Theory Comput Syst (2011) 49:246–272
DOI 10.1007/s00224-010-9260-0

On the Complexity of Matroid Isomorphism Problem

B.V. Raghavendra Rao · M.N. Jayalal Sarma

Published online: 13 April 2010

Abstract We study the complexity of testing if two given matroids are isomorphic. The problem is easily seen to be in Σ_2^p. In the case of linear matroids, which are represented over polynomially growing fields, we note that the problem is unlikely to be Σ_2^p-complete and is coNP-hard. We show that when the rank of the matroid is bounded by a constant, linear matroid isomorphism, and matroid isomorphism are both polynomial time many-one equivalent to graph isomorphism.

We give a polynomial time Turing reduction from graphic matroid isomorphism problem to the graph isomorphism problem. Using this, we are able to show that graphic matroid isomorphism testing for planar graphs can be done in deterministic polynomial time. We then give a polynomial time many-one reduction from bounded rank matroid isomorphism problem to graphic matroid isomorphism, thus showing that all the above problems are polynomial time equivalent.

Further, for linear and graphic matroids, we prove that the automorphism problems are polynomial time equivalent to the corresponding isomorphism problems. In addition, we give a polynomial time membership test algorithm for the automorphism group of a graphic matroid.

Keywords Computational complexity

Part of this work was done while the authors were graduate students at the Institute of Mathematical Sciences, Chennai, India. The second author's work was also supported in part by the National Natural Science Foundation of China Grant 60553001, and the National Basic Research Program of China Grant 2007CB807900, 2007CB807901.

B.V. Raghavendra Rao (✉)
Department of Computer Science, Saarland University, Saarbrücken, Germany
e-mail: bvrr@cs.uni-sb.de

M.N. Jayalal Sarma
Institute for Theoretical Computer Science, Tsinghua University, Beijing, China
e-mail: jayalal@tsinghua.edu.cn

1 Introduction

Isomorphism problems over various mathematical structures have been a source of intriguing problems in complexity theory (see [2]). The most important problem of this domain is the well-known graph isomorphism problem. Though the complexity characterization of the general version of this problem is still unknown, there have been various interesting special cases of the problem which are known to have polynomial time algorithms [4, 17]. In this paper we talk about isomorphism problems associated with matroids.

A matroid M is a combinatorial object defined over a finite set S (of size m) called the *ground set*, equipped with a non-empty family $\mathcal{I}$ of subsets of S (containing the empty subset) which is closed under taking of subsets and satisfies the *exchange axiom*: for any $I_1, I_2 \in \mathcal{I}$ such that $|I_1| > |I_2|$, $\exists x \in I_1 \setminus I_2$, $I_2 \cup \{x\} \in \mathcal{I}$. The sets in $\mathcal{I}$ are called *independent sets*. A set $B \subseteq S$ is *dependent* if and only if $B \notin \mathcal{I}$. The rank of the matroid is the size of the maximal independent set. This provides useful abstractions of many concepts in combinatorics and linear algebra [28]. The theory of matroids is a well studied area of combinatorics [22]. We study the problem of testing isomorphism between two given matroids.

Two matroids M_1 and M_2 are said to be isomorphic if there is a bijection between the elements of the ground set which maps independent sets to independent sets (or equivalently circuits to circuits, or bases to bases, see Sect. 2). Quite naturally, the representation of the input matroids is important in deciding the complexity of the algorithmic problem.

There are several equivalent representations of a matroid. For example, enumerating the maximal independent sets (called bases) or the minimal dependent sets (called circuits) also defines the matroid. These representations, although can be exponential in the size of the ground set, indeed exist for every matroid, by definition. With this enumerative representation, Mayhew [19] studied the matroid isomorphism problem, and showed that the problem is equivalent to the graph isomorphism problem. However, a natural question is whether the problem is difficult when the representation of the matroid is more implicit. In a black-box setting, one can also consider the input representation in the form of an oracle or a black-box, where the oracle answers whether a given set is independent or not.

More implicit (and efficient) representation of matroids have been studied. One natural way is to identify the given matroid with matroids defined over combinatorial or algebraic objects which have implicit descriptions. A general framework in this direction is the representation of a matroid over a field. A matroid $M = (S, \mathcal{I})$ of rank r is said to be *representable* over a field $\mathbb{F}$ if there is a map, $\phi : S \to \mathbb{F}^r$ such that, $\forall A \subseteq S$, $A \in \mathcal{I} \iff \phi(A)$ is linearly independent over $\mathbb{F}^r$ as a vector space. However, there are matroids which do not admit linear representations over any field. (For example, the Vamós Matroid, see Proposition 6.1.10, [22].) In contrast, there are matroids (called regular matroids) which admit linear representations over all fields.

Another natural representation for a matroid is over graphs. For any undirected graph X, we can associate a matroid $M(X)$ as follows: the set of edges of X is the ground set, and the acyclic subgraphs of the given graph form the independent sets. A matroid M is called a *graphic matroid* (also called polygon matroid or cyclic matroid) if it is isomorphic to $M(X)$ for some graph X. It is known that graphic matroids are linear. Indeed, the vertex-edge incidence matrix of the graph will give a representation over $\mathbb{F}_2$. There are linear matroids which are not graphic. (See [22] for more details.)

The above definitions themselves highlight the importance of testing isomorphism between two given matroids. We study the isomorphism problem for the case of linear matroids (Linear Matroid Isomorphism problem (LMI) and graphic matroids (Graphic Matroid Isomorphism problem (GMI)).

From a complexity perspective, the general case black-box of the problem is in Σ_2^p. However, it is not even clear a priori if the problem is in NP even in the restricted cases above where there are implicit representations. But we note that for the case of graphic matroids the problem admits an NP algorithm. Hence an intriguing question is about the comparison of this problem to the well studied graph isomorphism problem.

At an intuitive level, the graph isomorphism problem asks for a map between the vertices that preserves the adjacency relations, whereas the graphic matroid isomorphism problem asks for maps between the edges such that the set of cycles (or spanning trees) in the graph are preserved. As an example, in the case of trees, any permutation of the edges gives an isomorphism of the matroids, whereas testing for the isomorphism of trees is known to be L-complete. This indicates that the reduction between the problems cannot be obtained by a local replacement of edges with gadgets, and has to consider the global structure.

An important result in this direction, due to Whitney (see [26]), says that in the case of 3-connected graphs, the graphs are isomorphic if and only if the corresponding matroids are isomorphic (see Sect. 5). Thus the problem of testing isomorphism of graphs and the corresponding graphic matroids are equivalent for the case of 3-connected graphs. Despite this similarity between the problems, to the best of our knowledge, there has not been a systematic study of GMI and its relationships to graph isomorphism problem (GI). This immediately gives a motivation to study the isomorphism problem for 3-connected graphs. In particular, from the recent results on graph isomorphism problem for these classes of graphs [6, 25], it follows that graphic matroid isomorphism problem for 3-connected planar graphs is L-complete.

In this context we study the general, linear and graphic matroid isomorphism problems. Our main contributions in the paper are as follows:

- Black-box matroid isomorphism problem is easily seen to be in Σ_2^p. In the case of linear matroids where the field is also a part of the input we observe that the problem is coNP-hard (Proposition 3.4), and is unlikely to be Σ_2^p-complete (Proposition 3.2). We also observe that when the rank of the matroid is bounded, linear matroid isomorphism, and matroid isomorphism are both equivalent to GI (Theorem 3.5).[1]
- We develop tools to handle the coloring of ground set elements in the context of isomorphism problem. We show that the colored version of the linear matroid iso-

[1] We note that, although not explicitly stated, the equivalence of bounded rank matroid isomorphism and graph isomorphism also follows from the results of Mayhew [19]. However, it is not immediately clear if the GI-hard instances are linearly representable. Our proofs are different and extends this to linear matroids.

Table 1 Complexity of MI under various input representations

Representation of M_1, M_2	Complexity bounds for MI
List of Ind. sets	GI-complete [19]
Linear	GI-hard, coNP-hard [10, 21]
Linear (bounded rank)	GI complete
Graphic	Turing equivalent to GI
Planar	L-complete

morphism and graphic matroid isomorphism problem are as hard as their general versions (Lemmas 4.2, 4.1). As an immediate application of this, we show that the automorphism problems for graphic matroids and linear matroids are polynomial time Turing equivalent to the corresponding isomorphism problems. In this context, we also give a polynomial time membership test algorithm for the automorphism group of a graphic matroid (Theorem 7.5).

- We give a polynomial time Turing reduction from graphic matroid isomorphism problem to the graph isomorphism problem by developing an edge coloring scheme which algorithmically uses a decomposition given by [12] (and [5]) and reduce the graphic matroid isomorphism problem to the graph isomorphism problem (Theorem 5.3). Our reduction also implies efficient algorithms for isomorphism testing of graphic matroids in special cases such as planar graphs, bounded degree graphs, bounded genus graphs etc. (Corollary 6.1). In addition, we observe that, using recent developments in the planar graph isomorphism testing problem, we can give a log-space algorithm for planar graphic matroid isomorphism.
- Finally, we give a reduction from the bounded rank matroid isomorphism problem to graphic matroid isomorphism (Theorem 5.9), thus showing that all the above problems are poly-time Turing equivalent. Since the equivalence is only under a Turing reduction, we also study the closure properties of the graphic matroid isomorphism problem under $\wedge$ and $\vee$ operations.

Table 1 summarizes the complexity of matroid isomorphism problem under various input representations.

2 Notations and Preliminaries

All the complexity classes used here are standard and we refer the reader to any standard text book (for e.g. see [8]). Now we collect some basic definitions on matroids (see also [22]). Formally, a matroid M is a pair $(S, \mathcal{I})$, where S is a finite set called the ground set of size m and $\mathcal{I}$ is a collection of subsets of S such that: (1) the empty set ϕ, is in $\mathcal{I}$. (2) If $I_1 \in I$ and $I_2 \subset I_1$, then $I_2 \in \mathcal{I}$. (3) If $I_1, I_2 \in \mathcal{I}$ with $|I_1| < |I_2|$, then $\exists x \in I_2 \setminus I_1$ such that $I_1 \cup \{x\}$ is in $\mathcal{I}$. The subsets in $\mathcal{I}$ are called *independent sets*. A set $A \subseteq S$ is *dependent* if and only if $A \notin \mathcal{I}$.

The RANK function of a matroid is a map $\mathsf{rank}\colon 2^S \to \mathbb{N}$, is defined for a $T \subseteq S$, as the maximum size of any element of $\mathcal{I}$ that is contained in T. The rank of the matroid is the maximum value of this function. A *basis* is a maximal independent set.

A *circuit* is a minimal dependent set. Spanning sets are subsets of S which contain at least one basis as its subset. Notice that a set $X \subseteq S$ is spanning if and only if $\mathrm{rank}(X) = \mathrm{rank}(S)$. Moreover, X is a basis set if and only if it is a minimal spanning set. For any $F \subseteq S$, $cl(F) = \{x \in S : \mathrm{rank}(F \cup x) = \mathrm{rank}(F)\}$. A set $F \subseteq S$ is a flat if $cl(F) = F$. *Hyperplanes* are flats which are of rank $r - 1$, where $r = \textsc{Rank}(S)$. $X \subseteq S$ is a hyperplane if and only if it is a maximal non-spanning set.

An isomorphism between two matroids M_1 and M_2 is a bijection $\phi : S_1 \to S_2$ such that $\forall C \subseteq S_1 : C \in \mathcal{C}_1 \iff \phi(C) \in \mathcal{C}_2$, where $\mathcal{C}_1$ and $\mathcal{C}_2$ are the family of circuits of the matroids M_1 and M_2 respectively. Now we state the computational problems more precisely.

Problem 1 (Matroid Isomorphism (MI)) *Given two matroids $M_1 = (S_1, \mathcal{I}_1)$ and $M_2 = (S_2, \mathcal{I}_2)$ where I_1 and I_2 are given as black-box oracles, does there exist an isomorphism between M_1 an M_2?*

Given a matrix A over a field $\mathbb{F}$, we can define a matroid $M[A]$ with columns of A as the ground set and linearly independent columns as the independent sets of $M[A]$. A matroid $M = (E, \mathcal{I})$ with rank $= r$ is said to be *representable over* $\mathbb{F}$, if there is a map $\Phi : E \to \mathbb{F}^r$ such that $I \in \mathcal{I} \iff \Phi(I)$ is linearly independent in $\mathbb{F}^r$. *Linear matroids* are matroids representable over fields. Without loss of generality we can assume that the representation is of the form of a matrix where the columns of the matrix correspond to the ground set elements. We assume that the field on which the matroid is represented is also a part of the input, also that the field has at least m elements and at most $poly(m)$ elements, where $m = poly(n)$.

Problem 2 (Linear Matroid Isomorphism (LMI)) *Given two matrices A and B over a given field $\mathbb{F}$ does there exist an isomorphism between the two linear matroids represented by them?*

As mentioned in the introduction, given a graph $X = (V, E)$ ($|V| = n$, $|E| = m$), a classical way to associate a matroid $M(X)$ with X is to treat E as ground set elements, the bases of $M(X)$ are spanning forests of X. Equivalently circuits of $M(X)$ are simple cycles in X. A matroid M is called *graphic* iff $\exists X$ such that $M = M(X)$.

Evidently, adding vertices to a graph G with no incident edges will not alter the matroid of the graph. Without loss of generality we can assume that G does not have self-loops.

Problem 3 (Graphic Matroid Isomorphism (GMI)) *Given two graphs X_1 and X_2 does there exist an isomorphism between $M(X_1)$ and $M(X_2)$?.*

Another associated terminology in the literature is about 2-isomorphism. Two graphs X_1 and X_2 are said to 2-*isomorphic* (denoted by $X_1 \cong_2 X_2$) if their corresponding graphic matroids are isomorphic. Thus the above problem asks to test if two given graphs are 2-isomorphic. Recall that a *separating pair* in a graph X is a pair of vertices whose deletion leaves the graph disconnected.

In a rather surprising result, Whitney [27] came up with a combinatorial characterization of 2-isomorphic graphs. We briefly describe it here. Whitney defined the following operations.

- *Vertex Identification:* Let v and v' be vertices of distinct components of X. We modify X by identifying v and v' as a new vertex $\bar{v}$.
- *Vertex Cleaving:* This is the reverse operation of vertex identification so that a graph can only be cleft at a cut-vertex or at a vertex incident with a loop.
- *Twisting:* Suppose that the graph X is obtained from two disjoint graphs X_1 and X_2 by identifying vertices u_1 of X_1 and u_2 of X_2 as the vertex u of X, and identifying vertices v_1 of X_1 and v_2 of X_2 as the vertex v of X. In a *twisting* of X about $\{u, v\}$, we identify, instead u_1 with v_2 and u_2 with v_1 to get a new graph X'. Note that $\{u, v\}$ is a separating pair in X'.

Theorem 2.1 (Whitney's 2-Isomorphism Theorem) ([27], *see also* [22]) *Let X_1 and X_2 be two graphs having no isolated vertices. Then $M(X_1)$ and $M(X_2)$ are isomorphic if and only if X_1 can be transformed to a graph isomorphic to X_2 by a sequence of operations of vertex identification, cleaving and/or twisting.*

The graphic matroids of planar graphs are called *planar matroids*. We now define the corresponding isomorphism problem for graphic matroids,

Problem 4 (Planar Matroid Isomorphism (PMI)) *Given two planar graphs X_1 and X_2 does there exist an isomorphism between their graphic matroids?*

As a basic complexity bound, it is easy to see that MI $\in \Sigma_2^p$. Indeed, the algorithm will existentially guess a bijection $\sigma : S_1 \to S_2$ and universally verify if for every subset $C \subseteq S_1$, $C \in \mathcal{C}_1 \iff \sigma(C) \in \mathcal{C}_2$ using the independent set oracle.

3 Linear Matroid Isomorphism

In this section we present some observations and results on LINEAR MATROID ISOMORPHISM. Some of these follow easily from the techniques in the literature. We make them explicit in a form that is relevant to the problem that we are considering.

We first observe that using the arguments similar to that of [14] one can show $\overline{\mathrm{LMI}} \in \mathrm{BP} \cdot \Sigma_2^{\mathrm{P}}$. (Notice that an obvious upper bound for this problem is Π_2.) We include some details of this here while we observe some points about the proof.

Proposition 3.1 $\overline{\mathrm{LMI}} \in \mathrm{BP} \cdot \Sigma_2^{\mathrm{P}}$.

Proof Let M_1 and M_2 be the given linear matroids having m columns each. We proceed as in [14], for the case of GI. To give a $\mathrm{BP}.\Sigma_2^{\mathrm{P}}$ algorithm for $\overline{\mathrm{LMI}}$, define the following set:

$$N(M_1, M_2) = \{(N, \phi) : (N \cong M_1) \vee (N \cong M_2) \wedge \phi \in \mathrm{Aut}(N)\}$$

where $\mathrm{Aut}(N)$ contains all the permutations (bijections) which are isomorphisms of matroid N to itself. The key property that is used in [14] has the following easy counterpart in our context.

For any matroid M on a ground set of size m, if $\mathrm{Aut}(M)$ denotes the automorphism group of M, and $\#M$ denotes the number of different matroids isomorphic to M, then $|\mathrm{Aut}(M)| * (\#M) = |S_m|$.

$$M_1 \cong M_2 \implies |N(M_1, M_2)| = m!,$$

$$M_1 \ncong M_2 \implies |N(M_1, M_2)| = 2 \cdot m!.$$

As in [14], we can amplify this gap and then using a good hash family and utilize the gap to distinguish between the two cases. In the final protocol (before amplifying) the verifier chooses a hash function and sends it to the prover, the prover returns a tuple (N, ϕ) along with a proof that this belongs to $N(M_1, M_2)$. (Notice that this will not work over very large fields, especially over infinite fields.) Verifier checks this claim along with the hash value of the tuple. This can be done in Σ_2^p. Hence the entire algorithm gives an upper bound of $\mathrm{BP}.\exists \cdot \Sigma_2^p = \mathrm{BP} \cdot \Sigma_2^p$, and thus the result follows. □

Now, we know that [23], if $\Pi_2^p \subseteq \mathrm{BP} \cdot \Sigma_2^p$ then $\mathrm{PH} = \mathrm{BP}.\Sigma_2^p = \Sigma_3^p$. Thus we get the following:

Theorem 3.2 LMI $\in \Sigma_2^p$. *In addition*, LMI is Σ_2^{P}-hard $\implies$ $\mathrm{PH} = \Sigma_3^{\mathrm{P}}$.

We notice that a special case of this is problem already known to be CONP-hard. A matroid of rank k is said to be *uniform* if all subsets of size at most k are independent. Testing if a given linear matroid of rank k is uniform is known to be CONP-complete [21]. We denote by $U_{k,m}$, the uniform matroid whose ground set is of m elements. Now notice that the above problem is equivalent to checking if the given linear matroid of rank k is isomorphic to $U_{k,m}$. To complete the argument, we use a folklore result that $U_{k,m}$ is representable over any field $\mathbb{F}$ which has at least m non-zero elements. We give some details here since we have not seen an explicit description of this in the literature.

Claim 3.3 *Let* $|\mathbb{F}| > m$, $U_{k,m}$ *has a representation over* $\mathbb{F}$.

Proof Let $\{\alpha_1, \dots, \alpha_m\}$ be distinct elements of $\mathbb{F}$, and $\{s_1, \dots, s_m\}$ be elements of the ground set of $U_{k,m}$. Assign the vector $(1, \alpha_i, \alpha_i^2, \dots, \alpha_i^{k-1}) \in \mathbb{F}^k$ to the element s_i. Any k subset of these vectors forms a Vandermonde matrix, and hence linearly independent. Any larger set is dependent since the vectors are in $\mathbb{F}^k$. □

This gives us the following proposition.

Proposition 3.4 LMI *is* CONP-*hard.*

The above proposition also holds when the representation is over infinite fields. In this case, the proposition also more directly follows from a result of Hlinený [10], where it is shown that the problem of testing if a spike (a special kind of matroids) represented by a matrix over $\mathbb{Q}$ is the free spike is coNP complete. He also derives a linear representation for spikes.

Now we look at bounded rank variant of the problem. We denote by LMI_b (MI_b), the restriction of LMI (MI) for which the input matrices have rank bounded by b. In the following we use the following construction due to Babai [3] to prove $\mathrm{LMI}_b \equiv_m^p$ GI.

Given a graph $X = (V, E)$ and a number $3 \le k \le d$, where d is the minimum vertex degree of X, define a matroid $M = St_k(X)$ of rank k with the ground set E as follows: every subset of E containing $k - 1$ or less number of edges is independent in M and a subset A of E with k edges is independent if and only if there is no single vertex that is part of *all* the edges in A. Babai [3] proved that $\mathrm{Aut}(X) \cong \mathrm{Aut}(St_k(X))$ and also gave a linear representation for $St_k(X)$ (Lemma 2.1 in [3]) for all k in the above range.

Theorem 3.5 *For any constant $b \ge 3$, $\mathrm{LMI}_b \equiv_m^p$ GI.*

Proof GI $\le_m^p$ LMI_b: Let $X_1 = (V_1, E_1)$ and $X_2 = (V_2, E_2)$ be the given GI instance. We can assume that the minimum degree of the graph is at least 3 since otherwise we can attach cliques of size $n + 1$ at every vertex. We note that from Babai's proof we can derive the following stronger conclusion.

Lemma 3.6 $X_1 \cong X_2 \iff \forall k \in [3, d] St_k(X_1) \cong St_k(X_2)$.

Proof Suppose $X_1 \cong X_2$ via a bijection $\pi : V_1 \to V_2$. (The following proof works for any $k \in [3, d]$.) Let $\sigma : E_1 \to E_2$ be the map induced by π. That is $\sigma(\{u, v\}) = \{\pi(u), \pi(v)\}$. Consider an independent set $I \subseteq E_1$ in $St_k(X_1)$. If $|I| \le k - 1$ then $|\sigma(I)| \le k - 1$ and hence $\sigma(I)$ is independent in $St_k(X_2)$. If $|I| = k$, and let $\sigma(I)$ be dependent. This means that the edges in $\sigma(I)$ share a common vertex w in X_2. Since π is an isomorphism which induces σ, $\pi^{-1}(w)$ must be shared by all edges in I. Thus I is independent if and only if $\sigma(I)$ is independent.

Suppose $St_k(X_1) \cong St_k(X_2)$ via a bijection $\sigma : E_1 \to E_2$. By definition, any subset $H \subseteq E_1$ is a hyperplane of $St_k(X_1)$ if and only if $\sigma(H)$ is a hyperplane of $St_k(X_2)$. Now we use the following claim which follows from [3].

Claim 3.7 [3] *For any graph X, any dependent hyperplane in $St_k(X)$ is a maximal set of edges which share a common vertex (forms a star) in X, and these are the only dependent hyperplanes.*

Now we define the graph isomorphism $\pi : V_1 \to V_2$ as follows. For any vertex v, look at the star $E_1(v)$ rooted at v, we know that $\sigma(E_1(v)) = E_2(v')$ for some v'. Now set $\pi(v) = v'$. From the above claim, π is an isomorphism. □

It remains to show that representation for $St_k(X)$ ($X = (V, E)$) can be computed in polynomial time. We choose $k = 3$. (By the above proof, the universal quantifier in the Lemma 3.6 is equivalent to an existential quantification.) Now we show that the representation of $St_k(X)$ given in [3] is computable in polynomial time. The representation of $St_k(X)$ is over a field $\mathbb{F}$ such that $|\mathbb{F}| \geq |V|^{2k-1}$. For $e = \{u, v\} \in E$ assign a vector $b_e = [1, (x_u + x_v), (x_u x_v), y_{e,1}, \ldots, y_{e,k-3}] \in \mathbb{F}^k$, where x_u, x_v and $y_{e,i}$ are distinct unknowns. To represent $St_k(X)$ we need to ensure that the k-subsets of the columns corresponding to a basis form a linearly independent set, and all the remaining k-subsets form a dependent set. Babai [3] showed that by the above careful choice of b_e, it will be sufficient to ensure only the independence condition. He also proved the existence of a choice of values for the variables which achieves this if $|\mathbb{F}| \geq |V|^{2k-1}$.

We make this constructive. As k is a constant, the number of bases in $St_k(X)$ is bounded by $\text{POLY}(m)$. We can greedily choose the value for each variable at every step, such that on assigning this value, the resulting set of constant $(k \times k)$ size matrices are non-singular. Since there exists a solution, this algorithm will always find one. Thus we can compute a representation for $St_k(X)$ in polynomial time.

$\mathrm{LMI}_b \leq_m^p \mathrm{GI}$: Let $A_{k\times m}$ and $B_{k\times m}$ be two matrices of rank b at the input. Now define the following bipartite graph $X_A = (U_A, V_A, E_A)$ (similarly for X_B), where U_A has a vertex for each column of A, and V_A has a vertex for each maximal independent set of A (notice that there are at most $\binom{m}{b} = O(m^b)$ of them) and $\forall i \in U_A, I \in V_A$, $\{i, I\} \in E_A \iff i \in I$. Now we claim that $M(A) \cong M(B) \iff X_A \cong X_B$ where the isomorphism maps V_A to V_B, and which is reducible to GI. It is easy to see that the matroid isomorphism can be recovered from the map between the sets. □

Observe that the reduction $\mathrm{LMI}_b \leq_m^p \mathrm{GI}$ can be done even if the input representation is an independent set oracle. This gives the following corollary.

Corollary 3.8 $\mathrm{LMI}_b \equiv_m^p \mathrm{MI}_b \equiv_m^p \mathrm{GI}$.

4 Isomorphism Problem of Colored Matroids

Vertex or edge coloring is a classical tool used extensively in proving various results in graph isomorphism problem. We develop similar techniques for matroid isomorphism problems too.

An edge-k-coloring of a graph $X = (V, E)$ is a function $f : E \to \{1, \ldots, k\}$. Given two colored graphs $X_1 = (V_1, E_1, f_1)$ and $X_2 = (V_2, E_2, f_2)$, the COLORED-GMI asks for an isomorphism between the corresponding graphic matroids which preserves the colors of the edges. Not surprisingly, we can prove the following.

Lemma 4.1 COLORED-GMI *is* AC^0 *many-one reducible to* GMI.

Proof Let $X_1 = (V_1, E_1, f_1)$ and $X_2 = (V_2, E_2, f_2)$, be the two k-colored graphs at the input, with $n = |V_1| = |V_2|$. For every edge $e = (u, v) \in E_1$ (respectively E_2), add a path $P_e = \{(u, v_{e,1}), (v_{e,1}, v_{e,2}), \ldots, (v_{e,n+f_1(e)}, v)\}$ of length $n + f_1(e)$ (respectively $n + f_2(e)$), where $v_{e,1}, \ldots, v_{e,n+f_1(e)}$ are new vertices. Let X_1' and X_2' be the two new graphs thus obtained. By definition, any 2-isomorphism between X_1' and

X_2' can only map cycles of equal length to themselves. There are no simple cycles of length more than n in the original graphs. Thus, given any 2-isomorphism between X_1' and X_2', we can recover a 2-isomorphism between X_1 and X_2 which preserves the coloring and vice versa. □

Now we generalize the above construction to the case of linear matroid isomorphism. COLORED-LMI denotes the variant of LMI where the inputs are the linear matroids M_1 and M_2 along with color functions $c_i : \{1, \dots, m\} \to \mathbb{N}, i \in \{1, 2\}$. The problem is to test if there is an isomorphism between M_1 and M_2 which preserves the colors of the column indices. We have,

Lemma 4.2 COLORED-LMI *is* AC^0 *many-one reducible to* LMI.

Proof Let M_1 and M_2 be two colored linear matroids represented over a field $\mathbb{F}$. First we illustrate the reduction where only one column index of M_1 (resp. M_2) is colored. Without loss of generality, we assume that there are no two vectors in M_1 (resp.M_2) which are scalar multiples of each other. Otherwise, if V is a subset of vectors such that every pair of vectors in V are scalar multiples of each other, we replace the set of columns in V by a single representative vector with suitable color. This assumption also implies that there is no all-zeroes vectors in M_1 and M_2.

We transform M_1 and M_2 to get two matroids M_1' and M_2'. In the transformation, we add more columns to the matrix (vectors to the ground set) and create dependency relations in such a way that any isomorphism between the matroids must map these new vectors in M_1 to the corresponding ones M_2.

We describe this transformation in a generic way for a matroid M. Let $\{e_1, \dots, e_m\}$ be the column vectors of M, where $e_i \in \mathbb{F}^n$. Let $e = e_1$ be the colored vector in M.

Choose $m' > m$, we construct $\ell = m + m'$ vectors $f_1, \dots, f_\ell \in \mathbb{F}^{n+m'}$ as the columns of the following $(m + m') \times \ell$ matrix. The ith column of the matrix represents f_i.

$$\left[\begin{array}{cccc|ccccccc}
e_{11} & e_{21} & \dots & e_{m1} & e_{11} & 0 & \dots & 0 & 0 & \dots & 0 \\
e_{12} & e_{22} & \dots & e_{m2} & 0 & e_{12} & \dots & 0 & 0 & \dots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
e_{1m} & e_{2m} & \dots & e_{mm} & 0 & 0 & \dots & e_{1m} & 0 & \dots & 0 \\
\hline
0 & 0 & \dots & 0 & 1 & -1 & 0 & 0 & \dots & \dots & 0 \\
\vdots & \vdots & \dots & \vdots & 0 & 1 & -1 & 0 & \dots & \dots & 0 \\
\vdots & \vdots & \dots & \vdots & \vdots & \vdots & \ddots & \ddots & \dots & \dots & \vdots \\
0 & 0 & \dots & 0 & 0 & 0 & \dots & & 0 & 1 & -1 \\
0 & 0 & \dots & 0 & -1 & 0 & \dots & & 0 & 0 & 1
\end{array}\right]$$

where -1 denotes the additive inverse of 1 in $\mathbb{F}$. Denote the above matrix as $M' = \left(\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right)$, where the sub-matrices A, B, C, and D of M' are defined as indicated by the lines in the definition of M'. Let $S = \{f_{m+1}, \dots, f_{m+m'}\}$. We observe the following:

1. Columns of B generate e_1. Since C is a 0-matrix $f_1 \in Span(S)$.

2. Columns of D are minimal dependent. Any proper subset of columns of D will split the $1, -1$ pair in at least a row and hence will be independent.
3. S is linearly independent. Suppose not. Let $\sum_{i=m}^{m+m'} \alpha_i f_i = 0$. Restricting this to the columns of B gives that $\alpha_j = 0$ for first j such that $e_{1j} \neq 0$. Thus this gives a linearly dependent proper subset of columns of B, and contradicts observation 2.
4. If for any $f \notin S$, $f = \sum_{f_i \in S} \alpha_i f_i$, then $\alpha_{m+1} = \cdots = \alpha_{m+m'}$.

Now we claim that the newly added columns respect the circuit structure involving e_1. Let $\mathcal{C}$ and $\mathcal{C}'$ denote the set of circuits of M and M' respectively.

Claim 4.3

$$\begin{aligned} &\left\{e_1, e_{i_2}, \ldots, e_{i_k}\right\} \in \mathcal{C} \quad \iff \\ &\left\{f_1, f_{i_2} \ldots, f_{i_k}\right\} \in \mathcal{C}' \quad \textit{and} \quad \left\{f_{i_2}, \ldots, f_{i_k}, f_{m+1}, \ldots, f_{m+m'}\right\} \in \mathcal{C}'. \end{aligned}$$

Proof Suppose $c = \{e_1, e_{i_2}, \ldots, e_{i_k}\}$ is a circuit in M. Then clearly $\{f_1, f_{i_2}, \ldots, f_{i_k}\}$ is a circuit, since they are nothing but vectors in c extended with 0s. Since $\{f_{i_2}, \ldots, f_{i_k}\}$ and $\{f_{m+1}, \ldots, f_{m+m'}\}$ both generate f_1, the set $F = \{f_{i_2}, \ldots, f_{i_k}, f_{m+1}, \ldots, f_{m+m'}\}$ is a linearly dependent set. Now we argue that F is a minimal dependent set, and hence is a circuit. Let $G = \{f_{i_2}, \ldots, f_{i_k}\}$.

Suppose that F is not a minimal independent set and let $F' \subset F$ be linearly dependent. Since S is linearly independent (property 3 above), we have that $F' \not\subseteq \{f_{m+1}, \ldots, f_{m+m'}\}$. Therefore, $f_{i_j} \in F'$ for some $0 \leq j \leq k$. Since F' is dependent, express f_{i_j} in terms of the other elements in F':

$$f_{i_j} = \sum_{g \in G - \{f_{i_j}\}} \gamma_g g + \sum_{s \in S} \delta_s s.$$

Since G is linearly independent, at least one of the δ_s should be non-zero. Restrict this to the matrices C and D. This gives a non-trivial dependent proper subset of D and hence a contradiction. □

From Claim 4.3 and the fact that there is no other column in M which is a multiple of e, the set $f(e) = \{f_1, f_{m+1}, \ldots, f_{m+m'}\}$ is a unique circuit of length $m' + 1 > m$ in M', where e is the column which is colored.

Now we argue about the isomorphism between M_1' and M_2' obtained from the above operation. Since there is a unique circuit of length $m' + 1 > m$ in both M_1' and M_2' corresponding to two vectors $e \in M_1$ and $e' \in M_2$, any matroid isomorphism between M_1' and M_2' should map these circuits to each other. From such an isomorphism, we can recover a matroid isomorphism between M_1 and M_2 that maps between e and e', thus preserving the colors. Indeed, if there is a matroid isomorphism between M_1 and M_2, it can easily be extended to M_1' and M_2'.

For the general case, let k be the number of different color classes and c_i denote the size of the ith color class. Then for each vector e in the color class i, we add $l_i = m + m' + i$ many new vectors, which also increases the dimension of the space by l_i. Thus the total number of vectors in the new matroid is $\sum c_i(l_i) \leq m^3$. Similarly, the dimension of the space is bounded by m^3. Rest of the proof is analogous. □

We can further generalize the above idea to matroids given in the form of independent set oracles. We define COLORED-MI as the variant of MI where the inputs are matroids $M_1 = (S_1, \mathcal{I}_1)$ and $M_2 = (S_2, \mathcal{I}_2)$ given as independent set oracles along with color functions $c_i : \{1, \ldots, m\} \to \{1, \ldots, m\}, i \in \{1, 2\}$. (Here $m = |S_1| = |S_2|$.) We assume that the color functions are part of the input and not in the oracle. The problem is to test if there is an isomorphism between M_1 and M_2 which preserves the colors of the ground set elements. We have,

Lemma 4.4 COLORED-MI *is polynomial time many-one reducible to* MI.

Proof Let $M_1 = (S_1, \mathcal{I}_1)$ and $(S_2, \mathcal{I}_2)$ be the given matroids, c_1 and c_2 be their color classes. Let $m = |S_1| = |S_2|$. We demonstrate coloring for a singleton color class. Suppose $c_1(e) = i$. Let $m' = m + i$. As in Lemma 4.1, we need to introduce a large enough new circuit C that contains e. We construct matroid M_1' (resp. M_2') as follows.

1. Let $F_1 = \{f_1, \ldots, f_{m'}\}$ be new ground set elements. Let $S_1' = S_1 \cup F_1$.
2. All circuits of M_1 remain to be so in M_1'.
3. Let $\{f_1, \ldots, f_{m'}, e\}$ be a circuit in M_1'.
4. If C is a circuit in M_1 containing e, then $(C \setminus \{e\}) \cup F_1$ is a circuit in M_1'

To see that M_1' is a matroid, we need a circuit based characterization of matroids. A set $\mathcal{C}$ of subsets of S defines circuits of a matroid on S if and only if it satisfies the *circuit elimination axioms*, which are:

- $\emptyset \notin \mathcal{C}$;
- If $A \in \mathcal{C}$ then for all $B \subset A$, $B \notin \mathcal{C}$; and
- For all $C_1 \neq C_2 \in \mathcal{C}$ and $e \in C_1 \cap C_2$, the set $(C_1 \cup C_2) \setminus \{e\}$ contains a circuit.

It is known that the set of circuits uniquely defines a matroid. (See [22] for more details.) Now by doing a case analysis it is not hard to see that the sets of circuits of M_1' defined above satisfy the above properties. Hence M_1' is a matroid. We construct M_2' analogously. Now using the arguments from Lemma 4.1 it follows that M_1' and M_2' satisfy the required property: $M_1' \cong M_2' \iff$ there is a color-preserving isomorphism between M_1 and M_2. However we need to show how to implement independent set oracles for M_1' and M_2' in polynomial time using access to those of M_1 and M_2 respectively. This essentially involves a case analysis depending on whether F_1 is contained in the input set A to be tested for independence. This can be done by the following algorithm (this is shown for M_1', the case of M_2' can be handled analogously):

Input: $A \subseteq S_1'$
Output: YES if and only if A is independent in M_1'

1. If $A \subseteq S_1$, then return YES if and only if $A \in \mathcal{I}_1$.
2. If $F_1 \cup \{e\} \subseteq A$, then return NO.
3. If $F_1 \subseteq A$ but $e \notin A$, then return YES if and only if $(A \setminus F_1) \cup \{e\} \in \mathcal{I}_1$.
4. If $F_1 \cap A \neq \emptyset$ and F_1 is not contained in A, then return YES if and only if $(A \setminus F_1) \in \mathcal{I}_1$. □

5 Graphic Matroid Isomorphism

In this section we study GMI. Unlike in the case of the graph isomorphism problem, an NP upper bound is not so obvious for GMI. We start with the discussion of an NP upper bound for GMI.

As stated in Theorem 2.1, Whitney gave an exact characterization of when two graphs are 2-isomorphic, in terms of three operations; twisting, cleaving and identification. Note that it is sufficient to find 2-isomorphisms between 2-connected components of X_1 and X_2. In fact, any *matching* between the sets of 2-connected components identifying the 2-isomorphic components will serve the purpose. This is because, any 2-isomorphism preserves simple cycles, and any simple cycle of a graph is always within a 2-connected component. Hence we can assume that both the input graphs are 2-connected and in the case of 2-connected graphs, twist is the only possible operation.

The set of separating pairs does not change under a twist operation. Despite the fact that the twist operations need not commute, Truemper [24] gave the following bound.

Lemma 5.1 [24] *Let X be a 2-connected graph of n vertices, and let Y be a graph 2-isomorphic to X, then: X can be transformed to graph X' isomorphic to Y through a sequence at most $n-2$ twists.*

Using this lemma we get an NP upper bound for GMI. Given two graphs, X_1 and X_2, the NP machine just guesses the sequence of $n-2$ separating pairs corresponding to the 2-isomorphism. For each pair, guess the cut w.r.t. which the twist operation is to be done, and apply each of them in sequence to the graph X_1 to obtain a graph X_1'. Now ask if $X_1' \cong X_2$. For the converse, by Whitney's theorem, if X_1 and X_2 are not 2-isomorphic then for any sequence of twist operations, X_1' is not isomorphic to X_2. This gives an upper bound of $\exists.\mathrm{GI} \subseteq \mathrm{NP}$. Thus we have,

Proposition 5.2 GMI *is in* NP.

This can also be seen as an NP-reduction from GMI to GI. Now we will give a deterministic reduction from GMI to GI. Although, this does not improve the NP upper bound, it implies that it is unlikely that GMI is hard for NP (using methods similar to that of Proposition 3.2, one can also directly prove that if GMI is NP-hard, then PH collapses to the second level).

Now we state the main result of the paper:

Theorem 5.3 GMI $\leq_T^p$ GI.

Also, in another seminal paper [26], Whitney showed that in the case of 3-connected graphs the notion of isomorphism and 2-isomorphism coincide. First let us recall a few definitions: A *separating pair* is a pair of vertices whose deletion leaves the graph disconnected. A 3-connected graph is a connected graph which does not have any separating pairs. In the following, we state the theorem of Whitney [26].

Theorem 5.4 (Whitney, [26]) *Let X_1 and X_2 be 3-connected graphs. Then $X_1 \cong_2 X_2 \Longleftrightarrow X_1 \cong X_2$.*

Before giving a formal proof of Theorem 5.3, we describe the idea roughly here:

Basic Idea Let X_1 and X_2 be the given graphs. From the above discussion, we can assume that they are 2-connected.

In [12], Hopcroft and Tarjan proved that every 2-connected graph can be decomposed uniquely into a tree of 3-connected components, bonds or polygons.[2] Moreover, [12] showed that this decomposition can be computed in polynomial time. The idea is to then find the isomorphism classes of these 3-connected components using queries to GI (see Theorem 5.4), and then color the tree nodes with the corresponding isomorphism class, and then compute a colored tree isomorphism between the two trees produced from the two graphs.

A first mind block is that these isomorphisms between the 3-connected components need not map separating pairs to separating pairs. We overcome this by coloring the separating pairs (in fact the edge between them), with a canonical label of the two sub-trees which the corresponding edge connects. To support this, we observe the following. There may be many isomorphisms between two 3-connected components which preserve the colors of the separating pairs. However, the order in which the vertices are mapped within a separating pair is irrelevant, since any order will be canonical up to a twist operation with respect to the separating pair.

So with the new coloring, the isomorphism between 3-connected components maps a separating pair to a separating pair, if and only if the two pairs of sub-trees are isomorphic. However, even if this is the case, the colored sub-trees need not be isomorphic. This creates a simultaneity problem of coloring of the 3-connected components and the tree nodes and thus a second mind block.

We overcome this by coloring again using the code for colored sub-trees, and then finding the new isomorphism classes between the 3-connected components. This process is iterated till the colors stabilize on the tree as well as on the individual separating pairs (since there are only linear number of 3-connected components). Once this is ensured, we can recover the 2-isomorphism of the original graph by weaving the isomorphism of the 3-connected components guided by the tree adjacency relationship. In addition, if two 3-connected components are indeed isomorphic in the correctly aligned way, the above coloring scheme, at any point, does not distinguish between them.

Now we convert this idea into an algorithm and a formal proof.

Breaking into Tree of 3-Connected Components We use the algorithm of Hopcroft and Tarjan [12] to compute the set of 3-connected components of a 2-connected graph in polynomial time. We will now describe some details of the algorithm which we will exploit.

[2] Cunningham et al. [5] shows that any graphic matroid $M(X)$ is isomorphic to $M(X_1) \oplus M(X_2) \cdots \oplus M(X_k)/\{e_1, e_2, \ldots, e_k\}$, where $M(X_1), \ldots, M(X_k)$ are 3-connected components, bonds or polygons of $M(X)$ and $e_1, \ldots, e_k$ are the virtual edges. However, it is unclear if this can be turned into a reduction from GMI to GI using edge/vertex coloring.

Let $X = (V, E)$ be a 2-connected graph. Let Y be a connected component of $X \setminus \{a, b\}$, where $\{a, b\}$ is a separating pair in X. Y is an *excisable* component with respect to $\{a, b\}$ if $X \setminus Y$ has at least 2 edges and is 2-connected. The operation of excising Y from X results in two graphs: $C_1 = X \setminus Y$ plus a *virtual edge* joining (a, b), and $C_2 =$ the induced subgraph on $Y \cup \{a, b\}$ plus a *virtual edge* joining (a, b). This operation may introduce multiple edges.

The decomposition of X into its 3-connected components is achieved by the repeated application of the excising operation (we call the corresponding separating pairs as *excised pairs*) until all the resulting graphs are free of excisable components. This decomposition is represented by a graph G_X with the 3-connected components of X as its vertices and two components are adjacent in G_X if and only if they share a virtual edge.

In the above construction, the graph G_X need not be a tree as the components which share a separating pair will form a clique. To make it a tree T_X, [12] introduces new nodes in the graph G_X corresponding to every virtual edge e, which is adjacent to all 3-connected components containing e. However, these new vertices do not correspond to any 3-connected components of X. We construct a new graph X', such that $T_X = G_{X'}$ and hence there is a 1-1 correspondence between the vertices of $G_{X'}$ and 3-connected components of X'. Formally, the graph X' is obtained by simply adding an edge between every pair of vertices which are excised while obtaining G_X. The properties of T_X listed here essentially follow from the arguments in [12]. (1) For every node $t \in T_X$, there is exactly one 3-connected component in X'. We denote this by c_t. (2) For every edge $e = (u, v) \in T_X$, there are exactly two virtual edges, one in each of the 3-connected components c_u and c_v. We call these virtual edges the *twin edges* of each other. (3) For any given graph X, T_X is unique up to isomorphism (since G_X is unique [12]). In addition, T_X can be obtained from G_X in polynomial time.

In the following claim, we prove preserves the 2-isomorphism property.

Claim 5.5 $X_1 \cong_2 X_2 \Longleftrightarrow X'_1 \cong_2 X'_2$.

Proof Suppose $X_1 \cong_2 X_2$, via a bijection $\phi : E_1 \to E_2$. This induces a map ψ between the sets of 3-connected components of X_1 and X_2. By Theorem 5.4, for every 3-connected component c of X_1, $c \cong \psi(c)$ (via say τ_c; when c is clear from the context we refer to it as τ).

We claim that ψ is an isomorphism between G_1 and G_2. To see this, consider an edge $e = (u, v) \in T_1$. This corresponds to two 3-connected components c_u and c_v of X_1 which share a separating pair s_1. The 3-connected components $\psi(c_u)$ and $\psi(c_v)$ must share a separating pair say s_2; otherwise, the cycles spanning across c_u and c_v will not be preserved by ϕ which contradicts the fact that ϕ is a 2-isomorphism. Hence $(\psi(c_u), \psi(c_v))$ corresponds to an edge in G_2. Therefore, ψ is an isomorphism between G_1 and G_2. In fact, this also gives an isomorphism between T_1 and T_2, which in turn gives a map between the excised pairs of X_1 and X_2. To define the 2-isomorphism between X'_1 and X'_2, we extend the map ψ to the excised edges.

To argue the reverse direction, let $X'_1 \cong_2 X'_2$ via ψ. In a very similar way, this gives an isomorphism between T_1 and T_2. The edge map of this isomorphism gives the map between the excised pairs. Restricting ψ to the edges of X_1 gives the required 2-isomorphism between X_1 and X_2. This is because, the cycles of $X_1(X_2)$ are anyway contained in X_1' (X_2'), and the excised pairs do not interfere in the mapping. □

Thus it is sufficient to give an algorithm to test if $X_1' \cong_2 X_2'$, which we describe as follows.

```
Input: 2-connected graphs $X_1'$ and $X_2'$ and their trees of 3-connected components $T_1$ and $T_2$ respectively.
Output: Yes if $X_1' \cong_2 X_2'$, and No otherwise.
Algorithm:
Notation: code$(T)$ denotes the canonical label[3] for a tree $T$ by applying the algorithm of [15].
1. Initialize $T_1' = T_1$, $T_2' = T_2$.
2. Repeat
   (a) Set $T_1 = T_1'$, $T_2 = T_2'$.
   (b) For each edge $e = (u, v) \in T_i$, $i \in \{1, 2\}$:
         Let $T_i(e, u)$ and $T_i(e, v)$ be sub-trees of $T_i$ obtained by deleting the edge $e$, containing $u$ and $v$ respectively.
         Color virtual edges corresponding to the separating pairs in the components $c_u$ and $c_v$ with the set $\{\text{code}(T_i(e, u)), \text{code}(T_i(e, v))\}$. From now on, $c_t$ denotes the colored 3-connected component corresponding to node $t \in T_1 \cup T_2$.
   (c) Let $S_1$ and $S_2$ be the set of colored 3-connected components of $X_1'$ and $X_2'$ and let $S = S_1 \cup S_2$. Using queries to GI (see Observation 5.8) find out the isomorphism classes in $S$. Let $C_1, \dots, C_q$ denote the isomorphism classes.
   (d) Color each node $t \in T_i$, $i \in \{1, 2\}$, with color $\ell$ if $c_t \in C_\ell$. (This gives two colored trees $T_1'$ and $T_2'$.)
   Until $(\text{code}(T_i) \neq \text{code}(T_i'), \forall i \in \{1, 2\})$
3. Check if $T_1' \cong T_2'$ preserving the colors. Answer Yes if $T_1' \cong T_2'$, and No otherwise.
```

First we prove that the algorithm terminates after linear number of iterations of the repeat-until loop. Let q_i denote the number of isomorphism classes of the set of the colored 3-connected components after the ith iteration. We claim that, if the termination condition is not satisfied, then $|q_i| > |q_{i-1}|$. To see this, suppose the termination is not satisfied. This means that the colored tree T_1' is different from T_1. This can happen only when the color of a 3-connected component c_v, $v \in T_1 \cup T_2$ changes. In addition, this can only increase the isomorphism classes. Thus $|q_i| > |q_{i-1}|$. Since q can be at most $2n$, this shows that the algorithm exits the loop after at most $2n$ steps.

[3] When T is colored, code(T) is the code of the tree obtained after attaching the necessary gadgets to the colored nodes. Notice that even after coloring, the graph is still a tree. In addition, for any T, code(T) can be computed in L [15].

Now we prove the correctness of the algorithm. We follow the notation described in the algorithm.

Lemma 5.6 $X_1' \cong_2 X_2' \Longleftrightarrow T_1' \cong T_2'$.

Proof ($\Rightarrow$) Suppose $X_1' \cong_2 X_2'$, via a bijection $\phi : E_1 \to E_2$. This induces a map ψ between the sets of 3-connected components of X_1' and X_2'. By Theorem 5.4, for every 3-connected component c of X_1', $c \cong \psi(c)$ (via say τ_c; when c is clear from the context we refer to it as τ).

We claim that ψ is an isomorphism between T_1 and T_2. To see this, consider an edge $e = (u, v) \in T_1$. This corresponds to two 3-connected components c_u and c_v of X_1' which share a separating pair s_1. The 3-connected components $\psi(c_u)$ and $\psi(c_v)$ must share a separating pair say s_2; otherwise, the cycles spanning across c_u and c_v will not be preserved by ϕ which contradicts the fact that ϕ is a 2-isomorphism. Hence $(\psi(c_u), \psi(c_v))$ correspond to an edge in T_2. Therefore, ψ is an isomorphism between T_1 and T_2. So in what follows, we interchangeably use ψ to be a map between the set of 3-connected components as well as between the vertices of the tree. Note that ψ also induces (and hence denotes) a map between the edges of T_1 and T_2.

Now we prove that ψ preserves the colors attached to T_1 and T_2 after all iterations of the repeat-until loop in step 2. To simplify the argument, we do it for the first iteration and the same can be carried forward for any number of iterations. Let T_1' and T_2' be the colored trees obtained after the first iteration. We argue that ψ itself is an isomorphism between T_1' and T_2'.

To this end, we prove that for any vertex u in T_1, $c_u \cong \psi(c_u)$ even after coloring as in step 2b. That is, the map preserves the coloring of the virtual edges in step 2b.

Consider any virtual edge f_1 in c_u, we know that $f_2 = \tau(f_1)$ is a virtual edge in $\psi(c_u)$. Let $e_1 = (u_1, v_1)$ and $e_2 = (u_2, v_2)$ be the tree edges in T_1 and T_2 corresponding to f_1 and f_2 respectively. We know that, $e_1 = \psi(e_2)$. Since $T_1 \cong T_2$ via ψ, we have

$$\{\text{CODE}(T_1(e_1, u_1)), \text{CODE}(T_1(e_1, v_1))\} = \{\text{CODE}(T_2(e_2, u_2)), \text{CODE}(T_2(e_2, v_2))\}.$$

Thus, in step 2b, the virtual edges f_1 and f_2 get the same color. Therefore, c_u and $\psi(c_u)$ belong to the same color class after step 2b. Hence ψ is an isomorphism between T_1' and T_2'.

($\Leftarrow$) First, we recall some definitions needed in the proof. A *center* of a tree T is defined as a vertex v such that $\max_{u \in T} d(u, v)$ is minimized at v, where $d(u, v)$ is the number of edges in the unique path from u to v. It is known [9] that every tree T has a center consisting of a single vertex or a pair of adjacent vertices. The minimum achieved at the center is called the *height* of the tree, denoted by $ht(T)$.

Claim 5.7 *Let ψ be a color preserving isomorphism between T_1' and T_2', and let χ_t be an isomorphism between the 3-connected components c_t and $c_{\psi(t)}$. Then, $X_1' \cong_2 X_2'$ via a map σ such that $\forall t \in T_1'$, $\forall e \in c_t \cap E_1 : \sigma(e) = \chi_t(e)$ where E_1 is the set of edges in X_1'.*

Proof The proof is by induction on height of the trees $h = ht(T_1') = ht(T_2')$, where the height (and center) is computed with respect to the underlying tree ignoring colors on the vertices.

Base case is when $h = 0$; that is, T_1' and T_2' have just one node (3-connected component) without any virtual edges. Simply define $\sigma = \chi$. By Theorem 5.4, this gives the required 2-isomorphism.

Suppose that if $h = ht(T_1') = ht(T_2') < k$, the above claim is true. For the induction step, suppose further that $T_1' \cong T_2'$ via ψ, and $ht(T_1') = ht(T_2') = k$. Notice that ψ should map the center(s) of T_1 to that of T_2. We consider two cases:

In the first case, T_1' and T_2' have unique centers α and β. It is clear that $\psi(\alpha) = \beta$. Let c_1 and c_2 be the corresponding colored (as in step 2b) 3-connected components. Therefore, there is a color preserving isomorphism $\chi = \chi_\alpha$ between c_α and c_β. Let $f_1, \dots, f_k$ be the virtual edges in c_α corresponding to the tree edges $e_1 = (\alpha, v_1), \dots, e_k = (\alpha, v_k)$ where $v_1, \dots, v_k$ are neighbors of α in T_1'. Denote $\psi(e_i)$ by e_i', and $\psi(v_i)$ by v_i'.

Observe that only virtual edges are colored in the 3-connected components in step 2b while determining their isomorphism classes. Therefore, for each i, $\chi(f_i)$ will be a virtual edge in c_β, and in addition, with the same color as f_i. That is,

$$\{\text{CODE}(T_1(e_i, \alpha)), \text{CODE}(T_1(e_i, v_i))\} = \left\{\text{CODE}(T_2(e_i', \beta)), \text{CODE}(T_2(e_i', v_i'))\right\}.$$

Since α and β are the centers of T_1' and T_2', it must be the case that in the above set equality, $\text{CODE}(T_1(e_i, v_i)) = \text{CODE}(T_2(e_i', v_i'))$. From the termination condition of the algorithm, this implies that $\text{CODE}(T_1'(e_i, v_i)) = \text{CODE}(T_2'(e_i', v_i'))$. Hence, $T_1'(e_i, v_i) \cong T_2'(e_i', v_i')$. In addition, $ht(v_i) = ht(v_i') < k$. Let X'_{f_i} and $X'_{\chi(f_i)}$ denote the subgraphs of X_1' and X_2' corresponding to $T_1'(e_i, v_i)$ and $T_2'(e_i', v_i')$ respectively. By induction hypothesis, the graphs X'_{f_i} and $X'_{\chi(f_i)}$ are 2-isomorphic via σ_i which agrees with the corresponding χ_t for $t \in T_1'(e_i, v_i)$. Define π_i as a map between the edges, such that it agrees with σ_i on all edges of $X'_{f(i)}$ and with χ_t (for $t \in T_1'(e_i, v_i)$) on the colored virtual edges.

We claim that π_i must map the twin-edge of f_i to twin-edge of $\tau(f_i)$. Suppose not. By the property of the coloring, this implies that there is a sub-tree of $T_1'(e_i, v_i)$ isomorphic to $T_1' \setminus T_1'(e_i, v_i)$. This contradicts the assumption that c_α is the center of T_1'.

For each edge $e \in E_1$, define $\sigma(e)$ to be $\chi(e)$ when $e \in c_\alpha$ and to be $\pi_i(e)$ when $e \in E_{f_i}$ (edges of X_{f_i}). From the above argument, $\chi = \chi_\alpha$ and σ_i indeed agrees on where it maps f_i to. This ensures that every cycle passing through the separating pairs of c_α gets preserved. Thus σ is a 2-isomorphism between X_1' and X_2'.

For case 2, let T_1' and T_2' have two centers (α_1, α_2) and (β_1, β_2) respectively. It is clear that $\psi(\{\alpha_1, \alpha_2\}) = \{\beta_1, \beta_2\}$. Without loss of generality, we assume that $\psi(\alpha_1) = \beta_1$, $\psi(\alpha_2) = \beta_2$. Therefore, there are color preserving isomorphisms χ_1 from c_{α_1} to c_{β_1} and χ_2 from c_{α_2} and c_{β_2}. Define $\chi(e)$ as follows:

$$\chi(e) = \begin{cases} \chi_1(e), & e \in c_{\alpha_1}, \\ \chi_2(e), & e \in c_{\alpha_2}, \end{cases}$$

$$c_\alpha = \bigcup_i c_{\alpha_i}, \quad c_\beta = \bigcup_i c_{\beta_i}$$

With this notation, we can appeal to the proof in the case 1, and construct the 2-isomorphism σ between X_1' and X_2'. □

This completes the proof of correctness of the algorithm (Lemma 5.6). □

To complete the proof of Theorem 5.3, we need the following observation.

Observation 5.8 COLORED-GMI *for* 3-*connected graphs reduces to* GI.

To complete the equivalence of GI, MI_b, LMI_b and GMI, we give a polynomial time many-one reduction from MI_b to GMI.

Theorem 5.9 $\mathrm{MI}_b \leq_m^p$ GMI.

Proof Let M_1 and M_2 be two matroids of rank b over the ground set S_1 and S_2. Let $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively denote the set of circuits of M_1 and M_2. Note that $|\mathcal{C}_1|, |\mathcal{C}_2| \leq m^{b+1}$.

We define graphs $X_1 = (V_1, E_1)$ (respectively for $X_2 = (V_2, E_2)$) as follows. For each circuit $c = \{s_1, \dots, s_\ell\} \subseteq S_1$ in M_1, let G_c be the undirected graph (V_c, E_c) where $V_c = \{u_i, x_i, y_i \mid 1 \leq i \leq \ell\}$ and

$$E_c = \bigcup_{i=1}^{\ell} \{(u_i, u_{(i+1\,\mathrm{mod}\ell)+1}), (x_i, y_i), (u_i, x_i), (u_{(i+1\,\mathrm{mod}\ell)+1}, y_i)\}.$$

See Fig. 1 for an example. We say that x_i and y_i are the vertices corresponding to s_i in G_c. Color the edges $(u_i, u_{(i+1\,\mathrm{mod}\ell)+1})$ as BLUE for $1 \leq i \leq \ell$. The edges $(u_{i+1\,\mathrm{mod}\ell}, y_i)$ and (u_i, x_i) are colored YELLOW and (x_i, y_i) are colored GREEN for $1 \leq i \leq \ell$. Now, X_1 contains the disjoint union of G_c for all $c \in \mathcal{C}_1$ and additionally the following edges: For every $s \in S_1$, consider all the circuits $c \in \mathcal{C}_1$ that contain s. Let $x_{s,c}$ and $y_{s,c}$ denote the vertices that correspond to s in G_c. Then add all the

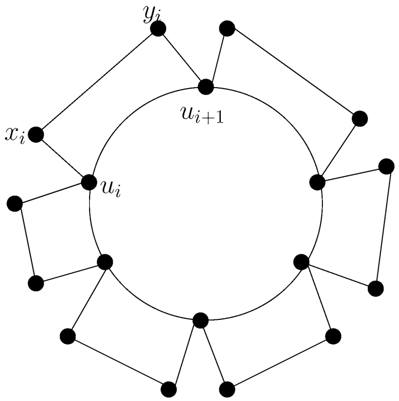

Fig. 1 An example of G_c when c is a six-vertex simple cycle

edges necessary so that the set $\{x_{s,c}, y_{s,c} \mid s \text{ is contained in } c\}$ is a clique in X_1; call this clique R_s. The new edges added to complete the clique are colored RED.

We list the properties of X_1 for further reference:

1. For every circuit $c \in \mathcal{C}_1$, there is a unique BLUE cycle in X_1 that is disjoint from all other BLUE cycles.
2. All the cliques with at least four vertices in X_1 are formed by edges colored RED and GREEN. Moreover, there is a one-one map from the set of all cliques of size at least four in X_1 to the ground set S_1.
3. For every circuit $c \in \mathcal{C}_1$, the union of all the cliques of X_1 corresponding to the elements of c defines a unique blue cycle whose associated GREEN edges are in the cliques.

Now we claim the following:

Lemma 5.10 *$M_1 \cong M_2$ if and only if $X_1 \cong_2 X_2$.*

Proof Suppose $M_1 \cong M_2$, via a map $\phi : S_1 \to S_2$.

This gives a map ψ between the BLUE edges of the graphs X_1 and X_2 which preserves BLUE cycles. Now it is not hard to see that we can extend this map to include the remaining edges.

Conversely, suppose $X_1 \cong_2 X_2$ via $\psi : E_1 \to E_2$. Define $\phi : S_1 \to S_2$ as follows: For $s \in S_1$ let R_s denote the clique in X_1 corresponding to s. R_s is either a single GREEN edge or a clique on at least 4 vertices (in the latter case it is 3-connected). Thus, by the property 2 of X_1 we can see that ψ maps R_s to $R'_{s'}$ for some s' in S_2. Define $\phi(s) = s'$.

Now we argue that ψ is an isomorphism between M_1 and M_2. Let $c \subseteq S_1$ be a circuit in M_1. Now using the property 2 of X_1, we have:

$$\begin{aligned} c \in \mathcal{C}_1 \quad &\Longleftrightarrow \quad \bigcup_i \psi(R_{s_i}) \quad \text{defines a unique BLUE cycle in } X_1 \\ &\Longleftrightarrow \quad \bigcup_i \psi(R'_{s'_i}) \quad \text{defines a unique BLUE cycle in } X_2 \\ &\Longleftrightarrow \quad \phi(c) \in \mathcal{C}_2. \end{aligned}$$ □

This completes the proof of Theorem 5.9. □

The following corollary summarizes the relationship between GMI, GI, LMI$_b$, and MI$_b$ proved so far,

Corollary 5.11 GMI $\leq_T^p$ GI $\equiv_m^p$ LMI$_b$ $\equiv_m^p$ MI$_b$ $\leq_m^p$ GMI.

6 Improved Upper Bounds for Special Cases of GMI

In this section we give improved upper bounds for special cases of GMI such as planar graphic matroids, matroids of graphs of bounded genus and bounded eigen value.

6.1 Planar Matroids

Recall that a graph is said to be planar if it can be drawn on a plane without any crossings. A matroid is called a planar matroid if it is the graphic matroid of a planar graph. Let PMI denote the computational problem of isomorphism testing for planar matroid. Observing that the construction used in the proof of Theorem 5.3 does not use any non-planar gadgets and the fact that isomorphism testing of planar graphs can be done in P [13], we get the following.

Corollary 6.1 PMI *is in* P.

Using the recent developments on the planar graph isomorphism problem, we improve the above bound to show that PMI $\in$ L. We adapt the log-space canonization procedure of [7] to the setting of planar matroids to obtain a log-space algorithm for PMI. The idea used in [7] is to build the canonization using the 3-connected component decomposition of the given 2-connected planar graph. We briefly describe the modifications to this procedure.

Theorem 6.2 PMI $\in$ L. *Moreover, a canonical encoding for planar matroids can be obtained in* log-*space*.

Proof As observed in Sect. 5, it is sufficient to consider the case of 2-connected graphs. Let $X_1 = (V_1, E_1)$ and $X_2 = (G_2, V_2)$ be the given 2-connected planar graphs. Let T_1 and T_2 be the unique decompositions of X_1 and X_2 into 3-connected components respectively. (This can be done in log- space [7].) Suppose T_1 (resp. T_2) is rooted at r_1 (resp. r_2). We proceed as in [7], the only difference being that we ignore the orientations of the virtual edges.

The modified definition of ordering of the 3-connected component tree is as follows:

$T_1 <_T T_2$ if one of the following holds,

(1) $|T_1| < |T_2|$
(2) $|T_1| = |T_2|$ and # of sub-trees of r_1 is less than that of r_2 or
(3) $|T_1| = |T_2|$ and # of sub-trees of r_1 is equal to that of r_2 and $(T_{1,1}, \dots, T_{1,l}) < (T_{2,1}, \dots, T_{2,l})$ where $T_{1,1} \le_T \cdots \le_T T_{1,l}$ (resp. $T_{2,1} \le_T \cdots \le_T T_{2,l}$) are subtrees of T_1 (resp. T_2) rooted at the children of r_1 (resp. r_2). Here $<$ refers to the lexicographic order.

Here is an outline of the algorithm:

(1) Compute T_1 (resp. T_2) rooted at r_1 (resp. r_2).
(2) Check if $T_1 =_T T_2$ using the algorithm of [7].

By Whitney's theorem (see Theorem 2.1), twist operations on G do not change the underlying matroid, and so we get the required correctness of the algorithm. The space complexity bound follows from the arguments in [7].

The canonization of planar matroids can also done in a similar fashion following [7]. □

6.2 Matroids of Bounded Genus and Bounded Degree Graphs

The *genus* of a graph is the minimum number k of handles that are required so that the graph can be drawn on a plane with k handles without any crossings of the edges. If we are given the guarantee that the input instances of GMI are graphs of bounded genus (resp. bounded degree), then in the decomposition of the graphs into 3-connected components the components obtained are themselves graphs of bounded genus (resp. bounded degree). Hence the queries made to GI are that of bounded genus (resp. bounded degree) instances which are known to be in P (see [17, 20]). Thus, as a corollary of Theorem 5.3, we have:

Corollary 6.3 *Isomorphism testing of matroids of graphs of bounded genus/degree can be done in* P.

7 Matroid Automorphism Problem

With any isomorphism problem, there is an associated automorphism problem i.e, to find a generating set for the automorphism group of the underlying object. Relating the isomorphism problem to the corresponding automorphism problem gives access to algebraic tools associated with the automorphism groups. In the case of graphs, studying automorphism problem has been fruitful (e.g. see [1, 4, 17]). In this section we turn our attention to the matroid automorphism problem.

An automorphism of a matroid $M = (S, \mathcal{C})$ (where S is the ground set and $\mathcal{C}$ is the set of circuits) is a permutation ϕ of elements of S such that $\forall C \subseteq S$, $C \in \mathcal{C} \iff \phi(C) \in \mathcal{C}$. $\mathrm{Aut}(M)$ denotes the group of automorphisms of the matroid M. When the matroid is graphic we denote by $\mathrm{Aut}(X)$ and $\mathrm{Aut}(M_X)$ the automorphism group of the graph and the graphic matroid respectively.

To begin with, we note that given a graph X, and a permutation $\pi \in S_m$, it is not clear a priori how to check if $\pi \in \mathrm{Aut}(M_X)$ efficiently. This is because we need to ensure that π preserves all the simple cycles, and there could be exponentially many of them. Note that such a membership test (given a $\pi \in S_n$) for $\mathrm{Aut}(X)$ can easily be done by testing whether π preserves all the edges. We provide an efficient algorithm for testing if $\pi \in \mathrm{Aut}(M_X)$.

We use the notion of a cycle basis of X. A *cycle basis* of a graph X is a minimal set of cycles $\mathcal{B}$ of X such that every cycle in X can be written as a linear combination (viewing every cycle as a vector in $\mathbb{F}_2^m$) of the cycles in $\mathcal{B}$. Let $\mathscr{B}$ denote the set of all cycle bases of the graph X.

Lemma 7.1 *Let* $\pi \in S_n$. *Then* $\exists \mathcal{B} \in \mathscr{B} : \pi(\mathcal{B}) \in \mathscr{B} \implies \forall \mathcal{B} \in \mathscr{B} : \pi(\mathcal{B}) \in \mathscr{B}$.

Proof Let $\mathcal{B} = \{b_1, \ldots, b_\ell\} \in \mathscr{B}$ such that $\pi(\mathcal{B}) = \{\pi(b_1), \ldots, \pi(b_\ell)\}$ is a cycle basis. Now consider any other cycle basis $\mathcal{B}' = \{b'_1, \ldots, b'_k\} \in \mathscr{B}$. Thus, $b_i = \sum_j \alpha_j b'_j$. This implies,

$$\pi(b_i) = \sum_j \alpha_j \pi(b'_j).$$

Thus, $\pi(B') = \{\pi(b'_1), \ldots, \pi(b'_\ell)\}$ forms a cycle basis. □

Lemma 7.2 *Let $\pi \in S_m$, and let $\mathcal{B} \in \mathscr{B}$, then $\pi \in \mathrm{Aut}(M_X) \Longleftrightarrow \pi(\mathcal{B}) \in \mathscr{B}$.*

Proof Let $\mathcal{B} = \{b_1, \ldots, b_\ell\}$ be the given cycle basis.

For the forward direction, suppose $\pi \in \mathrm{Aut}(M_X)$. That is, $C \subseteq E$ is a cycle in X if and only if $\pi(C)$ is also a cycle in X. Let C be any cycle in X, and let $D = \pi^{-1}(C)$. Since $\mathcal{B} \in \mathscr{B}$, we can write, $D = \sum_i \alpha_i b_i$, and hence $C = \sum_i \alpha_i \pi(b_i)$. Hence $\pi(\mathcal{B})$ forms a cycle basis for X.

For the reverse direction, suppose $\pi(\mathcal{B})$ is a cycle basis of X. Let C be any cycle in X. We can write $C = \sum_i \alpha_i b_i$. Hence, $\pi(C) = \sum_i \alpha_i \pi(b_i)$. As π is a bijection, we have $\pi(b_i \cap b_j) = \pi(b_i) \cap \pi(b_j)$. Thus $\pi(C)$ is also a cycle in X. Since π extends to a permutation on the set of cycles, we get that C is a cycle if and only if $\pi(C)$ is a cycle. □

Using Lemmas 7.1 and 7.2 it follows that, given a permutation π, to test if $\pi \in \mathrm{Aut}(M_X)$ it suffices to check if for a cycle basis $\mathcal{B}$ of X, $\pi(\mathcal{B})$ is also a cycle basis. Given a graph X a cycle basis $\mathcal{B}$ can be computed in polynomial time (see e.g. [11]). Now it suffices to show:

Lemma 7.3 *Given a permutation $\pi \in S_m$, and a cycle basis $\mathcal{B} \in \mathscr{B}$, testing whether $\pi(\mathcal{B})$ is a cycle basis, can be done in polynomial time.*

Proof To check if $\pi(\mathcal{B})$ is a cycle basis, it is sufficient to verify that every cycle in $\mathcal{B} = \{b_1, \ldots, b_\ell\}$ can be written as a $\mathbb{F}_2$-linear combination of the cycles in $\mathcal{B}' = \{b'_1, \ldots, b'_\ell\} = \pi(\mathcal{B})$. This can be done as follows.

For $b_i \in \mathcal{B}$, let $\pi(b_i) = b'_i$. View b_i and b'_i as vectors in $\mathbb{F}_2^m$. Let b_{ij} (resp. b'_{ij}) denote the jth component of b_i (resp. b'_i). Construct the set of linear equations, $b'_{ij} = \sum_{b_k \in \mathcal{B}} x_{ik} b_{kj}$ where x_{ik} are unknowns. There are exactly ℓ b_i's and each of them gives rise to exactly m equations like this. This gives a system I of ℓm linear equations in ℓ^2 unknowns such that, $\pi(B)$ is a cycle basis if and only if I has a non-trivial solution. This test can indeed be done in polynomial time. □

This gives us the following:

Theorem 7.4 *Given any $\pi \in S_m$, the membership test for π in $\mathrm{Aut}(M_X)$ is in* P.

Notice that similar arguments can also give another proof of Proposition 5.2. As in the case of graphs, we can define automorphism problems for matroids.

MATROID AUTOMORPHISM(MA): *Given a matroid M as independent set oracle, compute a generating set for* $\mathrm{Aut}(M)$.

We define GMA and LMA as the corresponding automorphism problems for graphic and linear matroids, when the input is a graph and matrix respectively. Using the coloring techniques from Sect. 4, we prove the following equivalence.

Theorem 7.5 LMI $\equiv_T^p$ LMA, *and* GMI $\equiv_T^p$ GMA.

Proof This proof follows a standard idea due to Luks [18]. We argue the forward direction as follows. Given two matrices M_1 and M_2, form the new matrix M as,

$$M = \begin{bmatrix} M_1 & 0 \\ 0 & M_2 \end{bmatrix}.$$

Now using queries to LMA construct the generating set of $\mathrm{Aut}(M)$. Check if at least one of the generators maps the columns in M corresponding to columns of M_1 to those corresponding to the columns of M_2.

To see the other direction, we use the coloring idea, and the rest of the details are standard. The idea is to find the orbits of each element of the ground set as follows: For every element of $e \in S$, for each $f \in S$, color e and f by the same color to obtain colored matroids M_1 and M_2. Now by asking queries to LMI we can check if f is in the orbit of e. Thus we can construct the orbit structure of $\mathrm{Aut}(M)$ and hence compute a generating set.

Using similar methods we can prove GMI $\equiv_T^p$ GMA. □

8 Closure Properties

In this section we consider taking and-function and or-functions of polynomial many instances of GMI. Following [14], we formally define and-functions and or-functions as follows:

Definition 8.1 (See [14, 16]) Let A be any language in $\{0,1\}^*$. An *or-function* for A is a function $f : \{0,1\}^* \to \{0,1\}^*$ such that for every sequence $x_1, \ldots, x_\ell \in \{0,1\}^*$ we have,

$$f(x_1, \ldots, x_\ell) \in A \quad \Longleftrightarrow \quad \exists i \in [\ell],\ x_i \in A.$$

Similarly, an and-function for A is a function $g : \{0,1\}^* \to \{0,1\}^*$ such that for all $x_1, \ldots, x_\ell \in \{0,1\}^*$ the following holds:

$$g(x_1, \ldots, x_\ell) \in A \quad \Longleftrightarrow \quad \forall i \in [\ell],\ x_i \in A.$$

We show that GMI restricted to 2-connected graphs has these closure properties.

Theorem 8.2 GMI *restricted to* 2*-connected graphs has polynomial time computable and-functions and or-functions.*

Proof Our proof follows closely the proof of closure properties of and/or-functions for GI given in [14].

AND-*Function* Let $(G_1, H_1), \ldots, (G_\ell, H_\ell)$ be ℓ different instances of GMI where all the graphs are 2-connected. We first demonstrate the construction for $\ell = 2$.

Let $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$, $H_1 = (V_1', E_1')$, $H_2 = (V_2', E_2')$, $|V_1| = |V_1'| = n_1$, $|V_2| = |V_2'| = n_2$ and $|E_1| = |E_1'| = m_1$, $|E_2| = |E_2'| = m_2$. We construct two graphs $G = \langle G_1, G_2 \rangle$ and $H = \langle H_1, H_2 \rangle$ such that $G \cong_2 H \Longleftrightarrow (G_1 \cong_2 H_1$ and

$G_2 \cong_2 H_2$). The vertex set V of G consists of V_1 and V_2 with four additional vertices u^1, u^2, v^1, v^2. Add (u^1, u^2) and (v^1, v^2) as edges. Now for every edge $e = (a, b) \in E_1$, add new edges so that the subgraph induced by $\{u^1, u^2, a, b\}$ is a 4-vertex clique. Similarly for every $e = (a, b) \in E_1 \cup E_2$, add new edges to G so that the subgraph induced by $\{v^1, v^2, a, b\}$ forms a 4-vertex clique.

Define $G = (V, E)$ as follows:

$$V = V_1 \cup V_2 \cup \{u^1, u^2, v^1, v^2\},$$

$$\begin{aligned} E = E_1 \cup E_2 &\cup \{(u^1, u^2), (v^1, v^2)\} \\ &\cup \{(u^i, a), (u^i, b) \mid (a, b) \in E_1, i \in \{1, 2\}\} \\ &\cup \{(v^i, a), (v^i, b) \mid (a, b) \in E_1 \cup E_2,\ i \in \{1, 2\}\}. \end{aligned}$$

We define H in a similar fashion using H_1 and H_2 instead of G_1 and G_2 respectively. We denote the four new vertices thus introduced in H by $\bar{u}^1, \bar{u}^2, \bar{v}^1, \bar{v}^2$.

Now the following claim completes the proof for the and-function:

Claim 8.3 $(G_1 \cong_2 H_1$ *and* $G_2 \cong H_2) \iff G \cong_2 H$.

Proof The forward direction is easy to see. To prove the converse, suppose $G \cong_2 H$ via a bijection $\phi : E \to E'$. Let $e_{m+1} = (u^1, u^2), e_{m+2} = (v^1, v^2)$ and $\bar{e}_{m+1} = (\bar{u}^1, \bar{u}^2), e_{m+2} = (\bar{v}^1, \bar{v}^2)$. Now, as e_{m+1} (resp. $\bar{e}_{m+1}$) is the unique edge in G that intersects with n_1 many 4-vertex cliques, we have $\phi(e_{m+1}) = \bar{e}_{m+1}$. Similarly we can argue that $\phi(e_{m+2}) = \bar{e}_{m+2}$. Also, all the newly introduced edges of G get mapped to those of H. Thus we can recover the required 2-isomorphisms between G_1, H_1 and G_2, H_2 respectively. □

Note that we introduced only 8 new vertices, 4 for each of G and H. In the case of $\ell > 2$ we do the above process iteratively. At each iteration we add 8 new vertices, hence the final graphs will have number of vertices bounded by $n + 8\ell$ (where n is the total number of vertices in the graphs we began with). As the graphs obtained are always simple, the number of edges is bounded by $O((n + 8\ell)^2)$. Also, it is straightforward to see that the computation of the resulting graphs can be done in polynomial time.

OR-FUNCTION Let (G_1, H_1) and (G_2, H_2) be two instances of GMI. Now define the function f as:

$$f((G_1, H_1), (G_2, H_2)) = (\langle G_1, G_2\rangle \cup \langle H_1, H_2\rangle, \langle G_1, H_2\rangle \cup \langle H_1, G_2\rangle).$$

From the arguments in the above paragraphs, it is easy to see that f represents the or-function of (G_1, H_1) and (G_2, H_2). However, extending this directly for polynomial many instances will cause an exponential blow up in size. We use the divide and conquer approach as done in Theorem 1.42 of [14].

Let $x_i = (G_i, H_i)$, $1 \le i \le \ell$ be the given sequence of instances of GMI. We define the function $\bar{f}$ as follows:

$$\bar{f}(x_1, \dots, x_\ell) = \begin{cases} x_1 & \text{if } \ell = 1, \\ f(\bar{f}(x_1, \dots, x_{\lceil \ell/2 \rceil}).\bar{f}(x_{\lceil \ell/2 \rceil + 1}, \dots, x_\ell)) & \text{otherwise} \end{cases}$$

From the definition, the depth of recursion is $O(\log \ell)$. At each step the application of f blows up the size by a constant factor. Thus the size of the graph $\bar{f}(x_1, \dots, x_\ell)$ is bounded by $\text{POLY}(\ell)$. Now using the arguments similar to the one in the proof of Theorem 1.42 of [14] we get the desired result. □

Remark 1 Note that the Theorem 8.2 cannot be directly applied to graphs that are not 2-connected. This is mainly because our reduction from the connected GMI instance to 2-connected instance is a Turing reduction and not a many-one reduction. (See discussions preceding Lemma 5.1.)

9 Conclusion and Open Problems

We studied the matroid isomorphism problem under various input representations and restriction on the rank of the matroid. We proved that graph isomorphism, graphic matroid isomorphism and bounded rank version of matroid isomorphism are all polynomial time equivalent.

In addition, we find it interesting that in the bounded rank case, MI_b and LMI_b are equivalent, though there exist matroids of bounded rank which are not representable over any field. Some of the open questions that we see are as follows:

- Our results provide new possibilities to attack the graph isomorphism problem. For example, it will be interesting to prove a coNP upper bound for LMI_b. Note that this will imply that GI $\in$ NP $\cap$ coNP. Similarly, are there special cases of GMI (other than what is translated from the bounds for GI) which can be solved in polynomial time?
- The representations of the matroid in the definition of LMI is over fields of size at least m and at most $\text{POLY}(m)$, where m is the size of the ground set of the matroid. This is critically needed for the observation of coNP-hardness. One could ask if the problem is easier over fixed finite fields independent on the input. However, we note that, by our results, it follows that this problem over $\mathbb{F}_2$ is already hard for GI. It will still be interesting to give a better (than the trivial Σ_2) upper bound for linear matroids represented over fixed finite fields (even for $\mathbb{F}_2$).
- Can we use the coloring technique of linear matroid isomorphism to reduce the general instances of linear matroid isomorphism to isomorphism testing of "simpler components" of the matroid?
- Can we make the reduction from GMI to GI many-one? Can we improve the complexity of this reduction in the general case?

Acknowledgements We thank V. Arvind and Meena Mahajan for providing us with useful inputs and many insightful discussions, James Oxley for sharing his thoughts while responding to our queries about matroid isomorphism. We also thank the anonymous referees for providing us many useful pointers to the literature.

References

1. Arvind, V., Kurur, P.P.: Graph isomorphism is in SPP. In: FOCS, pp. 743–750 (2002)
2. Arvind, V., Torán, J.: Isomorphism testing: perspective and open problems. Bull. Eur. Assoc. Theor. Comput. Sci. **86**, 66–84 (2005)
3. Babai, L.: Vector representable matroids of given rank with given automorphism group. Discrete Math. **24**, 119–125 (1978)
4. Babai, L., Grigoryev, D.Yu., Mount, D.M.: Isomorphism of graphs with bounded eigenvalue multiplicity. In: *STOC*, pp. 310–324 (1982)
5. Cunningham, W.H., Edmonds, J.: A combinatorial decomposition theory. Can. J. Math. **17**, 734–765 (1980)
6. Datta, S., Limaye, N., Nimbhorkar, P.: 3-connected planar graph isomorphism is in log-space. In: Proceedings of FSTTCS, pp. 155–162 (2008)
7. Datta, S., Limaye, N., Nimbhorkar, P., Thierauf, T., Wagner, F.: Planar graph isomorphism is in log-space. In: IEEE Conference on Computational Complexity, pp. 203–214 (2009)
8. Goldreich, O.: Computational Complexity: A Conceptual Perspective. Cambridge University Press, Cambridge (2008)
9. Harary, F.: Graph Theory. Addison-Wesley, Reading (1969)
10. Hlinený, P.: Some hard problems on matroid spikes. Theory Comput. Syst. **41**(3), 551–562 (2007)
11. Horton, J.D.: A polynomial-time algorithm to find the shortest cycle basis of a graph. SIAM J. Comput. **16**(2), 358–366 (1987)
12. Hopcroft, J.E., Tarjan, R.E.: Dividing a graph into triconnected components. SIAM J. Comput. **2**(3), 135–158 (1973)
13. Hopcroft, J.E., Wong, J.K.: Linear time algorithm for isomorphism of planar graphs (preliminary report). In: STOC '74: Proceedings of the Sixth Annual ACM Symposium on Theory of Computing, pp. 172–184. ACM, New York (1974)
14. Köbler, J., Schöning, U., Torán, J.: The Graph Isomorphism Problem: Its Structural Complexity. Birkhäuser, Basel (1993)
15. Lindell, S.: A logspace algorithm for tree canonization (extended abstract). In: STOC, pp. 400–404 (1992)
16. Lozano, A., Torán, J.: On the nonuniform complexity of the graph isomorphism problem. In: Structure in Complexity Theory Conference, pp. 118–129 (1992)
17. Luks, E.M.: Isomorphism of graphs of bounded valence can be tested in polynomial time. In FOCS, pp. 42–49 (1980)
18. Luks, E.M.: Permutation Groups and Polynomial-Time Computation. DIMACS, vol. 11, pp. 139–175. (1993)
19. Mayhew, D.: Matroid complexity and nonsuccinct descriptions. SIAM J. Discrete Math. **22**(2), 455–466 (2008)
20. Miller, G.L.: Isomorphism testing for graphs of bounded genus. In: STOC, pp. 225–235 (1980)
21. Oxley, J., Welsh, D.: Chromatic, flow and reliability polynomials: the complexity of their coefficients. Comb. Probab. Comput. **11**, 403–426 (2002)
22. Oxley, J.G.: Matroid Theory. Oxford University Press, New York (1992)
23. Schöning, U.: Probabilistic Complexity Classes and Lowness. J. Comput. Syst. Sci. **39**, 84–100 (1999)
24. Truemper, K.: On Whitney's 2-isomorphism theorem for graphs. J. Graph Theory, 43–49 (1980)
25. Thierauf, T., Wagner, F.: The isomorphism problem for planar 3-connected graphs is in unambiguous logspace. In: STACS, pp. 633–644 (2008)
26. Whitney, H.: Congruent graphs and connectivity of graphs. Am. J. Math. **54**(1), 150–168 (1932)
27. Whitney, H.: 2-isomorphic graphs. Am. J. Math. **55**, 245–254 (1933)
28. Whitney, H.: On the abstract properties of linear dependence. Am. J. Math. **57**(3), 509–533 (1935)