Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 9 (2009)

Checking Equality of Matroid Linear Representations and the Cycle Matching Problem*

T.C. Vijayaraghavan
Chennai Mathematical Institute
SIPCOT IT Park, Padur PO
Siruseri 603103, India.
email:vijay@cmi.ac.in

3rd August 2009

Abstract

Given linear representations M_1 and M_2 of matroids over a field $\mathbb{F}$, we consider the problem (denoted by ECLR[$\mathbb{F}$]), of checking if M_1 and M_2 represent the same matroid over $\mathbb{F}$. We show that when $\mathbb{F} = \mathbb{Z}_2$, ECLR[$\mathbb{Z}_2$] is complete for $\oplus$L under logspace many-one reductions. When $\mathbb{F} = \mathbb{Q}$, given linear representations $M_1, M_2 \in \mathbb{Q}^{m \times n}$ as input, any set of indexes $X \subseteq \{1, \ldots, n\}$ such that columns corresponding to these indexes in X are linearly dependent in one linear representation but are linearly independent in the other linear representation is a witness that M_1 and M_2 represent different matroids over $\mathbb{Q}$. We show that the decision and the search version of this problem are polynomial time equivalent.

We consider the CYCLE MATCHING problem of checking if for a pair of undirected graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ given as input with $n = |V_1| = |V_2|$, we check if the set of vertices having indexes in $X \subseteq \{1, \ldots, n\}$ form a cycle in G_1 if and only if the corresponding set of vertices having indexes in $X \subseteq \{1, \ldots, n\}$ form a cycle in G_2 for all $X \subseteq \{1, \ldots, n\}$. We show that CYCLE MATCHING is complete for L. Also the problem of counting the number of $X \subseteq \{1, \ldots, n\}$ such that vertices with indexes in X form a cycle in one of the input graphs but not in the other is shown to be #P-complete.

1 Matroid Linear Representations and ECLR

Matroids are combinatorial objects that generalize the notions of linear independence and dependence of vectors in a vector space. The study of computational problems related to matroids and providing efficient algorithms for solving them is an important branch of combinatorial optimization [8]. Matroids are also known to generalize connectivity properties between vertices in a graph. As a result several problems on graphs have been re-cast into problems on matroids to seek efficient algorithms for solving them. A classic example is the maximum matching problem for bipartite graphs which can be shown to reduce to the problem of intersection of two linearly representable matroids [8, Section 12.5]. Efficient algorithms for the matroid intersection problem (not necessarily linearly representable matroids) are known and in fact the first polynomial time algorithm for this problem was given by Edmonds in [4, 5].

As one of the main results in this paper, we consider a problem on linearly representable matroids denoted by ECLR. Given two linear representations over a field $\mathbb{F}$, the ECLR[$\mathbb{F}$]

*This article is the Revision 1 of the extended abstract available as ECCC Report TR 09-009. The previous version was submitted to the ISAAC 2008 and STACS 2009 conferences and this revised version is based on the comments of the referees of these conferences.

ISSN 1433-8092

problem is to check if the input linear representations represent the same matroid. Before defining the problem formally, we introduce necessary definitions and terminology on matroids. We refer to [7, Chapter 1] for details and clarifications.

2 Preliminaries

2.1 Definitions

Definition 2.1. *A* matroid M *is a pair* $(S, \mathcal{I})$*, where* S *is a finite set and* $\mathcal{I}$ *is a collection of subsets of* S *such that:*

1. *the empty set* $\emptyset$ *is in* $\mathcal{I}$*.*
2. *if* $X \in \mathcal{I}$ *and* $Y \subseteq X$ *then* $Y \in \mathcal{I}$*.*
3. *if* $X, Y \in \mathcal{I}$ *with* $|X| = |Y| + 1$*, then there exists* $x \in X - Y$ *such that* $Y \cup \{x\} \in \mathcal{I}$*. This condition is also called the* independence augmentation axiom*.*

We say that a subset X *of* S *is* independent *if* $X \in \mathcal{I}$*. Any subset of* S *not in* $\mathcal{I}$ *is said to be a* dependent set*.*

Definition 2.2. *Let* $M = (S, \mathcal{I})$ *be a matroid, and let* $X \in \mathcal{I}$*. We say that* X *is a base if* $X \not\subseteq Y$*, where* $Y \in \mathcal{I}$ *with* $X \neq Y$*. In other words, a* base *is a maximal independent set of* M*.*

Definition 2.3. *Let* $M = (S, \mathcal{I})$ *be a matroid, and let* $X \subseteq S$*. We say that* X *is a circuit if* $X \notin \mathcal{I}$*, but every proper subset* Y *of* X *is in* $\mathcal{I}$*. In other words, a* circuit *is a minimal dependent set of* M*.*

Definition 2.4. *Let* $M = (S, \mathcal{I})$ *be a matroid and* $\mathbb{F}$ *be a field. We say that* M *is linearly representable over* $\mathbb{F}$*, if for some positive integer* r *there exists a matrix* $A \in \mathbb{F}^{r \times |S|}$ *such that any set of columns in* A *is linearly independent over* $\mathbb{F}$ *if and only if the subset of* S *corresponding to this set of columns form an independent set in* $\mathcal{I}$*.*

Definition 2.5. [1] *We say that a function* $f : \{0,1\}^* \to \mathbb{Z}^+$ *is in* #L *if there exists a* NL *machine* M *such that* $f(x)$ *is the number of accepting computation paths of* M *on input* x*.*

Definition 2.6. [2] *We say that a language* $L \in \oplus$L *if there exists* $f \in$ #L *such that* $x \in L$ *if and only if* $f(x) \equiv 1 (mod\ 2)$*.*

Note 1. *We follow the Ruzzo-Simon-Tompa oracle access mechanism* [1] *in allowing non-deterministic Turing machines to query oracles. According to this mechanism a non-deterministic Turing machine is allowed to write its queries in the oracle tape in a deterministic manner only.*

2.2 Basic results

We now recall some basic results from [7] required to prove our results.

Lemma 2.7. [7, Lemma 1.2.2] *If* X_1 *and* X_2 *are two distinct bases of a matroid* $M = (S, \mathcal{I})$ *then there exists* $x, y \in S$ *such that* $x \in X_1 - X_2$ *and* $y \in X_2 - X_1$ *such that* $(X_1 - x) \cup \{y\}$ *is also a base of* M*. This result is also called the* base exchange axiom *of a matroid.*

Theorem 2.8. [7, Theorem 1.2.3] *Let* S *be a set and let* $\mathcal{B}$ *be a collection of subsets of* S *such that* $\mathcal{B}$ *is non-empty and elements of* $\mathcal{B}$ *satisfy the base exchange axiom stated in Lemma 2.7. Now if* $\mathcal{I}$ *is the collection of all subsets of* B *for all* $B \in \mathcal{B}$ *then* $M = (S, \mathcal{I})$ *is a matroid.*

As a consequence of the above theorem we obtain the following result.

Corollary 2.9. *Let $M_1 = (S, \mathcal{I}_1)$ and $M_2 = (S, \mathcal{I}_2)$ be two matroids such that any subset $B \subseteq S$ is a base in M_1 if and only if B is a base in M_2. Then both M_1 and M_2 are the same matroid.*

3 ECLR: A problem on matroid linear representations

If M is a matroid linearly representable over a field $\mathbb{F}$ then its linear representation need not be unique. For example the matroid represented by the $n \times n$ identity matrix I_n is the same as the matroid represented by any other $n \times n$ non-singular matrix over a field $\mathbb{F}$. This naturally raises the question of whether there is an efficient algorithm that can decide if two input linear representations over a field $\mathbb{F}$ represent the same matroid. We denote this problem as ECLR[$\mathbb{F}$] (surprisingly it seems that this problem has not been investigated so far).

> Equality Checking for Linear Representations (ECLR): Given two linear representations M_1, M_2 over a field $\mathbb{F}$, the ECLR[$\mathbb{F}$] problem is to check if M_1 and M_2 represent the same matroid.

Note 2. *While we deal with checking if two linear representations represent the same matroid, note that a notion of* equivalence *of two linear representations is also known* [7, Section 6.3]. *We wish to emphasize that in our* ECLR *problem we check if the matroids represented by the input linear representations are the same and not if they are* equivalent *as defined in* [7].

Let $M = (S, \mathcal{I})$ be a matroid such that every element of S is in at least one independent set in $\mathcal{I}$. In Theorem 3.1 to be proved, we show that if M is linearly representable by a matrix $A \in \mathbb{F}^{m \times n}$ over a field $\mathbb{F}$ then M is also linearly representable by a submatrix $B \in \mathbb{F}^{r \times n}$ of A such that $r = \text{rank } (B) = \text{rank } (A)$.

Theorem 3.1. *Let $\mathbb{F}$ be a field and let $A \in \mathbb{F}^{m \times n}$ be a linear representation of the matroid $M = (S, \mathcal{I})$ over $\mathbb{F}$. We assume that every element of S is in at least one independent set in $\mathcal{I}$. Also if $r = \text{rank } (A)$ and $B \in \mathbb{F}^{r \times n}$ is a submatrix of A such that $r = \text{rank } (B)$ then B is also a linear representation of the matroid M.*

Proof: We have assumed that every element of S is in at least one independent set in $\mathcal{I}$ of the matroid M. Therefore every column of the linear representation A has at least one non-zero element from $\mathbb{F}$. When $r = \text{rank } (A) = n$ the result immediately follows since B is a $r \times r$ non-singular submatrix of A over $\mathbb{F}$. We therefore assume $r < n = |S|$.

Claim **1.** *The matrix $B \in \mathbb{F}^{r \times n}$ does not contain any column containing only zeroes.*

Proof of Claim 1. Assume the claim is not true and that the i^{th} column of B, denoted by B_i contains only zeroes for some $1 \leq i \leq n$. However since B is a submatrix of A, the corresponding column in A, denoted by A_i is non-zero. We have also assumed rank $(B) = r$ and so there exists a non-singular $r \times r$ submatrix B' of B. Using B' and the column A_i we can then obtain a non-singular $(r+1) \times (r+1)$ submatrix of A. But this contradicts our assumption that $r = \text{rank } (A)$. This shows that the matrix B does not contain any column that contains only zeroes which proves our claim.

We use induction on r to prove our result. For the case when $r = 1$ it is easy to observe that any row of A is of the form $(\alpha, \ldots, \alpha)$ where $\alpha \in \mathbb{F}$ and there exists at least one row for which $\alpha \neq 0$. Clearly B is then one such non-zero row of A and so the result is true for the case when $r = 1$.

Inductively assume the result to be true for all matroids having linear representations $A \in \mathbb{F}^{m\times n}$ such that rank $(A) \leq (r-1)$, where $r \geq 2$.

Let $A \in \mathbb{F}^{m\times n}$ be a linear representation of a matroid $M = (S, \mathcal{I})$ over $\mathbb{F}$ such that $r =$ rank (A). Let B be a submatrix of A with rank $(B) = r$ as mentioned in the theorem statement. It follows from claim 1 that every column of B contains at least one non-zero element in $\mathbb{F}$. Now let $B' \in \mathbb{F}^{r\times(n-1)}$ be a submatrix of B obtained by removing the i^{th} column of B for some $1 \leq i \leq n$.

- when rank $(B') = (r-1)$: If $A' \in \mathbb{F}^{m\times(n-1)}$ is the submatrix of A obtained by removing the i^{th} column of A then since rank $(B) = r$ and since B' is a submatrix of A', we get rank (A') = rank $(B') = (r-1)$. It now follows from the inductive hypothesis that the matroid represented by A' and the matroid represented by B' are the same. Let this matroid be M'. Moreover any base of the matroid M represented by A is a base of M' augmented with the i^{th} element of S (corresponding to the i^{th} column of A) that we have removed in order to obtain the linear representation A'. Also M does not contain any other base. However since rank $(B) = r$ and rank $(B') = (r-1)$ it follows that any base of the matroid represented by B is a base of M' augmented with the i^{th} element of S (corresponding to the i^{th} column of B). Also the matroid represented by B does not contain any other base. Now using Corollary 2.9 it follows that the matroid represented by B is the matroid represented by A which is M.

- when rank $(B') = r$: We use induction on $n = |S|$ to prove the result. For the base case when $|S| = 1$, we have rank $(A) = 1$ for which the result trivially holds true. When $n = 2$ we either have rank $(A) = 1$ or rank $(A) = 2$ and correspondingly either $r = 1$ or $r = 2$. It is easy to see that the result is true for these cases also.

  Inductively assume the result to be true for all matroids $M = (S, \mathcal{I})$ such that if A is a linear representation of M then rank $(A) = r$ and $|S| = (n-1)$ where $n \geq 3$.

  Once again let $M = (S, \mathcal{I})$ be a matroid as in the theorem statement having a linear representation $A \in \mathbb{F}^{m\times n}$ such that rank $(A) = r$ and $|S| = n$. Now let $A' \in \mathbb{F}^{m\times(n-1)}$ be the submatrix of A obtained by removing the i^{th} column of A. Let the matroid represented by A' be M'. Clearly B' is a submatrix of A' and so we get rank (A') = rank $(B') = r$. Also it follows from the inductive hypothesis on $|S|$ that B' also linearly represents the matroid M'. However since rank (A) = rank $(A') = r$, it follows that any base of the matroid M is either a base of M' or it can be obtained from a base of A' by applying the base exchange axiom stated in Lemma 2.7 on the i^{th} element of S (corresponding to the i^{th} column of A) which we have removed in order to obtain the linear representation A'. Also there are no other bases in M. But since rank $(B) = r$ and since B' is a submatrix of B with rank $(B') = r$ this observation regarding the bases is true for the matroid linearly represented by B also. Moreover A' and B' linearly represent the same matroid M' and since A linearly represents matroid M, using Corollary 2.9 it follows that B is also a linear representation of the matroid M. This completes the proof of the theorem.

3.1 ECLR[$\mathbb{Z}_2$] is $\oplus$L-complete

We consider the ECLR problem when $\mathbb{F} = \mathbb{Z}_2$. Using results on linear algebraic subroutines such as computing a maximal set of linearly independent vectors from a given set of vectors over $\mathbb{Z}_2$ and a solving system of linear equations over $\mathbb{Z}_2$ from [2] and properties of $\oplus$L from [6] we show that ECLR[$\mathbb{Z}_2$] is $\oplus$L-complete under logspace many-one reductions. Computing the number of spanning trees modulo 2 of an arbitrary undirected graph has been recently shown to be complete for $\oplus$L under logspace many-one reductions in [3]. Also in [3] computing the

permanent of an integer matrix modulo 2^k for a fixed integer $k > 0$ is also shown to be complete for $\oplus$L under logspace many-one reductions. The completeness result for ECLR[$\mathbb{Z}_2$] that we obtain in Theorem 3.3 is a new addition to this list of problems complete for $\oplus$L.

We first show an equivalence relation that characterizes when two linear representations over $\mathbb{Z}_2$ represent the same matroid.

Theorem 3.2. *Given matrices $M_1, M_2 \in \mathbb{Z}_2^{m\times n}$ such that rank (M_1) = rank $(M_2) = m$, we say that M_1 and M_2 are related (denoted by $M_1 \sim M_2$) if there exists an invertible matrix $X \in \mathbb{Z}_2^{m\times m}$ such that $XM_1 = M_2$. Then*

1. *$\sim$ is an equivalence relation, and*
2. *M_1 and M_2 represent the same matroid M over $\mathbb{Z}_2$ if and only if $M_1 \sim M_2$.*

Proof: Let $M_1, M_2 \in \mathbb{Z}_2^{m\times n}$ such that rank (M_1) = rank $(M_2) = m$.

1. It is easy to note that the above relation is reflexive: for any matrix $M \in \mathbb{Z}_2^{m\times n}$ we can take X to be I_m, the $m \times m$ identity matrix. Also $\sim$ is symmetric, since $XM_1 = M_2$ for an invertible matrix X if and only if $M_1 = X^{-1}M_2$. Transitivity follows since given M_1, M_2 and M_3, where $M_3 \in \mathbb{Z}_2^{m\times n}$ with rank $(M_3) = m$, if $M_1 \sim M_2$ and $M_2 \sim M_3$ then there exists invertible matrices $X_1, X_2 \in \mathbb{Z}_2^{m\times m}$ such that $X_1M_1 = M_2$ and $X_2M_2 = M_3$. Now let $X_3 = X_2X_1$. Then $X_3 \in \mathbb{Z}_2^{m\times m}$ and is invertible. Moreover $X_3M_1 = M_3$ which implies $M_1 \sim M_3$.

2. Let $M_1, M_2 \in \mathbb{Z}_2^{m\times n}$ be linear representations having rank (M_1) = rank $(M_2) = m$ representing the same matroid M. We can identify the columns corresponding to the lexicographically least base containing m elements of the matroid M from these matrices. Let $Y_1, Y_2 \in \mathbb{Z}_2^{m\times m}$ be submatrices of M_1 and M_2 respectively corresponding to this base. Clearly Y_1 and Y_2 are invertible. Thus there exists an invertible matrix $X \in \mathbb{Z}_2^{m\times m}$ such that $XY_1 = Y_2$. More precisely we have $X = Y_2Y_1^{-1}$. Now any column in M_1 can be written as a $\mathbb{Z}_2$-linear combination of columns in Y_1 in a unique way. Also given any set of linearly independent vectors over $\mathbb{Z}_2$, there is exactly one vector in their $\mathbb{Z}_2$ span when all the coefficients of this set of vectors are non-zero. We have also assumed M_1 and M_2 represent the same matroid. As a result any set of columns in M_1 form a circuit if and only if the corresponding set of columns in M_2 also form a circuit. Therefore from these observations it follows that $XM_1 = M_2$.

   Conversely if for $M_1, M_2 \in \mathbb{Z}_2^{m\times n}$ we have $M_1 \sim M_2$, then M_1 and M_2 represent the same matroid. This is a consequence of the fact that the set of vectors obtained from the product of an invertible matrix with a set of linearly independent vectors is also linearly independent.

Remark 1. *It follows from Theorem 3.2 that the number of equivalence classes under the relation $\sim$ between the linear representations of a matroid M over $\mathbb{Z}_2$ is only one. This observation about the number of equivalence classes is true only for $\mathbb{Z}_2$. This is important since the problem of deciding* ECLR[$\mathbb{Z}_2$] *reduces to solving a system of linear equations over $\mathbb{Z}_2$. It is easy to see that when considering linear representations of a matroid M over any other field (even with characteristic* 2*) number of equivalence classes under the relation $\sim$ defined in Theorem 3.2 is more than one. We do not know if there is a polynomial time algorithm to solve the* ECLR *problem when the input linear representations are over a field that is not $\mathbb{Z}_2$.*

Theorem 3.3. ECLR[$\mathbb{Z}_2$] *is* $\oplus$L*-complete.*

Proof: We are given linear representations $M_1, M_2 \in \mathbb{Z}_2^{m\times n}$ as input. In [2] it has been shown that computing a maximal set of linearly independent vectors over $\mathbb{Z}_2$ from a given set of vectors over $\mathbb{Z}_2$ is complete for $\oplus$L under logspace many-one reductions. As a result we can compute rank (M_1) and rank (M_2) in $\oplus$L and check if they are equal. If rank $(M_1) \neq$ rank (M_2) then we output M_1 and M_2 do not represent the same matroid and stop.

Otherwise assume that $r =$ rank $(M_1) =$ rank (M_2). Now once again it follows from [2] that we can obtain a submatrix $M_1' \in \mathbb{Z}_2^{r\times n}$ of M_1 and a submatrix $M_2' \in \mathbb{Z}_2^{r\times n}$ of M_2 such that rank (M_1') = rank $(M_2)' = r$ in $\oplus$L. Since the linear representations are over $\mathbb{Z}_2$ using Theorem 3.1 it follows that the matroid represented by M_i' is the same as the matroid represented by M_i, for $i = 1, 2$ respectively.

It follows from Theorem 3.2 that M_1 and M_2 represent the same matroid if and only if there exists a non-singular $X \in \mathbb{Z}_2^{m\times m}$ such that $XM_1' = M_2'(\text{mod } 2)$. Essentially this step is to solve for a system of linear equations over $\mathbb{Z}_2$ which is also shown to be complete for $\oplus$L in [2].

It is easy to see that a logspace machine with access to two levels of $\oplus$L oracle can retrieve the entries of M_1' and M_2' and also check for a solution to the system $XM_1' = M_2'$ over $\mathbb{Z}_2$. Now using the observations on the $\oplus$L upper bound we have made above and the result of [6] it follows that ECLR$[\mathbb{Z}_2] \in \oplus$L. This shows the $\oplus$L upper bound.

Hardness for $\oplus$L follows from the following observation: given a matrix $M \in \mathbb{Z}_2^{n\times n}$ checking if M is non-singular over $\mathbb{Z}_2$ is hard for $\oplus$L. Therefore given a matrix M as input we output M and I_n, where I_n is the $n \times n$ identity matrix. Clearly $(M, I_n) \in$ ECLR$[\mathbb{Z}_2]$ if and only if M is non-singular which shows the hardness for $\oplus$L and hence the proof is complete.

3.2 An equivalence between search and decision for ECLR$[\mathbb{Q}]$

In this section we consider the ECLR problem when the input linear representations are over $\mathbb{Q}$, the set of rationals. Let $M_1, M_2 \in \mathbb{Q}^{m\times n}$ be the input linear representations. Given a set of indexes, columns corresponding to which are linearly dependent in one linear representation but are linearly independent in the other linear representation is a witness that M_1 and M_2 represent different matroids over $\mathbb{Q}$. We show that the search and the decision version of this problem are polynomial time equivalent. More precisely, assume that there is a polynomial time algorithm that decides ECLR$[\mathbb{Q}]$ and that the input linear representations M_1 and M_2 represent different matroids. The polynomial time procedure described below outputs a set of indexes such that columns corresponding to these indexes form a circuit in M_i but the corresponding columns do not form a circuit in M_j using ECLR(M_1, M_2) as an oracle, where $1 \leq i, j \leq 2$ and $i \neq j$.

Given linear representations $M_1, M_2 \in \mathbb{Q}^{m\times n}$, let ECLR$(M_1, M_2)$ be the function that outputs 1 if the matroid represented by M_1 and M_2 are the same, and it outputs 0 otherwise. Also if $X \subseteq S = \{1, \ldots, n\}$ and $j \in \{1, 2\}$, let $M_j^{(X)}$ denote the matrix obtained from M_j by retaining columns whose indexes correspond to integers in X. We denote the matroid so obtained from M_j by $(S, \mathcal{I}_j^{(X)})$, where $\mathcal{I}_j^{(X)} = \{X \cap I | \text{for all } I \in \mathcal{I}_j\}$. We start by assuming that M_1 and M_2 represent different matroids.

Let $i = 1$, $X = \{1, \ldots, i\}$, and $Y = \emptyset$. We query the ECLR$(M_1^{(X)}, M_2^{(X)})$ oracle for increasing values of i until the oracle outputs 0 for the smallest $1 \leq i \leq n$. Once we obtain this element $i \in S$ we re-initialize $Y = Y \cup \{i\}$. If the columns corresponding to elements in Y form a circuit in M_k but are linearly independent in M_l, where $1 \leq k, l \leq 2$ with $k \neq l$, we output Y and stop.

Otherwise if the set Y does not witness that the input linear representations represent different matroids then we find the smallest $j \in \{1, \ldots, i\} - Y$ for which the ECLR$(M_1^{(X\cup Y)}, M_2^{(X\cup Y)})$ oracle outputs 0, where $X = \{1, \ldots, j\} - Y$. Having obtained this $j \in S$ we once again re-

initialize $Y = Y \cup \{j\}$ and check if the columns corresponding to elements in Y form a circuit in M_k but are linearly independent in M_l, where $1 \leq k, l \leq 2$ with $k \neq l$. We iterate this step until we obtain the desired set $Y \subseteq \{1, \ldots, i\} \subseteq S$ that witnesses that the input linear representations represent different matroids over $\mathbb{Q}$.

The steps given above involve retaining some set of columns of the given input matrices and querying the ECLR oracle. Clearly these steps are polynomial time computable and hence the claim follows.

4 Cycle Matching Problem

The Cycle Matching Problem is defined as follows.

> Cycle Matching Problem (CYCLE MATCHING): Given a pair of undirected graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ as input with $n = |V_1| = |V_2|$, we check if the set of vertices having indexes in $X \subseteq \{1, \ldots, n\}$ form a cycle in G_1 if and only if the corresponding set of vertices having indexes in $X \subseteq \{1, \ldots, n\}$ form a cycle in G_2 for all $X \subseteq \{1, \ldots, n\}$.

It is not hard to view the CYCLE MATCHING as the graph theoretic analog of the ECLR problem. This follows from the fact that we can associate a linearly representable matroid over any given field $\mathbb{F}$ to a given simple undirected graph. The matroid so obtained is called a *cycle matroid* or a *graphic matroid* [7, pages 11 and 12]. Here the edges of the graph are the underlying elements and a set of edges form a cycle if and only if the corresponding columns of the linear representation form a minimal linearly dependent set of vectors over $\mathbb{F}$. In fact the incidence matrix of a simple undirected graph G is itself a linear representation for the cycle matroid of G over $\mathbb{Z}_2$. Thus it seems natural to expect a $\oplus$L upper bound for CYCLE MATCHING. However as one of the main results we show that the problem is in fact complete for L and the proof uses only elementary properties of the input graph.

Definition 4.1. *A cut-edge of a graph G is an edge whose deletion from the graph increases the number of components in G.*

Theorem 4.2. *[10, Theorem 1.2.14] Given a graph G, an edge is a cut-edge if and only if it does not belongs to any cycle in G.*

Our algorithm for the CYCLE MATCHING problem depends on identifying if the edges of the input graphs are cut-edges. We use the logspace undirected *st*-connectivity algorithm of [9] to identify such cut-edges and hence decide CYCLE MATCHING.

Lemma 4.3. *Given a graph G and an edge e in G, there is a logspace algorithm to check if e is a cut-edge in G.*

Proof: Let $e = (i, j) \in E(G)$. It follows from Theorem 4.2, that e is a cut-edge if and only if there is no path from vertex i to vertex j in the graph $G - \{e\}$. Now we can easily obtain $G - \{e\}$ from G in logspace. We then use the undirected *st*-connectivity algorithm of [9] to check if there is path from i to j in $G - \{e\}$ and hence output if e is a cut-edge.

Theorem 4.4. CYCLE MATCHING *is in* L.

Proof: Given a graph G, the following procedure obtains the subgraph of G induced by edges that are not cut-edges in G.

CUT-EDGE FREE SUBGRAPH(G)
for (each $e \in E(G)$)
  if (e is not a cut-edge) **then**
    output e.

It follows from Theorem 4.2 that an edge e in a graph G is a cut-edge if and only if e is not in any cycle in G. Let H be the subgraph of G induced by edges output by CUT-EDGE FREE SUBGRAPH(G). It is then clear that none of the cut-edges in G are in H. Moreover any isolated vertex formed in the process of excluding cut-edges in G is not in H. Thus any vertex or edge is contained in a cycle in G if and only if the same vertex or edge along with that cycle is in H also.

For the CYCLE MATCHING problem, we are given two graphs G_1 and G_2 as input. We obtain subgraphs H_1 of G_1 and H_2 of G_2 as mentioned above. From the observations made regarding H_1 and H_2, it is clear that $(G_1, G_2) \in$ CYCLE MATCHING if and only if $(H_1, H_2) \in$ CYCLE MATCHING. We also infer that $(H_1, H_2) \in$ CYCLE MATCHING if and only if the indexes of vertices in H_1 is the same as indexes of the vertices in H_2. This is a consequence of the fact that whenever a vertex with index k exists in H_i but not in H_j, there is a cycle containing the vertex with index k in H_i and hence in G_i also. However, there is no cycle containing the vertex with index k in H_j and hence not in G_j also, where $1 \leq i, j \leq 2$. This would then imply $(H_1, H_2) \notin$ CYCLE MATCHING or equivalently $(G_1, G_2) \notin$ CYCLE MATCHING. It is easy to note that the same argument carries over to edges of G_1 and G_2 that are not cut-edges. Therefore if $(H_1, H_2) \in$ CYCLE MATCHING then under the identity mapping between the indexes of vertices in H_1 and H_2, the adjacency relation between the vertices in H_1 should be the same as the adjacency relation between the vertices in H_2. In other words $(H_1, H_2) \in$ CYCLE MATCHING if and only if $V(H_1) = V(H_2)$ and $E(H_1) = E(H_2)$. Restating this, $(H_1, H_2) \in$ CYCLE MATCHING if and only if $H_1 = H_2$. It is clear from Lemma 4.3 that checking if any edge in G is a cut-edge or not is in L. To obtain H_1 and H_2, we make repeated calls to this subroutine in an iterative manner. Clearly this is logspace computable and hence we can also check if $H_1 = H_2$ in L.

Theorem 4.5. CYCLE MATCHING *is hard for* L

Proof: The *st*-connectivity problem for undirected graphs has been shown to be complete for L in [9]. Thus given a directed graph $G = (V, E)$ and vertices $s, t \in V$, we output $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, where $V_1 = V_2 = V$, $E_1 = E$, and $E_2 = E \cup \{(s, t)\}$. It is clear that if there does not exist any path between s and t in G, then the vertices corresponding to any set of indexes between $\{1, \ldots, |V|\}$ form a cycle in G_1 if and only if they form a cycle in G_2. Thus the pair (G_1, G_2) is an yes instance of the CYCLE MATCHING problem. On the contrary, if there is a path between s and t in G then there exists at least one set of vertices containing the edge (s, t) that form a cycle in G_2 but the corresponding set of vertices do not form a cycle in G_1. In this case the pair $(G_1, G_2) \notin$ CYCLE MATCHING. Given the input G we can construct G_1 and G_2 easily. The fact that L is closed under complement then completes the proof.

4.1 A hard counting problem from CYCLE MATCHING

Definition 4.6. *We say that a function $f : \{0,1\}^* \to \mathbb{Z}^+$ is in* #P *if there exists a* NP *machine M such that $f(x)$ is the number of accepting computation paths of M on input x.*

Definition 4.7. *We say that a function $f : \{0,1\}^* \to \mathbb{Z}^+$ is polynomial time many-one reducible to $g : \{0,1\}^* \to \mathbb{Z}^+$, if for any input $x \in \{0,1\}^*$ we can compute $f(x)$ when we are given $g(x)$ as input in time polynomial in $|x|$.*

In this section we show that given a pair of input graphs (G_1, G_2) with the same number of vertices, the problem of counting the number of $X \subseteq \{1, \ldots, n\}$ such that vertices corresponding to X form a cycle in one of the input graphs but not the other is #P-complete. We first give a proof sketch of the #P completeness of counting the number of cycles in an undirected graph. This problem is shown to reduce to our counting problem on CYCLE MATCHING.

Given a simple undirected graph $G = (V, E)$ the problem of counting the number of cycles in G is #P-complete under polynomial time many-one reductions. This is easy to observe. Given a graph $G = (V, E)$ we first replace each edge in G by a path of length $|V|^3$ to obtain a new graph $G_1 = (V_1, E_1)$. Then we replace each edge $(u, v) \in E_1$ of G_1 by two paths of length 2 each. More formally, we replace each $(u, v) \in E_1$ of G_1 by the four edges: $(u, x), (x, v), (u, y), (y, v)$. Let this new graph obtained after this replacement step from G_1 be denoted by $G_2 = (V_2, E_2)$. It is easy to observe that if there exists a Hamilton cycle in the input graph G, then correspondingly there exists a cycle of length $2|V|^3$ in G_2. Also any cycle in G_2 is of length at most $2|V|^3$. It can then be observed that the edges newly introduced in G to obtain G_2 create an exponential gap between the number of cycles of length $2|V|^3$ and the number of cycles of length strictly less than $2|V|^3$. As a consequence, each bit of the number of Hamilton cycles in G (which correspond to number of cycles of length $2|V|^3$ in G_2) occupies a distinct position in the number of cycles of the graph G_2. To be more precise, the leading polynomially many bits of the number of cycles in G_2 gives us the number of Hamilton cycles in G. Clearly in polynomial time we can retrieve the number of Hamilton cycles in G if the number of cycles of the graph G_2 is known. This shows that counting the number of cycles in an undirected graph is #P-hard under polynomial time many-one reductions. It is also easy to see that this counting problem is in #P and therefore we get completeness for #P under polynomial time many-one reductions.

We return back to the counting problem defined with respect to CYCLE MATCHING. Let G be the input graph on n vertices. Consider the graph G' formed by a path on n vertices. Clearly G' does not contain any cycle. Therefore the number of cycles in G is equal to the number of subsets of vertices that form a cycle in G, but the corresponding subset of vertices do not form a cycle in G'. Since the former is shown to be #P-complete we get the result.

Remark 2. *Assume that (G_1, G_2) is a no instance of the* CYCLE MATCHING *problem. Then counting the number of pairs of vertices that form an edge which is not a cut-edge in one of these input graphs but the corresponding pair of vertices either do not form an edge or it is a cut-edge in the other input graph can be easily seen to be in* #L. *However the problem of counting the number of subsets of vertices that form a cycle in one of the input graphs but the corresponding set of vertices do not form a cycle in the other input graph has been shown to be* #P*-complete. However any element of both these sets witnesses the fact that (G_1, G_2) is not in* CYCLE MATCHING.

Acknowledgments

I thank Samir Datta and Srikanth Srinivasan for useful discussions. I also thank the anonymous third referee of the STACS 2009 conference for correcting Remark 1 that the number of equivalence classes under the equivalence relation $\sim$ shown in Theorem 3.2 is more than one even for linear representations over a field $\mathbb{F}$ of characteristic 2 representing the same matroid M if $\mathbb{F}$ is not $\mathbb{Z}_2$.

References

[1] Eric Allender and Mitsunori Ogihara. Relationships among PL, #L and the Determinant. *RAIRO - Theoretical Informatics and Applications*, 30: 1-21, 1996.

[2] G. Buntrock, C. Damm, U. Hertrampf, and C. Meinel. Structure and Importance of Logspace-MOD Classes. *Mathematical Systems Theory*, 25(3): 223-237, 1992.

[3] Mark Braverman, Raghav Kulkarni, and Sambuddha Roy. Parity Problems in Planar Graphs. *Proceedings of the 22nd Annual IEEE Conference on Computational Complexity (CCC 2007)*, pages 222-235, 2007.

[4] J. Edmonds. Minimum partition of a matroid into independent subsets. *J. Res. National Bureau of Standards*, 69B: 67-72, 1965.

[5] J. Edmonds. Matroids and the Greedy Algorithm. *Mathematical Programming*, 1: 127-136, 1971.

[6] U. Hertrampf, S. Reith, and H. Vollmer. A Note on Closure Properties of Logspace-MOD Classes. *Information Processing Letters*, 75(3): 91-93, 2000.

[7] J. Oxley. *Matroid Theory.* Oxford University Press, 2006.

[8] C. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity* Prentice-Hall of India Private Limited, 2001.

[9] O. Reingold. Undirected connectivity in log-space. *Journal of the ACM*, 55(4): 1-24, 2008.

[10] D.B. West. *Introduction to Graph Theory, Second edition.* Prentice-Hall of India Private Limited, 2003.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de