On Hypergraph and Graph Isomorphism with Bounded Color Classes*

V. Arvind[1] and Johannes Köbler[2]

[1] The Institute of Mathematical Sciences, Chennai 600 113, India
arvind@imsc.res.in
[2] Institut für Informatik, Humboldt Universität zu Berlin, Germany
koebler@informatik.hu-berlin.de

Abstract. Using logspace counting classes we study the computational complexity of hypergraph and graph isomorphism where the vertex sets have bounded color classes for certain specific bounds. We also give a polynomial-time algorithm for hypergraph isomorphism for bounded color classes of arbitrary size.

1 Introduction

In this paper we explore the complexity of Graph Isomorphism (GI) and Hypergraph Isomorphism (HGI) in the bounded color class setting. This means that the vertices of the input graphs are colored and we are only interested in isomorphisms that preserve the colors. The restriction of graph isomorphism to graphs with n vertices where the number of vertices with the same color is bounded by $b(n)$ is very well studied (we call this problem GI_b). In fact, for $b(n) = O(1)$ it was the first restricted version of GI to be put in polynomial time using group-theoretic methods [5] (as usual we denote $\mathrm{GI}_{O(1)}$ by BCGI). Later Luks put BCGI in NC using nontrivial group theory [8], whereas Torán showed that BCGI is hard for the logspace counting classes $\mathrm{Mod}_k\mathrm{L}$, $k \geq 2$ [11]. Actually, Torán's proof shows that for $k \geq 2$, GI_{k^2} (as well as GA_{2k^2}) is hard for $\mathrm{Mod}_k\mathrm{L}$. More recently, in [1] it is shown by carefully examining Luks' algorithm and Torán's hardness result that BCGI is in the $\mathrm{Mod}_k\mathrm{L}$ hierarchy and is in fact hard for this hierarchy.

For a fixed constant b, there is still a gap between the general upper bound result of [1] for GI_b and Torán's hardness result. More precisely, if GI_b is upper bounded by, say, the t-th level of the $\mathrm{Mod}_j\mathrm{L}$ hierarchy, and is hard for the s-th level of the $\mathrm{Mod}_k\mathrm{L}$ hierarchy, the constants t and j are much larger than s and k respectively. In the absence of a general result closing this gap, it is interesting to investigate the complexity of GI_b for specific values of b. In [6] GI_2 and GI_3 are shown to be equivalent to undirected graph reachability implying that they are complete for L [10]. In the present paper, we take a linear-algebraic approach to proving upper and lower bounds for GI_b. This is natural because the $\mathrm{Mod}_k\mathrm{L}$

* Work supported by a DST-DAAD project grant for exchange visits.

classes for prime k have linear-algebraic complete problems. Using linear algebra over $\mathbb{F}_2$ we are able to show that GI_b is $\oplus\mathrm{L}$ complete for $b \in \{4, 5\}$. Our techniques involve a combination of linear algebra with a partial Weisfeiler-Lehman type of labeling procedure (see e.g. [4]).

Another natural question to investigate is the complexity of Hypergraph Isomorphism when the vertex set is divided into color classes of size at most $b(n)$ (call it HGI_b). Firstly, notice that even for constant b, it is not clear whether HGI_b is reducible to BCGI. At least the usual reduction from HGI to GI does not give a constant bound on the color classes. The reason is that hyperedges can be of unbounded size. Hence a hyperedge's orbit can be exponentially large even under color-preserving vertex permutations. Thus, we need to directly examine the complexity of HGI_b. We show using group theory and linear algebra that HGI_2 is $\oplus\mathrm{L}$ complete. Further we show that for any prime p, HGA_p (as well as GA_{p^2}) is $\mathrm{Mod}_p\mathrm{L}$ hard.

Next we consider HGI_b for arbitrary b. Since HGI is polynomial-time many-one equivalent to GI, complexity-theoretic upper bounds for GI like $\mathrm{NP} \cap \mathrm{coAM}$ and SPP hold for HGI. However, consider an instance of HGI: a pair of hypergraphs (X_1, X_2), with n vertices and m edges each. The reduction to GI maps it to a pair of graphs (Y_1, Y_2) with vertex sets of size $m + n$. The best known isomorphism testing algorithm (see [2]) which has running time $c^{\sqrt{n \lg n}}$ will take time $c^{\sqrt{(m+n)\lg(m+n)}}$ when combined with the above reduction and applied to HGI. The question whether HGI has a simply exponential time (i.e. c^n time) algorithm was settled positively by Luks by using a dynamic programming approach [9]. In Section 5 we give a polynomial-time upper bound for HGI_b when b is bounded by a constant. This result is based on Luks' polynomial-time algorithm [7] for the set stabilizer problem for a permutation group in the class Γ_d.

2 Preliminaries

We first fix some notation. Let $X = (V, E)$ denote a (finite) hypergraph, i.e., E is a subset of the power set $\mathcal{P}(V)$ of V, and let g be a permutation on V. We can extend g to a mapping on subsets $U = \{u_1, \ldots, u_k\}$ of V by

$$g(U) = \{g(u_1), \ldots, g(u_k)\}.$$

g is an *isomorphism* between hypergraphs $X = (V, E)$ and $X' = (V, E')$, if

$$\forall e \subseteq V : e \in E \Leftrightarrow g(e) \in E'.$$

We also say that g maps X to X' and write $g(X) = X'$. If $g(X) = X$, then g is called an *automorphism* of X. Note that the identity mapping on V is always an automorphism. Any other automorphism is called *nontrivial.*

A *coloring* of a hypergraph (V, E) is given by a partition $\mathcal{C} = (C_1, \ldots, C_m)$ of V into disjoint *color classes* C_i. We call $X = (V, E, \mathcal{C})$ a *colored hypergraph.* In case $\|C_i\| \leq b$ for all $i = 1, \ldots, m$, we refer to X as a *b-bounded hypergraph.*

Further, for $1 \leq k \leq b$, we use $\mathcal{C}_k = \{C \in \mathcal{C} \mid \|C\| = k\}$ to denote the set of all color classes having size exactly k.

A permutation g on V is called an *isomorphism* between two colored hypergraphs $X = (V, E, \mathcal{C})$ and $X' = (V, E', \mathcal{C}')$ with colorings $\mathcal{C} = (C_1, \ldots, C_m)$ and $\mathcal{C}' = (C'_1, \ldots, C'_m)$, if g preserves the hyperedges (i.e., $g(V, E_1) = (V, E_2)$) and g preserves the colorings (i.e., $g(C_i) = g(C'_i)$ for all $i = 1, \ldots, m$).

The decision problem HGI_b consists of deciding whether two given b-bounded hypergraphs X_1 and X_2 are *isomorphic*. A related problem is the hypergraph automorphism problem HGA_b of deciding if a given b-bounded hypergraph X has a nontrivial automorphism. For usual b-bounded graphs $X = (V, E)$ (i.e., each edge $e \in E$ contains exactly 2 nodes), we denote the isomorphism and automorphism problems by GI_b and GA_b, respectively.

Let $X = (V, E)$ be a graph. For a subset $U \subseteq V$, we use $X[U]$ to denote the *induced subgraph* $(U, E(U))$ of X, where $E(U) = \{e \in E \mid e \subseteq U\}$. Further, for disjoint subsets $U, U' \subseteq V$, we use $X[U, U']$ to denote the *induced bipartite subgraph* $(U \cup U', E(U, U'))$, where $E(U, U')$ contains all edges $e \in E$ with $e \cap U \neq \emptyset$ and $e \cap U' \neq \emptyset$. For a set U of nodes, we use $\Gamma_X(U)$ to denote the *neighborhood* $\{v \in V \mid \exists u \in U : (u, v) \in E\}$ of U in X.

We denote the *symmetric group* of all permutations on a set A by $\mathrm{Sym}(A)$ and by S_n in case $A = \{1, \ldots, n\}$. Let G be a subgroup of $\mathrm{Sym}(A)$ and let $a \in A$. Then the set $\{b \in A \mid \exists \pi \in G : \pi(a) = b\}$ of all elements $b \in A$ reachable from a via a permutation $\pi \in G$ is called the *orbit* of a in G.

3 Graphs with color classes of size 5

In this section we prove that GA_4 is contained in $\oplus\mathrm{L}$. The proof is easily extended to GI_4 as well as to GA_5 and GI_5. Let $X = (V, E, \mathcal{C})$ be a 4-bounded graph (an instance of GA_4) and let $\mathcal{C} = (C_1, \ldots, C_m)$. We use X_i to denote the graph $X[C_i]$ induced by C_i and X_{ij} to denote the bipartite graph $X[C_i, C_j]$ induced by the pair of color classes C_i and C_j. We assume that all vertices in the same color class have the same degree and that the edge set E_i of X_i is either empty or consists of two disjoint edges (only if $\|C_i\| = 4$), since otherwise we can either split C_i into smaller color classes or we can replace X_i by the complement graph without changing the automorphism group $\mathrm{Aut}(X)$ of X. Further, we assume that the edge set E_{ij} of X_{ij} is of size at most $\|C_i\| \cdot \|C_j\|/2$, since otherwise, we can replace X_{ij} by the complement bipartite graph without changing $\mathrm{Aut}(X)$.

Any $\pi \in \mathrm{Aut}(X)$ can be written as $\pi = (\pi_1, \ldots, \pi_m)$ in $\mathrm{Aut}(X_1) \times \cdots \times \mathrm{Aut}(X_m)$, where π_i is an automorphism of X_i. Furthermore, for any pair of color classes C_i and C_j, (π_i, π_j) has to be an automorphism of X_{ij}. These are precisely the constraints that any automorphism in $\mathrm{Aut}(X)$ must satisfy.

Since each $\mathrm{Aut}(X_i)$ is isomorphic to a subgroup of S_4, the only prime factors of $\|\mathrm{Aut}(X)\|$ (if any) are 2 and 3. Thus, $\mathrm{Aut}(X)$ is nontrivial if and only if it has either an automorphism of order 2 or of order 3. By a case analysis, we will show that the problem of testing whether X has a nontrivial automorphism can be reduced to either undirected graph reachability or to solving a system of linear equations over $\mathbb{F}_2$, implying that the problem is in $\oplus$L. In fact, as we will see, it is also possible to compute a generating set for $\mathrm{Aut}(X)$ in $\mathrm{FL}^{\oplus \mathrm{L}}$.

Let G_i be the intersection of $\mathrm{Aut}(X_i)$ with the projections of $\mathrm{Aut}(X_{ij})$ on C_i for all $j \neq i$. Any subgroup of the symmetric group $\mathrm{Sym}(C_i)$ of all permutations on C_i is called a *constraint* for C_i. We call G_i the *direct constraint* for C_i.

The algorithm proceeds in several preprocessing steps which progressively eliminate different cases and simplify the graph. We describe some base steps of the algorithm in a sequence of claims.

Claim 1. *The direct constraints can be determined in deterministic logspace.*

Proof. Follows easily from the fact that the color classes are of constant size. □

Next we consider a specific way in which the direct constraints get propagated to other color classes in X. To this end, we define a symmetric binary relation $\mathcal{T}$ on the set $\mathcal{C}$. Let $C_i, C_j \in \mathcal{C}$ such that $\|C_i\| = \|C_j\|$. Then $(C_i, C_j) \in \mathcal{T}$ if

– $\|C_i\| \in \{1, 2, 3\}$ and X_{ij} is a perfect matching or
– $\|C_i\| = 4$ and X_{ij} is either a perfect matching or an 8-cycle.

The following easy lemma states a specific useful way in which the constraints get propagated over color classes related via $\mathcal{T}$.

Lemma 2. *For each pair $(C_i, C_j) \in \mathcal{T}$ there is a bijection $f_{ij} : C_i \to C_j$ such that for any automorphism $\pi = (\pi_1, \ldots, \pi_m) \in \mathrm{Aut}(X)$ the permutations $\pi_i \in G_i$ and $\pi_j \in G_j$ are related as follows:*

$$\forall u, v \in C_i : \pi_i(u) = v \iff \pi_j(f_{ij}(u)) = f_{ij}(v).$$

In other words, if $(C_i, C_j) \in \mathcal{T}$ via f_{ij} and $\pi = (\pi_1, \ldots, \pi_m) \in \mathrm{Aut}(X)$ is an automorphism, then $\pi_j(f_{ij}(u)) = f_{ij}(\pi_i(u))$, i.e., π_j is the image of π_i under the bijection $g_{ij} : \mathrm{Sym}(C_i) \to \mathrm{Sym}(C_j)$ defined by $\pi \mapsto f_{ij} \circ \pi \circ f_{ij}^{-1}$ (here we use $g \circ h$ to denote the mapping $x \mapsto g(h(x))$).

We use Lemma 2 to define a symmetric relation on constraints. Let G and H be constraints of two different color classes C_i and C_j, respectively, where $(C_i, C_j) \in \mathcal{T}$. We say that G is *directly induced by* H, if g_{ij} is an isomorphism between G and H. Further, G *is induced by* H, if G is reachable from H via a chain of directly induced constraints. Note that the latter relation is an equivalence on the set of all constraints. We call the intersection of all constraints of C_i that are induced by some direct constraint the *induced constraint* of C_i and denote it by G'_i. Note that $\mathrm{Aut}(X)$ is a subgroup of the m-fold product group $\prod_{i=1}^{m} G'_i$ of all induced constraints.

Claim 3. *The induced constraints can be determined in deterministic logspace.*

Proof. Consider the undirected graph $X' = (V', E')$ where V' consists of all constraints G in X and $E' = \{(G, H) \mid G \text{ is directly induced by } H\}$. In this graph we mark all direct constraints computed by Claim 1 as special nodes. Now, the algorithm outputs for each color class C_i the intersection of all constraints for C_i that are reachable from some special node, and since SL = L [10], this can be done in deterministic logspace. □

We define two special types of constraints. We say that C_i is *split*, if G'_i has at least two orbits, and we call the partition of C_i in the orbits of G'_i the *splitting partition* of C_i. Further, a partition $\{H_0, H_1\}$ of C_i with $\|H_0\| = \|H_1\| = 2$ is called a *halving* of C_i, if any $\pi \in G'_i$ either maps H_0 to itself or to H_1. Any class C_i which has a halving, is called *halved*, and all color classes that are neither split nor halved are called *whole*. Now let $\mathcal{C}_s$, $\mathcal{C}_h$ and $\mathcal{C}_w$ denote the subclasses of $\mathcal{C}$ containing all split, halved, and whole color classes, respectively, and consider the following two cases:

Case A: All color classes in $\mathcal{C}_4$ are halved (i.e., $\mathcal{C} \subseteq \mathcal{C}_h \cup \mathcal{C}_3 \cup \mathcal{C}_2 \cup \mathcal{C}_1$).
Case B: All color classes in $\mathcal{C}_4$ are halved and $\mathcal{C}_3$ is empty (i.e., $\mathcal{C} \subseteq \mathcal{C}_h \cup \mathcal{C}_2 \cup \mathcal{C}_1$).

We first show how the general case logspace reduces to Case A. Then we reduce Case A to Case B in logspace, and finally we show how Case B is solved in logspace with a $\oplus$L oracle. We start by summarizing some properties of whole color classes which are easily proved by a case analysis.

Lemma 4. *Let $C_i, C_j \in \mathcal{C}$ be color classes, where C_i is whole and $E_{ij} \neq \emptyset$.*

- *All vertices in C_i have the same degree in X_{ij}. Likewise, all vertices in the neighborhood $\Gamma_{X_{ij}}(C_i)$ have the same degree in X_{ij}.*
- *If C_j is whole, then $\|C_i\| = \|C_j\|$ and $(C_i, C_j) \in \mathcal{T}$.*
- *If C_j is halved, then $\|C_i\| \leq 3$.*
- *If C_j is halved and $E_j \neq \emptyset$, then $\|C_i\| \leq 2$.*
- *If C_j is split and $\|C_j\| \leq \|C_i\|$, then all vertices in C_i have the same neighborhood in X_{ij}.*

Lemma 4 tells us that the action of an automorphism on a whole color class $C \in \mathcal{C}_4$ is not influenced by its action on color classes that are either smaller or halved or split, i.e., only other whole color classes in $\mathcal{C}_4$ can influence C. This means that we can write $\mathrm{Aut}(X)$ as the product $\mathrm{Aut}(X') \times \mathrm{Aut}(X'')$, where X' is the induced subgraph of X containing the nodes of all color classes in $\mathcal{C}_4 \cap \mathcal{C}_w$ and X'' is induced by the set of all other nodes. Clearly, it suffices to compute generating sets for $\mathrm{Aut}(X')$ and $\mathrm{Aut}(X'')$.

Claim 5. *A generating set for* $\mathrm{Aut}(X')$ *can be computed in* FL.

Proof. The algorithm will work by reducing the problem to reachability in undirected graphs. For each whole color class $C_i \in \mathcal{C}_4$ we create a set P_i of 4! nodes (one for each permutation of C_i). Consider $C_i, C_j \in \mathcal{C}_4 \cap \mathcal{C}_w$ such that $(C_i, C_j) \in \mathcal{T}$ and let f_{ij} be the bijection from Lemma 2. Recall that for each $\pi \in P_i$ the bijection f_{ij} induces a unique permutation $\psi = g_{ij}(\pi)$ on C_j and hence, we put an undirected edge between π and ψ. We thus get an undirected graph $\hat{X}$ with $4!\|\mathcal{C}_4 \cap \mathcal{C}_w\|$ nodes.

A connected component P in $\hat{X}$ that picks out at most one element π_i from each set P_i defines a valid automorphism π for the graph X', if P contains only elements $\pi_i \in \mathrm{Aut}(X_i)$. On the color classes C_i, for which P contains an element $\pi_i \in P_i$, π acts as π_i, and it fixes all nodes of the other color classes. By collecting these automorphisms we get a generating set for $\mathrm{Aut}(X')$ and since SL = L [10], this can be done in deterministic logspace. □

By using Claim 5 we can already assume that $\mathcal{C}_4$ only contains halved or split color classes. To fulfil the assumption of Case A, we now take all the split color classes and break them up into smaller color classes defined by the split. Then only halved color classes remain in $\mathcal{C}_4$. Further, we can assume that if C_i is halved, then E_i consists of two disjoint edges.

Next we consider the reduction of Case A to Case B. Since $\mathcal{C}_4$ only contains halved color classes with two disjoint edges, Lemma 4 now guarantees that a whole color class $C \in \mathcal{C}_3$ can only influence other whole color classes in $\mathcal{C}_3$. Hence, we can remove all whole color classes in $\mathcal{C}_3$ exactly as we removed the whole color classes in $\mathcal{C}_4$, by applying an analogue of Claim 5. Now we can again break up all split color classes yielding a graph that fulfils Case B. Observe that the splitting partition of a halved color class only contains sets of size 1, 2 or 4.

It remains to compute a generating set for a 4-bounded graph which only has color classes of size 1, 2 or 4, where all the size 4 color classes are halved. Clearly, in this case X can only have nontrivial automorphisms of orders 2 or 4. We continue by defining an encoding of the candidate automorphisms of $\mathrm{Aut}(X)$ as vectors over $\mathbb{F}_2$. Later we show how one can reduce the problem of finding $\mathrm{Aut}(X)$ to the problem of solving linear equations over $\mathbb{F}_2$.

An encoding trick

We now describe how to encode the automorphisms of X with vectors over $\mathbb{F}_2$. We introduce some $\mathbb{F}_2$ indeterminates for each color class C in X of size more than 1. More precisely, we encode each automorphism $\pi \in \mathrm{Aut}(X_i)$ by a vector $v_\pi \in \mathbb{F}_2^{n_i-1}$, where $n_i = \|C_i\|$.

1. To each color class $C_i = \{u_0, u_1\}$ of size 2 we introduce a single variable x. Here $x = 1$ denotes the transposition $(u_0\ u_1)$ and $x = 0$ denotes the identity mapping on C_i.
2. If C_i is a halved color class of size 4, let $e_0 = \{u_{00}, u_{01}\}$ and $e_1 = \{u_{10}, u_{11}\}$ be the two disjoint edges in E_i. Notice that each vertex index is encoded with two bits. The first bit encodes the edge and the second bit the vertex in that edge. Then we encode an automorphism $\pi \in \mathrm{Aut}(X_i)$ by a three bit vector $v_\pi = xyz$, where $\pi(u_{00}) = u_{xy}$ and $\pi(u_{10}) = u_{\bar{x}z}$. In other words, the permutation π encoded by v_π maps

$$u_{ab} \mapsto u_{a'b'}, \text{ where } a' = a + x \text{ and } b' = \begin{cases} b + y, & a = 0, \\ b + z, & a = 1. \end{cases}$$

Notice that the addition of the 3-bit representations in $\mathbb{F}_2^3$ *does not* capture the permutation group structure of $\mathrm{Aut}(X_i)$, since the latter is nonabelian.

Let $t = \sum_{i=1}^{m}(n_i - 1)$ denote the sum of the lengths of the $\mathbb{F}_2$ representations for each color class. Thus, every element $\pi = (\pi_1, \ldots, \pi_m)$ of $\mathrm{Aut}(X)$ is encoded as a vector $v_\pi = (v_{\pi_1}, \ldots, v_{\pi_m})$ in $\mathbb{F}_2^t$, and each vector in $\mathbb{F}_2^t$ represents a potential automorphism. Recall that a permutation $\pi = (\pi_1, \ldots, \pi_m) \in$

$\mathrm{Aut}(X_1) \times \cdots \times \mathrm{Aut}(X_m)$ is in $\mathrm{Aut}(X)$ if and only if for each C_i and C_j, (π_i, π_j) is an automorphism of X_{ij}. We now show that each of these constraints yields a set of linear equalities on the indeterminates that encode (π_i, π_j). Hence, we claim that a vector in $\mathbb{F}_2^t$ encodes an automorphism of X if and only if it satisfies the set of all these linear equalities.

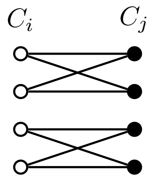
Lemma 6. *For any pair of color classes C_i and C_j in X the set*

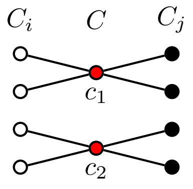
$$F_{ij} = \{v_\pi v_\varphi \mid \pi \in \mathrm{Aut}(X_i), \varphi \in \mathrm{Aut}(X_j) \text{ and } (\pi, \varphi) \in \mathrm{Aut}(X_{ij})\}$$

forms a subspace of $\mathbb{F}_2^{n_i+n_j-2}$.

Proofsketch. The proof is by a case analysis checking all the possibilities. First note that for any color class C_i of size more than 1, the encodings of all automorphisms in $\mathrm{Aut}(X_i)$ form the space $\mathbb{F}_2^{n_i-1}$. Hence, $F_{ij} = \mathbb{F}_2^{n_i+n_j-2}$ if $E_{ij} = \emptyset$.

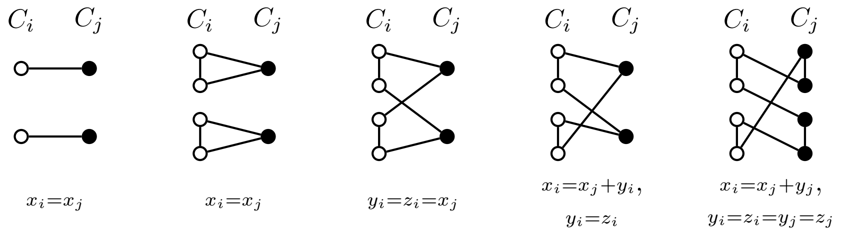
To simplify the analysis if E_{ij} is not empty, notice that if C_i, C_j are unsplit halved color classes, then either $(C_i, C_j) \in \mathcal{T}$ or E_{ij} is the union of two 4-cycles. In the latter case we can modify X_{ij} by removing the two 4-cycles and introducing a new color class $C = \{c_1, c_2\}$.





The nodes c_1 and c_2 represent the two deleted 4-cycles in the following sense: the node c_1 is adjacent to the 4 nodes that were part of one of the 4-cycles (two each in C_i and C_j). Similarly, c_2 is adjacent to the 4 nodes on the other 4-cycle (two each in C_i and C_j). Then the modified graph has the same automorphism group as X_{ij}, after we project out the new color class C. Hence, we can assume that if C_i and C_j are both halved, then $(C_i, C_j) \in \mathcal{T}$.

Now consider the case that both C_i and C_j are of size 2. This case is easy, since the addition of the representations $v_\pi v_\varphi$ captures the permutation group structure. In fact, the only interesting case is when E_{ij} is a perfect matching where $F_{ij} = \{x_i x_j \mid x_i = x_j\}$ (see the following picture).



Next we consider the case $n_i = 4$ and $n_j = 2$. It is easy to see by considering the different cases that the elements $x_iy_iz_ix_j$ of F_{ij} form a subspace of $\mathbb{F}_2^4$ (see the picture for three interesting cases).

Finally, if $n_i = n_j = 4$, we can assume that $(C_i, C_j) \in \mathcal{T}$. It is easy to check that exactly one of the following two possibilities can occur: either the projection of $(\mathrm{Aut}(X_i) \times \mathrm{Aut}(X_j)) \cap \mathrm{Aut}(X_{ij})$ on C_i contains all 8 automorphisms (this case is similar to the case $E_{ij} = \emptyset$), or it is a subgroup that contains no automorphisms of order four. An interesting property of the encoding is that all subgroups containing no order four elements are such that permutation composition coincides with vector addition in $\mathbb{F}_2^{n_i-1}$. It follows from this observation that the elements $x_iy_iz_ix_jy_jz_j$ of F_{ij} form a subspace of $\mathbb{F}_2^6$ (see the picture for an example). □

As a direct consequence of the above lemma and the fact that the solution space for a system of linear equations over $\mathbb{F}_2$ can be computed in $\mathrm{FL}^{\oplus\mathrm{L}}$ (see [3]) we get the following upper bound.

Claim 7. *Let X be a 4-bounded graph fulfilling the assumption of Case B. Then the problem of computing a generating set for its automorphism group is in* $\mathrm{FL}^{\oplus\mathrm{L}}$.

By refining Torán's proof [11] that GA is hard for $\oplus$L, it can be shown that also GA_4 is hard for $\oplus$L (see Section 5). By combining this hardness result with the $\oplus$L upper bound for GA_4 established in this section we get the following completeness result.

Theorem 8. GA_4 *is complete for* $\oplus$L.

We end this section by remarking that Theorem 8 easily extends to GA_5 as well as to GI_5. Color classes of size 5 are either whole or split. The halved classes can only be of size 4. Thus, the $\oplus$L upper bound goes through for GA_5 with minor changes.

4 Hypergraphs with color classes of size 2

In this section we show that HGA_2 and HGI_2 are complete for $\oplus$L under logspace reductions. We only give the proof for HGA_2. The proof for HGI_2 is similar. Let $X = (V, E)$ be a 2-bounded hypergraph. Thus V is partitioned into color classes C_i with $\|C_i\| \leq 2$ for $i = 1, \dots, m$. To each color class C_i of size 2 we associate an indeterminate x_i over $\mathbb{F}_2$ which indicates by its value whether the vertices of C_i flip or not. Thus we can represent any color preserving permutation of X by a vector $x = x_1 \cdots x_m$ in $\mathbb{F}_2^m$.

Our aim is to compute in deterministic logspace a set of linear constraints on $x_1, \dots, x_m$ over $\mathbb{F}_2$ that determines $\mathrm{Aut}(X)$. This will imply that HGA_2 is in $\oplus$L. Hyperedges e and e' have the same *type* if $\|e \cap C_i\| = \|e' \cap C_i\|$ for all i. We can partition E into subsets $E_1, \dots, E_t$ of distinct types. Clearly, automorphisms preserve edge types. Thus, $\mathrm{Aut}(X)$ is expressible as the intersection of $\mathrm{Aut}(V, E_j)$ over all edge types.

Proposition 9. *A vector x in $\mathbb{F}_2^m$ represents an element in* $\mathrm{Aut}(X)$ *if and only if it represents an element in* $\mathrm{Aut}(V, E_j)$ *for each E_j.*

If for each E_j we can compute in logspace a set of linear constraints on $x_1, \ldots, x_m$ that determines $\mathrm{Aut}(V, E_j)$, then the union of these constraints will determine $\mathrm{Aut}(X)$. Thus, it suffices to consider the case that all edges in E are of the same type. Further, by ignoring color classes C_i with $\|e \cap C_i\| \in \{0, 2\}$ for all e, we can assume that $\|e \cap C_i\| = 1$ for all $e \in E$ and $i = 1, \ldots, m$.

Let $C_i = \{u_{0i}, u_{1i}\}$ for $i = 1, \ldots, m$. We can represent the hyperedges $e \in E$ by vectors $v_e = v_1 \cdots v_m \in \mathbb{F}_2^m$ with $v_j = 1$ if $u_{1j} \in e$ and $v_j = 0$ if $u_{0j} \in e$. With this representation, a candidate automorphism $x \in \mathbb{F}_2^m$ acts on the hyperedges by vector addition in $\mathbb{F}_2^m$:

$$x : v_e \mapsto v_e + x.$$

Since every automorphism x maps v_e to some hyperedge $v_{e'}$, the candidate automorphisms are in $S = \{v_e + v_{e'} \mid e' \in E\}$, for a fixed $e \in E$. A logspace machine M can easily check whether each vector $x \in S$ represents an automorphism, by testing if $x + v_e \in E$ for each $e \in E$.

Thus, M can compute the set $F \subseteq S$ containing the encodings of all automorphisms. Notice that F is a subspace of $\mathbb{F}_2^m$.

Finally, we can easily see that M can compute the dual space in terms of a matrix A over $\mathbb{F}_2$ such that $x \in F$ if and only if $Ax = 0$. This matrix A provides the desired set of linear constraints. Combining the constraints of all edge types gives the overall system of linear constraints, whose solutions are the automorphisms. In summary, we have proved the following theorem.

Theorem 10. *Let X be a 2-bounded hypergraph. Then the problem of computing a generating set for its automorphism group is in* $\mathrm{FL}^{\oplus\mathrm{L}}$. *In particular, the problem* HGA_2 *is in* $\oplus\mathrm{L}$.

By modifying Torán's proof [11] that GA is hard for $\oplus$L, it can be shown that for any prime p, HGA_p is hard for $\mathrm{Mod}_p\mathrm{L}$ (see the next section). Combined with the above theorem this gives the following completeness result.

By modifying Torán's proof [11] that GA is hard for $\oplus$L, it can be shown that for any prime p, HGA_p is hard for $\mathrm{Mod}_p\mathrm{L}$. Combined with the above theorem this gives the following completeness result.

Corollary 11. HGA_2 *is complete for* $\oplus$L *under logspace many-one reductions.*

5 Hypergraphs with constant size color classes

In this section we give a polynomial time upper bound for the problem of computing a generating set of $\mathrm{Aut}(X)$ for a b-bounded hypergraph $X = (V, E)$. Further we show that for any prime $p \leq k$, HGA_k is hard for $\mathrm{Mod}_p\mathrm{L}$ (and hence also hard for $\mathrm{Mod}_j\mathrm{L}$ under logspace conjunctive truth-table reductions, by closure properties of $\mathrm{Mod}_j\mathrm{L}$ classes, where j is the product of all primes $p \leq k$ [3]). Since the orbit size of the hyperedges in the reduced hypergraphs is bounded by

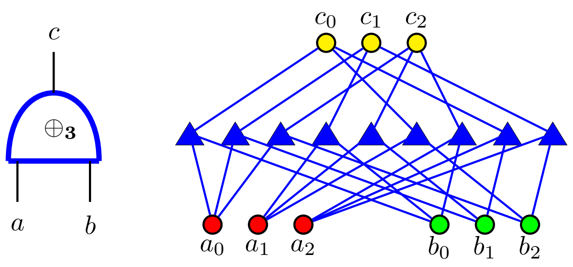
p^2, we also get that GA_{p^2} is $\mathrm{Mod}_p\mathrm{L}$ hard. For prime k, this slightly improves Torán's result that GA_{2k^2} is hard for $\mathrm{Mod}_k\mathrm{L}$.

Theorem 12. *For any prime p, HGA_p is logspace many-one hard for $\mathrm{Mod}_p\mathrm{L}$.*

Proofsketch. It is well-known that the evaluation problem CirVal_p for arithmetic circuits with $\oplus_p$-gates is complete for $\mathrm{Mod}_p\mathrm{L}$, where $\oplus_p$ denotes addition modulo p (see e.g. [11]). To evaluate a single $\oplus_p$-gate we consider the following hypergraph $H = (V, E)$, where

$$\begin{aligned} V =& \{a_i, b_i, c_i \mid i = 0, \ldots, p-1\}, \\ E =& \{\{a_i, b_j, c_k\} \mid i \oplus_p j = k\}. \end{aligned}$$

The following figure shows H for the case $p = 3$, where we use triangles to depict hyperedges.



Observe that for every pair of input nodes a_i, b_j, there is exactly one hyperedge e in E which contains a_i and b_j:

$$e = \{a_i, b_j, c_k\}, \text{ where } k = i \oplus_3 j$$

Hence, it follows for any automorphism f that

$$\begin{aligned} & f(a_0) = a_i \ \wedge \ f(b_0) = b_j \\ & \quad \Rightarrow f(\{a_0, b_0, c_0\}) = \{a_i, b_j, f(c_0)\} \in E \\ & \quad \Rightarrow f(c_0) = c_k, \text{ where } k = i \oplus_3 j. \end{aligned}$$

By replacing all $\oplus_p$-gates by such a gadget we can transform C into a hypergraph H which has a nontrivial automorphism g if and only if C evaluates to 1. Note that in order to force g to fix the p input nodes corresponding to a 0 input we can color them differently. Also, in order to force g to move the p input nodes corresponding to a 1 input we additionally insert p parallel feedback edges from the p output nodes to these nodes. Finally, by inserting additional vertices and edges we can force g to cyclically shift the p output nodes. $\square$

Next we give a polynomial time algorithm for computing a generating set of $\mathrm{Aut}(X)$ for a b-bounded hypergraph $X = (V, E)$. More precisely, we give an $n^{O(b)}$ algorithm for the problem, where $n = \|V\|$. We remark that if the hyperedges are all of constant size, i.e. $\|e\| \leq k$ for all $e \in E$ and constant k, then the problem is deterministic logspace reducible to BCGI which is known to be in NC [8]. However, when hyperedges are of unbounded size, it is not clear whether HGI_b is reducible to BCGI. Our polynomial time algorithm for HGI_b applies ideas from a different result of Luks [7]. We recall some definitions.

Definition 13. *Let $k \geq 1$. A finite group G is said to be in the class Γ_k if all nonabelian composition factors of G are isomorphic to subgroups of S_k.*

Theorem 14. [7] *Let $\Delta \subseteq \Omega$ and let $G \leq \mathrm{Sym}(\Omega)$ be given by a generating set S. If $G \in \Gamma_k$ then there is an $n^{O(k)}$ algorithm for computing the set stabilizer subgroup $G_\Delta = \{g \in G \mid g(\Delta) = \Delta\}$.*

For a colored hypergraph $X = (V, E, \mathcal{C})$ with $\mathcal{C} = (C', C'')$ let X' denote the hypergraph (C', E') where $E' = \{e \cap C' \mid e \in E\}$. X'' is defined accordingly. We first give an algorithm for the following problem, that we repeatedly use as a subroutine. Suppose $X = (V, E, \mathcal{C})$ is such a hypergraph where the automorphism groups $\mathrm{Aut}(X')$ and $\mathrm{Aut}(X'')$ are in Γ_k. The problem is to compute in polynomial time a generating set of $\mathrm{Aut}(X)$ from generating sets of $\mathrm{Aut}(X')$ and $\mathrm{Aut}(X'')$.

We first consider the embedding map

$$\varphi : \mathrm{Aut}(X') \to \mathrm{Sym}(C') \times \mathrm{Sym}(E') \text{ given by } \varphi(\pi) = (\pi, \tau),$$

where τ is simply defined as the action of π on E'. Similarly, we have the embedding map $\psi : \mathrm{Aut}(X'') \to \mathrm{Sym}(C'') \times \mathrm{Sym}(E'')$. Since φ and ψ are easy to compute, we can easily compute the generating set for $\varphi(\mathrm{Aut}(X'))$ as the image of the given generating set of $\mathrm{Aut}(X')$. Similarly, we can compute a generating set for $\psi(\mathrm{Aut}(X''))$. Thus, we can compute a generating set for the product group $\varphi(\mathrm{Aut}(X')) \times \psi(\mathrm{Aut}(X''))$ as a permutation group acting on $C' \cup E' \cup C'' \cup E''$.

Furthermore, since this permutation group action extends uniquely to $E' \times E''$, we can easily compute a generating set S for the product group $\varphi(\mathrm{Aut}(X')) \times \psi(\mathrm{Aut}(X''))$ as a permutation group acting on $C' \cup E' \cup C'' \cup E'' \cup (E' \times E'')$.

Notice that we can see $\mathrm{Aut}(X)$ as a subgroup of $\varphi(\mathrm{Aut}(X')) \times \psi(\mathrm{Aut}(X''))$. To compute $\mathrm{Aut}(X)$ we construct a bipartite graph Z with $W = E' \cup E''$ as the vertex set and edge set F, where for $e' \in E'$ and $e'' \in E''$ we include (e', e'') in F if and only if $e' \cup e'' \in E$.

Now, in order to invoke Theorem 14, let Ω denote the set

$$\Omega = C' \cup E' \cup C'' \cup E'' \cup (E' \times E'')$$

and let $G \leq \mathrm{Sym}(\Omega)$ be the group $\varphi(\mathrm{Aut}(X')) \times \psi(\mathrm{Aut}(X''))$. Since $\mathrm{Aut}(X')$ and $\mathrm{Aut}(X'')$ are in Γ_k and since Γ_k is closed under homomorphic images and products [7], it follows that $\varphi(\mathrm{Aut}(X')) \times \psi(\mathrm{Aut}(X''))$ is also in Γ_k. Hence, letting $\Delta = F$, Theorem 14 implies that we can compute G_Δ (which is $\mathrm{Aut}(X)$) in time $n^{O(k)}$. Notice that it suffices to retain the action of G_Δ on $V = C' \cup C''$ and we can discard the remaining part.

We now consider the problem of computing $\mathrm{Aut}(X)$ from scratch. For each $s = 1, \ldots, m$ we define hypergraphs $X_{1,s} = (C_{1,s}, E_{1,s})$, where $C_{1,s} = \bigcup_{i=1}^{s} C_i$ and $E_{1,s} = \{e \cap C_{1,s} \mid e \in E\}$. Notice that X itself is $X_{1,m}$. We also define $X_s = (C_s, E_s)$, where $E_s = \{e \cap C_s \mid e \in E\}$. Notice that X_s is a constant-size hypergraph for each s and $\mathrm{Aut}(X_s)$ can be computed easily in polynomial time. Our polynomial time algorithm starts by first computing $\mathrm{Aut}(X_{1,1}) = \mathrm{Aut}(X_1)$. Then for increasing values of $s = 1, \ldots, m$ it progressively computes the group $\mathrm{Aut}(X_{1,s})$ from the already computed groups $\mathrm{Aut}(X_{1,s-1})$ and $\mathrm{Aut}(X_s)$ by using the algorithm explained above. Notice that the groups $\mathrm{Aut}(X_{1,s-1})$ and $\mathrm{Aut}(X_s)$ are in Γ_k because their orbits are bounded by k. This completes the proof of the following result.

Theorem 15. *Given a hypergraph $X = (V, E)$ with color classes of size bounded by k, there is an $n^{O(k)}$ time algorithm that computes* $\mathrm{Aut}(X)$ *as a generating set in* $\mathrm{Sym}(V)$*. In particular,* HGI *and* HGA *are in* P*.*

References

1. V. Arvind, P. Kurur, and T. Vijayaraghavan. Bounded color multiplicity graph isomorphism is in the #L hierarchy. In *Proc. 20th Annual IEEE Conference on Computational Complexity*, pages 13–27. IEEE Computer Society Press, 2005.
2. L. Babai and E. Luks. Canonical labeling of graphs. In *Proc. 15th ACM Symposium on Theory of Computing*, 171–183, 1983.
3. G. Buntrock, C. Damm, U. Hertrampf, and C. Meinel. Structure and importance of logspace-MOD classes. *Mathematical Systems Theory*, 25:223–237, 1992.
4. J. Cai, M. Fürer, N. Immerman. An optimal lower bound on the number of variables for graph identifications. *Combinatorica*, 12(4):389-410, 1992.
5. M. Furst, J. Hopcroft, and E. Luks. Polynomial time algorithms for permutation groups. In *Proc. 21st IEEE Symposium on the Foundations of Computer Science*, pages 36–41. IEEE Computer Society Press, 1980.
6. B. Jenner, J. Köbler, P. McKenzie, and J. Torán. Completeness results for graph isomorphism. *Journal of Computer and System Sciences*, 66:549–566, 2003.
7. E. Luks. Isomorphism of bounded valence can be tested in polynomial time. *Journal of Computer and System Sciences*, 25:42–65, 1982.
8. E. Luks. Parallel algorithms for permutation groups and graph isomorphism. In *Proc. 27th IEEE Symposium on the Foundations of Computer Science*, pages 292–302. IEEE Computer Society Press, 1986.
9. E. Luks. Hypergraph isomorphism and structural equivalence of boolean functions. In *Proc. 31st ACM Symposium on Theory of Computing*, 652–658. ACM Press, 1999.
10. O. Reingold. Undirected st-connectivity in log-space. In *Proc. 37th ACM Symposium on Theory of Computing*, pages 376–385. ACM Press, 2005.
11. J. Torán. On the hardness of graph isomorphism. *SIAM Journal on Computing*, 33(5):1093–1108, 2004.