

Towards an Isomorphism Dichotomy for Hereditary Graph Classes

Pascal Schweitzer

RWTH Aachen University
Ahornstraße 55, 52074 Aachen, Germany
schweitzer@informatik.rwth-aachen.de

Abstract

In this paper we resolve the complexity of the isomorphism problem on all but finitely many of the graph classes characterized by two forbidden induced subgraphs. To this end we develop new techniques applicable for the structural and algorithmic analysis of graphs. First, we develop a methodology to show isomorphism completeness of the isomorphism problem on graph classes by providing a general framework unifying various reduction techniques. Second, we generalize the concept of the modular decomposition to colored graphs, allowing for non-standard decompositions. We show that, given a suitable decomposition functor, the graph isomorphism problem reduces to checking isomorphism of colored prime graphs. Third, we extend the techniques of bounded color valence and hypergraph isomorphism on hypergraphs of bounded color class size as follows. We say a colored graph has generalized color valence at most k if, after removing all vertices in color classes of size at most k , for each color class C every vertex has at most k neighbors in C or at most k non-neighbors in C . We show that isomorphism of graphs of bounded generalized color valence can be solved in polynomial time.

1998 ACM Subject Classification G.2.2 Graph Theory, F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases graph isomorphism, modular decomposition, bounded color valence, reductions, forbidden induced subgraphs

Digital Object Identifier 10.4230/LIPIcs.STACS.2015.689

1 Introduction

Given two graphs G_1 and G_2 , the graph isomorphism problem asks whether there exists a bijection from the vertices of G_1 to the vertices of G_2 that preserves adjacency and non-adjacency. In this paper we continue the systematic investigation of the complexity of graph isomorphism on hereditary graph classes with a focus on classes characterized by finitely many forbidden induced subgraphs as initiated in [21] (see also [22]).

Given a set of finite graphs H_1, \dots, H_t we define (H_1, \dots, H_t) -free to be the class of all graphs that do not contain any H_i as an induced subgraph. In the light of the unknown complexity status of the graph isomorphism problem, the goal in this context is typically to classify the complexity for the various graph classes into being polynomial time solvable or isomorphism complete (i.e., polynomially equivalent to graph isomorphism). Recently, in [26] it is shown that, assuming that graph isomorphism is not polynomial time solvable in general, there exist graph classes closed under taking (not necessarily induced) subgraphs which are of intermediate complexity. Trivially, this implies the same conditional existence of hereditary graph classes (i.e., graph classes characterized by forbidden induced subgraphs) of intermediate complexity. However, the construction in [26] intrinsically requires the use of infinitely many forbidden subgraphs. In contrast to this, in [26] it is also shown that there are



© Pascal Schweitzer;

licensed under Creative Commons License CC-BY

32nd Symposium on Theoretical Aspects of Computer Science (STACS 2015).

Editors: Ernst W. Mayr and Nicolas Ollinger; pp. 689–702

Leibniz International Proceedings in Informatics



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



SYMPOSIUM
ON THEORETICAL
ASPECTS
OF COMPUTER
SCIENCE

no intermediate graph classes characterized by finitely many forbidden subgraphs. However, this statement does not carry over to forbidden induced subgraphs and the question of the existence of intermediate graph classes characterized by finitely many forbidden induced subgraphs remains open. A more precise statement of the dichotomy result in [26] is that a graph class characterized by finitely many forbidden subgraphs has a polynomial time solvable graph isomorphism problem if one of the forbidden graphs is a union of subdivided stars, and is graph isomorphism complete otherwise. These graphs, the forests of subdivided stars, also play a central role in the complexity of hereditary graph classes.

With respect to classes defined by forbidden induced subgraphs, there is a dichotomy for the isomorphism problem on H_1 -free graphs into polynomially solvable and isomorphism complete cases. In [21] the complexity of the isomorphism problem on (H_1, H_2) -free graphs is determined for various pairs (H_1, H_2) and the results also follow the “polynomially solvable versus isomorphism complete” dichotomy. The crucial cases that were not resolved are those where either H_1 or H_2 is a complete graph. More specifically, except for finitely many cases, the unresolved cases were shown to be polynomially equivalent to a class where one of the graphs is a complete graph. In the light of all of these results we conjecture the following:

► **Conjecture 1.** *If $\mathcal{C} = (H_1, \dots, H_t)$ -free is a graph class defined by the finite set of forbidden induced subgraphs H_1, \dots, H_t then graph isomorphism of graphs in \mathcal{C} is polynomial time solvable or isomorphism complete.*

In this paper we continue the investigation of the complexity of the isomorphism problem on the graph classes focusing on the case of two forbidden subgraphs, where one of the graphs is complete. As mentioned above these are the crucial cases that were not resolved. For the resolution of those classes, techniques beyond those that were developed in [21] are required. These techniques are presented in this paper leading to the following theorem:

► **Theorem 2.** *On all but finitely many classes of the form (H_1, H_2) -free the graph isomorphism problem is polynomial time solvable or isomorphism complete.*

Contribution. In order to prove the theorem, three new techniques for the structural and algorithmic analysis of graphs are developed in this paper.

Firstly, we develop a methodology to show isomorphism completeness of the isomorphism problem on graph classes by providing a unifying framework for various reductions typically used for that purpose. The advantage of this framework is that it allows a streamlined abstract way to argue why some class is isomorphism complete, which boils down to an algorithmically checkable argument.

Secondly, we generalize the modular decomposition to colored graphs and define the concept of a colored modular decomposition with respect to a decomposition functor. This not only allows us to show that the graph isomorphism problem reduces to colored isomorphism of prime graphs but it also allows us to decompose graphs that are prime with respect to the classic modular decomposition. To show this reduction we also describe how to remodel an algorithm that has access to an oracle producing a complete invariant for a graph class into an algorithm that only has access to an isomorphism test of the graph class.

Thirdly, we extend the techniques of bounded color valence and hypergraph isomorphism on hypergraphs of bounded color class size as follows. We say a colored graph has generalized color valence at most k if, after removing all vertices in color classes of size at most k , for each color class C every vertex has at most k neighbors in C or at most k non-neighbors in C . We show that isomorphism of graphs of bounded generalized color valence can be solved in polynomial time. This generalization allows us to perform isomorphism tests for graphs

whose automorphism group cannot be forced into having bounded size composition factors even when using finitely many individualization steps. Since for such graphs alternating groups of unbounded size can appear among the composition factors, it seems that the standard group theoretic techniques cannot be directly applied. This shows that indeed new techniques were required to solve the particular cases.

We apply the three mentioned techniques to resolve the complexity of isomorphism on all but finitely many of the graph classes characterized by two forbidden induced subgraphs. For the resolved classes we either provide a reduction from the general problem or a polynomial time algorithm. The applications of the techniques include for example showing that bipartite graphs that are free from a fixed forbidden double star refine into graphs of bounded generalized color valence. This can be used to show that isomorphism of graphs of bounded clique number with a fixed forbidden double star can be solved in polynomial time. We in turn apply this in conjunction with the colored modular decomposition technique to solve isomorphism for graphs of bounded clique number which do not contain P_5 (a path of length 4) as an induced subgraph. We apply the general reductions to show that classes of graphs without cliques of size 4 and certain unions of paths are extensive enough to have an isomorphism problem that is isomorphism complete. Furthermore, we apply the modular decomposition techniques in conjunction with the bounded generalized color valence to analyze the structure of various graph classes of bounded clique number with certain forbidden forests of subdivided stars.

Related work. We refer the reader to [3, 18, 28] for an introduction to the diverse complexity-theoretic results related to the isomorphism problem. There are numerous results known on the complexity of graph isomorphism of hereditary graph classes. A collection showing the problem for many classes to be equivalent to the general problem is given by Booth and Colbourn [5]. In that paper, the complexity of classes characterized by one forbidden subgraph H is shown to depend on whether the graph H is an induced subgraph of P_4 (the path on 4 vertices). The systematic study of classes characterized by two forbidden subgraphs was initiated in [21].

With regard to algorithms, a recent very general result, implying many results devised earlier on more special graph classes, is a theorem of Grohe and Marx [13] that shows that isomorphism of graph classes defined by a forbidden topological minor can be solved in polynomial time.

There are several applications of modules or some form of modular decomposition in the context of graph isomorphism. For example, Goldberg's plain exponential algorithm [11] uses the concept of sections, which can be seen as colored modules. Furthermore, Junttila and Kaski [17] define non-uniform components within the individualization-refinement approach that also constitute colored modules. As described below, Rao [27] also exploits the classic modular decomposition to devise an isomorphism algorithm for gem and co-gem free graphs (i.e., $(P_4 \cup K_1, \overline{P_4 \cup K_1})$ -free graphs). His technique can be seen as a special case of the techniques using colored modular decompositions described in this paper. Other hereditary graph classes, for which isomorphism algorithms make use of modular decomposition, are for example subclasses of circular-arc graphs (see [7]).

In the early stages of the discovery of the group theoretic technique for the isomorphism problem, Luks [24] applied the technique to show that isomorphism of graphs of bounded maximum degree can be solved in polynomial time. Babai [2] applied the notion of bounded color valence in his algorithm for the general isomorphism problem. Miller (see [25]) applied this technique in a series of papers to perform isomorphism tests of k -separable and k -contactable graphs as well as isomorphism of hypergraphs of bounded color class size (see

also [1], [4] and [30]). In our generalization in this paper, only the subgraph induced by the color classes that are not of bounded size is required to exhibit bounded color valence.

There have been several studies aiming at dichotomy results for computational problems on graph classes characterized by forbidden induced subgraphs. For example, this has been done for the computation of the chromatic number [20], dominating sets [23], coloring [9], and list coloring [12] for two forbidden subgraphs.

Moreover there are numerous results analyzing whether the clique width of a graph class characterized by forbidden subgraphs is bounded (see [8] for an extensive list of references).

Structure of the paper. We mainly apply the three techniques developed in this paper to classes characterized by two forbidden induced subgraphs. However, the intention behind their presentation is to allow them to be applicable in a broader sense to practical and theoretical algorithms for isomorphism of general graphs or maybe for the classes of bounded clique width. The paper first presents the techniques and then hints at how to apply them to various classes of two forbidden subgraphs. The full version [29] provides details and proofs that have mostly been omitted.

In the first part of the paper we provide preliminaries such as introducing notation and recalling basic tools (Section 2). We then devise a methodology to prove isomorphism completeness results (Section 3) and briefly describe how to simulate a complete invariant given only an isomorphism algorithm (Section 4). After this we turn to techniques for isomorphism testing using modular decompositions (Section 5) and devise a polynomial time algorithm for graphs of bounded generalized color valence (Section 6).

The reduction techniques can be applied to show the isomorphism completeness of various graph classes characterized by forbidden induced subgraphs. The algorithmic techniques can be applied to graph classes with forbidden double stars and graph classes of graphs without induced paths of length 4. The techniques can also be applied to analyze specific triangle-free graphs and specific graphs of bounded clique number. (For details see [29].) We conclude by showing that together with the theorems in [21] this resolves the complexity of all but finitely many graph classes defined by two forbidden induced subgraphs (Section 7).

2 Preliminaries

In this paper all graphs are finite, simple, undirected graphs. For a graph G , by $V(G)$ and $E(G)$ we denote the vertex set and the edge set, respectively. By $N_G(S) = N(S)$ we denote the neighborhood of a set S , i.e., the vertices in $V(G) \setminus S$ adjacent to some vertex in S . By id we always denote the identity map. The bipartite complement of a bipartite graph G with bipartition classes A and B is obtained by replacing $E(G)$ with $\{\{a, b\} \mid a \in A, b \in B\} \setminus E(G)$.

We write $H \leq G$ if the graph G contains a graph H as an induced subgraph. A graph G is H -free if $H \not\leq G$. It is (H_1, \dots, H_k) -free, if it is H_i -free for all i . A graph class \mathcal{C} is H -free (respectively (H_1, \dots, H_k) -free) if this is true for all $G \in \mathcal{C}$. A graph class \mathcal{C} is *hereditary* if it is closed under taking induced subgraphs. The class (H_1, \dots, H_k) -free is the class of all (H_1, \dots, H_k) -free graphs. Note that each class (H_1, \dots, H_k) -free is hereditary. We say a graph G *contains* a graph H (as an induced subgraph) if an induced subgraph of G is isomorphic to H .

By I_t , K_t , P_t , and C_t we denote the independent set, the clique, the path, and the cycle on t vertices, respectively. The *clique number* of a graph G is the largest integer t such that G contains K_t . By $H \dot{\cup} H'$ we denote the disjoint union of H and H' ; we use tH for the disjoint union of t copies of the graph H . By \overline{G} we denote the (edge) complement of G . The graph $\overline{K_2 \dot{\cup} I_2}$, i.e., the graph obtained from K_4 by deleting an edge, is called the *diamond*.

A *star* is a graph isomorphic to the complete bipartite graph $K_{1,t}$ for some positive integer t . The *subdivision* of an edge is the replacement of the edge with a path of length two. A *subdivided star* is a possibly repeated subdivision of a star. If a subdivided star has a vertex of degree at least 3 then this vertex is unique and called the *center*. For non-negative integers a_0, \dots, a_t with $a_t > 0$, we define the graph $H(a_t, \dots, a_1, a_0)$ to be the disjoint union of an independent set of size a_0 with the following subdivided star H . The star H is the subdivided star that for $i \in \{1, \dots, t\}$ has exactly a_i leaves at distance i from the center and no other leaves. (If $\sum_{i=1}^t (a_i) < 3$ the center of H is defined so that the graph H is a path whose two leaves have suitable distances from that center.) In [21] it is shown that isomorphism of (H_1, \dots, H_t) -free graphs is isomorphism complete unless one of the forbidden graphs is a forest of subdivided stars.

In this paper a *colored graph* is a vertex colored graph whose coloring does not need to be proper (i.e., adjacent vertices can have the same color). Isomorphisms between colored graphs are required to respect the colors. The *naive vertex refinement* algorithm, or 1-dimensional Weisfeiler-Lehman algorithm, is a standard technique of repeatedly recoloring the vertices, refining the partition induced by the colors, by using the multiplicity of colors appearing among the neighbors of a vertex (see for example [28]). It has the property that after the refinement, the number of neighbors a vertex v has in a color class C only depends on the colors of v and C . Such a coloring is called *stable*. A graph has *color valence at most k* if for every vertex v and every color class C there are at most k neighbors of v in C or there are at most k non-neighbors of v in C .

3 Reductions

In this section we develop a systematic approach to proving isomorphism invariant reductions. This provides general means to construct isomorphism complete graph classes. Standard reductions like subdividing, taking complements, and adding isolated vertices fall into this framework. Likewise, most reductions performed in [21] and also various reductions in [5] fall into this framework.

► **Definition 3.** Let J be a finite set and $L: J \times J \rightarrow \{A, N\}$ be a labeling assigning every ordered pair of vertices the label A for adjacent or N for non-adjacent. Moreover let $L_N: J \times J \rightarrow \mathbb{N} \cup \{\infty\}$ be a labeling assigning every ordered pair of vertices a non-negative integer or infinity.

A graph G belongs to *the class encoded by the labeled graph (J, L, L_N)* if there exists a map $\phi: V(G) \rightarrow J$ such that the following hold:

1. If $v \in V(G)$ and $j \in J$ such that $L(\phi(v), j) = A$ and $L_N(\phi(v), j) \neq \infty$ then there are at most $L_N(\phi(v), j)$ vertices $v' \in V(G) \setminus \{v\}$ that are non-adjacent to v such that $\phi(v') = j$.
2. If $v \in V(G)$ and $j \in J$ such that $L(\phi(v), j) = N$ and $L_N(\phi(v), j) \neq \infty$ then there are at most $L_N(\phi(v), j)$ vertices $v' \in V(G) \setminus \{v\}$ that are adjacent to v such that $\phi(v') = j$.

In this definition, the triple (J, L, L_N) should be thought of as a generalized graph. The graph class encoded by (J, L, L_N) then contains graphs that can be obtained by replacing the elements of J with sets of vertices. In some contexts, this is referred to as blowing-up the elements of J . Adjacency of the new vertices is essentially governed by the adjacency of the original graph. However, the values of L_N control the number of exceptions to this rule that are allowed per vertex.

The definition captures various constructions used in graph theory. A first class of examples is formed by the complete multipartite graphs, which turn out to be graphs modeled

with the function L satisfying $L(x, y) = N$ if and only if $x = y$ and the function L_N being the constant function evaluating to 0. A second, more general class of examples modeled by the definition is the graphs of bounded color valence, which are graphs that frequently appear in the context of graph isomorphism. Such graphs are obtained whenever L_N is a bounded function. The coloring corresponds to the map ϕ . A third example is the class of graphs that map homomorphically onto a finite graph H . This class is obtained by letting (J, L) model the graph H (i.e., $V(H) = J$ and $L(x, y) = A$ if and only if $\{x, y\}$ is an edge of H) and setting $L_N(x, y) = 0$ if $L(x, y) = N$ and $L_N(x, y) = \infty$ otherwise.

Of interest to us is the complexity of the isomorphism problem of graph classes encoded by a triple (J, L, L_N) . It turns out that when L_N is a bounded function then isomorphism can be solved in polynomial time.

► **Theorem 4.** *Let (J, L, L_N) be a triple that encodes a graph class. If all values of L_N are finite then isomorphism of graphs encoded by (J, L, L_N) can be solved in polynomial time.*

The theorem implicitly shows that isomorphism of graphs that have bounded color valence for some coloring that uses a bounded number of color classes can be solved in polynomial time, even if the color classes are not given.

► **Corollary 5.** *For every positive integer c , graph isomorphism of graphs whose vertices can be partitioned into c color classes such that the graph has color valence at most c can be solved in polynomial time.*

While encodings can be used to show polynomial time solvability of certain graph classes, they can also be used to show hardness results as follows.

► **Definition 6.** An encoding (J, L, L_N) is a *simple path encoding* in case L is symmetric (i.e., if $L(j, j') = L(j', j)$ holds) and there is a sequence of vertices (p_1, \dots, p_t) of length at least 2 in J such that $L_N(p_1, p_2) = \infty$, $L_N(p_t, p_{t-1}) \geq 2$ and for all $k \in \{1, \dots, t-1\}$ we have $L_N(p_k, p_{k+1}) \geq 1$ and $L_N(p_{k+1}, p_k) \geq 1$.

Intuitively, a simple path encoding allows enough freedom to encode bipartite graphs with one bipartition class having vertices of degree two. We can formally prove this statement in the form of a reduction.

► **Theorem 7.** *A class of graphs encoded by a simple path encoding is isomorphism complete.*

The theorem can be applied to show that various classes are isomorphism complete.

► **Theorem 8.** *The classes $(2K_2 \dot{\cup} K_1, K_4)$ -free, $(P_6, P_4 \dot{\cup} P_2, K_4)$ -free, $(H(1, 0, 3, 0), K_4)$ -free, bipartite $(2P_3 \dot{\cup} K_1)$ -free, $(H(1, 0, 2, 0), K_5)$ -free are isomorphism complete.*

The theorem in turn implies that various classes (H_1, H_2) -free that are superclasses of one of these classes are isomorphism complete (such as $(K_3, 2P_3 \dot{\cup} K_1)$ -free).

4 Isomorphism, Invariants and Canonical Labeling

A *complete graph invariant* for a graph class \mathcal{C} is a map $\text{Inv}: \mathcal{C} \rightarrow \mathcal{D}$ into some class \mathcal{D} such that for graphs G_1 and G_2 in \mathcal{C} we have $\text{Inv}(G_1) = \text{Inv}(G_2)$ if and only if G_1 and G_2 are isomorphic. A *canonical labeling* is a map that assigns to every graph G a graph $C(G)$ with vertex set $V(C(G)) = \{1, \dots, |G|\}$ and an isomorphism $\phi: G \rightarrow C(G)$ such that the map assigning $C(G)$ to G is a complete invariant. There are relatively general techniques with which one can turn a complete invariant into a canonical labeling algorithm [14, 15, 19].

In this paper, we are mainly interested in isomorphism algorithms, as opposed to canonical labeling algorithms or complete invariants. We will therefore require a tool to simulate an invariant within one execution of our algorithm, given only an algorithm that performs isomorphism checks. Our simulated invariant will not be consistent across different calls of the same algorithm.

► **Theorem 9.** *Let A be a polynomial time algorithm with access to a complete invariant \mathcal{O} for a graph class \mathcal{C} given as oracle. Suppose the outputs of A are independent of the choice of the invariant \mathcal{O} . If isomorphism of graphs in \mathcal{C} can be solved in polynomial time then there is a polynomial-time algorithm B whose outputs coincide with those of A , which does not require access to an oracle.*

We use the theorem to replace an invariant with a isomorphism algorithm in the next section when dealing with modular decompositions (more precisely to prove Theorem 15).

5 Colored modular decomposition

In this section we are concerned with modular decompositions and their application to the isomorphism problem. We will work with colored graphs since this is convenient in the graph isomorphism context. However, we will also generalize the concept of a module to that of a colored module, since this is required by our later applications. Since we do not require previous knowledge about the uncolored modular decomposition, we will not define it. We refer the reader to the survey by Habib and Paul [16] for more information on the uncolored decomposition and its algorithmic applications. For the colors, we will assume that there is a linear order on the colors. In algorithmic applications such a linear order can always be obtained by comparing the bit-strings corresponding to the colors lexicographically.

► **Definition 10.** Let G be a colored graph. A *colored module* is a subset M of $V(G)$ such that for all $v \in V(G) \setminus M$, if $x, x' \in M$ are of the same color then either v is adjacent to both vertices x and x' or to neither x nor x' .

A module M is *non-trivial* if it contains at least two vertices that cannot be distinguished by vertices outside of the module. That is, M is non-trivial if there are $x, y \in M$ such that for all $v \in V(G) \setminus M$ the vertex v is adjacent to x if and only if it is adjacent to y . Note that every module that contains two vertices of the same color is non-trivial.

► **Definition 11.** A map assigning to every graph G a subset of the vertices $M(G)$ is said to be *isomorphism invariant* if for every isomorphism $\phi: G \rightarrow G'$ we have $\phi(M(G)) = M(G')$. A map that assigns to every graph G a partition of a subset of the vertices $M(G) = \{M_1, \dots, M_k\}$ is said to be *isomorphism invariant* if for every isomorphism $\phi: G \rightarrow G'$ we have $M(G') = \{\phi(M_1), \dots, \phi(M_k)\}$.

► **Definition 12.** A *decomposition functor* is a map assigning to every graph G a partition of a subset of the vertices into modules that is isomorphism invariant.

A graph G is *prime* with respect to a decomposition functor Mod if $\text{Mod}(G)$ does not contain non-trivial modules. While there is a standard decomposition functor for the uncolored case, for the colored case it is in general not clear whether we can find a useful functor to decompose the graphs, and we have to find such a functor for a given graph class first, in order to decompose the graphs. See [29] for an example illustrating that this allows us to decompose graphs that are prime with respect to classic modular decomposition.

In the remainder of this section, we argue for certain decomposition functors that graph isomorphism for a hereditary graph class can be solved in polynomial time if the isomorphism problem for colored prime graphs in the class can be solved in polynomial time. To facilitate the proof we can assume that we are given a complete invariant for the prime graphs in the hereditary graph class and then apply Theorem 9.

Our next goal is to define the concept of the quotient graph. In the uncolored case, the quotient graph is obtained by replacing each module with a single vertex whose adjacency to the rest of the graph is the same as that of every vertex of the module. However, for the colored case, the adjacency to the rest of the graph depends on the color of the vertex in the module. This means that for every adjacency type we need to retain a vertex that has the same adjacency type with respect to vertices outside the module. (For a vertex v , the adjacency type with respect to vertices outside the module is the set of vertices outside the module adjacent to v .)

A *replacement operator* is an isomorphism invariant map that assigns every non-trivial module M in a decomposition of a graph G an induced subgraph of M in which the vertices are possibly recolored. We require that for every adjacency type of vertices in M at least one vertex of M is maintained. Let $\text{Inv}(M)$ be a complete graph invariant. Given a family of modules $\{M_1, \dots, M_k\}$ that partitions the graph, the *quotient graph* is obtained by simultaneously replacing all modules using the replacement operator (i.e., removing from the module M_i all vertices not in the induced subgraph assigned to M_i by the replacement operator) and then recoloring every vertex v as the triple $(\chi(v), L, \text{Inv}(M_v))$, where $\chi(v)$ is the color of v after the replacement, L is a list of the colors of vertices with the same adjacency type as v , and $\text{Inv}(M_v)$ is the invariant of the module containing v . We say that the decomposition functor is *simple* with respect to a replacement operator if for every complete invariant every quotient graph is prime. Intuitively, this means that the decomposition functor provides us with maximal modules.

► **Definition 13.** Given a decomposition functor, we say a replacement operator is *reversible* if the following holds: two graphs G_1 and G_2 are isomorphic if and only if their colored quotient graphs with respect to the decomposition functor and the replacement operator are isomorphic.

Note that reversibility does not depend on the complete invariant that is used for the recoloring of the quotient graph.

We remark that for uncolored graphs the definitions of module, primality and the quotient graph coincide with the usual definition from the literature (see [16]). In that context, the decomposition functor is typically chosen to partition the graph into components, components of the complement graph, or maximal modules. The replacement operator simply replaces the entire module by one vertex. However, for the applications we have in mind, we require the more general concept of colored modules. Certain conditions immediately imply that a replacement operator is reversible.

► **Lemma 14.** *A replacement operator is reversible if 1.) there is only one trivial module, 2.) all replacements contain only one vertex, or 3.) non-trivial modules induce connected graphs, but the non-trivial modules are pairwise non-adjacent.*

A reversible simple decomposition functor can be used to test isomorphism by considering only isomorphisms between quotient graphs and modules. Iterating this yields an isomorphism test for decomposable graphs if one has access to a complete invariant for prime graphs. Using Theorem 9 we can replace the requirement for a complete invariant by an isomorphism algorithm.

► **Theorem 15.** *Let \mathcal{C} be a hereditary graph class and Mod a simple polynomial-time computable decomposition functor with polynomial-time computable, reversible replacement operator R for colored graphs in \mathcal{C} . If the isomorphism problem for colored prime graphs in \mathcal{C} can be solved in polynomial time then the isomorphism problem of all graphs in \mathcal{C} can be solved in polynomial time.*

The theorem in particular applies to the standard uncolored modular decomposition. This decomposition is associated with a simple decomposition functor with polynomial-time computable, reversible replacement. In his diploma thesis, Fuhlbrück [10] also describes a reduction for the standard uncolored decomposition functor. Already when trying to solve the isomorphism problem for uncolored graphs in a graph class \mathcal{C} with the described method, the fact that the quotient graph needs to be colored implies that we require an isomorphism algorithm for colored prime graphs. In [27], Rao describes a special case of the theorem for the uncolored modular decomposition, essentially considering hereditary graph classes in which every prime graph is of bounded size. Moreover, in the isomorphism context, the bi-join decomposition, also described in [27], can also be treated by using colored modules.

In our applications of Theorem 15 we also use another unrelated technique of dealing with small color classes that we describe next.

6 Bounded generalized color valence

In this section we show that isomorphism of graphs of bounded generalized color valence can be solved in polynomial time. However, the proofs of the lemmas and theorems within the section require familiarity with the computational group theoretic methods that have been developed within the context of the isomorphism problem. For example, when computing automorphism groups, cosets, and sets of isomorphisms between two combinatorial objects, we use a succinct representation by generators and a representative. We also use the set stabilizer theorem [24] for groups with composition factors of bounded size (see also [1, 4, 25]), which implies that for a permutation group with composition factors of bounded size acting faithfully on a set, we can compute the stabilizer of any given subset in polynomial time. For a good overview over the various computational group theoretic techniques, we refer the reader to [30].

Before we solve the isomorphism problem of graphs of bounded generalized color valence, let us consider some examples. Hypergraphs on sets of bounded color class size can directly be encoded as graphs by adding a vertex for every hyperedge that is adjacent to the elements of the hyperedge. Thus, a polynomial time algorithm for bounded generalized color valence also gives rise to a polynomial time algorithm for hypergraphs of bounded color class size. However, there are examples that are not captured by this. Consider a graph of maximum degree at most c consisting of a large number of components such that there is only a small number of isomorphism types among these components. Now add an arbitrary finite number of new vertices colored with new colors such that the added vertices form color classes of size at most c . The new vertices are connected via edges in an arbitrary way to the original vertices. The resulting graph has bounded generalized color valence. However, its automorphism group may contain composition factors of arbitrarily large size. The isomorphism algorithm for graphs of bounded degree exploits the fact that at least for components it is possible, by individualizing one vertex, to obtain a group with bounded composition factor. In the example graphs just described, it is not clear how the standard group theoretic arguments can be applied.

Nevertheless, our goal is to prove that graph isomorphism of graphs with bounded generalized color valence can be solved in polynomial time. Note that the classes of bounded generalized color valence are closed under refinement operations such as individualization and naive vertex refinement.

To solve the isomorphism problem, we need to deal with the small color classes. If there are only a bounded number of them, we can apply individualization to all vertices in small color classes. However, the number of small color classes can grow linearly in the number of vertices. To remedy this problem, we exploit the existence of certain colored modules and use group theoretic techniques.

We first define a decomposition functor that works well with graphs of bounded generalized color valence that have been refined with naive vertex refinement. Given a subset S of vertices, we say a vertex $v' \notin S$ contained in the color class C' is of degree dependence at most d with respect to S if there is a vertex v in S such that $v' \in N(v) \wedge |(N(v) \cap C')| \leq d$ or $v' \notin N(v) \wedge |C' \setminus (N(v))| \leq d$. Intuitively, this definition says that there is a vertex $v \in S$ such that individualization of this vertex followed by refinement with respect to adjacency towards v produces a set of size at most d within which v' can be found.

► **Definition 16.** A non-empty subset S is a d -degree dependence module, if no vertex outside S has degree dependence at most d with respect to S .

► **Lemma 17.** Let G be a stable graph of color valence at most c . If G does not contain color classes of size at most $2c$ then the minimal c -degree dependence modules partition G . Moreover, the map assigning every such graph the family of minimal c -degree dependence modules is a polynomial-time computable decomposition functor.

In Lemma 17, using naive vertex refinement is essential. Indeed, consider a wheel, i.e., a cycle with an added center adjacent to every other vertex. In this graph, there is only one non-trivial 3-degree dependence module, namely the set that contains only the center of the wheel. Thus the set of 3-degree dependence modules does not partition the graph.

By defining a faithful group action of the automorphism group of a graph on the c -degree dependence modules it is possible to gradually compute the automorphism group, yielding also an isomorphism test for such graphs.

► **Theorem 18.** Graph isomorphism for colored graphs of generalized color valence at most c can be solved in polynomial time.

We can combine Theorem 18 and Theorem 15 to show that several graph classes have a polynomial-time solvable isomorphism problem. These classes include various subcases for which the complexity of the isomorphism problem was also previously unresolved.

► **Theorem 19.** The isomorphism problem for the classes $(H(1, b, 0), K_s)$ -free, (P_5, K_t) -free, $(H(1, 0, b, 1), K_3)$ -free and $(K_{1,s} \cup K_{1,s}, K_t)$ -free can be solved in polynomial time.

7 Comprehensiveness of the case distinction

We now argue that the theorems developed throughout this paper together with the theorems from [21] resolve, except for finitely many cases, the complexity of graph isomorphism for (H_1, H_2) -free graphs. In fact, said theorems also either provide a polynomial time algorithm or a reduction from the general isomorphism problem to the respective classes.

Proof of Theorem 2. Let (H_1, H_2) -free be the graph class for which we want to determine the complexity. By the results in [21], we may assume that one of the graphs, H_2 say, is a

complete graph. Since graph isomorphism for K_2 -free graphs is polynomial-time solvable, we further assume that H_2 has at least 3 vertices. By [21, Lemma 2] we may consequently assume that H_1 is a forest of subdivided stars, since the problem is graph isomorphism complete otherwise. If H_1 contains 3 non-trivial components, then H_1 contains $3K_2$ and the problem is isomorphism complete [21, Lemma 5]. In the following, we can thus assume that H_1 has at most 2 non-trivial components. If H_1 has no vertex of degree 3 then if H_1 is sufficiently large, it is either of the form $P_i \dot{\cup} I_t$ with $i \leq 3$ or it contains the graph $2K_2 \dot{\cup} I_2$. In the former case, graph isomorphism is polynomial-time solvable by Theorem 19. In the latter case, the isomorphism problem is graph isomorphism complete [21, Lemma 5].

(*The case $H_2 = K_3$*). Suppose H_2 is the graph K_3 . Since H_2 is fixed, we can assume that H_1 is sufficiently large and we can thus assume by the observation above that H_1 contains a vertex of degree 3. If H_1 does not contain a P_4 then H_1 is a union of at most two stars plus isolated vertices. If there is at most one star, the problem is polynomial-time solvable by Theorem 19 (or by [21, Theorem 4]). Assuming there are two stars, if there is more than one isolated vertex then H_1 contains $2K_2 \dot{\cup} 2K_1$ and the problem is isomorphism complete by [21, Lemma 5]. If neither of the stars is only a single edge then if there exists an isolated vertex in H_1 the problem is isomorphism complete by Theorem 8 and if there is no isolated vertex in H_1 it is polynomial-time solvable by Theorem 19. Finally, if one of the stars is only an edge then by Theorem 19 the problem is polynomial-time solvable.

If H_1 contains a P_4 and there are two non-adjacent vertices not in the same connected component as the P_4 then H_1 contains $P_4 \dot{\cup} 2K_1$ and the problem is graph isomorphism complete [21, Lemma 5]. The assumption that this is not the case implies that the vertex of degree 3 is in the same component as the P_4 . Since isomorphism of $(2K_2 \dot{\cup} 2K_1)$ -free triangle-free graphs is isomorphism complete [21, Lemma 5], by assuming that H_1 is sufficiently large, we may further assume there is at most one additional vertex not in the connected component of the P_4 . If there is only one vertex of degree 1 non-adjacent to the vertex h of degree at least 3, and additionally this vertex has distance 2 from h , then H_1 is an induced subgraph of $H(1, 0, b, 1)$ for some positive integer b . This implies that the problem is polynomial-time solvable by Theorem 19. If H_1 contains two leaves of distance at least 3 from the center, then if H_1 is sufficiently large it also contains $2K_2 \dot{\cup} 2K_1$. If H_1 contains a leaf of distance at least 4 then H_1 contains $P_4 \dot{\cup} 2K_2$. In both cases the problem is graph isomorphism complete by [21, Lemma 5].

(*The case $H_2 = K_n$ with $n > 3$*). Suppose now H_2 is the graph K_n for some $n \geq 4$. Suppose first, H_1 contains two non-trivial components. If it contains an isolated vertex, then it contains $2K_2 \dot{\cup} K_1$ and the problem is isomorphism complete by Theorem 8. If one of the components contains P_4 and the graph is not connected, then the graph contains $P_4 \dot{\cup} K_1$ and the problem is isomorphism complete by [21, Theorem 3]. Otherwise, the two components form a double star and the problem is polynomial-time solvable by Theorem 19.

Thus we may assume now that there is only one non-trivial component. If there is no P_4 in H_1 , the problem is solvable by [21, Theorem 4]. Otherwise, by [21, Theorem 3], we can assume that H_1 is connected. If H_1 is isomorphic to P_5 , the problem is solvable by Theorem 19. If H_1 is isomorphic to P_6 the problem is isomorphism complete by Theorem 8. We can thus assume that H_1 contains a vertex of degree at least 3, which we call the center. If there is only one leaf not adjacent to the center and this leaf has distance 2 from the center, then the problem is polynomial-time solvable by Theorem 19. If there are two leaves not adjacent to the center then H_1 contains $P_4 \dot{\cup} K_1$ and the problem is isomorphism complete by [21, Theorem 3]. If there is a leaf of distance at least 3 from the center and the center has degree 4 then the problem is graph isomorphism complete by Theorem 8.

We can thus assume that the center has degree 3. Under these conditions, if $H_2 = K_4$ and H_1 is sufficiently large, then H_1 contains P_6 and the problem is isomorphism complete by Theorem 8. In the remaining cases $H_2 = K_n$ with $n \geq 5$ and H_1 has a leaf of distance at least 2 from the center. In this case the problem is isomorphism complete by Theorem 8. ◀

8 Conclusion

There is an intricate relationship between the boundedness of the clique width on a graph class and polynomial-time solvability of the isomorphism problem. While there are classes of unbounded clique width for which the isomorphism problem is solvable in polynomial time (for example the graphs of bounded degree or the graphs characterized by a forbidden star and a forbidden clique (see [21])), there are no classes known to be isomorphism complete and known to have bounded clique width. It is conceivable, but not known to be true, that isomorphism on graphs of bounded clique width is solvable in polynomial time. In fact, it seems that typically an isomorphism reduction to a graph class \mathcal{C} yields a proof for the unboundedness of the clique width. It is easy to see that every class of graphs encoded by a simple path encoding has unbounded clique width. The reason is that if G' is the graph produced from a graph G by the reduction detailed in the proof of Theorem 7 then G can be obtained from G' by a finite application of edge complementations between two sets, followed by an unbounded number of local complementations and vertex removals. This shows that the clique width of G is bounded by a function of the clique width of G' . Thus, if G has large clique width then G' has large clique width.

► **Corollary 20.** *The graphs encoded by a simple path encoding have unbounded clique width.*

This implies that many classes (H_1, H_2) -free have unbounded clique width. In particular the class $(P_4 \cup K_1, K_4)$ -free has unbounded clique width, which was previously an unknown case (see [8]). Conversely, when a graph class is shown to be polynomial-time solvable, often this is due to some structural insight that actually amounts to showing that the clique width of the class is bounded. On another note it appears that the solution of the generalized color valence problem presented in this paper, more precisely a generalization of the technique, is a first step towards showing polynomial-time solvability of graphs of bounded clique width. More concretely, the generalization applies to graphs with fixed rank decomposition.

As mentioned in the introduction, for various other computational problems, classification results have been considered for classes characterized by two forbidden graphs. For these problems, there had been extensive prior work and algorithmic techniques were already available. For the isomorphism problem, such techniques were lacking and the intention of this paper is to develop them. The fact that for new classes the complexity of isomorphism can be determined using these techniques, while other methods seem to fail, shows that these techniques provide something conceptually different.

Concerning a list of open cases that still remain for the classes characterized by two forbidden subgraphs, one has to analyze precisely to which classes the techniques can be applied. For example, Brandstädt and Kratsch [6] show that cycles of length 5 in $(P_5, \overline{P_4 \cup K_1})$ -free graphs are disjoint. By coloring vertices inside a 5-cycle depending on whether they have a neighbor outside the cycle, we obtain a decomposition functor. Furthermore, there is a description of the relevant prime graphs in [6] which allows us to apply the modular decomposition technique. This is just an example and a comprehensive description of the various cases that can actually be handled with the techniques remains as future work.

Acknowledgments I thank Matasha Mazis for inspiring comments and suggestions.

References

- 1 Vikraman Arvind, Bireswar Das, Johannes Köbler, and Seinosuke Toda. Colored hypergraph isomorphism is fixed parameter tractable. In *FSTTCS*, pages 327–337, 2010.
- 2 László Babai. Moderately exponential bound for graph isomorphism. In *FCT*, pages 34–50, 1981.
- 3 László Babai. *Handbook of Combinatorics (vol. 2)*, chapter Automorphism groups, isomorphism, reconstruction, pages 1447–1540. MIT Press, 1995.
- 4 László Babai and Eugene M. Luks. Canonical labeling of graphs. In *STOC*, pages 171–183, 1983.
- 5 Kellogg S. Booth and C. J. Colbourn. Problems polynomially equivalent to graph isomorphism. Technical Report CS-77-04, Comp. Sci. Dep., Univ. Waterloo, 1979.
- 6 Andreas Brandstädt and Dieter Kratsch. On the structure of (P_5, gem) -free graphs. *Discrete Applied Mathematics*, 145(2):155–166, 2005.
- 7 Andrew Curtis, Min Lin, Ross McConnell, Yahav Nussbaum, Francisco Soulignac, Jeremy Spinrad, and Jayme Szwarcfiter. Isomorphism of graph classes related to the circular-ones property. *Discrete Mathematics and Theoretical Computer Science*, 15(1):157–182, 2013.
- 8 Konrad Dabrowski and Daniël Paulusma. Clique-width of graph classes defined by two forbidden induced subgraphs. *CoRR*, abs/1405.7092, 2014.
- 9 Konrad K. Dabrowski, Petr A. Golovach, and Daniel Paulusma. Colouring of graphs with ramsey-type forbidden subgraphs. *Theoretical Computer Science*, 522(0):34–43, 2014.
- 10 Frank Fuhlbrück. Fixed-parameter tractability of the graph isomorphism and canonization problems. Diploma thesis, Humboldt-Universität zu Berlin, 2013.
- 11 Mark K. Goldberg. A nonfactorial algorithm for testing isomorphism of two graphs. *Discrete Applied Mathematics*, 6(3):229–236, 1983.
- 12 Petr A. Golovach and Daniël Paulusma. List coloring in the absence of two subgraphs. *Discrete Applied Mathematics*, 166:123–130, 2014.
- 13 Martin Grohe and Dániel Marx. Structure theorem and isomorphism test for graphs with excluded topological subgraphs. In *STOC*, pages 173–192, 2012.
- 14 Yuri Gurevich. From invariants to canonization. *Bulletin of the EATCS*, 63, 1997.
- 15 Yuri Gurevich. From invariants to canonization. In *Current Trends in Theoretical Computer Science*, pages 327–331. World Scientific, 2001.
- 16 Michel Habib and Christophe Paul. A survey of the algorithmic aspects of modular decomposition. *Computer Science Review*, 4(1):41–59, 2010.
- 17 Tommi A. Junttila and Petteri Kaski. Conflict propagation and component recursion for canonical labeling. In *TAPAS*, pages 151–162, 2011.
- 18 Johannes Köbler, Uwe Schöning, and Jacobo Torán. *The graph isomorphism problem: its structural complexity*. Birkhäuser Verlag, Basel, Switzerland, 1993.
- 19 Johannes Köbler and Oleg Verbitsky. From invariants to canonization in parallel. In *CSR*, pages 216–227, 2008.
- 20 Daniel Král, Jan Kratochvíl, Zsolt Tuza, and Gerhard J. Woeginger. Complexity of coloring graphs without forbidden induced subgraphs. In *WG*, pages 254–262, 2001.
- 21 Stefan Kratsch and Pascal Schweitzer. Graph isomorphism for graph classes characterized by two forbidden induced subgraphs. In *WG*, pages 34–45, 2012.
- 22 Stefan Kratsch and Pascal Schweitzer. Graph isomorphism for graph classes characterized by two forbidden induced subgraphs. *CoRR*, abs/1208.0142, 2012.
- 23 Vadim V. Lozin. A decidability result for the dominating set problem. *Theoretical Computer Science*, 411(44–46):4023–4027, 2010.
- 24 Eugene M. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *Journal of Computer and System Sciences*, 25(1):42–65, 1982.

- 25 Gary L. Miller. Isomorphism testing and canonical forms for k -contractable graphs (a generalization of bounded valence and bounded genus). In *FCT*, pages 310–327, 1983.
- 26 Yota Otachi and Pascal Schweitzer. Isomorphism on subgraph-closed graph classes: A complexity dichotomy and intermediate graph classes. In *ISAAC*, pages 111–118, 2013.
- 27 Michaël Rao. Decomposition of (gem,co-gem)-free graphs. Unpublished, available at <http://www.labri.fr/perso/rao/publi/decompgemcogem.ps>, 2007.
- 28 Pascal Schweitzer. *Problems of unknown complexity: Graph isomorphism and Ramsey theoretic numbers*. PhD thesis, Universität des Saarlandes, Germany, 2009.
- 29 Pascal Schweitzer. Towards an isomorphism dichotomy for hereditary graph classes. *CoRR*, abs/1411.1977, 2014.
- 30 Ákos Seress. *Permutation Group Algorithms*. Cambridge Tracts in Mathematics. Cambridge University Press, 2003.