Graph Isomorphism for Unit Square Graphs

Daniel Neuen

RWTH Aachen University, Aachen, Germany
neuen@informatik.rwth-aachen.de

Abstract

In the past decades for more and more graph classes the Graph Isomorphism Problem was shown to be solvable in polynomial time. An interesting family of graph classes arises from intersection graphs of geometric objects. In this work we show that the Graph Isomorphism Problem for unit square graphs, intersection graphs of axis-parallel unit squares in the plane, can be solved in polynomial time. Since the recognition problem for this class of graphs is NP-hard we can not rely on standard techniques for geometric graphs based on constructing a canonical realization. Instead, we develop new techniques which combine structural insights into the class of unit square graphs with understanding of the automorphism group of such graphs. For the latter we introduce a generalization of bounded degree graphs which is used to capture the main structure of unit square graphs. Using group theoretic algorithms we obtain sufficient information to solve the isomorphism problem for unit square graphs.

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

Keywords and phrases graph isomorphism, geometric graphs, unit squares

Digital Object Identifier 10.4230/LIPIcs.ESA.2016.70

1 Introduction

The Graph Isomorphism Problem is one of the most famous open problems in theoretical computer science. In the past three decades the problem was intensively studied but only recently the upper bound on the complexity could be improved to quasipolynomial time [2]. However, it is still open whether the Graph Isomorphism Problem can be solved in polynomial time. In this work we focus on geometric graph classes, that is, graph classes that arise as intersection graphs of geometric objects. In an intersection graph the vertices are identified with geometric objects and two vertices are connected if the corresponding objects intersect.

One of the most basic geometric graph classes is the class of interval graphs, intersection graphs of intervals on the real line. Although this graph class is quite restrictive there are a number of practical applications and specialized algorithms for interval graphs (see e.g. [15]). The Graph Isomorphism Problem on interval graphs can be solved in linear time [11] as well as in logarithmic space [18]. However, for several generalizations of interval graphs the complexity of the Graph Isomorphism Problem is unknown. This includes for example circular arc graphs (see [12]) and triangle graphs (see [30]). On the other hand a graph class is GI-complete if the Graph Isomorphism Problem for this class is as difficult as the general problem under polynomial time reductions. An example of a GI-complete geometric class is the class of grid intersection graphs, bipartite intersection graphs of horizontal and vertical line segments in the plane [29]. As an immediate consequence the class of intersection graphs of axis-parallel rectangles is also GI-complete. Unit square graphs, intersection graphs of axis-parallel unit squares, are a natural restriction for the rectangle graphs. This raises

24th Annual European Symposium on Algorithms (ESA 2016).
Editors: Piotr Sankowski and Christos Zaroliagis; Article No. 70; pp. 70:1–70:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

the question for the complexity of the isomorphism problem for unit square graphs. In this work we prove that the Graph Isomorphism Problem for unit square graphs can be solved in polynomial time. Besides being a natural restriction to rectangle graphs, another central motivation to study this problem comes from unit disk graphs, intersection graphs of unit circles in the plane. Unit disk graphs where first studied by Clark, Colbourn and Johnson in [10] and for several problems specialized algorithms have been proposed (see e.g. [13]). Practical applications arise for example from broadcast networks where each broadcast station is represented by a vertex and two stations communicate with each other if the distance between them does not exceed the broadcast range. In the work of Clark et al. two problems, namely the recognition problem and the isomorphism problem, were left open. While recognition of unit disk graphs proved to be NP-hard [8], the isomorphism problem for unit disk graphs is still open. Unit square graphs present a natural variant to unit disk graphs as we just replace the Euclidean norm by the Manhattan norm. Also, going from unit disks to unit squares removes geometric intricacies and tends to simplify the structure of graphs but maintains several key aspects of the problem. In particular, for unit disk as well as unit square graphs vertices only have a bounded number of independent neighbors (set of pairwise non-adjacent neighbors) and the structure of graphs seems to be a mixture of bounded degree and planarity. Hence, solving the isomorphism problem for unit square graphs might be a step towards solving the same problem for unit disk graphs. Furthermore, in [29, 30] Uehara also asked for the complexity of graph isomorphism for unit grid intersection graphs. Unit grid intersection graphs can be seen as bipartite versions of unit square graphs in the following sense: A bipartite graph is a unit grid intersection graph if and only if it is the intersection graph of unit squares where intersections between squares belonging to vertices on the same side of the bipartition are ignored. Hence, the result presented in this work shows that in some sense the difficulty for unit grid intersection lies in recreating the information which lines are close to each other.

Another interesting point arises from the fact that, like for unit disk graphs, recognition of unit square graphs is NP-hard [7]. Hence, we obtain an example of a natural graph class with the interesting property that isomorphism tests can be performed in polynomial time whereas recognition is NP-hard. Also, the hardness result rules out the classical approach to attack the isomorphism problem. Typically, isomorphism tests for geometric graphs are based on constructing a canonical geometric representation of the graph (see e.g. [18, 20]) but, as this would also solve the recognition problem, such an approach is not possible here. Instead, our algorithm combines group theoretic techniques with geometric properties of unit square graphs. For the group theoretic machinery we extend the results developed by Luks [22] to decide isomorphism for bounded degree graphs by also allowing for example large cycles in the neighborhood of a vertex. On a geometric level this coincides in some sense with the intuition that vertices in the neighborhood of some fixed vertex are cyclically arranged around the central vertex. Using geometric properties of unit square graphs and known algorithms for other geometric graph classes such as proper circular arc graphs we can canonically (in an isomorphism-invariant way) extract such circular orderings, which can then be used by the group theoretic machinery. For this, we show a series of results giving a deep insight into the structure of neighborhoods of single vertices and neighborhoods of cliques within unit square graphs. These results not only help us to understand the structure of unit square graphs, but also we obtain significant knowledge about the structure of the automorphism group of a unit square graph. However, an obvious obstacle to this approach comes from large cliques which are connected in a uniform way to the rest of the graph and do not contain any significant structure. More precisely, such cliques may be responsible for large symmetric groups which are subgroups of the automorphism group of the whole graph. Since large symmetric groups form a clear obstacle to the group theoretic machinery and can not be handled by Luks' algorithm we have to cope with these parts of the graph in a different way. Our second main result on the structure of unit square graphs characterizes connections between cliques which are stable with respect to the color refinement algorithm (see e.g. [6, 24]), and also establishes a close connection to interval graphs. Building on this characterization we show that the color refinement algorithm can be used to cope with the symmetric parts of the graph containing no significant structure. Finally, combining the color refinement algorithm with the group theoretic machinery, we obtain an algorithm to solve the isomorphism problem for unit square graphs.

2 Preliminaries

2.1 Graphs

A *graph* is a pair $G = (V, E)$ with vertex set $V = V(G)$ and edge set $E = E(G)$. In this paper all graphs are undirected, so $E(G)$ is always irreflexive and symmetric. The *(open) neighborhood* of a vertex $v \in V(G)$ is the set $N_G(v) = N(v) = \{w \in V(G) \mid vw \in E(G)\}$ and the size of $N(v)$ is the degree of v. The *closed neighborhood* is the set $N[v] = N(v) \cup \{v\}$. Two vertices $v, w \in V(G)$ are *connected twins* if $N[v] = N[w]$. The corresponding equivalence relation, where two vertices are related if they are connected twins, will be called the *connected twins relation*. A *path* from v to w of length m is a sequence $u_0, \ldots, u_m$ of distinct vertices with $u_0 = v$ and $u_m = w$, such that $u_{i-1}u_i \in E(G)$ for each $i \in [m] := \{1, \ldots, m\}$. The *distance* between v and w, $d(v, w)$, is the length of a shortest path from v to w. A colored graph is a tuple $G = (V, E, c)$ where $c\colon V(G) \to \mathbb{N}$ assigns each vertex a unique color. For each color $i \in \mathbb{N}$ let $V_i(G) = \{v \in V(G) \mid c(v) = i\}$. Two graphs G and H are *isomorphic* ($G \cong H$) if there is a bijection $\varphi\colon V(G) \to V(H)$, such that $vw \in E(G)$ if and only if $\varphi(v)\varphi(w) \in E(H)$ for all $v, w \in V(G)$. In this case the mapping φ is an *isomorphism* from G to H. In case the input graphs are colored it is demanded that the isomorphism also preserves the coloring of the vertices. The Graph Isomorphism Problem asks whether two given graphs G and H are isomorphic. An isomorphism from a graph to itself is called an *automorphism*. The set of automorphisms of a graph G, denoted by $\mathrm{Aut}(G)$, forms a subgroup of the symmetric group over the vertex set.

2.2 Color Refinement

A very basic and fundamental method, which is a basic building block of many isomorphism tests, is the color refinement algorithm (see e.g. [24]). The basic idea is to iteratively distinguish vertices if they have a different number of neighbors in some color. A partition $\mathcal{P}$ of the vertices is *stable* if for all $X, Y \in \mathcal{P}$ and all $v, w \in X$ it holds that $|N(v) \cap Y| = |N(w) \cap Y|$. Further a partition $\mathcal{P}$ *refines* another partition $\mathcal{Q}$ if for each $X \in \mathcal{P}$ there is some $Y \in \mathcal{Q}$ with $X \subseteq Y$. The color refinement algorithm computes the unique coarsest stable partition refining the initial color partition (i.e. the partition of the vertices according to their color). The coarsest stable partition can be computed in almost linear time (see [24, 6]). We say color refinement distinguishes two graphs if there is some class in the coarsest stable partition on the disjoint union of the graphs that contains a different number of vertices from the two graphs. In this case the two input graphs are not isomorphic.

The k-dimensional Weisfeiler-Leman algorithm is a generalization of the color refinement algorithm (cf. [9]). Instead of coloring only single vertices, the Weisfeiler-Leman algorithm colors k-tuples of vertices. Initially each tuple is colored with the isomorphism type of the underlying induced subgraph and then the coloring is refined in a similar way as by the color refinement algorithm (see [9] for a detailed description). We say k-dimensional Weisfeiler-Leman *identifies* a graph class $\mathcal{C}$ if for every pair of non-isomorphic graphs G, H with $G \in \mathcal{C}$ the k-dimensional Weisfeiler-Leman distinguishes between G and H.

2.3 Group Theory

In this subsection we briefly introduce the main group theoretic tools used in this work. For a general introduction to group theory we refer to [25] whereas several group theoretic algorithms are given in [16, 27]. Since we only deal with automorphism groups of graphs we can restrict ourselves to permutation groups which throughout this work will always be represented by generating sets of size polynomial in the size of the permutation domain. For a set Ω let $\mathrm{Sym}(\Omega)$ be the symmetric group over the set Ω. In particular we require a certain subclass of permutation groups, namely groups with bounded non-abelian composition factors. Let Γ be a group. A *normal series* is a sequence of subgroups $\Gamma = \Lambda_0 \trianglerighteq \Lambda_1 \trianglerighteq \cdots \trianglerighteq \Lambda_k = \{1\}$. The length of the series is k and the groups Λ_{i-1}/Λ_i are the factor groups of the series, $i \in [k]$. A *composition series* is a strictly decreasing normal series of maximal length. For every finite group Γ all composition series have the same family of factor groups considered as a multi-set (cf. [25]). A *composition factor* of a finite group Γ is a factor group of a composition series of Γ.

▶ **Definition 2.1.** Let $d \in \mathbb{N}$. The family Γ_d contains all finite groups Γ for which all non-abelian composition factors are isomorphic to subgroups of $S_d = \mathrm{Sym}([d])$.

The class of Γ_d-groups is closed under subgroups and homomorphic images. Furthermore, for groups $N \trianglelefteq \Gamma$ it holds that $\Gamma \in \Gamma_d$ if and only if $N \in \Gamma_d$ and $\Gamma/N \in \Gamma_d$ (cf. [22]). A group Γ is *solvable* if every composition factor is abelian. Two examples of solvable groups, that are particularly important for this work, are cyclic groups and dihedral groups which are the automorphism groups of directed cycles and undirected cycles. Note that every solvable group is a Γ_d-group for every $d \in \mathbb{N}$.

The Setwise Stabilizer Problem asks, given a permutation group $\Gamma \leq \mathrm{Sym}(\Omega)$ and $A \subseteq \Omega$, for a generating set of the group $\mathrm{Stab}_\Gamma(A) := \{\gamma \in \Gamma \mid A = A^\gamma\}$, where $A^\gamma = \{\alpha^\gamma \mid \alpha \in A\}$ and α^γ is the image of α under the permutation γ. A central motivation to consider Γ_d-groups is the following result.

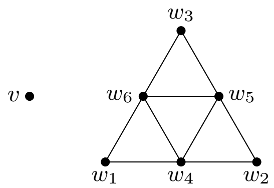
▶ **Theorem 2.2.** *Let $d \in \mathbb{N}$. The Setwise Stabilizer Problem for groups in Γ_d can be solved in polynomial time.*

A weaker version of this statement was proved by Luks in [22] considering only groups where all composition factors are isomorphic to subgroups of S_d. For the more general version stated above we refer to [1]. This result is for example used by Luks to solve graph isomorphism for graphs of bounded degree [22], but it can also be applied to more general graph classes such as t-bounded graphs (see e.g. [4]). For this work we introduce a slight variation, namely graphs which we call *t-circle-bounded* graphs. For a graph G and a set $X \subseteq V(G)$ we write $G[X]$ to denote the induced subgraph of G with vertex set X.

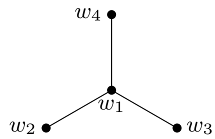
▶ **Definition 2.3.** A colored graph $G = (V, E, c)$ with $c \colon V(G) \to [k]$ is *t-circle-bounded* if for each $i \in [k]$ and $X \subseteq V_{<i} := \bigcup_{j<i} V_j(G)$ the graph

$$G_{i,X} := G[\{v \in V_i(G) \mid N(v) \cap V_{<i} = X\}]$$

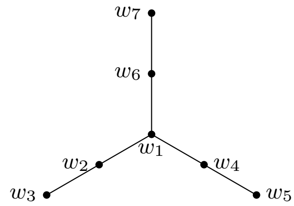
is the disjoint union of at most t connected graphs of maximum degree two.



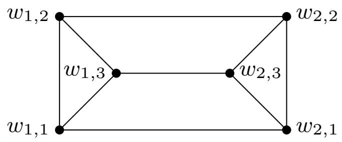
(a) $K_1 \cup S_3$



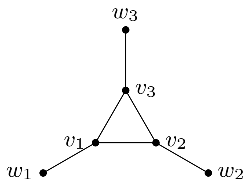
(b) $K_{1,3}$ (`claw`)



(c) T_2



(d) $\overline{C_6}$



(e) `net`

Figure 1 Some forbidden induced subgraphs of proper circular arc graphs.

The t-circle-bounded graphs are closely related to t-bounded graphs which are similarly defined with the size of $G_{i,X}$ bounded by t (see [4]). From an algorithmic point of view we can use very similar methods for the isomorphism problem on t-circle-bounded graphs as for t-bounded graphs.

▶ **Theorem 2.4.** *Let G be a t-circle-bounded graph. Then* $\mathrm{Aut}(G) \in \Gamma_t$.

▶ **Theorem 2.5.** *The Graph Isomorphism Problem for t-circle-bounded graphs can be solved in polynomial time.*

Both theorems can be proved in very similar fashion as the respective statements for t-bounded graphs. In fact the only additional argument required for t-circle-bounded graphs is that the automorphism groups of connected graphs of maximum degree two are solvable and thus elements of Γ_t.

2.4 Proper circular arc graphs

A graph G is a *unit interval graph* if G is the intersection graph of unit intervals on the real line. A graph G is a *proper circular arc graph* if G is the intersection graph of arcs on a circle, such that for no two arcs one is properly contained in the other. A characterization of unit interval graphs and proper circular arc graphs in terms of forbidden induced subgraphs is given in [28]. For our purposes the following statements are sufficient. Some relevant forbidden induced subgraphs are also depicted in Figure 1.

▶ **Theorem 2.6.** *A graph G is a unit interval graph if and only if there are no induced subgraphs isomorphic to C_{n+4} for $n \geq 0$, S_3, $K_{1,3}$ and* `net`.

Here, C_n denotes a cycle of length n. Furthermore we denote by $G \cup H$ the disjoint union of G and H and the graph $\overline{G}$ is the complement graph of G.

▶ **Lemma 2.7.** *Let G be a graph, such that $N_G[v]$ induces a unit interval graph for every $v \in V(G)$ and there are no induced subgraphs isomorphic to $K_1 \cup C_{n+4}$ for $n \geq 0$, $K_1 \cup S_3$, $\overline{T_2}$, $\overline{C_6}$ and* `net`. *Then G is a proper circular arc graph.*

We also require the following characterization for proper circular arc graphs. A vertex is *universal* if it is adjacent to all other vertices. A graph G is *twin-free* if it does not contain connected twins and G is *co-bipartite* if $\overline{G}$ is bipartite.

▶ **Theorem 2.8** ([19], Theorem 3)**.** *Let G be a graph without universal vertices. Then G is a proper circular arc graph if and only if there is a cycle H with $V(G) = V(H)$, such that*
1. *$N_G[v]$ induces a connected subgraph in H for each $v \in V(G)$*
2. *For all $v, w \in V(G)$ it holds that if $N_G[v] \subseteq N_G[w]$ then the two paths share an endpoint in H.*

Furthermore, if G is connected, twin-free and not co-bipartite, the cycle H is unique [17]. Additionally, given some proper circular arc graph, a cycle H can be computed in polynomial time (see [19]). This gives us the following result.

▶ **Theorem 2.9.** *Let G be a connected proper circular arc graph, such that $\overline{G}$ is not bipartite. Further let $\sim_G$ be the connected twins relation and $\mathcal{P}$ the corresponding partition into equivalence classes. Then one can compute in polynomial time a canonical connected graph H, such that $V(H) = \mathcal{P}$ and H has maximum degree two.*

3 Basic Properties

For unit square graphs there are several possible definitions. The most obvious one is to describe vertices by axis-parallel unit squares with edges connecting two vertices if the unit squares intersect. Alternatively it can also be demanded that vertices represented by unit squares are connected if the center of the first square is contained in the other square. Another possibility is to describe vertices by points in the plane. Note that two squares with unit side length intersect if and only if the distance between their centers using the maximum norm is at most one. Thus, two vertices are connected if the distance between the points is at most one using the maximum norm. Furthermore, a unit square contains the center point of another unit square if and only if the distance between both centers is at most one half using the maximum norm. By applying a scaling argument this also gives us the equivalence to the second definition. In this paper we work with the last definition, that is we represent vertices by points in the plane. For a point $p \in \mathbb{R}^k$ we denote by p_i the i-th component of p, $i \in [k]$. The L_∞-norm is defined as $\|p\|_\infty = \max_{i\in[k]} |p_i|$.

▶ **Definition 3.1.** A *k-dimensional L_∞-realization* of a graph G is a mapping $f : V(G) \to \mathbb{R}^k$ such that $vw \in E(G)$ if and only if $\|f(v) - f(w)\|_\infty \leq 1$ for all $v, w \in V(G)$. A *unit square graph* is a graph having a two-dimensional L_∞-realization.

Observe that graphs with 1-dimensional L_∞-realization are exactly the unit interval graphs. For the remainder of this paper we focus on unit square graphs and just use the term realization for a two-dimensional L_∞-realization. Following our previous notation, for a realization $f : V(G) \to \mathbb{R}^2$ and a vertex $v \in V(G)$, we denote by $(f(v))_i$ the i-th component of $f(v)$. This is also abbreviated by $f(v)_i$, $i \in [2]$. We start by listing some basic properties for unit square graphs.

▶ **Observation 3.2.** *Let G be a unit square graph and $f : V(G) \to \mathbb{R}^2$ a realization. Further let $X \subseteq V(G)$, such that there are $a_1, b_1, a_2, b_2 \in \mathbb{R}$ with $a_1 \leq b_1 \leq a_1 + 1$, $a_2 \leq b_2$ and $f(v) \in [a_1, b_1] \times [a_2, b_2]$ for every $v \in X$. Then $G[X]$ is a unit interval graph.*

▶ **Lemma 3.3.** *Let G be a unit square graph. Then the following properties hold:*
1. *There is some $v \in V(G)$, such that $G[N[v]]$ is a unit interval graph.*

2. *For every two non-adjacent $u, v \in V(G)$ the set $N(u) \cap N(v)$ induces a unit interval graph with at most two independent vertices.*
3. *G has no induced subgraph isomorphic to $K_{1,5}$, $K_{2,3}$ or $\overline{3K_2}$ ($3K_2$ is the disjoint union of three K_2).*

Proof. Let $f\colon V(G) \to \mathbb{R}^2$ be a realization. Pick $v = \operatorname{argmin}_{v \in V(G)} f(v)_1$. Further let $a_1 = f(v)_1$, $b_1 = a_1 + 1$, $a_2 = f(v)_2 - 1$ and $b_2 = f(v)_2 + 1$. Then $f(w) \in [a_1, b_1] \times [a_2, b_2]$ for every $w \in N[v]$. So $G[N[v]]$ is a unit interval graph by Observation 3.2.

Now let $u, v \in V(G)$ be two non-adjacent vertices. Without loss of generality assume $f(u)_1 + 1 \leq f(v)_1$. For $w \in N(u) \cap N(v)$ we obtain $f(u)_1 \leq f(w)_1 \leq f(v)_1$. Without loss of generality let $f(u) = (0, 0)$. Then $f(w) \in [0, 1] \times [-1, 1]$ for every $w \in N(u) \cap N(v)$. So $N(u) \cap N(v)$ defines a unit interval graph according to Observation 3.2. Furthermore in this area there can be at most two independent vertices.

For the third item first observe that the class of unit square graphs is hereditary (i.e. it is closed under taking induced subgraphs) so it suffices to show that the listed graphs are not unit square graphs. We first consider the graph $K_{1,5}$. Suppose towards a contradiction that there is a realization $f : V(G) \to \mathbb{R}^2$. Without loss of generality let $f(v) = (0, 0)$ where v is the center vertex connected to the other vertices $w_1, \dots, w_5$. Then there is some quadrant containing two vertices w_i and w_j for distinct $i, j \in [5]$. But then $w_i w_j \in E(G)$ which is a contradiction.

For $K_{2,3}$ the two vertices on the left side have three independent common neighbors. The graph $\overline{3K_2}$ contains two non-adjacent vertices whose common neighborhood is a 4-cycle. So in both cases it follows from the second statement that the graph is not a unit square graph. ◀

We also require some properties of maximal cliques. A clique is a set $C \subseteq V(G)$, such that $vw \in E(G)$ for every two distinct $v, w \in C$. A maximal clique is a clique so that there is no larger clique containing it. For a graph G the set of maximal cliques of G is denoted by $\mathcal{M}(G)$.

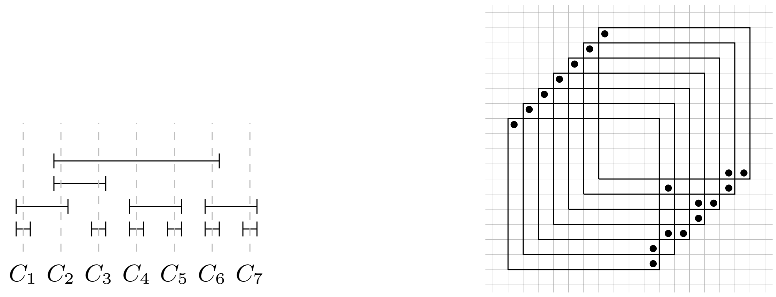
▶ **Lemma 3.4.** *Let G be a unit square graph and C be a maximal clique of G. Then there are $v_1, \dots, v_4 \in V(G)$, such that $C = \bigcap_{i \in [4]} N[v_i]$.*

Proof. Let $f : V(G) \to \mathbb{R}^2$ be a realization of G. Let $v_{2i-1} = \operatorname{argmin}_{v \in C} f(v)_i$ and $v_{2i} = \operatorname{argmax}_{v \in C} f(v)_i$ for $i \in [2]$. Clearly $C \subseteq \bigcap_{i \in [4]} N[v_i]$. So let $w \in \bigcap_{i \in [4]} N[v_i]$ and $v \in C$. In order to prove $w \in C$ it suffices to show that $\|f(v) - f(w)\|_\infty \leq 1$, since v is chosen arbitrarily from C. For $i \in [2]$ it holds that $f(v_{2i-1})_i \leq f(v_{2i})_i$. If $f(w)_i \leq f(v_{2i})_i$ then $-1 \leq f(v_{2i})_i - 1 - f(v)_i \leq f(w)_i - f(v)_i \leq f(v_{2i})_i - f(v)_i \leq 1$. Otherwise $f(w)_i \geq f(v_{2i-1})_i$ and $-1 \leq f(v_{2i-1})_i - f(v)_i \leq f(w)_i - f(v)_i \leq f(v_{2i-1})_i + 1 - f(v)_i \leq 1$. ◀

In particular all maximal cliques can be computed in polynomial time.

4 Local structure

The basic approach for our algorithm is group-theoretic. A main obstacle for group theoretic approaches comes from large symmetric or alternating groups that appear in the automorphism group of the given graph. For unit square graphs the central observation is that these groups can in a way only arise from cliques. In this section we show how to cope with possibly very symmetric parts of the graphs and give a corresponding reduction to get rid of them.



(a) An interval graph G

(b) The constructed realization for $G_{\mathcal{M}}$

Figure 2 From interval to unit square graphs.

We start by giving a central class of examples to obtain a better understanding of how the symmetric parts may look like. Let G be a graph. Define the colored graph $G_{\mathcal{M}} = (V(G) \cup \mathcal{M}(G), E(G_{\mathcal{M}}), c_{G_{\mathcal{M}}})$ with

$$E(G_{\mathcal{M}}) = \{vC \mid C \in \mathcal{M}(G), v \in C\} \cup \{vw \mid v \neq w \in V(G)\} \cup \{CD \mid C \neq D \in \mathcal{M}(G)\}$$

and $c_{G_{\mathcal{M}}}(v) = c_G(v) + 1$ for $v \in V(G)$, $c_{G_{\mathcal{M}}}(C) = 1$ for $C \in \mathcal{M}(G)$. For a group $\Gamma \leq \mathrm{Sym}(\Omega)$ and a set $A \subseteq \Omega$ the *pointwise stabilizer* is the group $\mathrm{Stab}^{\bullet}_{\Gamma}(A) := \{\gamma \in \Gamma \mid \forall \alpha \in A \colon \alpha^{\gamma} = \alpha\}$. If A is invariant under Γ (i.e. $A^{\gamma} = A$ for every $\gamma \in \Gamma$) we define the restriction of Γ to A as $\Gamma|_A := \{\gamma|_A \mid \gamma \in \Gamma\}$ where $\gamma|_A \colon A \to A$ with $\gamma|_A(\alpha) = \gamma(\alpha)$.

▶ **Observation 4.1.** *For every two graphs G and H it holds that*
1. $G \cong H$ *if and only if* $G_{\mathcal{M}} \cong H_{\mathcal{M}}$,
2. $\mathrm{Stab}^{\bullet}_{\mathrm{Aut}(G_{\mathcal{M}})}(V(G)) = \{1\}$ *(here* 1 *denotes the identify mapping),*
3. $\mathrm{Aut}(G_{\mathcal{M}})|_{V(G)} = \mathrm{Aut}(G)$.

In particular $\mathrm{Aut}(G) \cong \mathrm{Aut}(G_{\mathcal{M}})$ by combining the second and third part of the observation. We now show that for each interval graph G the graph $G_{\mathcal{M}}$ is a unit square graph. For this we use the following characterization of interval graphs: A graph G is an interval graph if and only if there is a linear order on the maximal cliques, such that each vertex appears in consecutive maximal cliques [14]. For a vertex $v \in V(G)$ let $\mathcal{M}_G(v) = \mathcal{M}(v) = \{C \in \mathcal{M}(G) \mid v \in C\}$.

▶ **Lemma 4.2.** *Let G be an interval graph. Then $G_{\mathcal{M}}$ is a colored unit square graph.*

Proof. Let $<$ be a linear order on $\mathcal{M}(G)$, such that each vertex appears in consecutive maximal cliques. Let $k = |\mathcal{M}(G)|$ and $C_1 < C_2 < \cdots < C_k$ be the maximal cliques of G. For each $v \in V(G)$ define $a_v, b_v \in [k]$ in such a way that $\mathcal{M}(v) = \{C_i \mid a_v \leq i \leq b_v\}$. Consider the following realization $f : V(G_{\mathcal{M}}) \to \mathbb{R}^2$ with $f(C_i) = (\frac{i}{k} - 1, \frac{i}{k})$ and $f(v) = (\frac{a_v}{k}, \frac{b_v}{k} - 1)$ for all $v \in V(G)$. Clearly all vertices are connected to each other as well as all maximal cliques. So let $v \in V(G)$. Then $|\frac{a_v}{k} - \frac{i}{k} + 1| = |\frac{a_v - i}{k} + 1| \leq 1$ if and only if $a_v \leq i$. Further $|\frac{i}{k} - \frac{b_v}{k} + 1| = |\frac{i - b_v}{k} + 1| \leq 1$ if and only if $i \leq b_v$. So there is an edge between $v \in V(G)$ and $C_i \in \mathcal{M}$ if and only if $a_v \leq i \leq b_v$ if and only if $v \in C_i$. ◀

A visualization of the presented realization is also given in Figure 2. In Figure 2b the vertices of $G_{\mathcal{M}}$ are represented by points. The squares are only for visualization purposes and indicate which vertices are connected. Each square describes a maximal clique of the original interval graph and contains exactly the vertex which corresponds to the given clique and the vertices being in the clique.

▶ **Corollary 4.3.** *For each colored interval graph G one can compute in polynomial time some colored unit square graph H with* $\mathrm{Aut}(G) \cong \mathrm{Aut}(H)$.

In particular this construction implies that there are twin-free unit square graphs where the automorphism group contains arbitrarily large symmetric groups which can not be handled by the group theoretic approach due to Luks. For example consider the following graph $G_{n,k}$ for $n, k \in \mathbb{N}$. The vertex set $V(G_{n,k}) = [n]^{\leq k}$ is the set of all words over the alphabet $[n]$ of length at most k and there is an edge $vw \in E(G_{n,k})$ if v is a prefix of w (this is interpreted for an undirected graph). First, $G_{n,k}$ is an interval graph. To verify this consider the set $[n]^k$ of words of length k with the natural lexicographic order. Then each vertex $v \in V(G_{n,k})$ can be represented by the interval $I_{n,k}(v) = \{w \in [n]^k \mid v \text{ is prefix of } w\}$. It is easy to check that two vertices are connected if and only if the corresponding intervals intersect. The automorphism group of $G_{n,k}$ is a wreath product of the automorphism group of $G_{n,k-1}$ by a symmetric group on n points.

One of the main contributions of this work is to show that within automorphism groups of unit square graphs large symmetric groups only appear in a local setting. Here, local refers to a small area in the realization of a unit square graph G. In the presented example the vertices of each color class are close together and in particular they induce a clique. The main target for this section is to present a method how to cope with the local parts of the graph that may admit large symmetric groups in the automorphism group. For this purpose we have to analyze the structure of clique-partitions of the vertices.

▶ **Definition 4.4.** Let G be a graph and $\mathcal{P}$ be a partition of the vertices. We call $\mathcal{P}$ a *clique-partition* if X is a clique for each $X \in \mathcal{P}$.

Let G be a unit square graph with realization $f : V(G) \to \mathbb{R}^2$. We first define the graph $G^*_{\mathcal{M}} = (V(G) \cup \mathcal{M}(G), E(G^*_{\mathcal{M}}), c_{G^*_{\mathcal{M}}})$ with $E(G^*_{\mathcal{M}}) = \{vC \mid C \in \mathcal{M}(G), v \in C\} \cup E(G)$ and $c_{G^*_{\mathcal{M}}} = c_{G_{\mathcal{M}}}$ as defined above. Let $\mathcal{P}$ be a clique-partition of $V(G)$, which is refined by the color refinement algorithm applied to the graph $G^*_{\mathcal{M}}$. More precisely let $\mathcal{P}^*$ be the unique coarsest partition of $V(G) \cup \mathcal{M}(G)$ that is stable with respect to $G^*_{\mathcal{M}}$ and refines the partition $\mathcal{P} \cup \{\mathcal{M}(G)\}$. The partition $\mathcal{P}$ is called *clique-stable* if $\mathcal{P}^* \cap 2^{V(G)} = \mathcal{P}$.

▶ **Lemma 4.5.** *Let G be a twin-free unit square graph and let $\mathcal{P}$ be a clique-stable partition. Further let $f\colon V(G) \to \mathbb{R}^2$ be a realization. Then the following properties hold:*

1. *For each $X \in \mathcal{P}$ there exists $b \in \{-1, 1\}$, such that*
$$f(v)_1 \leq f(w)_1 \;\Leftrightarrow\; b \cdot f(v)_2 \leq b \cdot f(w)_2 \tag{1}$$
for all $v, w \in X$. The value b is called the orientation of X, *which is denoted by* $\mathrm{ori}_{G,f}(X)$ *(the value b is unique unless $|X| = 1$, in this case we define* $\mathrm{ori}_{G,f}(X) = 1$*).*
2. *Let $X, Y \in \mathcal{P}$ with $\mathrm{ori}_{G,f}(X) \neq \mathrm{ori}_{G,f}(Y)$. Then either $xy \in E(G)$ for all $x \in X, y \in Y$ or there is no $x \in X, y \in Y$ with $xy \in E(G)$.*
3. *Let $X, Y \in \mathcal{P}$ with $\mathrm{ori}_{G,f}(X) = \mathrm{ori}_{G,f}(Y)$ and suppose $k = |\{xy \in X \times Y \mid xy \in E(G)\}| \geq 1$. Further let $X = \{x_1, \ldots, x_s\}$, such that $f(x_i)_1 \leq f(x_{i+1})_1$ for all $i \in [s]$, and $Y = \{y_1, \ldots, y_t\}$, such that $f(y_j)_1 \leq f(y_{j+1})_1$ for all $j \in [t]$. Then*
$$x_i y_j \in E(G) \;\Leftrightarrow\; \left\lceil \frac{i \cdot t}{k} \right\rceil = \left\lceil \frac{j \cdot s}{k} \right\rceil. \tag{2}$$

Now let $\mathcal{P}$ be a canonical, clique-stable partition of the graph G. We define the vertex and edge-colored graph $G[\mathcal{P}] = (\mathcal{P}, E, c_V, c_E)$ with $E = \{XY \mid X \neq Y \in \mathcal{P}\}$,

$c_V : \mathcal{P} \to \mathbb{N}\colon X \mapsto |X|$

and

$$c_E \colon E \to \mathbb{N} \colon XY \mapsto |\{xy \in X \times Y \mid xy \in E(G)\}|.$$

For $\gamma \in \mathrm{Aut}(G)$ define the permutation $\gamma^{\mathcal{P}} := \varphi(\gamma) \in \mathrm{Sym}(\mathcal{P})$ where $\varphi \colon \mathrm{Aut}(G) \to \mathrm{Sym}(\mathcal{P})$ is the natural action of $\mathrm{Aut}(G)$ on $\mathcal{P}$. Note that φ is well-defined since the partition $\mathcal{P}$ is canonical.

▶ **Theorem 4.6.** *Let G be a twin-free unit square graph and let $\mathcal{P}$ be a canonical, clique-stable partition. Further let $\delta \in \mathrm{Aut}(G[\mathcal{P}])$. Then there is some $\gamma \in \mathrm{Aut}(G)$ with $\gamma^{\mathcal{P}} = \delta$.*

Intuitively, the last theorem states that each automorphism of $G[\mathcal{P}]$ naturally extends to an automorphism of G. In particular, the graph G can be uniquely reconstructed from the graph $G[\mathcal{P}]$. This is the main result on the local structure of unit square graphs which allows us, for a canonical, clique-stable partition $\mathcal{P}$, to restrict to the graph $G[\mathcal{P}]$. For the remainder of this work the goal is to compute a canonical, clique-stable partition $\mathcal{P}$, such that the automorphism group of $G[\mathcal{P}]$ is a Γ_t-group for some constant t. To achieve this goal we require the graph G to have some singleton vertex v_0 (a vertex with a unique color). More precisely, for such a graph we construct the desired partition $\mathcal{P}$ and a canonical, t-circle-bounded graph H, such that $\mathcal{P} \subseteq V(H)$ and $\mathcal{P}$ is invariant under $\mathrm{Aut}(H)$. This results in a good supergroup of the group $\mathrm{Aut}(G[\mathcal{P}])$ which can be used by Luks' algorithm to compute the real automorphism group. To compute the graph H we devise an algorithm that iteratively extends H taking vertices with larger and larger distances to v_0 into account. While doing so the crucial subproblem is to compute canonical clique-partitions for neighborhoods of cliques. This problem is addressed in the next section.

5 Neighborhoods

In order to obtain canonical clique-partitions for neighborhoods we essentially proceed in two steps. First, we use some combinatorial partitioning techniques to obtain some initial coloring of the vertices. Then, considering each color class separately, the main contribution is to prove that each color class can either be described by a co-bipartite graph or a proper circular arc graph. In both cases it is easy to compute a canonical clique-partition.

5.1 Neighborhood graphs

Before considering neighborhoods of cliques we first restrict to neighborhoods of vertices. This occurs as a subcase when analyzing neighborhood of cliques. Also the structure of neighborhoods tends to be simpler than for neighborhoods of cliques.

▶ **Definition 5.1.** A unit square graph is a *neighborhood graph* if there is a realization $f \colon V(G) \to [-1, 1]^2$.

Note that every graph induced on a neighborhood of a vertex is indeed a neighborhood graph and every neighborhood graph can be turned into the neighborhood of a vertex by adding a universal vertex located at the origin. Let G be a neighborhood unit square graph. The goal is to prove that, after performing the k-dimensional Weisfeiler-Leman algorithm for sufficiently large k, each color class of vertices is co-bipartite or proper circular arc. We build on the characterization of proper circular arc graphs in terms of forbidden induced subgraphs. We start by giving two general graph-theoretic lemmas.

▶ **Lemma 5.2.** *Let G be a graph, such that*
1. *$G[N[v]]$ is a unit interval graph for each $v \in V(G)$,*
2. *G has no induced subgraph isomorphic to $C_4 \cup K_1$.*

Further let $X = \{w_1 \in V(G) \mid \exists w_2, \dots, w_6 \colon G[w_1, \dots, w_6] \cong \overline{C_6}\}$. Then $G[X]$ is co-bipartite.

▶ **Lemma 5.3.** *Let G be a graph, such that*
1. *$G[N[v]]$ is a unit interval graph for each $v \in V(G)$,*
2. *G has no induced subgraph isomorphic to $C_{n+4} \cup K_1$ for $n \geq 0$,*
3. *there are no $v, w \in V(G)$, such that $N[v] \subsetneq N[w]$.*

Then G has no induced subgraph isomorphic to `net`.

Now let G be a neighborhood unit square graph. In order to apply Lemma 2.7 we still need to consider $C_{n+4} \cup K_1$ and $S_3 \cup K_1$.

▶ **Lemma 5.4.** *Let G be a neighborhood unit square graph. Let $X = \{v \in V(G) \mid \exists \ell \geq 4\, \exists w_1, \dots, w_\ell : vw_i \notin E(G) \wedge G[w_1, \dots, w_\ell] \cong C_\ell\}$. Then $X \neq V(G)$.*

Proof. Let $f \colon V(G) \to [-1,1]^2$ be a realization and let $v = \text{argmin}_{v \in V(G)} |f(v)_1|$. Suppose towards a contradiction that $v \in X$. Then there is some $\ell \geq 4$ and $w_1, \dots, w_\ell \in V(G)$, such that $vw_i \notin E(G)$ for all $i \in [\ell]$ and $w_i w_j \in E(G)$ if and only if $i - j \equiv \pm 1 \mod \ell$ for all $i, j \in [\ell]$. Without loss of generality assume that $f(v) \in [-1,0] \times [-1,0]$. Let $i = \text{argmin}_{i \in [\ell]} f(w_i)_2$. Since $G[w_1, \dots, w_\ell]$ is not a unit interval graph it holds that $f(w_i) \in [0,1] \times [-1,0]$ by Observation 3.2. Without loss of generality assume $i = 2$. Now consider the two neighbors w_1 and w_3. Note that $w_1 w_3 \notin E(G)$ since $\ell \geq 4$. Then there is some $j \in \{1,3\}$, such that $f(w_j) \in [-1,0) \times [0,1]$. So in particular $f(w_j)_1 < 0$. Further $f(w_j)_1 + 1 \geq f(w_2)_1$ and $f(v)_1 + 1 < f(w_2)_1$. Altogether this means that $f(v)_1 < f(w_j)_1 < 0$ contradicting the definition of v. ◀

▶ **Lemma 5.5.** *Let G be a neighborhood unit square graph. Let $X = \{v \in V(G) \mid \exists w_1, \dots, w_6 : vw_i \notin E(G) \wedge G[w_1, \dots, w_6] \cong S_3\}$. Then $X \neq V(G)$.*

This lemma is proved in a similar fashion to Lemma 5.4. In order to prove the main partitioning result for neighborhood graphs we also require that for sufficiently large k the k-dimensional Weisfeiler-Leman algorithm identifies all interval graphs (cf. [21]).

▶ **Corollary 5.6.** *There is some $k \in \mathbb{N}$, such that for each neighborhood unit square graph the following holds: After performing k-dimensional Weisfeiler-Leman each equivalence class of vertices induces a graph which is co-bipartite or proper circular arc with at most four connected components.*

Proof. Choose k sufficiently large and let $X \subseteq V(G)$ be an equivalence class. Then $G[X]$ is a neighborhood unit square graph. By Lemma 3.3 there is some $v \in X$, such that $N_{G[X]}[v]$ induces a unit interval graph. Since k-dimensional Weisfeiler-Leman identifies all interval graphs this is true for all $v \in X$. From Lemma 5.4 it follows that there exists a vertex $v \in X$, such that every induced cycle contains at least one vertex being a neighbor of v. Again by stability of the set X with respect to k-dimensional Weisfeiler-Leman this is true for all $v \in X$ (note that the maximal length of an induced cycle is at most 8). So there is no induced subgraph isomorphic to $C_{n+4} \cup K_1$. Combining the same argument with Lemma 5.5 we get that $G[X]$ also has no induced subgraph isomorphic to $S_3 \cup K_1$. Since $G[X]$ is regular there are no vertices $v, w \in X$, such that $N_{G[X]}[v] \subsetneq N_{G[X]}[w]$. So we can apply Lemma 5.3 and obtain that there is no induced subgraph isomorphic to `net`. Furthermore $G[X]$ has no induced subgraph isomorphic to $\overline{3K_2}$ by Lemma 3.3 and therefore it has also no induced subgraph $\overline{T_2}$. Now suppose there is an induced subgraph isomorphic to $\overline{C_6}$. Then, by stability, every vertex is part of an induced subgraph $\overline{C_6}$ and thus, $G[X]$ is co-bipartite by Lemma 5.2. Otherwise $G[X]$ is proper circular arc by Lemma 2.7. The bound on the number of components follows from the fact that $K_{1,5}$ is not a unit square graph (cf. Lemma 3.3). ◀

▶ **Theorem 5.7.** *Let G be a neighborhood unit square graph. Then one can compute in polynomial time a canonical clique-partition $\mathcal{P}$ and a canonical colored graph H, such that*

1. $\mathcal{P} = V(H)$,
2. *H is 4-circle-bounded,*
3. $\mathrm{im}(\varphi) \leq \mathrm{Aut}(H)$ *where* $\varphi\colon \mathrm{Aut}(G) \to \mathrm{Sym}(\mathcal{P})$ *is the natural action of* $\mathrm{Aut}(G)$ *on* $\mathcal{P}$.

Proof. Choose k according to Corollary 5.6 and let $X \subseteq V(G)$ be an equivalence class after performing k-dimensional Weisfeiler-Leman. Further let c be the color of the equivalence class. First suppose $G[X]$ is co-bipartite. Let t be the number of non-trivial connected components of $\overline{G[X]}$. Then $t \leq 2$ by Lemma 3.3. Let $Y_{j,1}, Y_{j,2}$ be the unique bipartition of the j-th connected component, $j \in [t]$. Further let Y be the set of isolated vertices in $\overline{G[X]}$ and $\mathcal{Y} = \{Y\}$ if $Y \neq \emptyset$ and $\mathcal{Y} = \emptyset$ otherwise. Define $\mathcal{P}_X = \{Y_{j,j'} \mid j \in [t], j' \in [2]\} \cup \mathcal{Y}$. Further let $H_X = \{\mathcal{P}_X, E(H_X), c_X\}$ with $YZ \in E(H_X)$ if there are $v \in Y, w \in Z$ with $vw \in E\left(\overline{G[X]}\right)$ and $c_X(Y) = c$.

Otherwise $G[X]$ is proper circular arc according to Corollary 5.6. Let $X_1, \dots, X_t$ be the connected components of $G[X]$. Then $t \leq 4$ by Corollary 5.6. Let $i \in [t]$ and let $\mathcal{P}_{X,i}$ be the partition containing the equivalence classes of the connected twins relation for $G[X_i]$. Further let $H_{X,i}$ be the graph computed by Theorem 2.9 where each vertex is colored by c. Define $\mathcal{P}_X = \bigcup_{i \in [t]} \mathcal{P}_{X,i}$ and $H_X = \bigcup_{i \in [t]} H_{X,i}$.

Finally let $\mathcal{P} = \bigcup_X \mathcal{P}_X$ and $H = \bigcup_X H_X$. It can easily be checked that $\mathcal{P}$ and H have the desired properties. ◀

5.2 Clique neighborhoods graphs

Remember, that our goal is to compute a canonical clique-partition of a given unit square graph with singleton vertex v_0. We first group the vertices according to their distance to v_0. Then, for the first level of vertices which are all the neighbors of v_0, we use the previous theorem to compute a canonical clique-partition. For all other levels we want to build up on the partition computed in the previous level. More precisely, for a given clique in the partition of the previous level we want to partition its neighbors in the current level. Hence, we need to consider neighborhoods of cliques and extend the results of the previous subsection accordingly.

Let G be a colored unit square graph and let $X \subseteq V(G)$ be a clique, such that $V(G) = N[X] = \bigcup_{v \in X} N[v]$. Further suppose there is some color i, such that $X = V_i(G)$, and there is some $k \in [|X|]$, such that $|N[v] \cap X| = k$ for all $v \in V(G) \setminus X$. In this case G is called a *simple clique neighborhood graph with respect to X*. The next theorem extends the result of the previous subsection to simple clique neighborhood graphs. Note that we have to pay a price here, namely the constant for the circle-bounded graph increases from four to eight. This can be explained by the fact that a single vertex can have at most four independent neighbors whereas a clique can have eight independent neighbors (cf. Figure 3).

▶ **Theorem 5.8.** *Let G be a simple clique neighborhood graph with respect to $X \subseteq V(G)$. Then one can compute in polynomial time a canonical clique-partition $\mathcal{P}$ and a canonical colored graph H, such that*

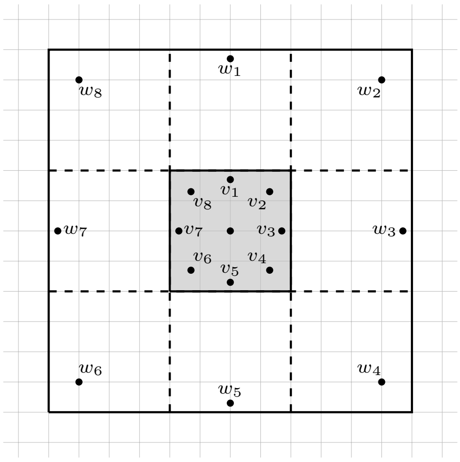


Figure 3 Neighborhood of a clique and realization for the graph G_8

1. $\mathcal{P} \subseteq V(H)$ *and* $\mathcal{P}$ *is* $\mathrm{Aut}(H)$*-invariant,*
2. H *is* 8*-circle-bounded,*
3. $\mathrm{im}(\varphi) \leq \mathrm{Aut}(H)|_{\mathcal{P}}$ *where* $\varphi\colon \mathrm{Aut}(G) \to \mathrm{Sym}(\mathcal{P})$ *is the natural action of* $\mathrm{Aut}(G)$ *on* $\mathcal{P}$.

The basic idea for the proof is similar to Theorem 5.7 but the technical details are far more involved. In particular Corollary 5.6 does not hold for neighborhoods of cliques. To circumvent this problem the basic idea is to consider an initial partition which is based on whether two vertices have the same neighbors in X. Then the single sets all define neighborhood graphs whereas on the sets considered as single elements we can define an auxiliary graph in a canonical way so that this auxiliary graph is again proper circular arc. From this point we can use similar arguments as for neighborhoods of single vertices. We omit the details here.

6 Global structure

In this section we are ready construct a canonical, clique-stable partition $\mathcal{P}$ together with some canonical 8-circle-bounded graph H, such that $\mathcal{P} \subseteq V(H)$ and $\mathcal{P}$ is $\mathrm{Aut}(H)$-invariant. This method is the central part of our algorithm and gives us a good supergroup of the natural action of the automorphism group on the computed partition. The computed supergroup is then given to the subroutine, that computes setwise stabilizers for groups in Γ_8, to obtain the automorphism group of $G[\mathcal{P}]$.

▶ **Theorem 6.1.** *Let* G *be a connected unit square graph with singleton vertex. Then one can compute in polynomial time a canonical, clique-stable partition* $\mathcal{P}$ *and a canonical colored graph* H*, such that*
1. $\mathcal{P} \subseteq V(H)$ *and* $\mathcal{P}$ *is invariant under* $\mathrm{Aut}(H)$,
2. H *is* 8*-circle-bounded,*
3. $\mathrm{im}(\varphi) \leq \mathrm{Aut}(H)|_{\mathcal{P}}$ *where* $\varphi\colon \mathrm{Aut}(G) \to \mathrm{Sym}(\mathcal{P})$ *is the natural action of* $\mathrm{Aut}(G)$ *on* $\mathcal{P}$.

The basic idea for the algorithm is to proceed in two steps. First, we compute a clique-partition $\mathcal{P}$, which is only canonical but not necessarily clique-stable, together with a corresponding graph H. For this part of the algorithm we make use of the partitioning algorithm for neighborhoods of cliques. More precisely we first group the vertices according to their distance to the singleton vertex v_0 and then we iteratively consider vertices with larger and larger distances to v_0. In the first iteration we only consider the neighbors of v_0 and compute a clique-partition and a canonical graph using Theorem 5.7. In the i-th iteration we partition the vertices with distance i to v_0 based on the partition of the vertices in the previous level. For each clique in the partition of the previous level we partition its neighbors in the current level using Theorem 5.8. Then we combine the computed partitions into one partition for the current level and use the computed graphs (which we obtained from Theorem 5.8) to update the graph H.

Then, in a second step, we refine the computed partition using the color refinement algorithm while simultaneously extending the graph H. The crucial idea for extending the graph H is to use additional layers which model the iterations of the color refinement algorithm.

▶ **Corollary 6.2.** *Let G be a connected unit square graph with singleton vertex. Then one can compute in polynomial time a canonical clique-partition $\mathcal{P}$, such that $\mathrm{im}(\varphi) \in \Gamma_8$ where $\varphi\colon \mathrm{Aut}(G) \to \mathrm{Sym}(\mathcal{P})$ is the natural action of $\mathrm{Aut}(G)$ on $\mathcal{P}$.*

▶ Remark. The constant $d = 8$ is tight for the previous corollary. In particular the graph G_8 with $V(G_8) = \{v_i \mid i \in [9]\} \cup \{w_i \mid i \in [8]\}$ and $E(G_8) = \{v_iv_j \mid i \neq j \in [9]\} \cup \{v_iw_i \mid i \in [8]\}$ is a unit square graph (the vertex v_9 may be a singleton vertex). A possible realization of G_8 is depicted in Figure 3.

Together with Theorem 4.6 this gives us sufficient structure to compute the natural action of the automorphism group on the computed partition. This can also be used to solve the isomorphism problem.

▶ **Theorem 6.3.** *Let G be a connected, twin-free unit square graph with a singleton vertex. Then one can compute in polynomial time a canonical, clique-stable partition $\mathcal{P}$ and a set $S \subseteq \mathrm{Sym}(\mathcal{P})$, such that $\langle S \rangle = \mathrm{im}(\varphi) \in \Gamma_8$ where $\varphi\colon \mathrm{Aut}(G) \to \mathrm{Sym}(\mathcal{P})$ is the natural action of $\mathrm{Aut}(G)$ on $\mathcal{P}$.*

Proof. Let $\mathcal{P}$ be the canonical, clique-stable partition and H the canonical, 8-circle-bounded graph obtained from Theorem 6.1. Then $\mathrm{Aut}(H)$ can be computed in polynomial time and $\mathrm{Aut}(H) \in \Gamma_8$ by Theorem 2.5 and 2.4. Further $\mathcal{P}$ is invariant under $\mathrm{Aut}(H)$. Since H is canonical this implies $\mathrm{im}(\varphi) \leq \mathrm{Aut}(H)|_{\mathcal{P}} \in \Gamma_8$. Furthermore $\mathrm{im}(\varphi) = \mathrm{Aut}(G[\mathcal{P}])$ by Theorem 4.6. A generating set for $\mathrm{Aut}(G[\mathcal{P}])$ can be computed in polynomial time using Theorem 2.2. ◀

▶ **Theorem 6.4.** *The Graph Isomorphism Problem for unit square graphs can be solved in polynomial time.*

Proof. Let G_1, G_2 be two unit square graphs. First, it can be assumed that G_1 and G_2 are connected by considering the connected components separately. Furthermore, the graphs can be assumed to be twin-free using modular decompositions of graphs (cf. [26]). Let $c \in \mathbb{N}$ be a fresh color (i.e. a color which does not appear in G_1 or G_2). For a graph G and a vertex $v \in V(G)$ we denote by $G^{v \mapsto c}$ the graph where vertex v is colored by c. Pick $v_1 \in V(G_1)$. For each $v_2 \in V(G_2)$ test whether $G_1^{v_1 \mapsto c} \cong G_2^{v_2 \mapsto c}$ by the following procedure. For $i \in [2]$ let $\mathcal{P}_i$ be the partition and H_i be the graph computed by Theorem 6.1 for the graph $G_i^{v_i \mapsto c}$. Let H be the disjoint union of H_1 and H_2. Note that $H_1 \cong H_2$ if $G_1^{v_1 \mapsto c} \cong G_2^{v_2 \mapsto c}$ because the graph H_i is canonical. Compute a generating set for $\mathrm{Aut}(H) \in \Gamma_8$. This can be done in polynomial time according to Theorem 2.5. Let G be the disjoint union of $G_1^{v_1 \mapsto c}[\mathcal{P}_1]$ and $G_2^{v_2 \mapsto c}[\mathcal{P}_2]$. Then $\mathrm{Aut}(G) \leq \mathrm{Aut}(H)|_{\mathcal{P}_1 \cup \mathcal{P}_2}$ and hence a generating set for $\mathrm{Aut}(G)$ can be computed in polynomial time using Theorem 2.2 (note that $\mathrm{Aut}(G)$ is the set of permutations which stabilize the edge set). By Theorem 4.6 it holds that $G_1^{v_1 \mapsto c} \cong G_2^{v_2 \mapsto c}$ if and only if there is an automorphism $\gamma \in \mathrm{Aut}(G)$ that maps $G_1^{v_1 \mapsto c}[\mathcal{P}_1]$ to $G_2^{v_2 \mapsto c}[\mathcal{P}_2]$. Since G is the disjoint union of of $G_1^{v_1 \mapsto c}[\mathcal{P}_1]$ and $G_2^{v_2 \mapsto c}[\mathcal{P}_2]$ and both of these graphs are connected it holds that if such an automorphism exists then there will also be one present in the generating set of $\mathrm{Aut}(G)$. Thus it can be checked in polynomial time whether $G_1^{v_1 \mapsto c} \cong G_2^{v_2 \mapsto c}$. ◀

▶ Remark. The running time of the presented algorithm is dominated by the running time for the subroutine computing setwise stabilizers for groups in Γ_8, which in turn depends on the maximal size of primitive Γ_8-groups.

The latter was analyzed by Babai, Cameron and Pálfy in [3] and proven to be polynomially bounded in the size of the permutation domain. For a complexity analysis of the setwise stabilizer subroutine we refer to [22, 23, 5]. Note that the setwise stabilizer subroutine is also used for computing the automorphism group of H and the graph H might be much larger than the original graph G.

▶ Remark. The presented algorithm also gives us some insight about the structure of the automorphism group of a unit square graph with singleton vertex. There is an invariant clique-partition, such that the natural action on the partition forms a Γ_8-group.

An interesting question is whether a similar statement still holds if the given graph does not have a singleton vertex. We leave this question open.

7 Discussion

We presented a polynomial time algorithm solving the Graph Isomorphism Problem for unit square graphs. Overall the presented algorithm heavily depends on group theoretic methods. This raises the question whether the problem can also be solved without the use of such methods. In fact, it might be that the k-dimensional Weisfeiler-Leman algorithm can identify every unit square graph for sufficiently large k. This is left as an open question.

Furthermore it is an interesting question whether the methods presented in this work can be adapted to other geometric classes for which the isomorphism problem is still open. At first glance a natural candidate seems to be the class of unit disk graphs. However, it turns out that there are some crucial structural differences to unit square graphs. In particular, there are unit disk graphs with singleton vertex, such that for each canonical clique-partition the natural action of the automorphism group contains a large symmetric group.

Finally we would like to address two natural generalizations of unit square graphs. The first one concerns the dimension of the realization, that is, what is the complexity of graph isomorphism for graphs with d-dimensional L_∞-realization for any constant number d. The second extension concerns squares of arbitrary size. This is still a natural restriction for the class of intersection graphs of rectangles, which is GI-complete, and the reduction does not directly extend to square graphs because it requires large complete bipartite graphs as induced subgraphs (cf. [29]). However, developing an efficient algorithm for this class of graphs would require some new ideas since the number of independent neighbors of a vertex is unbounded.

Acknowledgements. I want to thank Martin Grohe and Pascal Schweitzer for several helpful discussions and comments throughout this work. I also thank the anonymous referees for many comments improving the presentation of the results.

References

1 László Babai. On the automorphism groups of strongly regular graphs I. In Moni Naor, editor, *Innovations in Theoretical Computer Science, ITCS'14, Princeton, NJ, USA, January 12-14, 2014*, pages 359–368. ACM, 2014. URL: `http://dl.acm.org/citation.cfm?id=2554797`, `doi:10.1145/2554797.2554830`.

2 László Babai. Graph isomorphism in quasipolynomial time. *CoRR*, abs/1512.03547, 2015. URL: `http://arxiv.org/abs/1512.03547`.

3 László Babai, Peter J. Cameron, and Péter P. Pálfy. On the orders of primitive groups with restricted nonabelian composition factors. *Journal of Algebra*, 79(1):161–168, 1982. `doi:10.1016/0021-8693(82)90323-4`.

4 László Babai, Xi Chen, Xiaorui Sun, Shang-Hua Teng, and John Wilmes. Faster canonical forms for strongly regular graphs. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 157–166. IEEE Computer Society, 2013. `doi:10.1109/FOCS.2013.25`.

5 László Babai, William M. Kantor, and Eugene M. Luks. Computational complexity and the classification of finite simple groups. In *24th Annual Symposium on Foundations of Computer Science, Tucson, Arizona, USA, 7-9 November 1983*, pages 162–171. IEEE Computer Society, 1983. `doi:10.1109/SFCS.1983.10`.

6 Christoph Berkholz, Paul S. Bonsma, and Martin Grohe. Tight lower and upper bounds for the complexity of canonical colour refinement. In Hans L. Bodlaender and Giuseppe F. Italiano, editors, *Algorithms – ESA 2013 – 21st Annual European Symposium, Sophia Antipolis, France, September 2-4, 2013. Proceedings*, volume 8125 of *Lecture Notes in Computer Science*, pages 145–156. Springer, 2013. `doi:10.1007/978-3-642-40450-4_13`.

7 Heinz Breu. *Algorithmic Aspects of Constrained Unit Disk Graphs*. PhD thesis, University of British Columbia, Vancouver, Canada, 1996.

8 Heinz Breu and David G. Kirkpatrick. Unit disk graph recognition is np-hard. *Comput. Geom.*, 9(1-2):3–24, 1998. `doi:10.1016/S0925-7721(97)00014-X`.

9 Jin-yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identifications. *Combinatorica*, 12(4):389–410, 1992. `doi:10.1007/BF01305232`.

10 Brent N. Clark, Charles J. Colbourn, and David S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86(1-3):165–177, 1990. `doi:10.1016/0012-365X(90)90358-O`.

11 Charles J. Colbourn and Kellogg S. Booth. Linear time automorphism algorithms for trees, interval graphs, and planar graphs. *SIAM J. Comput.*, 10(1):203–225, 1981. `doi:10.1137/0210015`.

12 Andrew R. Curtis, Min Chih Lin, Ross M. McConnell, Yahav Nussbaum, Francisco J. Soulignac, Jeremy P. Spinrad, and Jayme Luiz Szwarcfiter. Isomorphism of graph classes related to the circular-ones property. *Discrete Mathematics & Theoretical Computer Science*, 15(1):157–182, 2013. URL: `http://www.dmtcs.org/dmtcs-ojs/index.php/dmtcs/article/view/2298`.

13 Fedor V. Fomin, Daniel Lokshtanov, and Saket Saurabh. Bidimensionality and geometric graphs. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 1563–1575. SIAM, 2012. `doi:10.1137/1.9781611973099`.

14 Delbert R. Fulkerson and Oliver A. Gross. Incidence matrices and interval graphs. *Pacific Journal of Mathematics*, 15(3):835–855, 1965.

15 Martin C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs (Annals of Discrete Mathematics, Vol 57)*. North-Holland Publishing Co., Amsterdam, The Netherlands, 2004.

16 Derek F. Holt, Bettina Eick, and Eamonn A. O'Brien. *Handbook of Computational Group Theory*. Discrete Mathematics and Its Applications. CRC Press, 2005.

17 Jing Huang. On the structure of local tournaments. *J. Comb. Theory, Ser. B*, 63(2):200–221, 1995. doi:10.1006/jctb.1995.1016.

18 Johannes Köbler, Sebastian Kuhnert, Bastian Laubner, and Oleg Verbitsky. Interval graphs: Canonical representations in logspace. *SIAM J. Comput.*, 40(5):1292–1315, 2011. doi:10.1137/10080395X.

19 Johannes Köbler, Sebastian Kuhnert, and Oleg Verbitsky. Solving the canonical representation and star system problems for proper circular-arc graphs in logspace. In Deepak D'Souza, Telikepalli Kavitha, and Jaikumar Radhakrishnan, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012, December 15-17, 2012, Hyderabad, India*, volume 18 of *LIPIcs*, pages 387–399. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2012. URL: http://www.dagstuhl.de/dagpub/978-3-939897-47-7, doi:10.4230/LIPIcs.FSTTCS.2012.387.

20 Johannes Köbler, Sebastian Kuhnert, and Oleg Verbitsky. Helly circular-arc graph isomorphism is in logspace. In Krishnendu Chatterjee and Jirí Sgall, editors, *Mathematical Foundations of Computer Science 2013 – 38th International Symposium, MFCS 2013, Klosterneuburg, Austria, August 26-30, 2013. Proceedings*, volume 8087 of *Lecture Notes in Computer Science*, pages 631–642. Springer, 2013. doi:10.1007/978-3-642-40313-2_56.

21 Bastian Laubner. Capturing polynomial time on interval graphs. In *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science, LICS 2010, 11-14 July 2010, Edinburgh, United Kingdom*, pages 199–208. IEEE Computer Society, 2010. doi:10.1109/LICS.2010.42.

22 Eugene M. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *J. Comput. Syst. Sci.*, 25(1):42–65, 1982. doi:10.1016/0022-0000(82)90009-5.

23 Eugene M. Luks. Permutation groups and polynomial-time computation. In Larry Finkelstein and William M. Kantor, editors, *Groups And Computation, Proceedings of a DIMACS Workshop, New Brunswick, New Jersey, USA, October 7-10, 1991*, volume 11 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 139–176. DIMACS/AMS, 1991. URL: http://dimacs.rutgers.edu/Volumes/Vol11.html.

24 Brendan D. McKay. Practical Graph Isomorphism. *Congressus Numerantium*, 30:45–87, 1981.

25 Joseph J. Rotman. *An Introduction to the Theory of Groups*. Graduate Texts in Mathematics. Springer, 1995.

26 Pascal Schweitzer. Towards an isomorphism dichotomy for hereditary graph classes. In Ernst W. Mayr and Nicolas Ollinger, editors, *32nd International Symposium on Theoretical Aspects of Computer Science, STACS 2015, March 4-7, 2015, Garching, Germany*, volume 30 of *LIPIcs*, pages 689–702. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2015. URL: http://www.dagstuhl.de/dagpub/978-3-939897-78-1, doi:10.4230/LIPIcs.STACS.2015.689.

27 Ákos Seress. *Permutation Group Algorithms:*. Cambridge Tracts in Mathematics. Cambridge University Press, 2003.

28 Alan Tucker. Structure theorems for some circular-arc graphs. *Discrete Math.*, 7(1-2):167–195, 1974. doi:10.1016/S0012-365X(74)80027-0.

29 Ryuhei Uehara. Simple geometrical intersection graphs. In Shin-Ichi Nakano and Md. Saidur Rahman, editors, *WALCOM: Algorithms and Computation, Second International Workshop, WALCOM 2008, Dhaka, Bangladesh, February 7-8, 2008.*, volume 4921 of *Lecture Notes in Computer Science*, pages 25–33. Springer, 2008. doi:10.1007/978-3-540-77891-2_3.

30 Ryuhei Uehara. The graph isomorphism problem on geometric graphs. *Discrete Mathematics & Theoretical Computer Science*, 16(2):87–96, 2014. URL: http://www.dmtcs.org/dmtcs-ojs/index.php/dmtcs/article/view/2528.