On the Complexity of Matroid Isomorphism Problems

Raghavendra Rao B.V.* and Jayalal Sarma M.N. **

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. 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 problem is 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.

1 Introduction

Isomorphism problems over various mathematical structures have been a source of intriguing problems in complexity theory (see [1]). The most important problem of this domain is the well-known graph isomorphism problem. Though the complexity characterisation 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 [13, 10] and many structural results are known [9, 19, 11]. In this paper we talk about isomorphism problem 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.* 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 and is well studied [16]. We study the problem of testing isomorphism between two given matroids.

* *The Institute of Mathematical Sciences, C.I.T. Campus, Chennai 600 113, India.* bvrr@imsc.res.in

** *Institute for Theoretical Computer Science, Tsinghua University, Beijing 100 084, China.* jayalal@tsinghua.edu.cn, *This work was supported in part by the National Natural Science Foundation of China Grant 60553001, and the National Basic Research Program of China Grant 2007CB807900,2007CB807901.*

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 section 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 [14] studied the matroid isomorphism problem, and shows that the problem is equivalent to 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, [16].). 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 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 incidence matrix of the graph will give a representation over $\mathbb{F}_2$. There are linear matroids which are not graphic. (See [16] 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)) where the inputs are in the implicit representation (matrices and graphs resp.).

From a complexity perspective, the general case of the problem (where the matroid is given as an independent set oracle) is in Σ_2^p. However, it is not even clear a priori if the problem is in NP even in the above restricted cases 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.

An important result in this direction, due to Whitney (see [21]), says that in the case of 3-connected graphs, the graphs are isomorphic if and only if the corresponding matroids are isomorphic (see section 5). Thus the problems of testing isomorphism of graphs and of 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 [4], 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:

- We prove that when the rank of the matroid is bounded, linear matroid isomorphism and matroid isomorphism are both equivalent to GI (Theorem 2)[1]
- We develop tools to handle colouring of ground set elements in the context of the isomorphism problem. We show that coloured versions of linear matroid isomorphism and graphic matroid isomorphism are as hard as the general version (Lemma 2, 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 8).
- We give a polynomial time Turing reduction (Theorem 3) from graphic matroid isomorphism problem to the graph isomorphism problem by developing an edge colouring scheme which algorithmically uses a decomposition given by [7] (and [3]). Our reduction, in particular implies that the graphic matroid isomorphism testing for planar graphs can be done in deterministic polynomial time (Corollary 2).
- Finally, we give a reduction from bounded rank matroid isomorphism problem to graphic matroid isomorphism (Theorem 5), thus showing that all the above problems are poly-time equivalent.

Due to space limitations we have omitted many proofs. The omitted proofs can be found in the full version [17].

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 [5]).

[1] We note that, although not explicitly stated, the equivalence of bounded rank matroid isomorphism and and graph isomorphism also follows from the results of Mayhew [14]. However, it is not immediately clear if the GI-hard instances of [14] are linearly representable. Our proof is different and extend this to linear matroids.

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. It is clear that for two matroids to be isomorphic the ground set has to be of the same size (say m) and they have to be of the same rank (say r). Now we state the computational problems more precisely.

*Problem 1 (*MATROID ISOMORPHISM(MI)*).* Given two matroids M_1 and M_2 as their ground sets and the independent set oracles, test if $M_1 \cong M_2$.

Given a matrix $A_{n \times m}$ over a field $\mathbb{F}$, we can define a matroid $M[A]$ with columns of A as the ground set (of m elements) and linearly independent columns as the independent sets of $M[A]$. A matroid $\mathcal{M} = (E, \mathcal{I})$ of rank r ($\leq n$) is said to be representable over $\mathbb{F}$, if there exists a matrix $A \in \mathbb{F}^{r \times m}$ such that $\mathcal{M}$ is isomorphic to the matroid $M[A]$. *Linear matroids* are matroids representable over fields. We assume that the field on which the matroid is represented is also a part of the input as the table for both operations, and that the field has at least m elements and at most $poly(m)$ elements.

*Problem 2 (*LINEAR MATROID ISOMORPHISM(LMI)*).* Given two matrices A and B over a given field $\mathbb{F}$, test if $M[A] \cong M[B]$.

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 $\mathcal{M}$ is called *graphic* iff $\exists X$ such that $\mathcal{M}$ is isomorphic to $M(X)$.

*Problem 3 (*GRAPHIC MATROID ISOMORPHISM(GMI)*).* Given two graphs X_1 and X_2, test if $M(X_1) \cong M(X_2)$?.

We denote by PMI, the version of GMI where the input graphs are planar. Another associated terminology in the literature is about 2-isomorphism. Two graphs X_1 and X_2 are said to be 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. In a rather surprising result, Whitney [22] came up with a combinatorial characterisation of 2-isomorphic graphs. See [16] for more details.

3 Linear Matroid Isomorphism

In this section we present some observations and results on LMI. 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.

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. We first observe that using the arguments similar to that of [11] one can show,

Theorem 1. LMI $\in \Sigma_2^p$. *In addition,* LMI is Σ_2^{P}-hard $\implies \mathsf{PH} = \Sigma_3^{\mathsf{P}}$.

Using the results of [15] and noting that uniform matroids are representable, we have the following,

Proposition 1. 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 [6], 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 variants 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 [2] to prove $\text{LMI}_b \equiv_m^p$ GI.

Given a graph $X = (V, E)$ and a $k \in [3, d]$, where d is the minimum vertex degree of X, define a matroid $M = St_k(X)$ of rank k with the ground set as E as follows: every subset of $k-1$ edges is independent in M and every subset of E with k edges is independent if and only if they do not share a common vertex. Babai proved that $Aut(X) \cong Aut(St_k(X))$ and also gave a linear representation for $St_k(X)$ (Lemma 2.1 in [2]) for all k in the above range. By tightening Babai's result, we obtain the following theorem, (See [17] for more details.)

Theorem 2. *For any constant $b \geq 3$,* $\text{LMI}_b \equiv_m^p$ GI.

The above reduction ($\text{LMI}_b \leq_m^p$ GI) works even if the matroids are not linear, provided they are given via an independent set oracle. This gives the following corollary.

Corollary 1. $\text{LMI}_b \equiv_m^p \text{MI}_b \equiv_m^p$ GI.

4 Isomorphism Problem of Coloured Matroids

Vertex or edge colouring 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-colouring of a graph $X = (V, E)$ is a function $f : E \to \{1, \ldots, k\}$. Given two graphs $X_1 = (V_1, E_1, f_1)$ and $X_2 = (V_2, E_2, f_2)$ with edge colourings, the Coloured-GMI asks for an isomorphism which preserves the colours of the edges. Not surprisingly, we can prove the following.

Lemma 1. Coloured-GMI *is* AC^0 *many-one reducible to* GMI.

Using linear algebraic constructions, which we defer to the full version due to shortage of space, we generalise the above construction to the case of linear matroid isomorphism. Coloured-LMI denotes the variant of LMI where the inputs are the linear matroids M_1 and M_2 along with colour functions $c_i : \{1, \ldots, m\} \to \mathbb{N}, i \in \{1, 2\}$. The problem is to test if there is an isomorphism between M_1 and M_2 which preserves the colours of the column indices. We have,

Lemma 2. Coloured-LMI *is* AC^0 *many-one reducible to* LMI.

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.

Whitney gave an exact characterisation of when two graphs are 2-isomorphic, in terms of three operations; twisting, cleaving and identification. (see [16].) 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 whose edges connect 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. Moreover, despite the fact that the twist operations need not commute, Truemper [20] proved : for any two 2-connected 2-isomorphic graphs X and Y (on n vertices), 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 which 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 $X_1' \cong X_2'$. This gives an upper bound of $\exists.\mathrm{GI} \subseteq \mathsf{NP}$. Thus we have,

Proposition 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 GMI cannot be NP-hard unless PH collapses. This deterministic reduction, stated in the theorem 3 below, is the main result of the paper.

Theorem 3. GMI $\le_T^p$ GI

Let us first look into the case of 3-connected graphs. 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. Whitney ([21]) proved the following equivalence,

Theorem 4 ([21]). *X_1 and X_2 be 3-connected graphs, $X_1 \cong_2 X_2 \iff X_1 \cong X_2$.*

Before giving a formal proof of Theorem 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 the given graph is 2-connected.

In [7], 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, [7] 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 4), and then colour the tree nodes with the corresponding isomorphism class, and then compute a coloured 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 colouring 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 preserves the colours 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 colouring, 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 coloured sub trees need not be isomorphic. This creates a simultaneity problem of colouring of the 3-connected components and the tree nodes and thus a second mind block.

We overcome this by colouring again using the code for coloured sub trees, and then finding the new isomorphism classes between the 3-connected components. This process is iterated till the colours stabilise 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 colouring scheme, at any point, does not distinguish between them.

Breaking into Tree of 3-connected components: We use the algorithm of Hopcroft and Tarjan [7] 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.

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. X is an *excisable* component w.r.t $\{a, b\}$ 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 $X \cup \{a, b\}$ plus a *virtual edge* joining (a, b). This operation may introduce multiple edges.

[2] Cunningham et al. [3] shows that any graphic matroid $M(X)$ is isomorphic to $M(X_1) \oplus M(X_2) \ldots \oplus M(X_k)/\{e_1, e_2, \ldots, e_k\}$, 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 colouring.

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 explanation, 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, [7] introduces another component corresponding to the virtual edges thus identifying all the virtual edges created in the same excising operation with each other.

Instead, we do a surgery on the original graph X and the graph G_X. We add an edge between all the *excised pairs* (excised while obtaining G_X) to get the graph X'. Notice that, following the same series of decomposition gives a new graph T_X which is the same as G_X except that the cliques are replaced by star centred at a newly introduced vertex (component) corresponding to the newly introduced excised edges in X'. The newly introduced edges form a 3-connected component themselves with one virtual edge corresponding to each edge of the clique they replace.

We list down the properties of the tree T_X for further reference. (1) For every node in $t \in T_X$, 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 each in the 3-connected components c_u and c_v. We call these virtual edges as the *twin edges* of each other. (3) For any given graph X, T_X is unique up to isomorphism (since G_X is unique [7]). In addition, T_X can be obtained from G_X in polynomial time.

The following claim states that (we omit the proof) this surgery in the graphs does not affect the existence of 2-isomorphisms.

Claim. $X_1 \cong_2 X_2 \iff X'_1 \cong_2 X'_2$.

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 tree of 3-connected components T_1 and T_2.
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.

```
1. Initialise $T'_1 = T_1$, $T'_2 = T_2$.
2. REPEAT
   (a) Set $T_1 = T'_1$, $T_2 = T'_2$.
   (b) For each edge $e = (u, v) \in T_i$, $i \in \{1, 2\}$:
```

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

```
      Let $T_i(e,u)$ and $T_i(e,v)$ be subtrees of $T_i$ obtained by deleting the edge $e$,
      containing $u$ and $v$ respectively.
      Colour virtual edges corresponding to the separating pairs in the components
      $c_u$ and $c_v$ with the set $\{\textsc{code}(T_i(e,u)), \textsc{code}(T_i(e,v))\}$. From now on, $c_t$ denotes
      the coloured 3-connected component corresponding to node $t \in T_1 \cup T_2$.
   (c) Let $S_1$ and $S_2$ be the set of coloured 3-connected components of $X_1'$ and $X_2'$
      and let $S = S_1 \cup S_2$. Using queries to GI (see Proposition 3) find out the
      isomorphism classes in $S$. Let $C_1, \ldots, C_q$ denote the isomorphism classes.
   (d) Colour each node $t \in T_i$, $i \in \{1,2\}$, with colour $\ell$ if $c_t \in C_\ell$. (This gives two
      coloured trees $T_1'$ and $T_2'$.)
   UNTIL ($\textsc{code}(T_i) \neq \textsc{code}(T_i')$, $\forall i \in \{1,2\}$)
3. Check if $T_1' \cong T_2'$ preserving the colours. Answer YES if $T_1' \cong T_2'$, and NO otherwise.
```

First we prove that the algorithm terminates in linear number of iterations of the repeat-until loop. Let q_i denote the number of isomorphism classes of the set of the coloured 3-connected components after the i^{th} 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 coloured tree T_1' is different from T_1. This can happen only when the colour 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.

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

Lemma 3. $X_1' \cong_2 X_2'. \iff T_1' \cong T_2'.$

Proof. We give a proof sketch here.
($\Rightarrow$) This dirction is easy and we omit the proof.
($\Leftarrow$) First, we recall some definitions needed in the proof. A *centre* of a tree T is defined as a vertex v such that $\max_{u \in T} d(u,v)$ is minimised at v, where $d(u,v)$ is the number of edges in the unique path from u to v. It is known that every tree T has a centre consisting of a single vertex or a pair of adjacent vertices. The minimum achieved at the centre is called the *height* of the tree, denoted by $ht(T)$.

Claim. Let ψ be a colour preserving isomorphism between T_1' and T_2', and χ_t is 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 centre) is computed with respect to the underlying tree ignoring colours 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 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 centre(s) of T_1 to that of T_2. We consider two cases.

In the first case, T'_1 and T'_2 have unique centres α and β. It is clear that $\psi(\alpha) = \beta$. Let c_1 and c_2 be the corresponding coloured (as in step 2b) 3-connected components. Therefore, there is a colour 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 neighbours 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 coloured 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 colour as f_i. That is, $\{\textsc{code}(T_1(e_i, \alpha)), \textsc{code}(T_1(e_i, v_i))\} = \{\textsc{code}(T_2(e'_i, \beta)), \textsc{code}(T_2(e'_i, v'_i)))\}$. Since α and β are the centres of T'_1 and T'_2, it must be the case that in the above set equality, $\textsc{code}(T_1(e_i, v_i)) = \textsc{code}(T_2(e'_i, v'_i))$. From the termination condition of the algorithm, this implies that $\textsc{code}(T'_1(e_i, v_i)) = \textsc{code}(T'_2(e'_i, v'_i))$. 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 set of all 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 coloured 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 colouring, this implies that there is a subtree 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 centre 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 centres (α_1, α_2) and (β_1, β_2) respectively. An essentially similar argument works in this case too. ■

This completes the proof of correctness of the algorithm (Lemma 3). ■

To complete the proof of Theorem 3, we need the following proposition:

Proposition 3. Coloured-GMI *for 3-connected graphs reduces to* GI

Observe that the above construction does not use non-planar gadgets. It is known that isomorphism testing for planar 3-connected graphs can be done in linear time [7] (in fact in L [4]) we get the following.

Corollary 2. PMI $\in \mathsf{P}$

Now we give a polynomial time many-one reduction from MI_b to GMI.

Theorem 5. $\mathrm{MI}_b \leq^p_m$ GMI.

Combining Corollary 1, Theorem 3 and Theorem 5 we have,

Theorem 6. GI $\equiv^p_T$ GMI $\equiv^p_T$ $\mathrm{MI}_b \equiv^p_T \mathrm{LMI}_b$

6 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 [13].) In this section we turn our attention to 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 $\forall C \subseteq S,\ C \in \mathcal{C} \iff \phi(C) \in \mathcal{C}$. $Aut(M)$ denotes the group of automorphisms of the matroid M. When the matroid is graphic we denote by $Aut(X)$ and $Aut(M_X)$ 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 $\pi \in Aut(M_X)$ 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 $Aut(X)$ can be done easily by testing whether π preserves all the edges. We provide an efficient test for this problem, i.e.,

Theorem 7. *Given any $\pi \in S_m$, testing if $\pi \in Aut(M_X)$ can be done in* P.

To prove the above theorem, we use the notion of a cycle bases 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 basis of the graph X.

Lemma 4. *Let $\pi \in S_n$, $\exists \mathcal{B} \in \mathscr{B} : \pi(\mathcal{B}) \in \mathscr{B} \implies \forall \mathcal{B} \in \mathscr{B} : \pi(\mathcal{B}) \in \mathscr{B}$*

Lemma 5. *Let $\pi \in S_m$, and let $\mathcal{B} \in \mathscr{B}$, then $\pi \in Aut(M_X) \iff \pi(\mathcal{B}) \in \mathscr{B}$.*

Using Lemmas 4 and 5 it follows that, given a permutation π, to test if $\pi \in Aut(M_X)$ 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, [8]). Now it suffices to show:

Lemma 6. *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.*

Notice that similar arguments can also give another proof of Proposition 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 $Aut(M)$.*

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 colouring techniques from Section 4, we prove the following.

Theorem 8. LMI $\equiv_T^p$ LMA, *and* GMI $\equiv_T^p$ GMA.

References

1. V. Arvind and J. Torán. Isomorphism testing: Perspective and open problems. *Bulletin of the EATCS*, 86:66–84, 2005.
2. L. Babai. Vector Representable Matroids of Given Rank with Given Automorphism Group. *Discrete Math.*, 24:119–125, 1978.
3. W. H. Cunningham and J. Edmonds. A Combinatorial Decomposition Theory. *Canad. Jl. of Math.*, 17:734–765, 1980.
4. S. Datta, N. Limaye, and P. Nimbhorkar. 3-connected planar graph isomorphism is in log-space. In *FSTTCS*, 2008. To appear.
5. O. Goldreich. *Computational Complexity: A Conceptual Perspective.* Cambridge Univ. Press, 2008.
6. P. Hlinený. Some hard problems on matroid spikes. *Theory of Computing Sys.*, 41(3):551–562, 2007.
7. J. Hopcroft and R. Tarjan. Dividing a graph into triconnected components. *SIAM Jl. of Comp.*, 2(3):135–158, 1973.
8. J. D. Horton. A Poly-time Algorithm to Find the Shortest Cycle Basis of a graph. *SIAM Jl. of Comp.*, 16(2):358–366, 1987.
9. B. Jenner, J. Köbler, P. McKenzie, and J. Torán. Completeness results for graph isomorphism. *J. Comput. Syst. Sci.*, 66(3):549–566, 2003.
10. J. Köbler. On graph isomorphism for restricted graph classes. In *CiE*, pages 241–256, 2006.
11. J. Köbler, U. Schöning, and J. Torán. *The Graph Isomorphism Problem: its Structural Complexity.* Birkhauser Verlag, 1993.
12. E. Luks. *Permutation groups and polynomial-time computation*, volume 11 of *DIMACS*, pages 139–175. 1993.
13. E. M. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. In *FOCS*, pages 42–49, 1980.
14. D. Mayhew. Matroid Complexity and Nonsuccinct Descriptions. *SIAM Jl.D.Math.*, 22(2):455, 2008.
15. J. Oxley and D. Welsh. Chromatic, flow and reliability polynomials: The complexity of their coefficients. *Comb. Prob. and Comp.*, 11:403–426, 2002.
16. J. G. Oxley. *Matroid theory.* Oxford University Press, New York, 1992.
17. R. Rao B.V. and J. Sarma M.N. On the complexity of matroid isomorphism problem. http://arxiv.org/abs/cs.CC/0811.3859, 24Nov. 2008.
18. U. Schöning. Probablistic Complexity Classes and Lowness. *JCSS*, 39:84–100, 1999.
19. J. Toran. On the Hardness of Graph Isomorphism. *SIAM Jl. of Comp.*, 33(5):1093–1108, 2004.
20. K. Truemper. On Whitney's 2-isomorphism theorem for graphs. *Jl. of Graph Th.*, pages 43–49, 1980.
21. H. Whitney. Congruent graphs and connectivity of graphs. *American Journal of Mathematics*, 54(1):150–168, 1932.
22. H. Whitney. 2-isomorphic graphs. *American Journal of Mathematics*, 55:245–254, 1933.