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

Checking Equality of Matroid Linear Representations and the Cycle Matching Problem

(Extended Abstract)

T.C. Vijayaraghavan

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

Abstract. Given linear representations M_1 and M_2 of matroids over a field $\mathbb{F}$, we consider the problem (denoted by ECLR), of checking if M_1 and M_2 represent the same matroid. We show that when $\mathbb{F} = \mathbb{Z}_2$, ECLR$[\mathbb{Z}_2]$ is complete for $\oplus$L. Let $M_1, M_2 \in \mathbb{Q}^{m\times n}$ be two matroid linear representations given as input. Then any set of indexes, columns corresponding to which are linearly dependent in one representation but are linearly independent in another 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 $|V_1| = |V_2| = n$, whether any 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 form a cycle in G_2. 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 [6]. Matroids are also known to generalize connectivity properties between vertices in a graph. As a result several problems on graphs have be re-cast into problems on matroids to seek efficient algorithms for solving them. A classic example is the maximum matching problem on bipartite graphs, which can be shown to reduce to the problem of intersection of two linearly representable matroids [6, 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 [3].

As one of the main results in this paper, we consider a problem on linearly representable matroids denoted by ECLR. Given two linear representations, we

ISSN 1433-8092

want to check if they represent the same matroid. Before defining the problem formally, we introduce necessary definitions and terminology on matroids. We refer to [5, Chapter 1] for details and clarifications.

Definition 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 $Y \cup \{x\} \in \mathcal{I}$. We refer to this condition as 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. *Let $M = (S, \mathcal{I})$ be a matroid, and let $X \in \mathcal{I}$. We say that X is a base if $X \nsubseteq Y$, where $Y \in \mathcal{I}$ with $X \neq Y$. In other words, a* base *is a maximal independent set of the given matroid M.*

Definition 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 the given matroid M.*

Definition 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 corresponding set of column indexes in S is in $\mathcal{I}$.*

It is easy to note that if a matroid M is linearly representable over a field $\mathbb{F}$, then its representation need not be unique. For instance, the matroid represented by the $n \times n$ identity matrix I_n is the same as the matroid represented by any $n \times n$ non-singular matrix over a field $\mathbb{F}$. Thus the following problem seems natural in the context of linearly representable matroids.

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

Definition 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 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 $f(x) \equiv 1 (mod\ 2)$.*

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 and solving systems of linear equations over $\mathbb{Z}_2$ from [2] and properties of $\oplus$L from [4] we show that ECLR[$\mathbb{Z}_2$] is $\oplus$L-complete under logspace many-one reductions.

Note 1. In this paper, we deal with checking if two linear representations represent the same matroid. However, note that a notion of *equivalence* of two linear representations is also known [5, Section 6.3]. We say that *two linear representations M_1 and M_2 are equivalent* if we can obtain M_2 from M_1 using any of the following operations: elementary row operations, multiplying a column of M_1 by a non-zero element of the base field $\mathbb{F}$, replacing each entry of M_1 by its image under some automorphism of $\mathbb{F}$ and permuting columns of M_1 (moving indexes along with their columns). It is not hard to note that M_1 and M_2 can be equivalent even if they do not represent the same matroid. Also two linear representations representing the same matroid need not be equivalent. We once again emphasise that we are interested in the ECLR problem which checks if the matroids represented by the input linear representations are equal, rather than their equivalence. We hope this terminology is not misleading. We are also unaware of any reference where the ECLR problem has been previously taken up for study.

Theorem 1. *Given matrices $M_1, M_2 \in \mathbb{Z}_2^{m\times n}$ of full row rank, 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.
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, if $M_1 \sim M_2$ and $M_2 \sim M_3$ then there exists invertible matrices $X_1, X_2 \in \mathbb{Z}_2^{m\times m}$ with $X_1M_1 = M_2$ 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 $M_1, M_2 \in \mathbb{Z}_2^{m\times n}$ of full row rank representing the same matroid M. We can identify the lexicographically least base (refer Definition 2) of the matroid M from these matrices. Let Y_1 and Y_2 be the sub matrices of M_1 and M_2 corresponding to this base. Also 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_2 can be written as a $\mathbb{Z}_2$-linear combination of columns in Y_2 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 are non-zero. Moreover M_1 and M_2 represent the same matroid. Thus any set of columns form a circuit in M_1 if and only if the corresponding set of columns form a circuit in M_2 also. As a result we have $XM_1 = M_2$ whenever M_1 and M_2 represent the same matroid.
   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 any set of linearly independent vectors is also linearly independent.

Remark 1. In fact this property regarding the number of equivalence classes holds true for all linear representations over any field $\mathbb{F}$ whose characteristic is 2. As we observe in the next result that this property is crucial in solving ECLR$[\mathbb{Z}_2]$. However when considering linear representations of matroids with more than one element over a field $\mathbb{F}$ of characteristic not equal to 2, the number of such equivalence classes is more than one. It remains unknown even if there is a polynomial time algorithm to solve the ECLR problem when the input is over such a field of characteristic not equal to 2.

Theorem 2. ECLR$[\mathbb{Z}_2]$ *is* $\oplus$L*-complete.*

Proof. We are given linear representations $M_1, M_2 \in \mathbb{Z}_2^{l\times n}$ as input. In [2, Theorem 10] it has been shown that computing a maximal set of linearly independent vectors from a given set is $\oplus$L-complete. As a result we can obtain submatrices M_1' of M_1 and M_2' of M_2 of full row rank using a $\oplus$L computation. Choosing a sub matrix M' of full rank from a given matrix M does not affect the linear independence and dependence of columns of that matrix. In other words, columns indexed by $i_1, \ldots, i_l$ are linearly independent (or dependent) in M if and only if they are linearly independent (or dependent) in M' also. Thus the matroid represented by M_i' is the same as the matroid represented by M_i, for $i = 1, 2$.

It then follows from Theorem 1 that M_1 and M_2 represent the same matroid if and only if there exists a solution to the system $XM_1' = M_2'$ over $\mathbb{Z}_2$. Checking if a system of linear equations over $\mathbb{Z}_2$ is feasible is shown to be complete for $\oplus$L in [2, Theorem 10]. Now, we can retrieve the entries of M_1' and M_2' using a $\oplus$L oracle, and check for a solution to the above system using another $\oplus$L computation. Thus ECLR$[\mathbb{Z}_2]$ is in $\oplus\mathrm{L}^{\oplus\mathrm{L}}$, which is once again $\oplus$L using the result of [4]. This shows the $\oplus$L upper bound.

Hardness for $\oplus$L follows from the following observation: given a matrix $A \in \mathbb{Z}_2^{n\times n}$, checking if A is invertible or not is hard for $\oplus$L. Thus given a matrix A as input, we output A and I_n, where I_n is the $n \times n$ identity matrix. Clearly, $(A, I_n) \in$ ECLR$[\mathbb{Z}_2]$ if and only if A is invertible which completes the proof.

1.1 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 rationals, $\mathbb{Q}$. Let $M_1, M_2 \in \mathbb{Q}^{m\times n}$ be of full row rank. Given a set of indexes, columns corresponding to which are linearly dependent in one representation but are linearly independent in another 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. More precisely, assume that there is a polynomial time algorithm that decides ECLR$[\mathbb{Q}]$. Then 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 is the same, and outputs 0 otherwise. Assume the inputs 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 (refer Definition 3) in M_i, but corresponding columns do not form a circuit in M_j, using ECLR(M_1, M_2) as an oracle.

Given any $X \subseteq S = \{1, \ldots, n\}$ 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 $\mathcal{I}_j^{(X)} = \{X \cap I | \text{for } I \in \mathcal{I}_j\}$. We start by assuming that M_1 and M_2 represent different matroids. Let $i = 1$, $X = \{2, \ldots, n\}$, and $Y = \emptyset$. We now query the ECLR oracle if $M_1^{(X)}$ and $M_2^{(X)}$ represent the same matroid. If the oracle outputs 1, then it is clear that the i^{th} element of S, represented by the i^{th} column in M_1 and M_2, is in every subset of S that forms a circuit in $M_k^{(X)}$ but is linearly independent in $M_l^{(X)}$, where $1 \leq k, l \leq 2$ with $k \neq l$. In this case, we include i in the set Y, increment i, and re-initialize $X = Y \cup \{(i + 1), \ldots, n\}$. However, if the ECLR oracle outputs 0 upon receiving input $M_1^{(X)}$ and $M_2^{(X)}$, it is clear that there exists some subset of X that forms a circuit in one of the input linear representations but is linearly independent in the other. In this case we do not include i in Y, but just increment i, and re-initialize $X = Y \cup \{(i + 1), \ldots, n\}$. We repeat the above procedure until $i \leq n$. It is easy to note that the set Y that we finally obtain is a set of indexes such that columns corresponding to it form a circuit in one of the linear representations but not in the other. 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.

2 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 $|V_1| = |V_2| = n$, we check if any 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 form a cycle in G_2.

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, defined over any given field $\mathbb{F}$, to a given undirected graph. The matroid so obtained is called a *cycle matroid* or a *graphic matroid* [5, page 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 are linearly dependent over $\mathbb{F}$. In fact the incidence matrix of a 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 7. *A cut-edge of a graph G is an edge whose deletion from the graph increases the number of components in G.*

Theorem 3. *[8, 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 [7] to identify such cut-edges and hence decide CYCLE MATCHING.

Lemma 1. *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 3, 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, $G - \{e\}$ can be easily constructed from G in logspace. We then use the undirected *st*-connectivity algorithm of [7], to check if there is path from i to j in $G - \{e\}$, and hence output if e is a cut-edge.

Theorem 4. CYCLE MATCHING *is in* L.

Proof. Given a graph G, the following procedure obtains the sub graph 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 3 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 sub graph 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 sub graphs 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 vertex 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, 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, adjacency relation between the vertices should be preserved. 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 1 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 5. CYCLE MATCHING *is hard for* L

Proof. The *st*-connectivity problem for undirected graphs has been shown to be complete for L [7]. 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 is no path from s to t, then 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 from s to t 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.

2.1 A hard counting problem from CYCLE MATCHING

Definition 8. *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 9. *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 $f(x)$ from $g(x)$ in polynomial time.*

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 connected 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 can be easily observed 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 newly introduced edges in 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 completes the #P-completeness proof.

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 edges that are cut-edges in one of these input graphs but not the other can be shown 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 not the other has been shown to be #P-complete. But any element of both these sets witness the fact that (G_1,G_2) is not in CYCLE MATCHING.

Acknowledgments. I thank Samir Datta and Srikanth Srinivasan for useful discussions.

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. J. Edmonds. Minimum partition of a matroid into independent subsets. *J. Res. National Bureau of Standards*, 69B: 67-72, 1965.
4. U. Hertrampf, S. Reith, and H. Vollmer. A note on closure properties of logspace-MOD classes. *Information Processing Letters*, 75(3): 91-93, 2000.
5. J. Oxley. *Matroid Theory.* Oxford University Press, 2006.
6. C. Papadimitriou and K. Steiglitz. *Combinatorial Optimization, Second edition.* Dover Publications, 1998.
7. O. Reingold. Undirected connectivity in log-space. *Journal of the ACM*, 55(4): 1-24, 2008.
8. D.B. West. *Introduction to Graph Theory, Second edition.* Prentice-Hall of India private limited, 2003.