MATROID COMPLEXITY AND NONSUCCINCT DESCRIPTIONS

DILLON MAYHEW

Abstract. We investigate an approach to matroid complexity that involves describing a matroid via a list of independent sets, bases, circuits, or some other family of subsets of the ground set. The computational complexity of algorithmic problems under this scheme appears to be highly dependent on the choice of input type. We define an order on the various methods of description, and we show how this order acts upon 10 types of input. We also show that under this approach several natural algorithmic problems are complete in classes thought not to be equal to P.

1. Introduction

The study of matroid-theoretical algorithmic problems and their complexity has been dominated by two approaches. The first approach was implicitly used by Edmonds [1] and later developed by Hausmann and Korte [4, 5]. It uses the idea of an ordinary Turing machine augmented with an oracle. Suppose that the subject of the computation is a matroid on the ground set E. When queried about a subset, X, of E, the oracle returns in unit time some information about X. That information might be the rank of X, or an answer to the question, "Is X independent?" In this case the oracle is known as the rank or independence oracle, respectively.

The second dominant approach to matroid complexity uses the standard model of a Turing machine but considers as its input only a restricted class of matroids that can be represented by some "succinct" structure, for instance, a graph, or a matrix over a field.

A third approach to the study of matroid complexity would, in some ways, be more natural. A matroid is essentially a finite set with a structured family of subsets (we shall not consider infinite matroids). An obvious way to describe a matroid to a Turing machine is to simply list the subsets that belong to this family. The advantage of this approach is that any matroid could be received as input, and not merely those matroids which belong to some restricted class.

The reason this approach has not received as much attention as the latter two is the concern that the input will be too large. A result of Knuth [8] says that if $f(n)$ is the number of nonisomorphic matroids on a set of size n, then there is a constant c such that

$$\log_2 \log_2 f(n) \geq n - \frac{3}{2} \log_2 n + c \log_2 \log_2 n$$

for sufficiently large values of n. It follows that if Σ is a finite alphabet and $\sigma : \mathcal{M} \to \Sigma^*$ is an injective encoding scheme that takes the set of all matroids to words in Σ, then there can be no polynomial function p such that $|\sigma(M)|$, the length of the word $\sigma(M)$, is bounded above by $p(|E(M)|)$ for every matroid M. Thus if we wish to use this type of encoding function we must in some sense abandon the cardinality of the ground set of a matroid as a measure of its "size." Historically the concern with this type of scheme has been that, because of the large size of the input, the class of problems that can be solved in polynomial time will be artificially inflated, and that therefore, in a trivial way, all algorithmic problems for matroids will be tractable.

In this paper we show that the situation is apparently more subtle than this. We show that several natural matroid problems are complete in classes thought not to be equal to P, even using an encoding scheme that works for all matroids; for instance, the scheme that describes a matroid by listing its independent sets.

A quirk of this approach to matroid complexity is that the difficulty of a computational problem appears to vary widely according to the type of input. A problem may be solvable in polynomial time if the input takes the form of a list of bases, but if the input is a list of circuits, the same problem may be NP-complete.

Before we examine the complexity of matroid-theoretical problems, we define an order on types of input, and we show how this order acts upon a set of 10 natural methods of description. This work is an analogue of that done by Hausmann and Korte [6] and Robinson and Welsh [12]. Both of these works compare different types of oracles.

Our references for basic concepts, notation, and terminology will be Oxley [11] with regards to matroids and Garey and Johnson [2] with regards to complexity theory.

2. Various types of inputs

The 10 types of input that we consider are as follows: Rank, Independent Sets, Spanning Sets, Bases, Flats, Circuits, Hyperplanes, NonSpanning Circuits, Dependent Hyperplanes, and Cyclic Flats.

Of these, some need little explanation. A list of the independent sets, spanning sets, bases, flats, circuits, or hyperplanes of a matroid uniquely specifies that matroid. Thus the corresponding forms of input will consist simply of lists of the appropriate subsets.

The Rank input will list each subset of the ground set, along with its rank. The NonSpanning Circuits input for a matroid M will specify the rank of M and list all its nonspanning circuits. Dually, the Dependent Hyperplanes input will specify the rank of M and list its dependent hyperplanes.

A *cyclic flat* is a flat that is also a (possibly empty) union of circuits. It is known that listing the cyclic flats and specifying their ranks completely determines a matroid. Therefore the Cyclic Flats input will list each cyclic flat, along with its rank.

(Note that we have not specified how to describe some exceptional cases, such as a matroid M with no nonspanning circuits. In this particular case we will assume that the NonSpanning Circuits description lists only the rank of M. Other exceptional cases are easily dealt with in a similar way.)

Suppose that f and g are two functions on the positive integers. If there exist constants, c_1 and c_2, and an integer, N, such that $c_1 g(n) \leq f(n) \leq c_2 g(n)$ for every positive integer $n \geq N$, then we shall write $f = \Theta(g)$. Equivalently, $f = \Theta(g)$ if and only if $f = O(g)$ and $g = O(f)$.

Suppose that M is a matroid on a ground set of size n, and that Input is one of the types of input discussed above. Let (M, Input) be a word that describes M via Input. We shall assume that a subset of the ground set is specified by its characteristic vector. Thus if the Input description involves listing i subsets of $E(M)$, then $|(M, \text{Input})| \geq ni$. We shall consider only "reasonable" encoding schemes (i.e., those that do not allow, for instance, padding of words). It follows easily that $|(M, \text{Input})| = \Theta(ni)$.

Obviously there are many other natural ways of describing a matroid, but many are related in a fairly trivial way to one of the methods we have already discussed. For example, a matroid can be described by listing its cocircuits, but the cocircuits are exactly the complements of the hyperplanes.

3. A comparison of inputs

It is natural to ask whether one form of description is intrinsically more compact than another. In this section we attempt to answer that question.

Definition 3.1. Suppose that Input1 and Input2 are two methods for describing a matroid. Then Input1 $\leq$ Input2 if there exists a polynomial-time Turing machine which will produce (M, Input2) given (M, Input1) for any matroid M.

Suppose that Input1 $\leq$ Input2. If a problem is in P for descriptions via Input2, then clearly it is in P for Input1. Similarly, if a problem is NP-complete for Input1, then the same problem is NP-hard for Input2.

It is clear that $\leq$ is both reflexive and transitive. The rest of this section will be devoted to proving Theorem 3.2.

Theorem 3.2. *The* 10 *types of input listed in section* 2 *are ordered by* $\leq$ *according to the Hasse diagram in Figure* 1.

Theorem 3.2 will follow from the subsequent lemmas.

Lemma 3.3. Rank $\leq$ Spanning Sets *and* Rank $\leq$ Independent Sets.

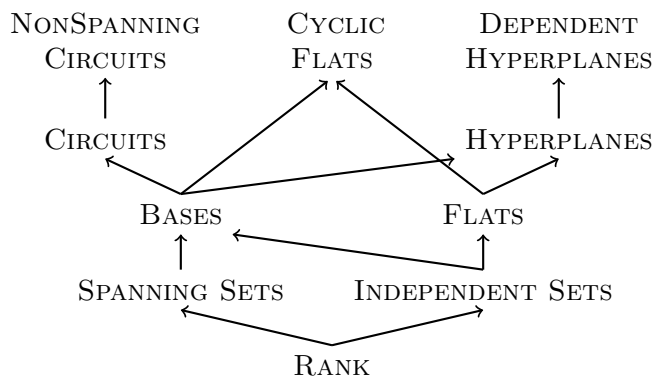

Figure 1. An ordering of input types.

Proof. To find (M, Spanning Sets) given (M, Rank), a Turing machine need only write onto its output those subsets whose rank equals $r(M)$. Similarly, to produce (M, Independent Sets) the machine need only write those subsets of $E(M)$ whose rank is equal to their size. □

Lemma 3.4. Spanning Sets $\leq$ Bases *and* Independent Sets $\leq$ Bases.

Proof. The bases of a matroid are exactly the minimal spanning sets and the maximal independent sets, so the result follows easily. □

Lemma 3.5. Independent Sets $\leq$ Flats.

Proof. Given the list of independent sets of M, we can create the list of flats by finding, in turn, the closure of each independent set, and then eliminating duplications. To find the closure of the independent set I, we check each set of the form $I \cup e$, where $e \notin I$, to see if it is independent. The element e belongs to $\mathrm{cl}(I)$ if and only if $I \cup e$ is dependent. Since $|(M, \text{Independent Sets})| = \Theta(ni)$, where $|E(M)| = n$ and the number of independent sets is i, it is easy to see that this entire procedure can be accomplished in polynomial time. □

Lemma 3.6. Bases $\leq$ Circuits.

Proof. If B is a basis of a matroid M and $e \in E(M) - B$, then $B \cup e$ contains a unique circuit $C(e, B)$, known as the *fundamental circuit of* e *with respect to* B. Since every circuit is a fundamental circuit with respect to some basis, we can construct the list of circuits of M, given the list of bases, by creating the list of fundamental circuits with respect to each of the bases, and then eliminating duplications.

If B is a basis of M, and $e \in E(M) - B$, we find $C(e, B)$ by comparing $(B \cup e) - f$ against the list of bases for each $f \in B$. The element f is in $C(e, B)$ if and only if $(B \cup e) - f$ is a basis. □

The next result follows easily using duality and Lemma 3.6.

Lemma 3.7. Bases $\leq$ Hyperplanes.

Let the number of bases and cyclic flats in the matroid M be denoted by $b(M)$ and $z(M)$, respectively.

Proposition 3.8. *Let M be a matroid. Then $z(M) \leq b(M)$.*

Proof. The proof will be by induction on $|E(M)|$. If $|E(M)| = 0$, then $b(M) = z(M) = 1$, so the proposition holds.

Let M be a matroid such that $|E(M)| = n > 0$, and assume that the proposition holds for all matroids on ground sets of $n-1$ elements. We may assume that there exists an element $e \in E(M)$ such that e is neither a loop nor a coloop, for otherwise $b(M) = z(M) = 1$.

Let $b_e(M)$ be the number of bases of M that contain e, and let $b_{\bar{e}}(M)$ be the number of bases of M that avoid e. Then $b_e(M) = b(M/e)$ and $b_{\bar{e}}(M) = b(M\backslash e)$. Similarly, let $z_e(M)$ be the number of cyclic flats of M that contain e, and let $z_{\bar{e}}(M)$ be the number of cyclic flats that do not contain e. It is easy to see that any cyclic flat that does not contain e is also a cyclic flat of $M\backslash e$, so $z_{\bar{e}}(M) \leq z(M\backslash e)$. Moreover, if Z is a cyclic flat of M, and $e \in Z$, then $Z - e$ is a cyclic flat of M/e. Thus $z_e(M) \leq z(M/e)$, and hence

$$z(M) = z_e(M) + z_{\bar{e}}(M) \leq z(M/e) + z(M\backslash e).$$

By the inductive hypothesis, $z(M/e) \leq b(M/e)$ and $z(M\backslash e) \leq b(M\backslash e)$, so

$$z(M) \leq b(M/e) + b(M\backslash e) = b_e(M) + b_{\bar{e}}(M) = b(M).$$

Lemma 3.9. Bases $\leq$ Cyclic Flats.

Proof. Our algorithm for generating the list of cyclic flats, given the list of bases, will start by constructing the closures of all circuits. At each repetition of the loop the algorithm will find the closure of the union of every pair of cyclic flats already on the list.

Suppose that M is a matroid on the ground set E. Note that we can check in polynomial time whether a set is independent by comparing it against the family of bases. Hence if $A \subseteq E$, we can use the greedy algorithm to find a basis I of A. The element $e \notin A$ is in $\mathrm{cl}(A)$ if and only if $I \cup e$ is dependent. It follows that we can find $\mathrm{cl}(A)$ in polynomial time.

From Lemma 3.6 we see that we can find the list of circuits in polynomial time, given the list of bases. Hence we can also construct the list of closures of circuits, add the closure of the empty set, and then eliminate duplications from the list in polynomial time. This completes the preprocessing that the algorithm will do before entering the loop.

Suppose that $Z_1, \ldots, Z_t$ is the list of cyclic flats that has been constructed after the loop has been repeated i times. In the next repetition of the loop the algorithm will take each pair $\{Z_j, Z_k\}$ of cyclic flats and find $\mathrm{cl}(Z_j \cup Z_k)$. At the completion of the loop the algorithm will add these new cyclic flats to the list and then eliminate duplications. By Proposition 3.8 the length of the list at the beginning of the loop will never exceed b, the number of bases.

Thus in each repetition of the loop the algorithm must find the closure of at most b^2 unions of flats. It is easy to see that after $r(M)$ repetitions of the loop the algorithm will have found every cyclic flat. Finding the rank of these flats can clearly be accomplished in polynomial time, using the greedy algorithm. Therefore the algorithm can construct (M, Cyclic Flats) in polynomial time. □

Lemma 3.10. Flats $\leq$ Hyperplanes.

Proof. The flat F is a hyperplane of M if and only if there is no flat, F', such that $F \subset F' \subset E(M)$. This clearly leads to a polynomial-time algorithm. □

Lemma 3.11. Flats $\leq$ Cyclic Flats.

Proof. A flat F fails to be a cyclic flat if and only if there is an element $e \in F$ such that $F - e$ is also a flat. This clearly leads to a polynomial-time algorithm. □

Lemma 3.12. Circuits $\leq$ NonSpanning Circuits.

Proof. Given (M, Circuits) a Turing machine can check in polynomial time whether a subset $A \subseteq E(M)$ is independent. Thus such a machine could find a basis of M by using the greedy algorithm. Once the rank of M is known the rest follows easily. □

The next lemma follows easily using duality and Lemma 3.12.

Lemma 3.13. Hyperplanes $\leq$ Dependent Hyperplanes.

To complete the proof of Theorem 3.2 we must show that if Input1 and Input2 are two types of input and the preceding results do not imply that Input1 $\leq$ Input2, then Input1 $\not\leq$ Input2. It is clear that if Input1 $\leq$ Input2, then there must exist a polynomial p such that

$$|(M, \text{Input2})| \leq p(|(M, \text{Input1})|)$$

for any matroid M. Thus if there exists a family of matroids $M_1, M_2, M_3, \ldots$ such that $\{|(M_i, \text{Input2})|\}_{i\geq 1}$ is not polynomially bounded by

$$\{|(M_i, \text{Input1})|\}_{i\geq 1},$$

then Input1 $\not\leq$ Input2.

The following proposition follows immediately from transitivity.

Proposition 3.14. *Suppose that* Input1 $\not\leq$ Input2*. If* Input1 $\leq$ Input3*, and* Input4 $\leq$ Input2*, then* Input3 $\not\leq$ Input4*.*

Using Proposition 3.14, it is a relatively simple matter to check that the proof of Theorem 3.2 will be completed by verifying the following cases:

1 Spanning Sets $\not\leq$ Flats.
2 Independent Sets $\not\leq$ Spanning Sets.
3 Flats $\not\leq$ NonSpanning Circuits.

4 Circuits $\not\leq$ Dependent Hyperplanes.
5 Hyperplanes $\not\leq$ Cyclic Flats.
6 Circuits $\not\leq$ Cyclic Flats.
7 NonSpanning Circuits $\not\leq$ Circuits.
8 Dependent Hyperplanes $\not\leq$ Hyperplanes.
9 Cyclic Flats $\not\leq$ Dependent Hyperplanes.

Lemma 3.15 (Case 1)**.** Spanning Sets $\not\leq$ Flats.

Proof. For $n \geq 1$ let M_n be isomorphic to $U_{n-1,n}$, the n-element uniform matroid of rank $n-1$. The number of spanning sets of M_n is $n+1$, whereas the number of flats is $2^n - n$. Thus $|(M_n, \text{Spanning Sets})| = \Theta(n^2)$, while $|(M_n, \text{Flats})| = \Theta(n2^n)$. □

Lemma 3.16 (Case 2)**.** Independent Sets $\not\leq$ Spanning Sets.

Proof. For $n \geq 1$ let M_n be isomorphic to $U_{1,n}$. The number of independent sets in M_n is $n+1$, while the number of spanning sets is $2^n - 1$. □

We denote the truncation of the matroid M by $T(M)$. If m is a positive integer, then we define $mU_{r,n}$ to be the matroid obtained by replacing each element in $U_{r,n}$ with a parallel class of size m.

Lemma 3.17 (Case 3)**.** Flats $\not\leq$ NonSpanning Circuits.

Proof. For $n \geq 3$ define M_n to be $T(nU_{n-1,n} \oplus U_{2,2})$. Note that M_n contains $n^2 + 2$ elements. There are $n+2$ parallel classes in M_n. It follows that the number of flats of M_n is at most 2^{n+2}. However, the number of nonspanning circuits of M_n is exactly $n^n + n\binom{n}{2}$. Thus $|(M_n, \text{Flats})| = O(n^2 2^{n+2})$. On the other hand $|(M_n, \text{NonSpanning Circuits})| = \Theta(n^{n+2})$, so the result follows. □

Lemma 3.18 (Case 4)**.** Circuits $\not\leq$ Dependent Hyperplanes.

Proof. It is not difficult to see that a polynomial-time algorithm that constructs $(M, \text{Dependent Hyperplanes})$ from $(M, \text{Circuits})$ for any matroid M can be used to show that Hyperplanes $\leq$ NonSpanning Circuits. This contradicts Proposition 3.14, as Flats $\leq$ Hyperplanes by Lemma 3.10, and Flats $\not\leq$ NonSpanning Circuits by Lemma 3.17. □

Lemma 3.19 (Case 5)**.** Hyperplanes $\not\leq$ Cyclic Flats.

Proof. For $n \geq 3$, let M_n be isomorphic to $2U_{n-1,n}$. The hyperplanes of M_n are exactly the sets of $n-2$ parallel classes, while any nonempty set of parallel classes is a cyclic flat as long as it does not contain exactly $n-1$ such classes. Thus the number of hyperplanes is $(n^2 - n)/2$, while the number of cyclic flats is $2^n - n - 1$. □

Lemma 3.20 (Case 6)**.** Circuits $\not\leq$ Cyclic Flats.

Proof. Given (M, Hyperplanes) we can certainly find (M^*, Circuits) in polynomial time. Also, given (M^*, Cyclic Flats) we can find the cyclic flats of M in polynomial time, since the cyclic flats of M^* are the complements of the cyclic flats of M. Moreover, it is easy to see that given (M^*, Circuits) we can find the rank of any subset in M in polynomial time.

Suppose that Circuits $\leq$ Cyclic flats. The above discussion implies that Hyperplanes $\leq$ Cyclic Flats, in contradiction to Lemma 3.19. □

Lemma 3.21 (Case 7)**.** NonSpanning Circuits $\not\leq$ Circuits.

Proof. If M_n is isomorphic to $U_{n,2n}$, then M_n contains no nonspanning circuits, so by definition $|(M_n, \text{NonSpanning Circuits})| = O(n)$. On the other hand, the number of circuits is $\binom{2n}{n+1}$, which is exponential in n. □

The next lemma follows using duality and Lemma 3.21.

Lemma 3.22 (Case 8)**.** Dependent Hyperplanes $\not\leq$ Hyperplanes.

Lemma 3.23 (Case 9)**.** Cyclic Flats $\not\leq$ Dependent Hyperplanes.

Proof. For $n \geq 2$ let M_n be the matroid obtained by adding a single parallel element to a member of the ground set of $U_{n,2n}$. The only cyclic flats of M_n are the empty set, the nontrivial parallel class, and the entire ground set. Thus $|(M_n, \text{Cyclic Flats})| = \Theta(n)$. However, any hyperplane that contains the nontrivial parallel class is dependent, so the number of such hyperplanes is $\binom{2n-1}{n-2}$. □

With this lemma we have completed the proof of Theorem 3.2.

4. Matroid intersection

It is easy to see that given (M, Input), it is possible to determine in polynomial time whether or not a subset of $E(M)$ is independent in M. (Henceforth we assume Input to be one of the types of input discussed in section 2.) Hausmann and Korte [6] and Robinson and Welsh [12] show that the standard matroid oracles can be efficiently simulated by an independence oracle. It follows from these observations that if a computational problem can be solved by an oracle Turing machine in time that is bounded by $p(n)$ for any n-element matroid, where p is a fixed polynomial, then the same problem can be solved in polynomial time by a Turing machine which receives (M, Input) as its input.

The converse is not true. Consider the problem of deciding whether a matroid is uniform. Robinson and Welsh [12] note that a Turing machine equipped with an oracle cannot solve this problem in time bounded by a polynomial function of the size of the ground set. In contrast, deciding whether M is uniform given (M, Input) is trivial.

However, there do exist matroid-theoretical problems which are probably not solvable in polynomial time, even when the input consists of a list of some family of subsets. One of these is 3-MATROID INTERSECTION.

3-MATROID INTERSECTION.
Instance: An integer k and (M_i, INPUT) for $1 \leq i \leq 3$, where M_1, M_2, and M_3 are matroids with a common ground set E.
Question: Does there exist a set $A \subseteq E$ such that $|A| = k$ and A is independent in M_1, M_2, and M_3?

The fact that this problem is NP-complete was first observed by Lawler [9]. He does not specify a form of matroid input, but he remarks that the problem is NP-complete for partition matroids (direct sums of rank-one uniform matroids). It is clear from his comments that a partition matroid is to be described via the partition of its ground set into connected components. We sketch a modified version of his proof here.

Theorem 4.1. *If* CIRCUITS $\leq$ INPUT *or if* HYPERPLANES $\leq$ INPUT*, then* 3-MATROID INTERSECTION *is* NP*-complete. However, if* INPUT $\leq$ BASES*, then* 3-MATROID INTERSECTION *is in* P*.*

Proof. Obviously the problem is in NP. It suffices to prove NP-completeness only in the case that either INPUT = CIRCUITS or INPUT = HYPERPLANES. We provide a reduction from the following NP-complete problem.

3-DIMENSIONAL MATCHING.
Instance: A set of triples, $M \subseteq X_1 \times X_2 \times X_3$, where X_1, X_2, and X_3 are pairwise disjoint sets having the same cardinality.
Question: Does M contain a *matching*? (A subset $M' \subseteq M$, such that every element in $X_1 \cup X_2 \cup X_3$ is contained in exactly one triple in M'.)

Let $M \subseteq X_1 \times X_2 \times X_3$ be an instance of 3-DIMENSIONAL MATCHING, and suppose that $X_i = \{x_1^i, \ldots, x_s^i\}$ for $1 \leq i \leq 3$. Suppose that M contains t triples, $p_1, \ldots, p_t$. We construct three partition matroids, M_1, M_2, and M_3, on the ground set $E = \{e_1, \ldots, e_t\}$. Each matroid M_i contains s connected components, corresponding to the elements of X_i. The connected component corresponding to x_j^i is equal to $\{e_k \mid p_k \text{ contains } x_j^i,\ 1 \leq k \leq t\}$. It is clear that M contains a matching if and only if M_1, M_2, and M_3 contain a common independent set of size s.

It remains to show that (M_i, CIRCUITS) and (M_i, HYPERPLANES) can be constructed in polynomial time. Since the number of circuits or hyperplanes in a partition matroid is at most quadratic in the size of the ground set, this is easily done.

If INPUT = BASES, then we can find a common independent set of maximum size by considering the intersection of every triple of bases from the three matroids. This can clearly be done in polynomial time, so 3-MATROID INTERSECTION is in P if INPUT $\leq$ BASES. □

Theorem 4.1 shows that 3-MATROID INTERSECTION is either NP-complete or in P for all but two of the methods of input described in section 2: The status of the problem is open for the case that INPUT = CYCLIC FLATS or INPUT = FLATS.

5. The isomorphism problem

The following computational problem has attracted much attention.

GRAPH ISOMORPHISM.
Instance: Two graphs, G and G'.
Question: Are G and G' isomorphic?

GRAPH ISOMORPHISM is thought to be a good candidate for a problem in NP that is neither NP-complete nor in P (see [2]).

A decision problem that is polynomially equivalent to GRAPH ISOMORPHISM is *isomorphism-complete*. In this section we show that the analogous matroid problem is in general isomorphism-complete.

MATROID ISOMORPHISM.
Instance: (M, Input) and (M', Input).
Question: Are M and M' isomorphic?

Lemma 5.1. MATROID ISOMORPHISM *is polynomially reducible to* GRAPH ISOMORPHISM.

Proof. A proof can be found in [10]; we give an outline here. We must construct a polynomial-time computable transformation that takes descriptions of matroids to graphs in such a way that isomorphism is preserved. There are many ways in which this can be accomplished. The key idea is that a list of characteristic vectors, representing subsets of the ground set, can be seen as the rows of the vertex-adjacency matrix of a bipartite graph.

The rest of the demonstration involves refining the transformation so that, given the unlabeled bipartite graph, it is possible to reconstruct the matroid description, up to relabeling. This guarantees that the transformation preserves isomorphism. Thus we must somehow distinguish the vertices that correspond to subsets of the ground set from the vertices that correspond to elements of the ground set. In the case that Input relies upon ranks being assigned to sets, as is the case when Input = Cyclic Flats, we must find a method of encoding binary representations of integers in the form of graphs. Constructing a transformation that satisfies these criteria, and confirming that it is polynomial-time computable, is an easy exercise. □

Next we develop a polynomial transformation from graphs to matroid descriptions. Suppose that G is a simple graph with n vertices, $\{v_1, \ldots, v_n\}$, and m edges, $\{e_1, \ldots, e_m\}$. Assume that $n \geq 3$. Let $X = \{x_1, \ldots, x_n\}$, $X' = \{x'_1, \ldots, x'_n\}$, and $Y = \{y_1, \ldots, y_m\}$ be disjoint sets. The matroid $\Phi(G)$ has rank 3, and the ground set of $\Phi(G)$ is $X \cup X' \cup Y$. The nonspanning circuits of $\Phi(G)$ are sets of the form $\{x_i, x'_i\}$, where $i \in \{1, \ldots, n\}$, and the sets

$$\{\{z_i, z_j, y_k\} \mid z_i \in \{x_i, x'_i\}, z_j \in \{x_j, x'_j\}, \text{ and } e_k \text{ joins } v_i \text{ to } v_j\}.$$

Thus $\Phi(G)$ can be formed by placing n parallel pairs, corresponding to the n vertices of G, in the plane in general position, and placing elements between

parallel pairs that correspond to adjacent vertices, in such a way that no additional dependencies are formed.

Let G be a graph. The *cyclomatic number of* G is $|V(G)| - |E(F)|$, where F is a spanning forest of G. The *bicircular matroid of a graph,* G, denoted by $B(G)$, has the edge set of G as its ground set. The circuits of $B(G)$ are exactly the minimal connected edge sets of G with cyclomatic number two, known as *bicycles*. Thus a set of edges is independent in $B(G)$ if and only if the subgraph of G it induces contains at most one cycle in every connected component.

Suppose that G is simple and has at least three vertices. Let $G^{\circ\circ}$ be the graph which is obtained by adding two loops at each vertex of G. Then $\Phi(G)$ is isomorphic to the matroid obtained by repeatedly truncating $B(G^{\circ\circ})$ so that its rank is reduced to three.

Suppose that G is a simple graph with $n \geq 3$ vertices and m edges. Then $|\Phi(G)| = 2n + m$. Since $m \leq n^2$, the number of independent sets in $\Phi(G)$ is $O(n^6)$. Thus ($\Phi(G)$, Input) can be constructed in polynomial time from a description of G when Independent Sets $\leq$ Input.

It is easy to demonstrate that if G and G' are simple graphs on at least three vertices, then G and G' are isomorphic if and only if $\Phi(G)$ and $\Phi(G')$ are isomorphic. Since GRAPH ISOMORPHISM is generally defined in terms of simple graphs, we have proved the following result.

Lemma 5.2. *If* Independent Sets $\leq$ Input*, then* GRAPH ISOMORPHISM *is polynomially reducible to* MATROID ISOMORPHISM.

Using duality we see that ($\Phi(G)^*$, Spanning Sets) can also be constructed in polynomial time from a description of G, where G is any simple graph with at least three vertices. Since G and G' are isomorphic graphs if and only if $(\Phi(G))^*$ and $(\Phi(G'))^*$ are isomorphic matroids, we have proved the following.

Theorem 5.3. *If* Spanning Sets $\leq$ Input*, or* Independent Sets $\leq$ Input*, then* MATROID ISOMORPHISM *is isomorphism-complete.*

6. Detecting minors

It is a well-known observation of Seymour's that an oracle Turing machine cannot decide whether a matroid has a $U_{2,4}$-minor in time that is polynomially bounded by the size of the ground set [13]. We will show that, in general, deciding whether a matroid contains a minor isomorphic to some fixed matroid can be done in polynomial time, given a list of the independent sets or some similar input. It is routine to verify the following result.

Proposition 6.1. *Suppose that M is a matroid and that X and Y are disjoint subsets of $E(M)$. It is possible to construct* ($M/X\backslash Y$, Input) *in polynomial time given* (M, Input).

Proposition 6.2. *Let N be a matroid and let t be the number of circuits of N, where $t > 0$. If M is a matroid on the ground set E, and M has a minor*

isomorphic to N on the set $A \subseteq E$, then there exist circuits, $C_1, \ldots, C_t$, of M such that, if $X = (C_1 \cup \cdots \cup C_t) - A$ and $Y = E - (A \cup X)$, then $M/X\backslash Y \cong N$.

Proof. Suppose that $M/X'\backslash Y'$ is isomorphic to N, where (X', Y') is a partition of $E - A$. We may assume that X' is independent. There are exactly t circuits, $C'_1, \ldots, C'_t$, in $M/X'\backslash Y'$. For each circuit, C'_i, there exists a circuit C_i of M such that $C_i \subseteq C'_i \cup X'$ and $C'_i = C_i - X'$. Let X be the set $(C_1 \cup \cdots \cup C_t) - A$ and let Y be $E - (A \cup X)$. It remains to show that $M/X\backslash Y \cong N$. Note that $X \subseteq X'$. If $X = X'$, then we are done, so suppose that x is an element in $X' - X$. Let $M' = M/(X' - x)\backslash(Y' \cup x)$. To complete the proof it will suffice to show that $M' = M/X'\backslash Y'$. This is elementary. □

Let N be some fixed matroid. It is a consequence of Propositions 6.1 and 6.2 that the following problem is, in general, in P.

DETECTING AN N-MINOR.
Instance: (M, INPUT).
Question: Does M have a minor isomorphic to N?

Theorem 6.3. *If* INPUT $\leq$ CIRCUITS *or* INPUT $\leq$ HYPERPLANES, *then* DETECTING AN N-MINOR *is in* P *for any fixed matroid N.*

Proof. By duality it will suffice to prove the theorem only when INPUT is equal to CIRCUITS. We may assume that we have the list of circuits of N, for we can construct it in constant time.

Suppose that the ground set of M is E, where $|E| = n$, and that M contains c circuits, so that $|(M, \text{CIRCUITS})| = \Theta(nc)$. Suppose also that $|E(N)| = s$, and that N has exactly t circuits. We may assume that $t > 0$, for otherwise the problem is trivially in P.

An algorithm to check whether M has an N-minor could simply work its way through every subset A of E such that $|A| = s$, and every set $\{C_1, \ldots, C_t\}$ of t circuits of M. By Proposition 6.1 it is possible to construct $(M/X\backslash Y, \text{CIRCUITS})$, where $X = (C_1 \cup \cdots \cup C_t) - A$ and $Y = E - (A \cup X)$, and then check whether $M/X\backslash Y \cong N$ in polynomial time. Proposition 6.2 guarantees that if M does have a minor isomorphic to N, then this procedure will find such a minor. Checking whether an isomorphism exists between N and $M/X\backslash Y$ will take some constant time, so the running time of the algorithm is determined by the number of subsets, $A \subseteq E$, and the number of families of circuits we must check. The first quantity is $\binom{n}{s}$, and the second is $\binom{c}{t}$, so the running time of the algorithm is $O(n^s c^t)$. Since s and t are fixed constants, the algorithm runs in polynomial time. □

In contrast to Theorem 6.3, Hliněný shows that the problem of deciding whether M has a minor isomorphic N for a fixed matroid N is NP-complete when the input consists of a representation of M over the rational numbers [7]. On the other hand, Geelen, Gerards, and Whittle conjecture that

the problem is in P when the input consists of a representation of M over a finite field [3].

The exponent in the running time of the algorithm described in Theorem 6.3 depends upon N. It is natural to ask whether there is a *fixed-parameter tractable* (FPT) algorithm for DETECTING AN N-MINOR, that is, an algorithm which runs in time $f(N)|(M, \textsc{Input})|^k$, where f is a function which depends only on N and k is a fixed constant. Certainly such an algorithm exists when $\textsc{Input} = \textsc{Rank}$, for we can simply consider minors of the form $M/X\backslash Y$ for all disjoint sets X and Y, and check by brute force whether any such minor is isomorphic to N. This leads to an FPT algorithm. The existence of such an algorithm when M is described via an input that lies above Rank is an open problem.

We have shown that detecting a fixed minor can be solved in polynomial time, in general. However, the problem of detecting a minor which forms part of the input is, in general, NP-complete.

MINOR ISOMORPHISM.
Instance: $(M, \textsc{Input})$ and $(N, \textsc{Input})$.
Question: Does M have a minor isomorphic to N?

Theorem 6.4. *If* Independent Sets $\leq$ Input *or* Spanning Sets $\leq$ Input*, then* MINOR ISOMORPHISM *is* NP*-complete.*

Proof. It is easy to see that MINOR ISOMORPHISM is in NP. The following problem is well known to be NP-complete.

SUBGRAPH ISOMORPHISM.
Instance: Two graphs, G and H.
Question: Does G have a subgraph isomorphic to H?

Suppose that the graphs G and H correspond to an instance of SUBGRAPH ISOMORPHISM. We may assume that G and H are simple and that both have at least three vertices. Since we can construct either $(\Phi(G), \textsc{Input})$ or $(\Phi(G)^*, \textsc{Input})$ in polynomial time from the description of G (and the same statement applies for H), the proof will be complete if we can demonstrate that G contains a subgraph isomorphic to H if and only if $\Phi(G)$ contains a minor isomorphic to $\Phi(H)$. This is easily done. $\square$

Suppose that $\mathcal{M}$ is a family of matroids. A natural computational problem is to ask whether a matroid M has a minor isomorphic to a member of $\mathcal{M}$ with specified size. The proof of Theorem 6.4 can be used to show that if $\mathcal{M} = \{\Phi(K_n) \mid n \geq 1\}$, then this problem is, in general, NP-complete. We will conclude by showing that the problem is, in general, also NP-complete when $\mathcal{M} = \{U_{r,n} \mid n \geq r\}$, where r is a fixed constant.

$U_{r,n}$-MINOR.
Instance: $(M, \textsc{Input})$ and an integer n.
Question: Does M have a minor isomorphic to $U_{r,n}$?

Theorem 6.5. *If $r > 2$ is a fixed integer and* Independent Sets $\leq$ Input*, then $U_{r,n}$-*MINOR *is* NP*-complete.*

Proof. It is easy to see that $U_{r,n}$-MINOR is in NP. We construct a reduction from the following NP-complete problem.

INDEPENDENT SET.
Instance: An integer k and a graph G.
Question: Does G contain an independent set of k vertices?

Let $r > 2$ be a fixed integer. Suppose that the integer k and the simple graph G are an instance of INDEPENDENT SET. Let m be the number of edges in G. Let t be $\lceil (r-1)/2 \rceil$, and let tG° be the graph obtained by adding a loop to each vertex of G and then replacing each nonloop edge with a path of length t. The matroid $\Phi_r(G)$ is the bicircular matroid of tG°, repeatedly truncated so that its rank is equal to r.

Checking whether a set of edges of tG° is independent in $B(tG^\circ)$ can certainly be done in polynomial time. Since r is a fixed integer, and no independent set of $\Phi_r(G)$ exceeds r in size, it follows that $(\Phi_r(G), \textsc{Input})$ can be constructed in polynomial time as long as Independent Sets $\leq$ Input.

We complete the proof by showing that G has an independent set of k vertices if and only if $\Phi_r(G)$ has a rank-r uniform minor of size $k + mt$.

Every nonloop cycle of tG° contains at least $3t$ edges, so if a bicycle of tG° contains at most one loop, then it has size at least $3t+1$. This quantity is greater than r, as $r \geq 3$. If a bicycle contains two loops, then either it contains exactly $t+2$ elements, or its size is at least $2t+2$. It is straightforward to confirm that $t+2 \leq r$ and that $2t+2 > r$. This shows that the nonspanning circuits of $\Phi_r(G)$ are exactly the sets containing the t edges in a path joining two vertices of G and the two loops incident with those vertices.

Suppose that G contains an independent set of k vertices. These vertices correspond to k loops of tG°. The set containing these loops and all the nonloop edges of tG° cannot contain a nonspanning circuit of $\Phi_r(G)$, by the above discussion. Thus restricting $\Phi_r(G)$ to this set of $k+mt$ elements gives a uniform minor. We may assume that $m \geq 3$, so $k + mt$ is certainly no less than r. Therefore $\Phi_r(G)$ contains a rank-r uniform minor with $k + mt$ elements.

For the converse suppose that $\Phi_r(G)$ has a rank-r uniform minor on the set A, where $|A| = k + mt$, and assume that G has no independent set of k vertices. The number of nonloop edges in A is at most mt. Suppose that A has been chosen so that it contains as many nonloop edges as possible. Now A contains at least k loops, so there must be a pair of loops, l and l', in A that correspond to adjacent vertices in G. Therefore one of the t edges that join l to l' in tG° does not belong to A. Let us call this edge e. Then $(A - l) \cup e$ contains no nonspanning circuits of $\Phi_r(G)$, and our assumption on A is contradicted. Therefore G has an independent set of k vertices. $\square$

If $r \leq 2$, then $U_{r,n}$-MINOR is trivially in P. Using duality, we can show that the problem of deciding whether M has a minor isomorphic to $U_{n-r,n}$ is NP-complete for fixed values of $r > 2$ as long as Spanning Sets $\leq$ Input.

7. Open problems

In this summary section we collect some open problems. 3-MATROID INTERSECTION is known to be either in P or NP-complete for all but two of the 10 types of input mentioned in section 2. The status of the problem is open when Input is either Cyclic Flats or Flats. MATROID ISOMORPHISM is NP-complete for all forms of input, except possibly Rank. Deciding if MATROID ISOMORPHISM can be solved in polynomial time when the matroids are described via the rank of each of their subsets is an open problem. More generally, it would be interesting to know if there is any "natural" computational problem which is NP-complete for the Rank input.

DETECTING AN N-MINOR is known to be in P for all forms of input except NonSpanning Circuits, Cyclic Flats, and Dependent Hyperplanes. The status of the problem for these types of input is unknown. The existence or otherwise of an FPT algorithm for MINOR ISOMORPHISM is known only when Input = Rank. Otherwise the problem is open.

Acknowledgments

I thank my supervisor, Professor Dominic Welsh, and Professors Colin McDiarmid and James Oxley for very useful advice and discussions. I also thank the referees for their constructive comments. In particular, the question of the existence of FPT algorithms for DETECTING AN N-MINOR was raised by one of the referees.

References

[1] J. Edmonds, *Submodular functions, matroids, and certain polyhedra*, in Combinatorial Structures and Their Applications (Proc. Calgary International Conference, Calgary, Alberta, 1969), Gordon and Breach, New York, 1970, pp. 69–87.

[2] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of* NP-*Completeness*, W. H. Freeman and Co., San Francisco, CA, 1979.

[3] J. Geelen, B. Gerards, and G. Whittle, *Towards a structure theory for matrices and matroids*, in International Congress of Mathematicians, Vol. III, Eur. Math. Soc., Zürich, 2006, pp. 827–842.

[4] D. Hausmann and B. Korte, *Oracle algorithms for fixed-point problems—an axiomatic approach*, in Optimization and Operations Research (Proc. Workshop, University of Bonn, Bonn, Germany, 1977), Lecture Notes in Econom. and Math. Systems 157, Springer, Berlin, 1978, pp. 137–156.

[5] D. Hausmann and B. Korte, *Worst-case behaviour of polynomial bounded algorithms for independence systems*, Z. Angew. Math. Mech., 58 (1978), pp. T477–T479.

[6] D. Hausmann and B. Korte, *Algorithmic versus axiomatic definitions of matroids*, Math. Programming Stud., (1981), pp. 98–111.

[7] P. Hliněný, *On matroid representability and minor problems*, in Mathematical Foundations of Computer Science 2006, Lecture Notes in Comput. Sci. 4162, Springer, Berlin, 2006, pp. 505–516.

[8] D. E. Knuth, *The asymptotic number of geometries*, J. Combin. Theory Ser. A, 16 (1974), pp. 398–400.

[9] E. L. Lawler, *Polynomial-bounded and (apparently) nonpolynomial-bounded matroid computations*, in Combinatorial Algorithms (Courant Computer Science Symposiums 9, New York University, New York, 1972), Algorithmics Press, New York, 1973, pp. 49–57.

[10] D. Mayhew, *Matroids and Complexity*, Ph.D. thesis, University of Oxford, Oxford, UK, 2005.

[11] J. G. Oxley, *Matroid Theory*, The Clarendon Press, Oxford University Press, New York, 1992.

[12] G. C. Robinson and D. J. A. Welsh, *The computational complexity of matroid properties*, Math. Proc. Cambridge Philos. Soc., 87 (1980), pp. 29–45.

[13] P. D. Seymour, *Recognizing graphic matroids*, Combinatorica, 1 (1981), pp. 75–78.

Mathematical Institute, University of Oxford, St. Giles, OX1 3LB Oxford, UK, and School of Mathematics, Victoria University of Wellington, P.O. Box 600, Wellington 6140, New Zealand

E-mail address: dillon.mayhew@mcs.vuw.ac.nz