results but regarding any way of storing enough information to compute requested values.

For an arbitrary graph with uncertain arcs F it is not clear whether $2^{|F|}$ bits are sufficient information to compute all flow values. Nevertheless, it is clearly enough space to store the *parities* of the maximum flows, and we can show that this is tight: By reinspecting our proof we can see that it can be adapted to the parity question. The key point is that the matrix used in the proof of Theorem 1 also has full rank over $GF(2)$, which can be easily verified.

▶ **Corollary 2.** *The lower bound of Theorem 1 is tight for* MAXIMUM FLOW PARITY*, i.e., given a graph $G = (V, A)$, $c \colon A \to \mathbb{N}^+$, and vertices $s, t \in V$, there is an encoding of $2^{|F|}$ bits, from which we can correctly extract the parity of the value of a maximum s,t-flow in $G - (F \setminus F')$ for all $F' \subseteq F$. There is no algorithm that returns a smaller encoding.*

The construction in the proof of Theorem 1 relies on uncertain edges with a high capacity. The following positive result shows that this is necessary since we can achieve an encoding to size polynomial in $|F| + l$ where l is the maximum capacity of any uncertain edge. Moreover, this also works if the capacity of edges in F can be instantiated to any value in $\{0, 1, \ldots, l\}$. The preprocessing can be performed by a randomized polynomial-time algorithm.

Let us first observe that we may easily reduce this question to the case that $l = 1$, which is equivalent to having a set $\hat{F}$ of edges of capacity 1 each that may or may not be present in the final instance: It suffices to replace each edge of F by l parallel edges of capacity 1 in $\hat{F}$. Setting the capacity of some $e \in F$ to a value $c(e) \in \{0, 1, \ldots, l\}$ is equivalent to making exactly $c(e)$ copies of e in $\hat{F}$ available (We can avoid getting a multigraph by subdividing the edges and putting one of the newly obtained edges in $\hat{F}$). Note that we make no assumption about the capacity of other edges, but the returned graph will have unit capacities.

▶ **Theorem 3.** *Let $G = (V, A)$ be a directed graph, $s, t \in V$, $F \subseteq A$, and $c\colon A \to \mathbb{N}$ be a capacity function such that $c(F) \equiv 1$. There exists a randomized polynomial-time algorithm that, given a network (G, s, t, c) and a set $F \subseteq A$ as above, returns a network $(G' = (V', A'), s, t, c')$ with $F \subseteq A'$, $c' \equiv 1$, $|V'| \in \mathcal{O}(|F|^3)$, and an integer $\alpha \in \mathbb{N}$ such that for any $F' \subseteq F$, the network $(G - F', s, t, c|_{A \setminus F'})$ has a maximum s,t-flow of value β if and only if the network $(G' - F', s, t, c'|_{A' \setminus F'})$ has maximum s,t-flow of value $\beta' = \beta - \alpha$. Here, α is the value of a maximum s,t-flow in $G - F$.*

As mentioned before, differing from the work of Assadi et al. [2] we are interested in finding a small instance of the same problem. In the full version we discuss how our resulting network (G', s, t, c') can be compressed into size $\mathcal{O}(|F|^3)$ bits which improves upon the upper bound of Assadi et al. [2].

The result is obtained by analyzing the *residual graph* with respect to any maximum flow f of $G - F$, i.e., not using any arc of F. We transform this graph in such a way that the cut-covering results of Kratsch and Wahlström [16] can be applied. Crucially, the residual graph has a small minimum s,t-cut size and its maximum flow with respect to deletion of $F' \subseteq F$ is equal to the possible additional flow in $G - F'$ as compared to f. Using the cut-covering set, a small equivalent instance can be obtained. A special case of Theorem 3, which is used for the gammoid intersection result, is the following.

▶ **Corollary 4.** *There exists a randomized polynomial-time algorithm that, given a directed graph $D = (V, A)$, two vertices $s, t \in V$ and a set $F \subseteq V \setminus \{s, t\}$, returns a directed graph $D' = (V', A)$ with $F \subseteq V'$, $|V'| \in \mathcal{O}(|F|^3)$ and an integer $\alpha \in \mathbb{N}$ such that $F \cup \{s, t\} \subseteq V'$ and for any $F' \subseteq F$, the maximum number of internally vertex-disjoint s,t-paths in $D - F'$ is β if and only if the maximum number of internally vertex-disjoint s,t-paths in $D' - F'$ is $\beta - \alpha$. Here, α is the maximum number of vertex-disjoint paths in $D - F$.*

4 Matroid intersection

In this section, we consider the well-known Matroid Intersection problem. In this problem, we are given two matroids $\mathcal{M}_1 = (E, \mathcal{I}_1)$ and $\mathcal{M}_2 = (E, \mathcal{I}_2)$ over the same ground set E. Our task is to determine the largest size of a set $I \subseteq E$ which is independent in both $\mathcal{M}_1$ and $\mathcal{M}_2$. We are interested in the case where the availability of a set of elements $F \subseteq E$ is uncertain and we want to know how much we can compress the input without knowing F. We obtain a positive result for the linear matroid intersection problem, where we construct a matrix from which we can compute the size of a common independent set. Furthermore, for two special cases of matroid intersection, gammoid intersection and rooted arborescence, we show how to obtain small instances of the same problem.

4.1 Linear matroid intersection

In this section we discuss preprocessing for the intersection problem of linear matroids over the same ground set $E = \{e_1, e_2, \ldots, e_m\}$ with respect to a set $F = \{e_1, e_2, \ldots, e_k\} \subseteq E$ of

elements whose presence in the final instance is uncertain. Throughout, we will refer to the two matroids in question as M_1 and M_2, and let A and B be $r \times m$ matrices over some field $\mathbb{F}$ that represent M_1 and M_2; here $r \leq m$ is an upper bound on the ranks of M_1 and M_2.

In previous work, e.g., for MINIMUM SPANNING TREE, it was possible to compute for each $F' \subseteq F$ an optimal solution that avoids F': The preprocessing would identify a set $X \subseteq E$ such that there exist solutions $X \cup Y_{F'}$ for each $F' \subseteq F$ where $Y_{F'}$ can be obtained from the outcome of the preprocessing. Unfortunately, this can be easily ruled out for LINEAR MATROID INTERSECTION: Let $G = (A \dot\cup B, E)$ be a bipartite graph and $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$ be two linear matroids over E with $\mathcal{I}_1 = \{E' \subseteq E \colon |\delta(v) \cap E'| \leq 1 \, \forall v \in A\}$ and $\mathcal{I}_2 = \{E' \subseteq E \colon |\delta(v) \cap E'| \leq 1 \, \forall v \in B\}$. It can be checked that a set E' is independent in M_1 and M_2 if and only if E' is a matching in G. Consider the bipartite graph that is a cycle of length $2n$. This graph has two disjoint maximum matchings E_1 and E_2. Let $F = \{e_1, e_2\}$ with $e_2 \in E_1$ and $e_1 \in E_2$. Now, the unique maximum common independent set in $M_1 - \{e_i\}$ and $M_2 - \{e_i\}$ is E_i for $i = 1, 2$. Thus, we cannot hope to identify $X \subseteq E$ that is shared by optimal solutions. Generally, the size of E cannot be bounded in terms of $|F|$ and as just seen the union of two maximum independent sets in two different instantiations can be the set E; hence we cannot hope to report for each $F' \subseteq F$ a largest common independent set $I \subseteq E \setminus F'$ from any preprocessed instance of size bounded in terms of $|F|$.

Instead, we will show that the *size* of a maximum common independent set in $M_1 - F'$ and $M_2 - F'$, for all $F' \subseteq F$, can be computed from (the rank of) an appropriate $2|F| \times 2|F|$ matrix M that is derived from A and B, which represent M_1 and M_2. To construct M we use a theorem due to Harvey [15]. Before stating the theorem, we need to introduce some notation. For each $J \subseteq E$ we define an $|E| \times |E|$ matrix $T(J)$ by

$$T(J)_{ij} := \begin{cases} 0 & \text{if } i \neq j \text{ or } i = j \in J, \\ t_i & \text{if } i = j \notin J, \end{cases}$$

where each t_i is an indeterminate. Next we define the matrix $Z(J)$ as

$$Z(J) := \begin{pmatrix} 0 & A \\ B^T & T(J) \end{pmatrix}.$$

By $\lambda(J)$ we denote the maximum cardinality of a set that is independent in the contracted matroids M_1/J and M_2/J. Later we consider these two matrices for the case that $J = \emptyset$; we define the shorthands $T = T(\emptyset)$, $Z = Z(\emptyset)$, and $\lambda = \lambda(\emptyset)$.

▶ **Theorem 5** (Harvey [15]). *Let M_1 and M_2 two linear matroids of rank r over the same ground set E. Let A (resp. B) be the $r \times m$ matrix that represents M_1 (resp. M_2). Let $r_1 \colon E \to \mathbb{N}$ and $r_2 \colon E \to \mathbb{N}$ the rank functions of M_1 (resp. M_2). For any $J \subseteq E$, we have* $\operatorname{rank}(Z(J)) = m + r_1(J) + r_2(J) - |J| + \lambda(J)$.

To determine the maximum cardinality of a set that is independent in M_1 and M_2, we use Theorem 5 for the case where $J = \emptyset$. For this case it implies $\operatorname{rank}(Z) = m + \lambda$. For $J = \emptyset$, this result was also obtained by Geelen [14] and it follows from the connection between matroid intersection and the Cauchy-Binet formula [24] (see also Murota [20, Remark 2.3.37]).

During the construction of the desired $2|F| \times 2|F|$ matrix M, we will perform many elementary row and column operations. This can lead to entries which are polynomials of large degree, because matrix T contains m indeterminates. To avoid this we replace some indeterminates by random elements from a field $\mathbb{F}$; this was also used in previous work [2, 25]. Performing row and column operations on the resulting matrix can cause elements to vanish

over $\mathbb{F}$, which can reduce the rank and thus lead to a wrong result. We bound the resulting error probability by using the Zippel-Schwartz lemma [23, 27].

Essentially, the idea of the proof is to derive an equivalent matrix whose rank can easily be computed from the rank of a couple of submatrices, where one of the submatrices is small and captures the uncertainty. Thus, we only need to keep the latter and store the rank of the other submatrices. A similar idea is used by Wahlström [25] and by Assadi et al. [2]. In the dynamic sketching scheme of Assadi et al. they have as initial matrix the Tutte matrix, where the uncertainty is contained in a $k \times k$ sub-matrix that contains indeterminates which can be set to zero in the final instance. In our case the initial matrix the is matrix Z which contains the sub-matrices A and B^T. The crucial difficulty is that we need the matrix Z from Harvey's theorem for each choice of $F' \subseteq F$ (or at least a matrix of same rank relative to an offset). Since each F' corresponds to a deletion of pairs of rows and columns, this is not simply handled by a small number of indeterminates that can be set to zero. Also, we cannot avoid using elements of these rows and columns for cancellation. We have to prove that our construction is *independent of the choice of* $F' \subseteq F$. This means, that taking the matrix Z for the matroids without elements of F' (which is same as deleting those rows and columns from Z) and applying our transformation yields the same result as first applying the transformation and then deleting rows and columns corresponding to F'.

To formally state our theorem and to describe the transformation steps let us denote by $W[F'^C, F'^C]$ the sub-matrix of W that contains all rows and columns that do not correspond to elements in $F' \subseteq F$, where W is the matrix in the current step; this means we delete the rows and columns that contain an indeterminate t_i with $e_i \in F'$.

▶ **Theorem 6.** *Let M_1 and M_2 be two linear matroids of rank r over the same ground set $E = \{e_1, e_2, \ldots, e_m\}$. Let A and B be $r \times m$ matrices over the same field $\mathbb{F}$ with $|\mathbb{F}| \geq N = 2^p(r+k)2^k$ that represent M_1 and M_2. Let $F = \{e_1, e_2, \ldots, e_k\} \subseteq E$ be the set of uncertain elements. There exists a randomized polynomial-time algorithm that, given the representations A and B of matroids M_1 and M_2 and $F \subseteq E$, returns a $2k \times 2k$ matrix M, and $\alpha \in \mathbb{N}$ such that with probability at least $1 - 2^{-p}$ for all $F' \subseteq F$, the maximum cardinality of a set that is independent in $M_1 - F'$ and $M_2 - F'$ is equal to* $\operatorname{rank}(M[F'^C, F'^C]) + \alpha - |F \setminus F'|$.

Proof sketch. In our case columns of A (resp. B) correspond to elements in E. For a set $X \subseteq E$ we denote by $A[X, \cdot]$ (resp. $B^T[\cdot, X]$) the matrix that contains the columns (resp. rows) that correspond to set X. Note that both row i and column i of matrix T correspond to the element $e_i \in E$, since $T[i,i] = t_i$. Therefore, by $T[X,X]$ we denote the submatrix of T that is induced by the rows an columns that correspond to the set X. To make sure that our preprocessing works for all choices of $F' \subseteq F$ we have to treat columns from A and B that correspond to elements in F differently from the remaining ones. To this end let $A_F = A[\cdot, F]$, $A_{E\setminus F} = A[\cdot, F^C]$, $B_F = B[\cdot, F]$, $B_{E\setminus F} = B[\cdot, F^C]$, $T_F = T[F,F]$ and $T_{E\setminus F} = T[F^C, F^C]$, i.e., $A = (A_F\, A_{E\setminus F})$, $B = (B_F\, B_{E\setminus F})$ and $T = \operatorname{diag}(T_F, T_{E\setminus F})$. We construct the matrix M in five steps, which we outline below. Due to space constrains, we only show how the construction of matrix M looks like; the crucial point of the proof, which is to show that the construction is independent on the choice of $F' \subseteq F$, is deferred to full version.

$$Z = \begin{pmatrix} 0 & A_F & A_{E\setminus F} \\ B_F^T & T_F & 0 \\ B_{E\setminus F}^T & 0 & T_{E\setminus F} \end{pmatrix} \xrightarrow{\text{step 1}} Z_1 = \begin{pmatrix} T_F & B_F^T \\ A_F & -A_{E\setminus F} T_{E\setminus F}^{-1} B_{E\setminus F}^T \end{pmatrix}$$

$$\xrightarrow{\text{step 2}} Z_2 = \begin{pmatrix} T_F & B_F^T \\ A_F & \widetilde{X} \end{pmatrix} \xrightarrow{\text{step 3}} Z_3 = \begin{pmatrix} T_F & {B_F}^T C \\ R A_F & D \end{pmatrix}$$

$$\xrightarrow{\text{step 4}} Z_4 = \begin{pmatrix} T_F - {B'}^T A' & {B''}^T \\ A'' & 0 \end{pmatrix} \xrightarrow{\text{step 5}} Z_5 = \begin{pmatrix} T_F - {B'}^T A' & \widetilde{B}^T \\ \widetilde{A} & 0 \end{pmatrix}$$

Step 1 In the first step, we reduce Z to an $(r+k)\times(r+k)$ matrix Z_1 such that the uncertain rows and columns remain unaffected; furthermore it holds that the rank of matrix Z equals the rank of matrix Z_1 plus $m-k$. We obtain matrix Z_1 by zeroing out the matrices $A_{E\setminus F}$ and $B_{E\setminus F}^T$ using the invertible matrix $T_{E\setminus F}$. Afterwards, we delete the rows and columns that contain matrix $T_{E\setminus F}$; this reduces the rank by $m-k$.

Step 2: We replace the indeterminate t_i for $k < i \leq m$ by random elements to avoid polynomials of large degree; the entries of matrix Z_1 are polynomials of degree at most one and this leads to an error probability of at most 2^{-p}. Denote the resulting sub-matrix by $\widetilde{X}$ and the complete matrix by Z_2. Note that we could replace the indeterminates before Step 1; but then we have to choose elements from a set of size at least $2^p(r+m)2^k$ to obtain the same error probability (instead of a set of size $N = 2^p(r+k)2^k$).

Step 3: Let $h = \operatorname{rank}(\widetilde{X})$. We apply elementary row and column operations to turn $\widetilde{X}$ into a diagonal matrix $D = \operatorname{diag}(1,\ldots,1,0,\ldots,0)$ with only h non-zero entries. It is well known that there exist two matrices R and C such that $D = R\widetilde{X}C$, where R is the product of all row operations and C the product of all column operations. Let Z_3 be the matrix after applying these row and column operations to Z_2. Matrix Z_2 and Z_3 have the same rank, because we only apply elementary row and column operations. Note that neither matrix R nor matrix C depend on the choice of $F' \subseteq F$, because $\widetilde{X}$ does not depend on the choice of $F' \subseteq F$. One can show that a column j of $R \cdot A_F$ (resp. $C^T \cdot B_F$) only depends on entries of column j of A_F (resp. B_F) and matrix entries that do not depend on the choice of $F' \subseteq F$; thus this transformation is independent on the choice of $F' \subseteq F$.

Step 4: Let $A' = (R \cdot A_F)[[h],\cdot]$ and let $A'' = (R \cdot A_F)[\{h+1, h+2, \ldots, r\},\cdot]$, i.e. $R \cdot A_F = \begin{pmatrix} A' & A'' \end{pmatrix}^T$. Analog we define the $h \times k$ sub-matrix B' and the $(r-h)\times k$ sub-matrix B'' of $B_F^T \cdot C$, i.e. $B_F^T \cdot C = \begin{pmatrix} B'^T & B''^T \end{pmatrix}$. Note that A' (resp. ${B'}^T$) correspond to the rows (resp. columns) where matrix D has a non-zero entry. We zero-out matrices A' and ${B'}^T$ using the identity matrix I_h. Afterwards, we delete the rows and columns that contain matrix I_h. We denote the resulting matrix by Z_4. By a well-known fact from linear algebra we have that matrix Z_4 has rank l if and only if matrix Z_3 has rank $l+h$.

Step 5: Since A'' (resp. B'') is an $(r-h)\times k$ matrix, at most k rows can be linear independent. We pick a maximum set of linear independent rows from A'' (resp. B''); if less than k rows are independent, then we arbitrarily pick from the remaining rows or add zero rows until we have k rows. Denote this matrix by $\widetilde{A}$ (resp. $\widetilde{B}$). This results in the $2k \times 2k$ matrix Z_5. Note that matrix Z_4 and matrix Z_5 have the same rank, because deleting rows that are dependent on the rows in $\widetilde{A}$ (resp. columns in $\widetilde{B}^T$) corresponds to row (resp. column) operations that zero-out these rows (resp. columns) and this does not change the rank of a matrix.

Altogether, we have constructed a $2|F| \times 2|F|$ matrix $M = Z_5$ and, as we show in the full version, for all $F' \subseteq F$ the equation $\operatorname{rank}(M[{F'}^C, {F'}^C]) + \beta = \operatorname{rank}(Z[{F'}^C, {F'}^C])$ holds with an error probability of at most 2^{-p} where $\beta = h + m - k$. Since the matroid $M_1 - F'$ (resp.

the matroid $M_2 - F'$) is represented by the matrix $A[\cdot, {F'}^C]$ (resp. the matrix $B[\cdot, {F'}^C]$), and the matroids M_1 and M_2 are defined over the same ground set of size $m - |F'|$ it follows from Theorem 5 that $\text{rank}(Z[{F'}^C, {F'}^C]) = (m - |F'|) + \lambda_{F'}$, where $\lambda_{F'}$ denotes the maximum cardinality of a set that is independent in $M_1 - F'$ and $M_2 - F'$. Combining these two equations result in equation $\lambda_{F'} = \text{rank}(M[{F'}^C, {F'}^C]) + h - |F \setminus F'|$. All operations required to obtain the matrix M can be performed in polynomial time and the matrix M together with integer $\alpha = h$ satisfy the required properties in Theorem 6. This completes the proof. ◀

4.2 Gammoid intersection

In this section, we consider the Matroid Intersection problem for the case that both matroids are gammoids. Since gammoids are also linear matroids (cf. [19]) we could apply Theorem 6 to obtain an encoding of size polynomial in the uncertain part. We show how to compute an instance of the gammoid intersection problem instead of an arbitrary encoding. In the same way, the following preprocessing result for the problem of intersecting two gammoids can also be applied to special cases of gammoids such as *transversal matroids* and *partition matroids*, but extra work would be needed to ensure that the output is again, e.g., a pair of transversal matroids rather than general gammoids.

▶ **Theorem 7.** *There exists a randomized polynomial-time algorithm that, given two gammoids $\mathcal{M}_1 = \mathcal{M}(G_1, S_1, T)$ and $\mathcal{M}_2 = \mathcal{M}(G_2, S_2, T)$ together with a subset $F \subseteq T$, returns two new gammoids $\mathcal{M}'_1 = \mathcal{M}(G'_1, S'_1, T')$ and $\mathcal{M}'_2 = \mathcal{M}(G'_2, S'_2, T')$ over a ground set T' of size $\mathcal{O}(|F|^3)$ together with an integer $\alpha \in \mathbb{N}$ such that $F \subseteq T'$ and for every $F' \subseteq F$, $\mathcal{M}_1 - F'$ and $\mathcal{M}_2 - F'$ have a maximum common independent set of size l if and only if $\mathcal{M}'_1 - F'$ and $\mathcal{M}'_2 - F'$ have a maximum common independent set of size $l - \alpha$.*

Proof. Because T appears in both graphs, G_1 and G_2, for each vertex $v \in T$ we rename it in G_1 by v_1 and G_2 by v_2 respectively. We obtain two sets T_1, T_2 where $T_i = \{v_i : v \in T\}$ plays the role of T in G_i and $\mathcal{M}_i$. In order to apply Corollary 4, we construct a graph G as follows. We first reverse all arcs in G_2 to obtain $\overleftarrow{G_2}$ and take the union of G_1 and $\overleftarrow{G_2}$. For each $v \in T$, we create a new vertex in G, also named v, and add two arcs $(v_1, v), (v, v_2)$ to G. Then we create a source s and a sink t together with arcs from s to each vertex in S_1 and arcs from each vertex in S_2 to t. Thus, we obtain a graph $G = (V, E)$ such that $T \subseteq V$ is an (s, t)-cut and there is no arc from $\overleftarrow{G_2}$ to G_1 in G.

▶ **Claim 1.** *For all $F' \subseteq F$, the maximum cardinality of a common independent set of $\mathcal{M}_1 - F'$ and $\mathcal{M}_2 - F'$ is equal to the maximum number of internally vertex-disjoint s, t-paths in $G - F'$.*

We apply the algorithm given by Corollary 4 for G and F to compute a graph $G' = (V', A')$ with $\mathcal{O}(|F|^3)$ vertices and an integer $\alpha' \in \mathbb{N}$ such that $F \subseteq V'$ and for any $F' \subseteq F$, the maximum number of internally vertex-disjoint s, t-paths in $G - F'$ is l if and only if the maximum number of internally vertex-disjoint s, t-paths in $G' - F'$ is $l - \alpha'$. Using graph G' we construct the two new gammoids $\mathcal{M}'_1$ and $\mathcal{M}'_2$. We obtain the graph G'_1 by adding two new vertices $s_v, \hat{v}$ as well as the arcs $(s_v, v), (s_v, \hat{v})$ to G' for all vertices $v \in F$. Let $S_F := \{s_v : v \in F\}$, and let N^+ (resp. N^-) denote the out-neighbors of s (resp. in-neighbors of t). The first gammoid $\mathcal{M}'_1 = \mathcal{M}(G'_1, S'_1, T')$ is defined by the graph G'_1, the set of sources $S'_1 = N^+ \cup S_F$ and the ground set $T' = N^- \cup F \cup \hat{F}$ and the second gammoid $\mathcal{M}'_2 = \mathcal{M}(G'_2, S'_2, T')$ is defined by the graph $G'_2 = (S'_2 \cup T', \{(s_v, v), (s_v, \hat{v}) : v \in F\})$, the set of sources $S'_2 = S_F \cup N^-$ and the ground set T'. For each $F' \subseteq F$, let $\hat{F}' := \{\hat{v} : v \in F'\}$.

▶ **Claim 2.** *For every $F' \subseteq F$, the maximum number of internally vertex-disjoint s,t-paths in $G' - F'$ is h if and only if the maximum cardinality of a common independent set of $\mathcal{M}'_1 - \hat{F}'$ and $\mathcal{M}'_2 - \hat{F}'$ is $h + |F|$.*

We conclude that for every $F' \subseteq F$, the maximum cardinality of a common independent set in $\mathcal{M}_1 - F'$ and $\mathcal{M}_2 - F'$ is l if and only if the maximum cardinality of a common independent set in $\mathcal{M}'_1 - \hat{F}'$ and $\mathcal{M}'_2 - \hat{F}'$ is $l - \alpha' + |F|$. Note that in the construction of $\mathcal{M}'_1$ and $\mathcal{M}'_2$, $\hat{F}$ is a copy of F, so if we identify $\hat{F}$ with the set F in the input gammoids and set $\alpha = \alpha' - |F|$, then we obtain the desired result. ◀

4.3 Rooted arborescence

In this subsection we consider the Rooted Arborescence problem, where we are given a directed graph $D = (V, A)$ with root r and we need to determine whether there exists an r-arborescence in D. An *arborescence* with root $r \in V$, or an *r-arborescence*, is an arc set $A' \subseteq A$ such that A' is a spanning tree if considered as an undirected subgraph and every $v \in V$ is reachable from r via arcs of A', i.e., there is a directed path from r to v using only arcs of A'. We are again interested in the case that there is uncertainty in the input, more precisely, that there are some arcs or vertices whose presence is not known. Rooted Arborescence can be considered as a special case of Matroid Intersection. Indeed, let $\mathcal{M}_1$ be the graphic matroid defined on the undirected underlying graph corresponding to D and $\mathcal{M}_2 = (A, \mathcal{I})$ where $\mathcal{I} = \{S \subseteq A : |S \cap \delta^-(v)| \leq 1 \text{ for all } v \in V \setminus r\}$. It can be checked that D has an r-arborescence if and only if $\mathcal{M}_1$ and $\mathcal{M}_2$ have a common independent set of size $|V| - 1$. Uncertainty of some elements in the two matroids corresponds to uncertain appearance of some arcs in D. By constructing reduction rules based on a well-known property of arborescences, we obtain the following result.

▶ **Theorem 8.** *There exists a polynomial-time algorithm that, given a directed graph $D = (V, A)$ with a root $r \in V$ and an arc set $F \subseteq A$, returns a directed graph $D' = (V(F) \cup \{r\}, A')$ with $F \subseteq A'$ such that for every $F' \subseteq F$, the graph $D - F'$ has an r-arborescence if and only if $D' - F'$ has an r-arborescence.*

In the next theorem, we consider the case that there are some uncertain vertices in our input graph and prove a lower bound for it. The construction is similar to the one used for maximum flow by Assadi et al. [2]. Note that this is not a special case of Matroid Intersection with uncertainty about ground set elements.

▶ **Theorem 9.** *There is no algorithm that, given an instance of* Rooted Arborescence *with k uncertain vertices, returns an encoding that requires fewer than $\binom{k}{\lceil k/2 \rceil}$ bits from which we can correctly extract the answer for all 2^k instantiations of the input instance.*

5 Conclusion

We have continued the study of preprocessing under uncertainty initiated by Assadi et al. [2] (who called their notion *dynamic sketching*) and Fafianie et al. [12]. Our main focus was on preprocessing for matroid intersection problems when the presence of certain ground set elements is uncertain. We obtained positive results for (i) intersecting two linear matroids, (ii) intersecting two gammoids, and (iii) the Rooted Arborescence problem. For the latter two problems our preprocessing returns an instance of the respective problem; for the former, the output is in form of a matrix. Additionally, we have revisited maximum flow, also studied by Assadi et al. [2]. We have tightened a lower bound construction and

gave a variant of the result for preserving the parity of maximum flows. Furthermore, we obtained a positive result for the case that a small amount of capacity is uncertain, with output again an instance of MAXIMUM FLOW, which is used for the gammoid intersection result. Deriving a matrix encoding from this yields bitsize $\mathcal{O}(|F|^3)$, improving slightly over $\mathcal{O}((k+C')^4 \log n)$ [2].

It would be interesting whether our result for LINEAR MATROID INTERSECTION can be generalized to the weighted case, possibly with uncertain weights. Similarly, one can try to extend the result to arbitrary matroids that are given by an independence oracle. Generally, preprocessing under uncertainty (or dynamic sketching) in the oracle setting is interesting, as the presence of oracles precludes the lower bounds based on MEMBERSHIP.

References

1 Martin Aigner and Thomas A Dowling. Matching theory for combinatorial geometries. *Transactions of the American Mathematical Society*, 158(1):231–245, 1971.

2 Sepehr Assadi, Sanjeev Khanna, Yang Li, and Val Tannen. Dynamic sketching for graph optimization problems with applications to cut-preserving sketches. In Prahladh Harsha and G. Ramalingam, editors, *35th IARCS Annual Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS 2015, December 16-18, 2015, Bangalore, India*, volume 45 of *LIPIcs*, pages 52–68. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015. doi:10.4230/LIPIcs.FSTTCS.2015.52.

3 Dimitris Bertsimas, David B. Brown, and Constantine Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, 2011. doi:10.1137/080734510.

4 Hans-Georg Beyer and Bernhard Sendhoff. Robust optimization–a comprehensive survey. *Computer methods in applied mechanics and engineering*, 196(33):3190–3218, 2007.

5 Leonora Bianchi, Marco Dorigo, Luca Maria Gambardella, and Walter J Gutjahr. A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing: an international journal*, 8(2):239–287, 2009.

6 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

7 George B. Dantzig. Linear programming under uncertainty. *Management Science*, 50(12-Supplement):1764–1769, 2004. doi:10.1287/mnsc.1040.0261.

8 Reinhard Diestel. *Graph theory (Graduate texts in mathematics)*. Springer Heidelberg, 2005.

9 Randall Dougherty, Chris Freiling, and Kenneth Zeger. Network coding and matroid theory. *Proceedings of the IEEE*, 99(3):388–405, 2011.

10 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.

11 Jack Edmonds. Submodular functions, matroids, and certain polyhedra. *Combinatorial structures and their applications*, pages 69–87, 1970.

12 Stefan Fafianie, Stefan Kratsch, and Vuong Anh Quyen. Preprocessing under uncertainty. In Nicolas Ollinger and Heribert Vollmer, editors, *33rd Symposium on Theoretical Aspects of Computer Science, STACS 2016, February 17-20, 2016, Orléans, France*, volume 47 of *LIPIcs*, pages 33:1–33:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPIcs.STACS.2016.33.

13 Harold N Gabow and Ying Xu. Efficient theoretic and practical algorithms for linear matroid intersection problems. *Journal of Computer and System Sciences*, 53(1):129–147, 1996.

14 J. F. Geelen. Matching theory. *Lecture Notes from the Euler Institute for Discrete Mathematics and Its Applications*, 2001.

15 Nicholas J. A. Harvey. Algebraic structures and algorithms for matching and matroid problems. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 531–542. IEEE Computer Society, 2006. doi:10.1109/FOCS.2006.8.

16 Stefan Kratsch and Magnus Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 450–459. IEEE Computer Society, 2012. doi:10.1109/FOCS.2012.46.

17 Stefan Kratsch and Magnus Wahlström. Compression via matroids: A randomized polynomial kernel for odd cycle transversal. *ACM Transactions on Algorithms*, 10(4):20:1–20:15, 2014. doi:10.1145/2635810.

18 Eugene L Lawler. Matroid intersection algorithms. *Mathematical programming*, 9(1):31–56, 1975.

19 Dániel Marx. A parameterized view on matroid optimization problems. *Theor. Comput. Sci.*, 410(44):4471–4479, 2009. doi:10.1016/j.tcs.2009.07.027.

20 Kazuo Murota. *Matrices and matroids for systems analysis*, volume 20. Springer Science & Business Media, 2009.

21 James Oxley. *Matroid Theory*. Oxford University Press, 2011.

22 Marcin Pilipczuk, Michal Pilipczuk, Piotr Sankowski, and Erik Jan van Leeuwen. Network sparsification for Steiner problems on planar and bounded-genus graphs. In *FOCS 2014*, pages 276–285. IEEE Computer Society, 2014. doi:10.1109/FOCS.2014.37.

23 Jacob T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, 1980. doi:10.1145/322217.322225.

24 Nobuaki Tomizawa and Masao Iri. Algorithm for determining rank of a triple matrix product axb with application to problem of discerning existence of unique solution in a network. *Electronics & Communications in Japan*, 57(11):50–57, 1974.

25 Magnus Wahlström. Abusing the Tutte Matrix: An Algebraic Instance Compression for the K-set-cycle Problem. In Natacha Portier and Thomas Wilke, editors, *30th International Symposium on Theoretical Aspects of Computer Science (STACS 2013)*, volume 20 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 341–352, Dagstuhl, Germany, 2013. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.STACS.2013.341.

26 Dominic JA Welsh. *Matroid theory*. Courier Corporation, 2010.

27 Richard Zippel. Probabilistic algorithms for sparse polynomials. In Edward W. Ng, editor, *Symbolic and Algebraic Computation, EUROSAM '79, An International Symposiumon Symbolic and Algebraic Computation, Marseille, France, June 1979, Proceedings*, volume 72 of *Lecture Notes in Computer Science*, pages 216–226. Springer, 1979. doi:10.1007/3-540-09519-5_73.