

# The Parameterized Complexity of Fixing Number and Vertex Individualization in Graphs\*

Vikraman Arvind<sup>1</sup>, Frank Fuhlbrück<sup>2</sup>, Johannes Köbler<sup>3</sup>,  
Sebastian Kuhnert<sup>4</sup>, and Gaurav Rattan<sup>5</sup>

- 1 The Institute of Mathematical Sciences, Chennai, India  
arvind@imsc.res.in
- 2 Institut für Informatik, Humboldt-Universität zu Berlin, Germany  
fuhrbfra@informatik.hu-berlin.de
- 3 Institut für Informatik, Humboldt-Universität zu Berlin, Germany  
koebler@informatik.hu-berlin.de
- 4 Institut für Informatik, Humboldt-Universität zu Berlin, Germany  
kuhnert@informatik.hu-berlin.de
- 5 The Institute of Mathematical Sciences, Chennai, India  
grattan@imsc.res.in

---

## Abstract

In this paper we study the complexity of the following problems:

1. Given a colored graph  $X = (V, E, c)$ , compute a minimum cardinality set of vertices  $S \subset V$  such that no nontrivial automorphism of  $X$  fixes all vertices in  $S$ . A closely related problem is computing a minimum base  $S$  for a permutation group  $G \leq S_n$  given by generators, i.e., a minimum cardinality subset  $S \subset [n]$  such that no nontrivial permutation in  $G$  fixes all elements of  $S$ . Our focus is mainly on the *parameterized complexity* of these problems. We show that when  $k = |S|$  is treated as parameter, then both problems are MINI[1]-hard. For the dual problems, where  $k = n - |S|$  is the parameter, we give FPT algorithms.
2. A notion closely related to fixing is called individualization. Individualization combined with the Weisfeiler-Leman procedure is a fundamental technique in algorithms for Graph Isomorphism. Motivated by the power of individualization, in the present paper we explore the complexity of individualization: what is the minimum number of vertices we need to individualize in a given graph such that color refinement “succeeds” on it. Here “succeeds” could have different interpretations, and we consider the following: It could mean the individualized graph becomes: (a) discrete, (b) amenable, (c) compact, or (d) refinable. In particular, we study the parameterized versions of these problems where the parameter is the number of vertices individualized. We show a dichotomy: For graphs with color classes of size at most 3 these problems can be solved in polynomial time, while starting from color class size 4 they become W[P]-hard.

**1998 ACM Subject Classification** F.2.0 Analysis of Algorithms and Problem Complexity – General

**Keywords and phrases** Parameterized complexity, graph automorphism, fixing number, base size, individualization

**Digital Object Identifier** 10.4230/LIPIcs.MFCS.2016.13

---

\* This work was supported by the Alexander von Humboldt Foundation in its research group linkage program. The third and fourth authors are supported by DFG grant KO 1053/7-2.



© Vikraman Arvind, Frank Fuhlbrück, Johannes Köbler, Sebastian Kuhnert, and Gaurav Rattan; licensed under Creative Commons License CC-BY

41st International Symposium on Mathematical Foundations of Computer Science (MFCS 2016).

Editors: Piotr Faliszewski, Anca Muscholl, and Rolf Niedermeier; Article No. 13; pp. 13:1–13:14

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

## 1 Introduction

A permutation  $\pi$  on the vertex set  $V$  of a (vertex) colored graph  $X = (V, E, c)$  is an *automorphism* if  $\pi$  preserves edges and colors. Uncolored graphs can be seen as the special case where all vertices have the same color. The automorphisms of  $X$  form the group  $\text{Aut}(X)$ , which is a subgroup of the symmetric group  $\text{Sym}(V)$  of all permutations on  $V$ .

A *fixing set* for a colored graph  $X = (V, E, c)$  is a subset  $S$  of vertices such that there is no nontrivial automorphism of  $X$  that fixes every vertex in  $S$ . The *fixing number* of  $X$  is the cardinality of a smallest size fixing set of  $X$ . This notion was independently studied in [10, 15, 16]. A nice survey on this and related topics is by Bailey and Cameron [8].

In this paper, one of the problems of interest is the computational complexity of computing the fixing number of graphs:

► **Problem 1.1.** *k*-RIGID

*Input:* A colored graph  $X$  and an integer  $k$

*Parameter:*  $k$

*Question:* Is there a subset  $S$  of  $k$  vertices in  $V$  such that there are no nontrivial automorphisms of  $X$  that fix each vertex of  $S$ ?

There is a closely related problem that has received some attention. Let  $G \leq S_n$  be a permutation group on  $[n]$ . A *base* of  $G$  is a subset  $S \subset [n]$  such that no nontrivial permutation of  $G$  fixes each point in  $S$ , i.e., the pointwise stabilizer subgroup  $G_{[S]} = \{g \in G \mid i^g = i \forall i \in S\}$  of  $G$  is the trivial subgroup  $\{1\}$ .

► **Problem 1.2.** *k*-BASE-SIZE

*Input:* A generating set for a permutation group  $G$  on  $[n]$  and an integer  $k$

*Parameter:*  $k$

*Question:* Is there a subset  $S \subset [n]$  of size  $k$  such that no nontrivial permutation of  $G$  fixes each point in  $S$ ?

Note that a graph  $X$  is in *k*-RIGID if and only if  $\text{Aut}(X)$  is in *k*-BASE-SIZE.

Computing a minimum cardinality base for  $G \leq S_n$  given by generators is shown to be NP-hard by Blaha [9]. The same paper also gives a polynomial-time  $\log \log n$  factor approximation algorithm for the problem, i.e., the algorithm outputs a base of size bounded by  $b(G) \log \log n$ , where  $b(G)$  denotes the optimal base size. We show that this approximation factor cannot be improved unless  $P = NP$ ; see Theorem 2.7.

In this paper our focus is on the parameterized complexity of these problems. Arvind has shown that *k*-BASE-SIZE is in FPT for transitive groups and groups with constant orbit size [2], and raised the question whether this can be extended to more general permutation groups. We show that both *k*-RIGID and *k*-BASE-SIZE are MINI[1]-hard, even when the automorphism group of the given graph  $X$  (resp., the given group  $G$ ) is an elementary 2-group; see Section 2.

We also consider the dual problems  $(n - k)$ -RIGID and  $(n - k)$ -BASE-SIZE, which ask whether the given graph or group have a fixing set or base that consists of all but  $k$  vertices or points and  $k$  is the parameter. We show that these problems are fixed parameter tractable. More precisely, we give an  $k^{O(k^2)} + n^{O(1)}$  time algorithm for  $(n - k)$ -BASE-SIZE and an  $k^{O(k^2)} n^{O(1)}$  time algorithm for  $(n - k)$ -RIGID in Section 3.

**Color refinement and individualization.** A broader question that arises is in the context of the Graph Isomorphism problem: Given two colored graphs  $X = (V, E, c)$  and  $X' =$

$(V', E', c')$  the problem is to decide if they are *isomorphic*, i.e., whether there is a bijection  $\pi: V \rightarrow V'$  such that for all  $x \in V$ ,  $c'(x^\pi) = c(x)$  and for all  $x, y \in V$ ,  $(x, y) \in E$  if and only if  $(x^\pi, y^\pi) \in E'$ . Color refinement is a classical heuristic for Graph Isomorphism, and in combination with other tools (group-theoretic/combinatorial) it has proven successful in Graph Isomorphism algorithms (e.g. in the two most important papers in the area [7, 6]). The basic color refinement procedure works as follows on a given colored graph  $X = (V, E, c)$ . Initially each vertex has the color given by  $c$ . The refinement step is to color each vertex by the tuple of its own color followed by the colors of its neighbors (in color-sorted order). The refinement procedure continues until the color classes become stable. If the multisets of colors are different for two graphs  $X$  and  $X'$ , we can conclude that they are not isomorphic. Otherwise, more processing needs to be done to decide if the input graphs are isomorphic. One important approach in this area is to combine *individualization* of vertices with color refinement: Given a graph  $X = (V, E)$  and  $k$  vertices  $v_1, v_2, \dots, v_k \in V$ , first these  $k$  vertices are assigned distinct colors  $c_1, c_2, \dots, c_k$ , respectively. Then, with this as initial coloring, the color refinement procedure is carried out as before. Individualization is used both in the algorithms with the best worst case complexity [7, 6] and in practical isomorphism solvers [21]. Note that individualizing a vertex  $v$  results in fixing  $v$ , as every automorphism must preserve the unique color of  $v$ .

In [5] we have examined several classes of colored graphs in connection with the color refinement procedure. They form a hierarchy:

$$\text{DISCRETE} \subsetneq \text{AMENABLE} \subsetneq \text{COMPACT} \subsetneq \text{REFINABLE} \quad (1)$$

- $X \in \text{DISCRETE}$  if running color refinement on  $X$  results in singleton color classes.
- $X \in \text{AMENABLE}$  if for any  $X'$  that is non-isomorphic to  $X$ , color refinement on  $X$  and  $X'$  results in different stable colorings [5].
- $X \in \text{COMPACT}$  if every fractional automorphism of  $X$  is a convex combination of automorphisms of  $X$  [25]. Here, automorphisms are viewed as permutation matrices that commute with the adjacency matrix  $A$  of  $X$ , and fractional automorphisms are doubly stochastic matrices that commute with  $A$ .
- $X \in \text{REFINABLE}$  if two vertices  $u$  and  $v$  of  $X$  receive the same color in the stable coloring if and only if there is an automorphism of  $X$  that maps  $u$  to  $v$  [5].

For these graph classes, various efficient isomorphism and automorphism algorithms are known. Motivated by the power of individualization in relation to color refinement, we consider the following type of problems.

► **Problem 1.3.**  $k\text{-}\mathcal{C}$  (where  $\mathcal{C}$  is a class of colored graphs)

*Input:* A colored graph  $X = (V, E, c)$  and an integer  $k$

*Parameter:*  $k$

*Question:* Are there  $k$  vertices of  $X$  so that individualizing them results in a graph in  $\mathcal{C}$ ?

It turns out that for each class  $\mathcal{C}$  in the hierarchy (1), the problem  $k\text{-}\mathcal{C}$  is  $W[P]$ -hard, even when the input graph has color class size at most 4. For color class size at most 3 however, the problems become polynomial time solvable. For the class  $\text{DISCRETE}[\ell]$  of all colored graphs where  $\ell$  rounds of color refinement turn all color classes into singletons, we show that  $k\text{-DISCRETE}[\ell]$  is  $W[2]$ -hard. These results are in Section 4.

Additionally, we give an FPT algorithm for the dual problem  $(n - k)\text{-DISCRETE}$  that asks whether there is a way to individualize all but  $k$  vertices so that the input graph becomes discrete; see Section 5.

**Color valence.** A beautiful observation due to Zemlyachenko [27], that plays a crucial role in [7], concerns the color valence of a graph. Given a colored graph  $X = (V, E, c)$ , the *color degree*  $\deg_C(v)$  of a vertex  $v$  in a color class  $C = \{v \in V \mid c(v) = c_0\}$  is the number of neighbors of  $v$  in  $C$ . The *color co-degree* of  $v$  in  $C$  is  $\text{co-deg}_C(v) = |C| - \deg_C(v)$ . The *color valence* of  $X$  is defined as  $\max_{v,C} \min\{\deg_C(v), \text{co-deg}_C(v)\}$ . Zemlyachenko has shown [27] that in any  $n$ -vertex graph  $X = (V, E)$  we can individualize  $O(n/d)$  vertices so that the vertex colored graph obtained after color refinement has color valence at most  $d$ . This gives rise to the following natural algorithmic problem:

► **Problem 1.4.**  $k$ -COLOR-VALENCE

*Input:* A colored graph  $X = (V, E, c)$  and two numbers  $k$  and  $d$

*Parameter:*  $k$

*Question:* Is there a set of  $k$  vertices such that when these are individualized, the graph obtained after color refinement has color valence bounded by  $d$ ?

We show that this problem is  $W[P]$ -complete; see Corollary 4.4.

## 2 The number of fixed vertices as parameter

In this section we show that the parameterized problems  $k$ -RIGID and  $k$ -BASE-SIZE are both  $\text{MINI}[1]$ -hard. The class  $\text{MINI}[1]$  contains all parameterized problems that are FPT-reducible to  $\text{MINI-3SAT}$ . Both were defined in [12, 14].

► **Problem 2.1** ([12, 14]).  $\text{MINI-3SAT}$

*Input:* A formula  $F$  in 3-CNF of size bounded by  $k \log n$  and the number  $n$  in unary

*Parameter:*  $k$

*Question:* Is there a boolean assignment to the variables that satisfies the formula  $F$ ?

It turns out that  $\text{MINI}[1]$  is contained in the class  $W[1]$  [14] and has a variety of complete problems in it. Moreover, it has been linked to the exponential time hypothesis.

► **Lemma 2.2** ([12, 14]). *If  $\text{MINI}[1] = \text{FPT}$  then there is a  $2^{o(n)}$  time algorithm for 3SAT.*

► **Theorem 2.3.** *The problem  $k$ -BASE-SIZE is  $\text{MINI}[1]$ -hard, even for elementary 2-groups.*

**Proof.** It is easy to see that  $\text{MINI-3SAT}$  in which each variable occurs at most 3 times is also  $\text{MINI}[1]$ -complete, by modification of a standard NP-completeness proof. This only increases the size by a constant factor. We will give an FPT many-one reduction from this variant of  $\text{MINI-3SAT}$  to  $k$ -BASE-SIZE. Let  $F = C_1 \wedge C_2 \wedge \dots \wedge C_m$ , and  $n$  in unary, be a  $\text{MINI-3SAT}$  instance with variable occurrences bounded by 3. Since the size of  $F$  is bounded by  $k \log n$ , we have  $m \leq k \log n$ . Let  $V$  denote the set of distinct variables in  $F$ . We also have  $|V| \leq k \log n$ . We partition  $V$  as  $V = \bigsqcup_{i=1}^k V_i$ , where  $|V_i| \leq \log n$  for  $1 \leq i \leq k$ . For each  $i$ , the set  $T_i = \{0, 1\}^{V_i}$  consisting of all truth assignments to variables in  $V_i$  has size  $|T_i| \leq n$ . Define the universe  $U = \{1, 2, \dots, m, m+1, \dots, m+k\}$ . For each truth assignment  $a \in T_i$  we define the subset  $S_{i,a} \subset U$  consisting of  $m+i$  along with all  $j$  such that  $a$  satisfies  $C_j$ , i.e.,

$$S_{i,a} = \{m+i\} \cup \{j \mid C_j \text{ contains a literal that is true under } a\}.$$

Clearly, since each variable occurs at most 3 times in  $F$  and since  $|a| = |V_i| \leq \log n$ , it follows that  $|S_{i,a}| \leq 1 + 3 \log n$ . The following claim is straightforward.

► **Claim 2.4.** The collection of sets  $\{S_{i,a} \mid 1 \leq i \leq k, a \in T_i\}$  with universe  $U$  has a set cover of size  $k$  if and only if  $F$  is satisfiable.

We will now transform this special set cover instance into an instance of  $k$ -BASE-SIZE. The group we shall consider is  $\mathbb{F}_2^{m+k}$ , i.e., the product of  $m+k$  copies of the group on  $\{0, 1\}$  defined by addition modulo 2. Treating each set  $S_{i,a}$  as a subset of the coordinates  $1, 2, \dots, m+k$ , we can associate a copy of  $\mathbb{F}_2^{|S_{i,a}|}$  with it. Consider the set  $\Omega = \bigsqcup_{i,a} \mathbb{F}_2^{|S_{i,a}|}$ . Note that  $|\Omega| = \sum_{i,a} 2^{|S_{i,a}|} \leq nk$ . The group  $\mathbb{F}_2^{m+k}$  acts faithfully on  $\Omega$  as follows. Given an element  $u \in \mathbb{F}_2^{m+k}$  and a point  $v \in \mathbb{F}_2^{|S_{i,a}|}$ , let  $u_{i,a}$  denote the projection of  $u$  to the coordinates in  $S_{i,a}$ . Then  $u$  maps  $v$  to  $v \oplus u_{i,a}$ . Thus,  $\mathbb{F}_2^{m+k}$  is a permutation group acting on  $\Omega$  given by the standard basis of  $m+k$  unit vectors as generating set. The following straightforward claim completes the reduction.

► **Claim 2.5.** The group  $\mathbb{F}_2^{m+k}$  acting on  $\Omega$ , as defined above, has a base of size  $k$  if and only if the set cover instance  $(U, \{S_{i,a} \mid 1 \leq i \leq k, a \in T_i\})$  has a set cover of size  $k$ .

To see the claim, observe that  $V \subseteq \Omega$  is a base if and only if the sets  $S_{i,a}$  with  $V \cap \mathbb{F}_2^{|S_{i,a}|} \neq \emptyset$  form a set cover for  $U$ . Indeed, a point  $p \in U$  is covered by these sets if and only if all  $u \in \mathbb{F}_2^{m+k}$  with  $u_p = 1$  move an element of  $V$ . ◀

► **Theorem 2.6.** *The problem  $k$ -RIGID is MINI[1]-hard.*

**Proof.** It suffices to encode the  $k$ -BASE-SIZE instance constructed in the proof of Theorem 2.3 as a  $k$ -RIGID instance  $(X, k)$  with the following properties. The graph  $X$  has  $|\Omega| + 2(m+k)$  vertices and at most  $|\Omega|(1 + 3 \log n)$  edges. Further, the above  $k$ -BASE-SIZE instance has a base of size  $k$  if and only if the graph  $X$  has a fixing set of size  $k$ .

We explain the construction of  $X$ . Let  $l = m+k$ . The vertex set of  $X$  is  $\Omega \cup I_1 \cup \dots \cup I_l$  where each set  $I_j = \{a_j^0, a_j^1\}$  is a distinct color class of size 2. The edge set of  $X$  is defined as follows. Let  $v = (b_1, \dots, b_p) \in \mathbb{F}_2^{|S_{i,a}|}$  be a vertex in  $\Omega$  and let  $S_{i,a} = \{i_1, i_2, \dots, i_p\}$  be the set of coordinates occurring in  $v$ . Then we connect  $v$  to the vertices  $a_{i_q}^{b_q}$ , for each  $q = 1, \dots, p$ . This finishes the construction of  $X$ .

We claim a one-to-one correspondence between the permutation group  $\mathbb{F}_2^{m+k}$  acting on  $\Omega$  and  $\text{Aut}(X)$ . Indeed, any vector  $v = (b_1, \dots, b_l) \in \mathbb{F}_2^{m+k}$  can be associated with a unique automorphism  $\sigma$  of  $X$  as follows. The automorphism  $\sigma$  flips the color class  $I_j$  if and only if  $b_j = 1$ . For a vertex  $u \in \Omega$ , define  $\sigma(u) = v(u)$  using the action of  $\mathbb{F}_2^{m+k}$  on  $\Omega$ . It is easy to check that  $\sigma$  respects the adjacencies inside  $X$ . Note that the action of an automorphism of  $X$  is determined by its action on  $I_1, \dots, I_l$ , so this is a one-to-one correspondence.

Consequently, a set  $J \subset \Omega$  is a base for the original  $k$ -BASE-SIZE instance if and only if  $J$  is a fixing set for the graph  $X$ . We observe that we can always avoid fixing a vertex  $u$  inside  $I_1 \cup \dots \cup I_l$  by instead fixing some neighbor of  $u \in \Omega$ . Therefore, the original  $k$ -BASE-SIZE instance has a base of size  $k$  if and only if the graph  $X$  has a fixing set of size  $k$ . ◀

We end this section with some consequences of our hardness proofs on the approximability of the minimum base size of a group. There is a  $\log \log n$  factor approximation algorithm due to Blaha [9] for the minimum base problem (in fact, a careful analysis yields a  $\ln \ln n$ -factor approximation). In this connection we have an interesting observation about the set cover problem instances that arise in Theorem 2.3 (Claim 2.4). A more general version is the  $B$ -SET-COVER problem: we are given a collection of subsets of size at most  $B$  of some universe  $U$  and the problem is to find a minimum size set cover. Trevisan [26] has shown that there is no approximation algorithm for this problem with approximation factor smaller than  $\ln B - O(\ln \ln B)$  unless  $\text{P} = \text{NP}$ . This leads us to the following theorem.

► **Theorem 2.7.** *The approximation factor of  $\ln \ln n$  in Blaha's approximation algorithm for minimum base cannot be improved, even for elementary abelian 2-groups, unless  $\text{P} = \text{NP}$ .*

**Proof.** The reduction from  $(\log n)$ -SET-COVER to the minimum base problem that is explained in the proof of Theorem 2.3 preserves the optimal solution size. Furthermore, it is easy to see that this reduction carries over to all  $(\log n)$ -SET-COVER instances. Combined with Trevisan's result, this completes the proof.  $\blacktriangleleft$

### 3 The number of non-fixed vertices as parameter

In this section we show that the problems  $(n - k)$ -RIGID and  $(n - k)$ -BASE-SIZE are in FPT with running time  $k^{O(k^2)}n^{O(1)}$ . We will show this first for  $(n - k)$ -BASE-SIZE. We need some permutation group theory.

Let  $G \leq \text{Sym}(\Omega)$  be a permutation group acting on a set  $\Omega$ . The *support* of a permutation  $g \in G$  is  $\text{supp}(g) = \{i \in \Omega \mid i^g \neq i\}$ . The *orbit* of a point  $i \in \Omega$  is the set  $i^G = \{i^g \mid g \in G\}$ . The group  $G$  is *transitive* if it has a single orbit in  $\Omega$ . Let  $G \leq \text{Sym}(\Omega)$  be transitive. A subset  $\Delta \subseteq \Omega$  is a *block* if for every  $g \in G$  its image  $\Delta^g = \{i^g \mid i \in \Delta\}$  is either  $\Delta^g = \Delta$  or  $\Delta^g \cap \Delta = \emptyset$ . Clearly,  $\Omega$  and singleton sets are blocks for any  $G$ . All other blocks are called nontrivial. A transitive group  $G$  is *primitive* if it has no nontrivial blocks.

There are polynomial-time algorithms that take as input a generating set for some  $G \leq \text{Sym}(\Omega)$  and compute its orbits and maximal nontrivial blocks [19]. We can test if  $G$  is primitive in polynomial time. If  $G$  is transitive on  $\Omega$  we can compute a maximal nontrivial block  $\Delta_1$ . It is easy to see that  $\Delta_1^g$  is also a block for each  $g \in G$ . This yields a partition of  $\Omega$  into blocks (which are said to constitute a block system for  $G$ ):  $\Omega = \Delta_1 \sqcup \Delta_2 \sqcup \dots \sqcup \Delta_\ell$ . The group  $G$  acts transitively on the blocks  $\{\Delta_1, \Delta_2, \dots, \Delta_\ell\}$ . Furthermore, since these are maximal blocks, the group action is primitive. The following classic result is useful for our algorithm.

► **Lemma 3.1.** [13, Lemma 3.3D] *Suppose  $G \leq S_n$  is primitive and  $G$  is neither  $A_n$  nor  $S_n$  itself. If there is an element  $g \in G$  such that  $|\text{supp}(g)| \leq k$ , then  $|\Omega| \leq (k - 1)^{2k}$ .*

Here,  $A_n = \text{Alt}([n])$  denotes the subgroup of  $S_n$  that consists of those permutations that can be written as the product of an even number of transpositions.

► **Theorem 3.2.** *There is a  $k^{O(k^2)} + n^{O(1)}$  time algorithm for the  $(n - k)$ -BASE-SIZE problem.*

**Proof.** Let  $G \leq S_n$  be the input group given by a generating set and let  $k$  be the parameter. We call a set  $S \subseteq [n]$  a *co-base* for  $G$ , if  $[n] \setminus S$  is a base for  $G$ . The algorithm finds a co-base  $S$  of size  $k$  if it exists. During its execution, the algorithm may decide to fix some points. Since in this case the actual group  $G$  is replaced by the pointwise stabilizer subgroup, there is no need to store these points. The algorithm proceeds as follows.

1. Let  $O_1, O_2, \dots, O_\ell$  be the orbits of the group  $G$ . If  $\ell \geq k$  then the set  $S$  obtained by picking one point from each of the orbits  $O_1, O_2, \dots, O_k$  is a co-base for  $G$ .
2. Suppose  $\ell < k$ , and there is an orbit  $O_i$  of size more than  $k^{2k}$  on which  $G$ 's action is not primitive. In this case compute a maximal block system of  $G$  in  $O_i$ ,  $O_i = \Delta_{i1} \sqcup \dots \sqcup \Delta_{ir_i}$ , and deal with the following subcases:
  - a. If  $r_i > k$ , then the set  $S$  obtained by picking one point from each block  $\Delta_{i1}, \dots, \Delta_{ik}$  is a co-base for  $G$ .
  - b. If  $r_i \leq k$ , then each block  $\Delta_{ij}$  is of size at least  $k^{2k-1}$  which is strictly more than  $k$ . Thus any  $n - k$  sized subset of  $[n]$  intersects each block  $\Delta_{ij}$  and hence the support of any permutation that moves the blocks. Let  $H$  be the subgroup of  $G$  that setwise stabilizes all blocks  $\Delta_{ij}$ . The subgroup  $H$  can be computed from  $G$  in polynomial time

using the Schreier-Sims algorithm [19]. Replace  $G$  by  $H$  and go to Step 1. This step is invoked at most  $k$  times since each invocation increases the number of orbits.

3. Suppose  $\ell < k$ , and there is an orbit  $O_i$  of size more than  $k^{2k}$  such that  $G$  is primitive on  $O_i$ , but different from  $\text{Sym}(O_i)$  and  $\text{Alt}(O_i)$ . Then any  $k$  points of  $O_i$  form a co-base for  $G$  (by Lemma 3.1).
4. Suppose there is an orbit  $O_i$  of size more than  $k^{2k}$  such that  $G$  restricted to  $O_i$  is either  $\text{Sym}(O_i)$  or  $\text{Alt}(O_i)$ . Then fix the first  $|O_i| - k$  elements of  $O_i$  (the choice of the subset of points fixed does not matter as both  $\text{Sym}(O_i)$  and  $\text{Alt}(O_i)$  are  $t$ -transitive for  $t \leq |O_i| - 2$ ). Replace  $G$  by the subgroup  $H$  that fixes the first  $|O_i| - k$  elements of  $O_i$  and go to Step 1. This step is invoked at most once.
5. This step is only reached if all orbits are of size at most  $k^{2k}$ , implying that the entire domain size is at most  $k^{2k+1}$ . Hence, the algorithm can find a co-base  $S$  of size  $k$  by brute-force search in  $k^{O(k^2)}$  time if it exists.

The brute-force computation (done in the last step), when the search space is bounded by  $k^{2k+1}$ , costs  $k^{O(k^2)}$ . The rest of the computation uses the standard group-theoretic algorithms [19] whose running time is polynomially bounded by  $n$ . Therefore, the overall running time of the algorithm is bounded by  $k^{O(k^2)} + kn^{O(1)}$ . As  $k \leq n$ , the theorem follows.

We note that the algorithm is in fact a kernelization algorithm. It computes in  $n^{O(1)}$  time a kernel of size  $k^{2k+1}$  (where size refers to the size of the domain on which the group acts). ◀

We now show the main result of this section, i.e., that  $(n - k)$ -RIGID is in FPT.

► **Theorem 3.3.** *There is a  $k^{O(k^2)}n^{O(1)}$  time algorithm for the  $(n - k)$ -RIGID problem.*

**Proof.** Let  $X = (V, E, c)$  be a colored  $n$ -vertex graph and  $k$  as parameter be an instance of  $(n - k)$ -RIGID. If we can use a subroutine for the Graph Isomorphism problem then we can compute a generating set for the automorphism group  $\text{Aut}(X)$  of  $X$  with polynomially many calls to this subroutine [20]. With this generating set as input we can then run the algorithm of Theorem 3.2 to compute an  $(n - k)$  size fixing set for  $X$ , if it exists, in time  $k^{O(k^2)}n^{O(1)}$ .

However, it turns out that we can avoid using the Graph Isomorphism subroutine and still solve the problem in  $k^{O(k^2)}n^{O(1)}$  time with the following observations:

1. We note that any set of size  $n - k$  will intersect the support of any element  $\sigma \in \text{Aut}(X)$  if  $|\text{supp}(\sigma)| > k$ . Thus, we only need to collect all elements of support bounded by  $k$ .
2. An automorphism  $\sigma \in \text{Aut}(X)$  is defined to be a *minimal support* automorphism of  $X$  if there is no nontrivial automorphism  $\varphi \in \text{Aut}(X)$  such that  $\text{supp}(\varphi) \subsetneq \text{supp}(\sigma)$ . For any nontrivial automorphism  $\pi \in \text{Aut}(X)$  such that  $|\text{supp}(\pi)| \leq k$ , there is a minimal support automorphism  $\varphi \in \text{Aut}(X)$  such that  $|\text{supp}(\varphi)| \leq k$  and  $\text{supp}(\varphi) \subseteq \text{supp}(\pi)$ .
3. We observe that Schweitzer's algorithm in [24] can be used to compute, in  $k^{O(k)}n^{O(1)}$  time, the set  $M$  of all minimal support automorphisms  $\sigma \in \text{Aut}(X)$  such that  $|\text{supp}(\sigma)| \leq k$ .
4. Let  $G'$  be the subgroup of  $\text{Aut}(X)$  generated by  $M$ . It follows from the above discussion that an  $n - k$  sized subset of  $V$  is a base for  $\text{Aut}(X)$  (and thus a fixing set for  $X$ ) if and only if it is a base for  $G'$ . We can apply the algorithm of Theorem 3.2 to compute such a base if it exists. ◀

## 4 The number of individualized vertices as parameter

In this section, we show that the problem  $k$ - $\mathcal{C}$  is W[P]-hard for all classes  $\mathcal{C}$  of the color refinement hierarchy (1). To this end, we give a reduction from WEIGHTED MONOTONE CIRCUIT SATISFIABILITY, which is known to be W[P]-complete [1].

► **Problem 4.1.** WEIGHTED MONOTONE CIRCUIT SATISFIABILITY

*Input:* A monotone boolean circuit  $C$  on  $n$  inputs and an integer  $k$

*Parameter:*  $k$

*Question:* Is there an assignment  $x \in \{0, 1\}^n$  of Hamming weight  $k$  so that  $C(x) = 1$ ?

► **Theorem 4.2.** For all classes  $\mathcal{C}$  of the color refinement hierarchy (1),  $k$ - $\mathcal{C}$  is  $\text{W[P]}$ -hard, even for graphs of color class size at most 4.

**Proof.** We will give a parameter-preserving reduction that maps positive instances of WEIGHTED MONOTONE CIRCUIT SATISFIABILITY to positive instances of  $k$ -DISCRETE, while negative instances are mapped to negative instances of  $k$ -REFINABLE. A similar reduction was used to show that the classes from the color refinement hierarchy (1) are all P-hard [5], which in turn builds on ideas of Grohe [17].

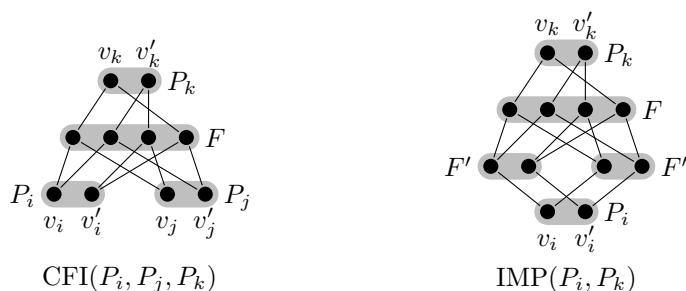
Let  $\langle C, k \rangle$  be the given instance of WEIGHTED MONOTONE CIRCUIT SATISFIABILITY, and let  $n$  be the number of inputs of the circuit  $C$ . We define a graph  $X_C$ . For each gate  $g_k$  of  $C$  (including the input gates),  $X_C$  contains a vertex pair  $P_k = \{v_k, v'_k\}$ , which forms a color class. If a pair corresponds to an input gate, we call it an *input pair*. The intention is that setting an input  $g_i$  to 1 corresponds to individualizing the vertex  $v_i$ ; we will add gadgets to  $X_C$  so that after color refinement it holds also for each non-input gate  $g_k$  that  $g_k = 1$  if and only if  $v_k$  and  $v'_k$  have different colors.

To achieve this, we use the gadgets given in Figure 1. The basic building block is the gadget  $\text{CFI}(P_i, P_j, P_k)$  introduced by Cai, Fürer, and Immerman [11]. It connects the three pairs  $P_i$ ,  $P_j$ , and  $P_k$  using four additional vertices as depicted. These four vertices form a color class  $F$ ; each instance of the gadget uses its own copy of  $F$ . This gadget has the property that every automorphism flips either none or exactly two of the pairs  $P_i$ ,  $P_j$  and  $P_k$ ; thus the CFI-gadget implements the XOR function in the sense that any automorphism must flip  $P_k$  if and only if it flips exactly one of  $P_i$  and  $P_j$ . In our case, however, the CFI-gadget implements the AND function: If both  $P_i$  and  $P_j$  are distinguished (either by direct individualization or in previous rounds of color refinement), the vertices of the inner color class  $F$  and consequently  $P_k$  will be distinguished in two rounds of color refinement. Conversely, if at most one of the pairs  $P_i$  and  $P_j$  is distinguished, then the color class  $F$  is split into two color classes of size 2 and color refinement stops at this point, leaving the other two pairs non-distinguished. For each AND gate  $g_k = g_i \wedge g_j$  in  $C$ , we add the gadget  $\text{CFI}(P_i, P_j, P_k)$  to  $X_C$ .

The second gadget we use is  $\text{IMP}(P_i, P_k)$ . It consists of the gadget  $\text{CFI}(F', F'', P_k)$ , where  $F'$  and  $F''$  are vertex pairs that form color classes of size two, and perfect matchings that connect these pairs to  $P_i$ ; see Fig. 1. Again, each instance of this gadget gets its own copy of the color classes  $F$ ,  $F'$  and  $F''$ . There is an automorphism of  $\text{IMP}(P_i, P_k)$  that flips the vertices in  $P_i$ , but none that flips the vertices in  $P_k$ . In the color refinement setting, this gadget implements the IMPLICATION function: When  $P_i$  is distinguished, this will propagate to both  $F'$  and  $F''$ , and consequently also to  $F$  and  $P_k$ . Conversely, distinguishing  $P_k$  will only split  $F$  into two color classes of size 2 before color refinement stops. For each OR gate  $g_k = g_i \vee g_j$  in  $C$ , we add the gadgets  $\text{IMP}(P_i, P_k)$  and  $\text{IMP}(P_j, P_k)$  to  $X_C$ . For the output gate  $g_\ell$  of  $C$ , we add a second vertex pair  $Q$  and the gadget  $\text{IMP}(P_\ell, Q)$  to  $X_C$ .

Our above analysis of the gadgets ensures that the following invariant holds when running color refinement on  $X_C$  after individualizing a subset of its input pairs: For each IMPLICATION gadget  $\text{IMP}(P_i, P_k)$  in  $X_C$  the pair  $P_k$  can only be distinguished if  $P_i$  is distinguished, and for each AND gadget  $\text{CFI}(P_i, P_j, P_k)$  the pair  $P_k$  can only be distinguished if both  $P_i$  and  $P_j$  are distinguished. This implies the following.





■ **Figure 1** Gadgets used in the reduction of Theorem 4.2.

► **Claim 4.3.** Running color refinement on  $X_C$  after individualizing some input pairs will distinguish exactly those pairs  $P_k$  for which the gate  $g_k$  evaluates to 1 under the assignment that sets exactly those input gates to 1 whose corresponding pairs were initially individualized.

Let  $X'_C$  be the graph that is obtained from  $X_C$  by adding implication gadgets from  $Q$  to each pair  $P_i$  that corresponds to an input gate  $g_i$ . If  $C$  has a satisfying assignment  $x \in \{0, 1\}^n$  of weight  $k$ , individualizing the vertices  $v_i$  with  $x_i = 1$  and subsequently running color refinement will assign distinct colors to all vertices of  $X_C$ . Indeed, the gadgets of  $X_C$  ensure that the pair  $Q$  becomes distinguished, the additional gadgets in  $X'_C$  propagate this to all input pairs  $P_i$ , and the gates of  $X_C$  in turn make sure that all remaining color classes become distinguished. Conversely, if  $C$  does not have a weight  $k$  satisfying assignment, there is no way to individualize  $k$  input vertices such that color refinement distinguishes  $Q$ . However, we already noted that there is no automorphism that transposes the output pair of the  $\text{IMP}(P_\ell, Q)$  gadget, so no way of individualizing  $k$  input vertices makes  $X'_C$  refinable.

In  $X'_C$ , it always suffices to individualize one vertex from  $Q$  to make it discrete. To drop the assumption that each of the  $k$  individualized vertices must correspond to an input gate, we construct a graph  $X''_C$ . It consists of  $n$  input pairs  $P_i = \{v_i, v'_i\}$  and  $n$  copies of  $X_C$ , to which we will refer to as  $X_C^{(1)}, \dots, X_C^{(n)}$ . We also add the gadgets  $\text{IMP}(P_i, P_i^{(j)})$  for all  $i, j \in \{1, \dots, n\}$  and the gadgets  $\text{IMP}(Q^{(i)}, P_i)$  for all  $i \in \{1, \dots, n\}$ . It is not hard to see that  $\langle C, k \rangle \mapsto \langle X''_C, k \rangle$  is the desired reduction; see the full version [3] for its correctness. ◀

As a corollary to this proof we can derive the  $\text{W[P]}$ -hardness of the  $k$ -COLOR-VALENCE problem.

► **Corollary 4.4.**  $k$ -COLOR-VALENCE is  $\text{W[P]}$ -hard.

**Proof.** In the previous reduction we mapped instances of WEIGHTED MONOTONE CIRCUIT SATISFIABILITY to instances of  $k$ -DISCRETE such that the given boolean circuit  $C$  has a satisfying assignment of weight  $k$  if and only if the resulting graph  $X''_C$  can be made discrete by individualizing  $k$  vertices. Note that individualizing  $k$  vertices in  $X''_C$  and subsequently running color refinement results in singleton color classes if and only if it brings the color valence down to 0. Thus,  $k$ -COLOR-VALENCE is  $\text{W[P]}$ -hard even for  $d = 0$ . ◀

### 4.1 Graphs of color class size at most 3

We call a vertex-colored graph  $b$ -bounded if all its color classes are of size at most  $b$ . In this section, we show that for any 3-bounded graph, we can compute in polynomial time the minimum number of vertices that have to be individualized so that the resulting colored

graph becomes rigid, discrete, amenable, compact, or refinable. We will use the following two lemmas; their proofs can be found in the full version of this article [3].

► **Lemma 4.5.** *Let  $X$  be a 3-bounded graph whose color classes are stable. If  $\text{Aut}(X)$  restricted to any color class  $C_i$  of  $X$  is the full symmetric group on  $C_i$ , then  $X$  is compact.*

► **Lemma 4.6.** *Let  $X$  be a connected 3-bounded graph whose color classes are stable. If some  $\sigma \in \text{Aut}(X)$  is cyclic (i.e.,  $\sigma$  acts cyclically on each color class  $C_i$ ), then  $X$  is compact.*

► **Theorem 4.7.** *For any 3-bounded graph we can compute in polynomial time a vertex set  $S$  of minimum size such that individualizing (or fixing) all the vertices in  $S$  makes the graph discrete, amenable, compact, refinable (or rigid).*

**Proof.** Let  $X = (V, E, c)$  be the given 3-bounded graph. We first compute the color partition  $\{C_1, \dots, C_m\}$  of the stable coloring of  $X$ . We can assume that each induced graph  $X_i = X[C_i]$  is empty and each induced bipartite graph  $X_{ij} = X[C_i, C_j]$  has at most  $|C_i| \cdot |C_j|/2$  edges, as otherwise we can complement these subgraphs. Since the partition  $\{C_1, \dots, C_m\}$  is stable and the color classes have size at most 3, it follows that there are no edges between color classes having different sizes, and that between color classes  $C_i$  and  $C_j$  of the same size we either have a perfect matching or no edges at all.

We say that two color classes  $C_i$  and  $C_j$  are *linked* if there is a path between some vertex  $u \in C_i$  and some vertex  $v \in C_j$ . Since this is an equivalence relation, it partitions the color classes into equivalence classes. This induces a partition  $V = V_1 \sqcup \dots \sqcup V_l$  of the vertices such that each set  $V_i$  is a union of linked color classes having the same size and there are no edges between  $V_i$  and  $V_j$  whenever  $i \neq j$ . Hence, it suffices to solve the problem separately for each of the induced subgraphs  $X[V_i]$ .

If all color classes of  $X[V_i]$  are of size 2, then  $\text{Aut}(X[V_i])$  contains exactly one non-trivial automorphism flipping all the color classes, implying that  $X[V_i]$  is compact (see Lemma 4.5). In this case it suffices to individualize (or fix) an arbitrary vertex to make the graph discrete (or rigid). Further,  $X[V_i]$  is already amenable if and only if it is a forest [4].

If all color classes of  $X[V_i]$  are of size 3, then we compute its connected components as well as  $\text{Aut}(X[V_i])$  (which is even possible in logspace [18, 23]) and consider the following subcases.

- If  $X[V_i]$  has 6 automorphisms (or, equivalently, consists of three connected components), then  $X[V_i]$  is compact (see Lemma 4.5) and it suffices to individualize two vertices inside an arbitrary color class to make the graph discrete. On the other hand, if we individualize only one vertex, then the graph does not become discrete (not even rigid). Further,  $X[V_i]$  is amenable if and only if it is a forest [4]. If  $X[V_i]$  contains cycles then we need to individualize 2 vertices to make the graph amenable.
- If  $X[V_i]$  has 3 automorphisms, then it follows that these automorphisms act cyclically on each color class and  $X[V_i]$  is connected as well as compact (see Lemma 4.6). In this case it suffices to individualize an arbitrary vertex to make the graph discrete.
- If  $X[V_i]$  has 2 automorphisms (or, equivalently, consists of two connected components), then  $X[V_i]$  is not refinable and it suffices to individualize an arbitrary vertex in the larger of the two components to make the graph discrete.
- Finally, if  $X[V_i]$  is rigid, then it follows that  $X[V_i]$  is connected and not refinable. In this case it suffices to individualize an arbitrary vertex to make the graph discrete. ◀

We can actually strengthen Theorem 4.7 and show that these problems are in logspace. Since the case analysis in the proof can be done in logspace, it suffices to show that the stable

color partition of a 3-bounded graph can be computed in logspace. The proof of this result is given in the full version of this article [3].

## 4.2 Bounded number of refinement steps

In this section, we consider (colored) graphs in which all color classes become singletons after  $\ell$  rounds of color refinement. We denote the class of these graphs by  $\text{DISCRETE}[\ell]$ .

► **Theorem 4.8.** *The  $k$ -DISCRETE $[\ell]$  problem is  $W[2]$ -hard for any constant  $\ell \geq 1$ , even for uncolored and for 2-bounded graphs.*

**Proof.** We prove this by providing a reduction from the  $W[2]$ -complete problem DOMINATING SET that is inspired by [22, Theorem 7]. The input to this problem is a graph  $X = (V, E)$  and a number  $k$  (treated as parameter) and the question is whether there exists a *dominating set*  $D \subseteq V$  of size  $k$  in  $X$ , meaning that each vertex  $v \in V \setminus D$  is adjacent to at least one vertex in  $D$ . We transform the DOMINATING SET instance  $(X, k)$  with  $X = (V, E)$  into an equivalent instance  $(X', k)$  where  $X = (V', E', c')$  for  $k$ -DISCRETE $[\ell]$ . For every  $v \in V$ , the colored graph  $X'$  contains the vertices  $v_1, \dots, v_\ell$  and  $v'_1, \dots, v'_\ell$  as well as the edges  $\{v_i, v_{i+1}\}$  and  $\{v'_i, v'_{i+1}\}$  for all  $i$  in  $\{1, \dots, \ell - 1\}$ . Furthermore, we add the edges  $\{v_1, u_1\}$  and  $\{v'_1, u'_1\}$  for every edge  $\{u, v\}$  of  $X$ . We choose  $c'$  in such a way that for all  $v \in V$  the set  $\{v_1, v'_1\}$  is a color class and  $c'(v_i) = c'(v'_i)$  for all  $i \in \{2, \dots, \ell\}$ .

Let  $D$  be a dominating set in  $X$ . Individualizing all the vertices  $v_1$  in  $X'$  with  $v \in D$  will distinguish the pairs  $\{v_1, v'_1\}$  for all  $v \in V$  after one round of color refinement. Thus after at most  $\ell - 1$  more rounds all color classes of  $X'$  will be singletons.

For the converse direction, let  $I$  be a set of vertices in  $X'$ , such that individualizing them and running  $\ell$  rounds of color refinement produces singleton color classes. If  $I$  contains vertices  $v_i$  or  $v'_i$  for  $i > 1$ , we can replace them by  $v_1$  and this still puts  $X'$  in  $\text{DISCRETE}[\ell]$ . It is easy to see that this replacement does not decrease the number of color classes that become singletons after  $\ell$  rounds. Hence, we can assume that  $I$  only contains vertices of the form  $v_1$ , implying that the set  $D = \{v \in V \mid v_1 \in I\}$  is a dominating set of size at most  $|I|$  in  $X$ . To see this it suffices to observe that the vertices  $u_\ell$  and  $u'_\ell$  can only be distinguished by color refinement within  $\ell$  rounds if either  $u_1$  is in  $I$  or  $u$  has a neighbor  $v$  for which  $v_1$  is in  $I$ , implying that either  $u$  or some neighbor of  $u$  is in  $D$ .

For uncolored graphs we simulate the colors with degrees using a global gadget which in turn is distinguished from the rest by four special vertices of which two need to be individualized in any case. This increases the parameter from  $k$  to  $k + 2$ . See the full version [3] for details. ◀

## 5 The number of non-individualized vertices as parameter

In this section, we show that the problem  $(n - k)$ -DISCRETE is in FPT. In fact, we show a linear kernel and consequently, a  $k^{O(k)} n^{O(1)}$  time algorithm for this problem.

► **Theorem 5.1.** *There exists a kernel of size  $2k$  for  $(n - k)$ -DISCRETE that can be computed in polynomial time.*

We begin with some notation. Given a colored graph  $X = (V, E, c)$ , let  $S$  be a subset of vertices. Let  $\mathcal{C}[S]$  denote the stable partition obtained by individualizing every vertex in  $V \setminus S$  and performing color refinement. We denote the number of color classes in  $\mathcal{C}[S]$  by  $|\mathcal{C}[S]|$ . We can partition the vertices  $u$  in  $V \setminus S$  by their neighborhood  $N(u) \cap S$  inside the set  $S$ . We denote this partition of  $V \setminus S$  by  $\mathcal{N}[S]$  and the number of sets in it by  $|\mathcal{N}[S]|$ . We call two

vertices  $u$  and  $v$  *twins* if  $N(u) \setminus \{v\} = N(v) \setminus \{u\}$ . This relation is an equivalence relation and the corresponding equivalence classes are called *twin classes*. A graph is said to be *twin-free* if each twin class is of size 1.

The following lemma shows that sufficiently large twin-free graphs are YES instances of the  $(n - k)$ -DISCRETE problem.

► **Lemma 5.2.** *Let  $X = (V, E)$  be a twin-free graph. Suppose  $|V| > 2k$ . There exists a set  $S \subset V$  of size  $k$  such that  $\mathcal{C}[S]$  is discrete. Moreover, we can compute such a set in  $(nk)^{O(1)}$  time.*

**Proof.** We describe the algorithm for computing  $S$ . Initially, we pick an arbitrary subset  $T_0 \subset V$  of size  $k$  and run color refinement to compute the stable partition  $\mathcal{C}[T_0]$ . Let  $C_1, \dots, C_l$  be the color classes in  $\mathcal{C}[T_0]$  that are contained in  $T_0$ . If  $\mathcal{C}[T_0]$  is already discrete, we output the set  $S = T_0$  and stop.

Otherwise we rename the color classes such that  $|C_1| \geq |C_i|$  for  $i = 2, \dots, l$ . Then we compute the partition  $\mathcal{N}[S] = \{B_1, \dots, B_m\}$  of  $V \setminus S$ , where we assume that  $|B_1| \geq |B_i|$  for  $i = 2, \dots, m$ . If  $m \geq k$ , then we form  $S$  by picking an arbitrary vertex from each of the sets  $B_1, \dots, B_k$ . To see that  $\mathcal{C}[S]$  is discrete it suffices to observe that individualizing all the vertices in  $V \setminus S$  causes the separation of the sets  $B_1, \dots, B_m$  and individualizing all but at most one vertex in each set  $B_i$  makes the graph discrete.

It remains to handle the case that  $m < k$ . We show that in this case it is possible to compute in polynomial time a set  $T_1$  of size  $k$  such that  $|\mathcal{C}[T_1]| > |\mathcal{C}[T_0]|$ . By repeating this procedure  $i \leq k - 1$  times, we end up with a set  $T_i$  for which  $\mathcal{C}[T_i]$  is discrete. Let  $u$  and  $v$  be two vertices inside the color-class  $C_1$ . Since  $X$  is twin-free, there must be a vertex  $a$  witnessing the fact that  $u$  and  $v$  are not twins. Since  $u$  and  $v$  have the same color,  $a$  cannot be individualized, implying that  $a \in T_0$ . Let  $C_j$  be the color class containing  $a$ . Since  $C_1$  and  $C_j$  are stable color classes, there must exist a vertex  $b \in C_j$  such that  $\{u, a\}$  and  $\{v, b\}$  are edges and  $\{u, b\}$  and  $\{v, a\}$  are non-edges. Clearly, individualizing  $a$  refines the color class  $C_1$ . Therefore, the set  $T' = T_0 - \{a\}$  has the desired property  $|\mathcal{C}[T']| > |\mathcal{C}[T_0]|$  but is of size  $k - 1$ .

Since  $|V| > 2k$  and  $m < k$ , it follows that  $|B_1| \geq 2$ . Let  $x$  and  $y$  be two vertices inside  $B_1$ . Since  $X$  is twin-free, there must be a vertex  $z$  witnessing the fact that  $x$  and  $y$  are not twins. Since all vertices in  $T_0$  either have both vertices  $x$  and  $y$  as neighbors or none of them (otherwise,  $x$  and  $y$  would have different neighborhoods inside  $T_0$ , contradicting the fact that  $x, y \in B_1$ ), it follows that  $z \notin T_0$ . We claim that the set  $T_1 = T' \cup \{z\}$  yields the same stable partition as  $T'$ , i.e.,  $\mathcal{C}[T_1] = \mathcal{C}[T']$ . In fact, color refinement anyway assigns a unique color to  $z$ , since it is the only non-individualized vertex that is adjacent to exactly one of the two individualized vertices  $x$  and  $y$ . This completes the proof of the lemma. ◀

**Proof of Theorem 5.1.** We outline a simple kernelization algorithm for  $(n - k)$ -DISCRETE. Let  $X$  be the given graph and let  $k$  be the given parameter. The algorithm first makes the graph  $X$  twin-free by removing all but one vertex from each twin-class.

If the resulting graph  $X'$  has at most  $2k$  vertices, it outputs the instance  $(X', k)$  as the kernel. Since in each twin class of  $X$ , all but one vertices have to be individualized to make the graph discrete, the two instances  $(X, k)$  and  $(X', k)$  are indeed equivalent with respect to the  $(n - k)$ -DISCRETE problem.

If  $X'$  has more than  $2k$  vertices, the algorithm computes in polynomial time a set  $S$  of size  $k$  such that individualizing every vertex outside of  $S$  makes the graph  $X'$  discrete (see Lemma 5.2). Clearly this set  $S$  is also a solution for  $X$ , so the kernelization algorithm can output a trivial YES instance. ◀

**Acknowledgements.** We thank the anonymous referees for their helpful comments.

---

## References

---

- 1 Karl A. Abrahamson, Rodney G. Downey, and Michael R. Fellows. Fixed-parameter tractability and completeness IV: On completeness for  $W[P]$  and PSPACE analogues. *Annals of Pure and Applied Logic*, 73(3):235–276, 1993. doi:10.1016/0168-0072(94)00034-Z.
- 2 V. Arvind. The parameterized complexity of fixpoint free elements and bases in permutation groups. In *Proceedings of 8th International Symposium on Parameterized and Exact Computation (IPEC)*, pages 4–15. Springer, 2013. doi:10.1007/978-3-319-03898-8\_2.
- 3 V. Arvind, Frank Fuhlbrück, Johannes Köbler, Sebastian Kuhnert, and Gaurav Rattan. The parameterized complexity of fixing number and vertex individualization in graphs. arXiv:1606.04383, 2016.
- 4 V. Arvind, Johannes Köbler, Gaurav Rattan, and Oleg Verbitsky. On the power of color refinement. In *Proceedings of 20th International Symposium Fundamentals of Computation Theory (FCT)*, pages 339–350. Springer, 2015. doi:10.1007/978-3-319-22177-9\_26.
- 5 V. Arvind, Johannes Köbler, Gaurav Rattan, and Oleg Verbitsky. On Tinhofer’s linear programming approach to isomorphism testing. In *Proceedings of 40th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 26–37. Springer, 2015. doi:10.1007/978-3-662-48054-0\_3.
- 6 László Babai. Graph Isomorphism in quasipolynomial time. arXiv:1512-03547, 2015.
- 7 László Babai and Eugene M. Luks. Canonical labeling of graphs. In *Proceedings of 15th Annual ACM Symposium on Theory of Computing (STOC)*, pages 171–183, 1983. doi:10.1145/800061.808746.
- 8 Robert F. Bailey and Peter J. Cameron. Base size, metric dimension and other invariants of groups and graphs. *Bulletin of the London Mathematical Society*, 43(2):209–242, 2011. doi:10.1112/blms/bdq096.
- 9 Kenneth D. Blaha. Minimum bases for permutation groups: The greedy approximation. *Journal of Algorithms*, 13(2):297–306, 1992. doi:10.1016/0196-6774(92)90020-D.
- 10 Debra L. Boutin. Identifying graph automorphisms using determining sets. *Electronic Journal of Combinatorics*, 13:R78, 2006. URL: <http://www.combinatorics.org/ojs/index.php/eljc/article/view/v13i1r78>.
- 11 Jin-Yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12(4):389–410, 1992. doi:10.1007/BF01305232.
- 12 Liming Cai and David Juedes. On the existence of subexponential parameterized algorithms. *Journal of Computer and System Sciences*, 67(4):789–807, 2003. doi:10.1016/S0022-0000(03)00074-6.
- 13 John D. Dixon and Brian Mortimer. *Permutation groups*. Springer, 1996. doi:10.1007/978-1-4612-0731-3.
- 14 Rodney G. Downey, Vladimir Estivill-Castro, Michael R. Fellows, Elena Prieto, and Frances A. Rosamund. Cutting up is hard to do: The parameterised complexity of  $k$ -cut and related problems. *Electronic Notes in Theoretical Computer Science*, 78:209–222, 2003. doi:10.1016/S1571-0661(04)81014-4.
- 15 David Erwin and Frank Harary. Destroying automorphisms by fixing nodes. *Discrete Mathematics*, 306(24):3244–3252, 2006. doi:10.1016/j.disc.2006.06.004.
- 16 Gašper Fijavž and Bojan Mohar. Rigidity and separation indices of paley graphs. *Discrete Mathematics*, 289(1-3):157–161, 2004. doi:10.1016/j.disc.2004.09.004.
- 17 Martin Grohe. Equivalence in finite-variable logics is complete for polynomial time. *Combinatorica*, 19(4):507–532, 1999. doi:10.1007/s004939970004.

- 18 Birgit Jenner, Johannes Köbler, Pierre McKenzie, and Jacobo Torán. Completeness results for graph isomorphism. *Journal of Computer and System Sciences*, 66(3):549–566, 2003. doi:10.1016/S0022-0000(03)00042-4.
- 19 Eugene M. Luks. Permutation groups and polynomial-time computation. In *Groups and Computation*, pages 139–175. American Mathematical Society, 1993. URL: <http://www.cs.uoregon.edu/~luks/dimacs.pdf>.
- 20 Rudolf Mathon. A note on the graph isomorphism counting problem. *Information Processing Letters*, 8(3):131–136, 1979. doi:10.1016/0020-0190(79)90004-8.
- 21 Brendan D. McKay and Adolfo Piperno. Practical graph isomorphism, II. *Journal of Symbolic Computation*, 60:94–112, 2014. doi:10.1016/j.jsc.2013.09.003.
- 22 Joanna Raczek. Distance paired domination numbers of graphs. *Discrete Mathematics*, 308(12):2473–2483, 2008. doi:10.1016/j.disc.2007.05.018.
- 23 Omer Reingold. Undirected st-connectivity in log-space. In *Proceedings of 37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 376–385, 2005. doi:10.1145/1060590.1060647.
- 24 Pascal Schweitzer. Isomorphism of (mis)labeled graphs. In *Proceedings of 19th European Symposium on Algorithms (ESA)*, pages 370–381, Berlin, 2011. Springer. doi:10.1007/978-3-642-23719-5\_32.
- 25 Gottfried Tinhofer. A note on compact graphs. *Discrete Applied Mathematics*, 30(2–3):253–264, 1991. doi:10.1016/0166-218X(91)90049-3.
- 26 Luca Trevisan. Non-approximability results for optimization problems on bounded degree instances. In *Proceedings of 33rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 453–461. ACM, 2001. doi:10.1145/380752.380839.
- 27 Viktor N. Zemlyachenko, Nikolay M. Kornienko, and Regina I. Tyshkevich. Graph isomorphism problem. *Zapiski Nauchnykh Seminarov Leningradskogo Otdeleniya Matematicheskogo Instituta*, 118:83–158, 1982. Russian. Translation to English: [28].
- 28 Viktor N. Zemlyachenko, Nikolay M. Kornienko, and Regina I. Tyshkevich. Graph isomorphism problem. *Journal of Mathematical Sciences*, 29(4):1426–1481, 1985. English translation of [27]. doi:10.1007/BF02104746.