Discrete Mathematics and Theoretical Computer Science DMTCS vol. **16:2**, 2014, 87–96

The graph isomorphism problem on geometric graphs

Ryuhei Uehara*

School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), Ishikawa, Japan.

received 29th Nov. 2013, revised 7th July 2014, accepted 8th July 2014.

The graph isomorphism (GI) problem asks whether two given graphs are isomorphic or not. The GI problem is quite basic and simple, however, it's time complexity is a long standing open problem. The GI problem is clearly in NP, no polynomial time algorithm is known, and the GI problem is not NP-complete unless the polynomial hierarchy collapses. In this paper, we survey the computational complexity of the problem on some graph classes that have geometric characterizations. Sometimes the GI problem becomes polynomial time solvable when we add some restrictions on some graph classes. The properties of these graph classes on the boundary indicate us the essence of difficulty of the GI problem. We also show that the GI problem is as hard as the problem on general graphs even for grid unit intersection graphs on a torus, that partially solves an open problem.

Keywords: graph isomorphism, intersection graph, graph recognition, unit grid intersection graph

1 Introduction

For any two given graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ with $|V_1| = |V_2|$, the graph isomorphism (GI) problem asks whether there exists a one-to-one mapping ϕ between two given graphs. That is, G_1 and G_2 are isomorphic if and only if there exists a bijective function $\phi : V_1 \to V_2$ such that $\{u, v\} \in E_1$ if and only if $\{\phi(u), \phi(v)\} \in E_2$. Although the GI problem is quite natural, determining its time complexity is a long standing open problem. The problem is clearly in NP, but it is not known to be NP-complete or not.

From the viewpoint of the computational complexity, it is not likely that the GI problem is NP-complete. If it is NP-complete, the polynomial hierarchy collapses to its second level (Boppana et al. (1987)). On the other hand, it is known that the GI problem is hard for the class DET (and hence NL) (Torán (2004)). Some comprehensive survey of the structural complexity of the GI problem can be found in Köbler et al. (1993). When we turn to develop an exponential time algorithm, it is known that the GI problem can be solvable in $2^{O(\sqrt{n \log n})}$ time (Zemlyachenko et al. (1985); Babai and Luks (1983)). Recently, Grohe investigates the GI problem from the viewpoint of a group theoretic approach with structural graph theory (Grohe (2012, 2013)): for example, the Weisfeiler-Lehman algorithm (a simple combinatorial algorithm)

*Email: uehara@jaist.ac.jp. This work was supported by JSPS KAKENHI Grant Number 23500013, 24106001

solves the GI problem for graph classes with excluded topological subgraphs in polynomial time (Grohe and Marx (2012)).

We here remark that the GI problem has a hereditary property. More precisely, for any graph classes $\mathcal{C}_1$ and $\mathcal{C}_2$ with $\mathcal{C}_1 \subset \mathcal{C}_2$, if the GI problem is polynomial time solvable for the class $\mathcal{C}_2$, so is for $\mathcal{C}_1$. On the other hand, if it is as hard as on general graphs for the class $\mathcal{C}_1$, so is for $\mathcal{C}_2$. (We note that this property does not hold on the recognition problem for these graph classes. This fact will be discussed in the concluding remarks.) The hereditary property leads us to the notion of GI-completeness: The GI problem on the class $\mathcal{C}$ is said to be *GI-complete* if it is as hard as on general graphs under polynomial time reduction (see, e.g., Uehara et al. (2004)). Last few decades, many graph classes have been proposed and investigated (Brandstädt et al. (1999); McKee and McMorris (1999); Spinrad (2003); Golumbic (2004)). Typical examples are interval graphs, that are intersection graphs of intervals, and chordal graphs, that are intersection graphs of subtrees of a tree. The GI problem can be solved in linear time for interval graphs, while the problem for chordal graphs is GI-complete. If we can clarify the gap between these graph classes, it indicates the essential difficulty of the GI problem.

In 1970s, some efficient algorithms were developed for the GI problem on basic graph classes, which include planar graphs (Hopcroft and Tarjan (1974)), interval graphs (Booth and Lueker (1976)). (We note that Myrvold and Kocay pointed out that some early papers for planar graphs share common errors Myrvold and Kocay (2011).) Since the Reingold's log-space algorithm for undirected connectivity (Reingold (2008)), some log-space algorithms for planar graphs (Datta et al. (2009)) and interval graphs (Köbler et al. (2011)) are also developed. For circular arc graphs, once Hsu reported $O(nm)$ time algorithm in 1995 (Hsu (1995)), however, a counterexample for the algorithm is found recently (Curtis et al. (2012)), and the time complexity of the GI problem for the circular arc graphs becomes open again.

In this paper, we focus on some graph classes that have geometric representations. That is, these graph classes consist of intersection graphs of geometric objects (e.g., intervals, trees, orthogonal rays). We introduce some graph classes including relatively new ones, and the current status of the GI problem for these graph classes. To demonstrate some basic techniques, we prove that the GI problem is GI complete for unit grid intersection graphs on a torus, which partially solves an open problem.

2 Preiminalies

The *neighborhood* of a vertex v in a graph $G = (V, E)$ is the set $N_G(v) = \{u \in V \mid \{u, v\} \in E\}$, and the *degree* of a vertex v is $|N_G(v)|$ denoted by $d_G(v)$. If no confusion can arise we will omit the index G. For a subset U of V, the subgraph of G induced by U is denoted by $G[U]$. Given a graph $G = (V, E)$, its *complement* $\bar{G} = (V, \bar{E})$ is defined by $\bar{E} = \{\{u, v\} \mid \{u, v\} \notin E\}$. A vertex set I is an *independent set* if and only if $G[I]$ contains no edge, and a vertex set C is a *clique* if and only if $G[C]$ contains all possible edges.

For a graph $G = (V, E)$, a sequence of distinct vertices $v_0, v_1, \ldots, v_l$ is a *path*, denoted by $(v_0, v_1, \ldots, v_l)$, if $\{v_j, v_{j+1}\} \in E$ for each $0 \le j < l$. The *length* of a path is the number of edges on the path. For two vertices u and v, the *distance* of the vertices, denoted by $dist(u, v)$, is the minimum length of the paths joining u and v. A *cycle* consists of a path $(v_0, v_1, \ldots, v_l)$ of length at least 2 with an edge $\{v_0, v_l\}$, and denoted by $(v_0, v_1, \ldots, v_l, v_0)$. The *length* of a cycle is the number of edges on the cycle (equal to the number of vertices).

Two graphs $G = (V, E)$ and $G' = (V', E')$ are *isomorphic* if and only if there is a one-to-one mapping $\phi : V \to V'$ such that $\{u, v\} \in E$ if and only if $\{\phi(u), \phi(v)\} \in E'$ for every pair of vertices $u, v \in V$. We denote by $G \sim G'$ if G and G' are isomorphic. The *graph isomorphism (GI) problem* is to determine if $G \sim G'$ for given graphs G and G'. A graph class $\mathcal{C}$ is said to be *GI-complete* if there is a polynomial time reduction from the GI problem for general graphs to the GI problem for $\mathcal{C}$. Intuitively, the GI problem for the class $\mathcal{C}$ is as hard as the problem for general graphs if $\mathcal{C}$ is GI-complete.

An edge that joins two vertices of a cycle but is not itself an edge of the cycle is a *chord* of that cycle. A graph is *chordal* if every cycle of length at least 4 has a chord. A graph $G = (V, E)$ is *bipartite* if and only if V can be partitioned into two sets X and Y such that every edge joins a vertex in X and the other vertex in Y. A bipartite graph is *chordal bipartite* if every cycle of length at least 6 has a chord.

3 Geometric graph classes and their relationship

In this paper, we will discuss about intersection graphs of geometrical objects. Representatively, interval graphs are characterized by intersection graphs of intervals, and it is well known that chordal graphs are intersection graphs of subtrees of a tree (see, e.g., Spinrad (2003)).

The GI problem is GI-complete for several graph classes including chordal bipartite graphs and strongly chordal graphs (Uehara et al. (2004)). On the other hand, the GI problem can be solved efficiently for many graph classes; for example, interval graphs (Booth and Lueker (1976)), permutation graphs (Colbourn (1981)), directed path graphs (Babel et al. (1996)), and distance hereditary graphs (i. Nakano et al. (2009)).

A bipartite graph $G = (X, Y, E)$ is a *grid intersection graph* if every vertex $x \in X$ and $y \in Y$ can be assigned to line segments I_x and J_y in the plane, parallel to the horizontal and vertical axis respectively, so that for all $x \in X$ and $y \in Y$, $\{x, y\} \in E$ if and only if I_x and J_y cross each other. We call $(\mathcal{I}, \mathcal{J})$ a *grid representation* of G, where $\mathcal{I} = \{I_x \mid x \in X\}$ and $\mathcal{J} = \{J_y \mid y \in Y\}$. A grid representation is *unit* if all line segments in the representation have the same (unit) length. A bipartite graph is a *unit grid intersection graph* if it has a unit grid representation. Otachi, Okamoto, and Yamazaki show some relationship between (unit) grid intersection graphs and other graph classes (Otachi et al. (2007)); one of them is that interval bigraphs are included in the intersection of unit grid intersection graphs and chordal bipartite graphs.

In a grid intersection graph, if each line segment can be replaced by a *ray*, or half-infinite line, we obtain an *orthogonal ray graph*. The notion of orthogonal ray graphs is recently introduced by Ueno (Shrestha et al. (2010)), motivated by an application of VLSI design. He also proposed the notions of $2D$ and $3D$ orthogonal ray graphs which are the restricted orthogonal ray graphs with respect to the directions of rays ($2D$ allows two directions, and $3D$ allows three directions among four possible directions). Based on their characterization of 2D orthogonal ray graphs, the GI problem for 2D orthogonal ray graphs can be solved in polynomial time. We here note that in (Shrestha et al., 2010, Corollary 14), they use the result for circular arc graphs by Hsu (Hsu (1995)), which contains a bug as mentioned in Introduction. However, their characterization in (Shrestha et al., 2010, Theorem 12) can avoid the bug. More precisely, the characterization in Shrestha et al. (2010) shows that 2D orthogonal ray graphs are characterized by circular arc graphs with two cliques (that is, each vertex belongs to one of these two cliques), and Eschen proposed a polynomial time algorithm for the GI problem in this case (see Curtis et al. (2012) further details). Moreover, recently, Chaplick et. al also give alternative characterizations supporting that the GI problem can be solved in polynomial time for 2D orthogonal ray graphs (Chaplick et al. (2013)).

We summary the situation in Fig. 1.

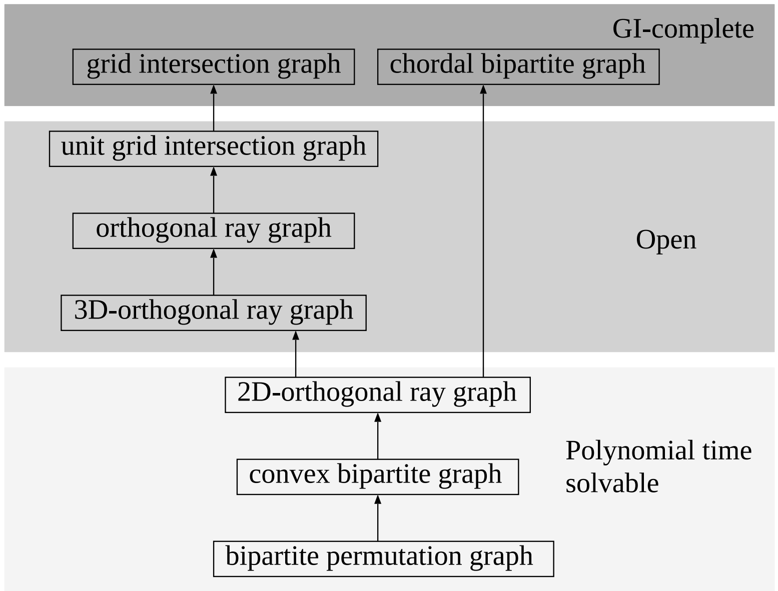

Fig. 1: Hierarchy of the graph classes and computational complexity of the GI problem.

4 GI completeness of unit grid intersection graph on a torus

The main theorem in this section is the following:

Theorem 1 *The GI problem is GI-complete even for the class of connected unit grid intersection graphs on a torus.*

The proof is done by a reduction from the GI problem for general connected graphs to the GI problem for connected unit grid intersection graphs on a torus. Similar idea can be found in Babel et al. (1996); Uehara et al. (2004); Uehara (2008, 2013).

Proof: We first start the GI problem for general connected graph $G_0 = (V_0, E_0)$ with $|V_0| = n$ and $|E_0| = m$. (we will refer the graph G_0 in Fig. 2(1) as an example). From G_0, we first add extra vertices V_1 with associated edges E_1 as follows: (1) For each vertex v of degree 1 in V_0, we add v' and v'' into V_1 and $\{v, v'\}$ and $\{v, v''\}$ into E_1. (2) For each vertex v of degree 2 in V_0, we add v' into V_1 and $\{v, v'\}$ into E_1. Let $G_1 = (V_0 \cup V_1, E_0 \cup E_1)$ be the resulting graph, $n' = |V_0 \cup V_1|$, and $m' = |E_0 \cup E_1|$. It is easy to see that $n' \leq 3n$, $m'm + 2n$. For the resulting graph $G_1 = (V_0 \cup V_1, E_0 \cup E_1)$, we define a bipartite graph $G_2 = (V_0 \cup V_1, E_0 \cup E_1, E_2)$ by $E_2 := \{\{v, e\} \mid v \text{ is one endpoint of } e\}$. (Intuitively, each edge in G_1 is divided into two edges joined by a new vertex; see Fig. 2(2)(3)). Then, $e \in E_0$ have degree 2 in G_2 by its two endpoints in $V_0 \cup V_1$. Moreover, by the definition of V_1, we can observe that v is in V_1 if and only if v has degree 1 in G_2, and v is in V_0 if and only if v has degree at least 3 in G_2. It is easy to see that $G_0 \sim G_0'$ if and only if $G_2 \sim G_2'$, for any graphs G_0 and G_0' with resulting graphs G_2 and G_2', respectively.

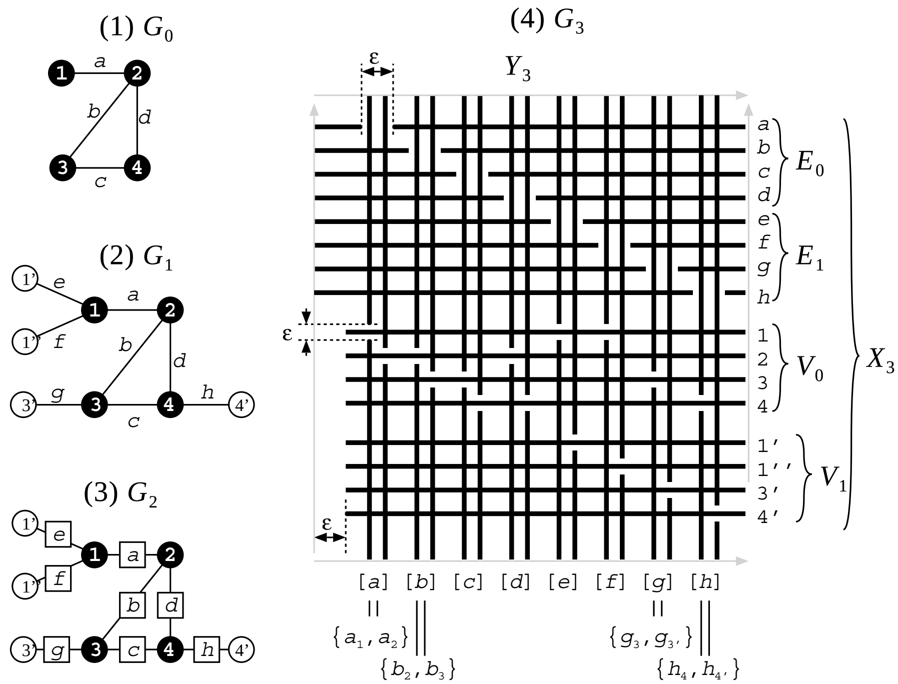

Fig. 2: Reductions from G_0 to G_1, from G_1 to G_2, and from G_2 to G_3.

Now, we construct a new bipartite graph $G_3 = (X_3, Y_3, E_3)$ that is a unit grid intersection graph on a torus from the bipartite graph $G_2 = (V_0 \cup V_1, E_0 \cup E_1, E_2)$ such that $G_2 \sim G_2'$ if and only if $G_3 \sim G_3'$ in the same manner. (For the G_2 in Fig. 2(3), the resulting graph is shown in Fig. 2(4). We illustrate G_3 in the figure by unit grid intersection model since its corresponding intersection graph has too many edges. Intuitively, each connection in G_2 is represented by a non-crossing in G_3.)

The vertex set X_3 is defined by $V_0 \cup V_1 \cup \cup E_0 \cup E_1$. The other vertex set Y_3 consists of $|E_2|$ pairs of vertices. For each edge $e = \{u, v\} \in E_2$, Y_3 contains a pair of e_u and e_v. Precisely, Y_3 is defined by the set $\{e_u | u \in e \in E_2\}$. To make the idea clearer, we consider Y_2 as the set consists of the pairs $[e] = \{e_u, e_v\}$ for each edge in E_2.

Now we define the edge set E_3 of G_3. First we mention that each of $X_3 = V_0 \cup V_1 \cup E_0 \cup E_1$ and Y_3 is independent set in G_3 since G_3 is bipartite. For each pair $[e] = \{e_u, e_v\}$ in Y_3, e_u is joined to all vertices in $(V_0 \cup V_1) \subset X_3$ but $u \in (V_0 \cup V_1)$ itself, and e_u is also joined to all vertices in $(E_0 \cup E_1) \subset X_3$ but $e \in (E_0 \cup E_1)$. The other vertex e_v is joined to almost all vertices in the same manner. This completes the construction of G_3. It is easy to see that the resulting graph has $n' + m' + 2m' + 1 = n' + 3m' + 1$ vertices, hence it is a polynomial time reduction. Now we show that (1) $G_0 \sim G_0'$ if and only if $G_3 \sim G_3'$, and (2) G_3 is a unit grid intersection graph on a torus.

(1) We show that when G_3 is given, we can reconstruct G_0 from G_3 up to isomorphism. Since G_3 is connected and bipartite, two vertex sets X_3 and Y_3 are easy to find. Now, all vertices in Y_3 has the same

degree $|X_3| - 2$. On the other hand, by the construction of G_1 and G_3, each vertex in $E_0 \cup E_1$ has degree $|Y_3| - 2$, each vertex in V_0 has degree at most $|Y_3| - 3$, and each vertex in V_1 has degree $|Y_3| - 1$. Moreover, since G_0 is not trivial, X_3 contains at least one vertex from V_0, which has degree at most $|Y_3| - 3$. Therefore, checking the degrees, we can distinguish Y_3 from X_3, and we can extract V_0 and V_1 from X_3. This is enough information to reconstruct G_0 from G_3. We first determine the vertices in V_1 by their degree equal to 1, and remove with associate edges in E_1. Then the remaining graph contains only V_0 and E_0.
(2) Now we show that G_3 is a unit grid intersection graph on a torus (see Fig. 2(4)). We first arrange the vertices in Y_3 as the set of vertical lines. For each pair $[e] = \{e_u, e_v\}$, the corresponding lines are arranged to be adjacent. The ordering among the pairs is arbitrary. Next we arrange the vertices in $X_3 = V_0 \cup V_1 \cup E_0 \cup E_1$ as horizontal lines. We here split V_0, V_1, E_0, and E_1 on the representation, but their ordering is arbitrary. Now we determine the positions of two endpoints for each line segments. For each $[e] = \{e_u, e_v\}$, two endpoints of the vertical line corresponding to e_u are arranged to avoid crossing the vertex u itself in $(V_0 \cup V_1) \subset X_3$. The other line corresponding to e_v is arranged in the same manner. It is easy to see that it is possible for all vertices in Y_3 (as shown in bottom half area of Fig. 2(4)). For each vertex $e \in (E_0 \cup E_1) \subset X_3$, two endpoints of the horizontal line corresponding to e are arranged to avoid crossing the pair $[e]$ itself in Y_3. It is possible for all vertices in $(E_0 \cup E_1) \subset X_3$ since two corresponding line segments are adjacent (as shown in top half area of Fig. 2(4)). We also arrange two endpoints of the horizontal lines representing the vertices in $(V_0 \cup V_1) \subset X_3$ in a trivial way (as shown in bottom left area of Fig. 2(4)).

We can put this arrangement onto a torus (by gluing the corresponding gray arrows in Fig. 2(4)). Changing its scale, all gaps can be the same length $\epsilon > 0$. Thus all line segments are of unit length.

Hence the GI problem for unit grid intersection graphs on a torus is as hard as the GI problem for general graphs. Thus the GI problem is GI-complete for unit grid intersection graphs on a torus. □

5 Concluding Remarks

In this paper, we show a hierarchy of graph classes with respect to the computational complexity of the GI problem. We give a partial answer to the unit grid intersection graphs: it is GI-complete if they are on a torus. The computational complexity of the GI problem for the following classes are still open: unit grid intersection graphs on a plane, orthogonal ray graphs, and 3D orthogonal ray graphs.

One of basic problems for graph classes is the recognition problem. That is, for any given graph G and some graph class $\mathcal{C}$, the recognition problem asks if G is a member of $\mathcal{C}$ or not. The recognition problem is not hereditary, but we need useful structures to recognize a class, and they are also helpful for solving the GI problem. From this viewpoint, it is worth listing the computational complexity of the recognition problem for these graph classes. The class of bipartite permutation graphs is well-investigated, and it can be recognized in polynomial time (see, e.g., Spinrad et al. (1987); Kratsch et al. (2003)). It is also known that convex bipartite graphs can be recognized in linear time (see Brandstädt et al. (1999)). The class of 2D orthogonal ray graphs is also recognized in polynomial time (Shrestha et al. (2010); Chaplick et al. (2013)). The class of chordal bipartite graph can be recognized in $O(\min\{m \log n, n^2\})$ time (Hoffman et al. (1985); Lubiw (1987); Paige and Tarjan (1987); Spinrad (1993)). On the other hand, the recognition problem of grid intersection graphs is NP-complete (Kratochvil (1994)), and recently, the NP-completeness for unit grid intersection graphs is also shown by Mustaţă and Pergel (2013). The recognition problems for the classes of orthogonal ray graphs and 3D orthogonal ray graphs are open.

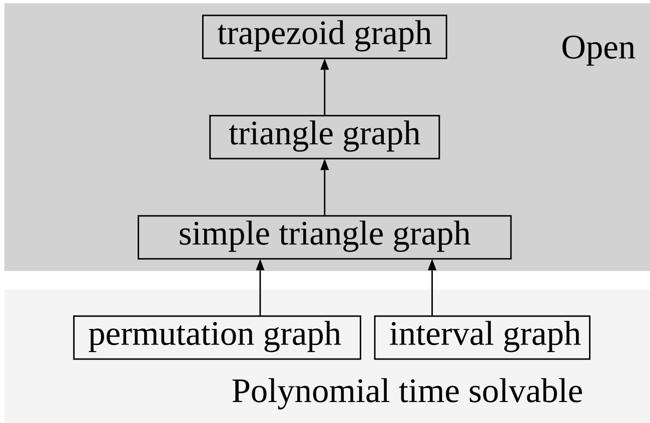

Fig. 3: Another hierarchy of the graph classes.

When we discuss computational complexity of the GI problem for a graph class, we have to take care of computational complexity of the recognition problem, especially, when it is NP complete. Now suppose that the recognition problem is NP complete for some class $\mathcal{C}$. Then we have to suppose that given graph is in $\mathcal{C}$ a priori without geometric representation. In the case, even if we know a graph G is in a class $\mathcal{C}$, it is not clear if we can construct its geometric representation efficiently, and if we can use some property of the class provided by the representation. Therefore, we may not be able to use its geometric property to solve the GI problem according to the assumption of the input of the problem (it can be natural to assume that graphs are given in some explicit representations).

Another interesting hierarchy of graph classes is shown in Fig. 3. Both of permutation graphs and interval graphs are recognizable in linear time; precisely, their geometric representations can be constructed in linear time (see Kratsch et al. (2003) for recent results with references). The classes of simple triangle graphs, triangle graphs, and trapezoid graphs are natural generalization of these graph classes. However, computational complexity of the GI problem are still open for these classes. Surprisingly, though the recognition problems for the classes of trapezoid graphs and simple triangle graphs are polynomial time solvable (see Mertzios and Corneil (2009)), the recognition problem for the class of triangle graphs is NP-complete (Mertzios (2012)).

For permutation graphs and interval graphs, using the canonical tree structures based on their geometric representations, the GI problem can be solved in linear time. However, the computational complexity of the GI problem of the other three superclasses are still open.

References

L. Babai and E. Luks. Canonical Labeling of Graphs. In *Proc. 15th ACM Symp. on the Theory of Computing*, pages 171–183. ACM, 1983.

L. Babel, I. Ponomarenko, and G. Tinhofer. The Isomorphism Problem For Directed Path Graphs and For Rooted Directed Path Graphs. *Journal of Algorithms*, 21:542–564, 1996.

K. Booth and G. Lueker. Testing for the Consecutive Ones Property, Interval Graphs, and Graph Planarity Using PQ-Tree Algorithms. *Journal of Computer and System Sciences*, 13:335–379, 1976.

R. Boppana, J. Håstad, and S. Zachos. Does co-NP Have Short Interactive Proofs? *Information Processing Letters*, 25:127–132, 1987.

A. Brandstädt, V. Le, and J. Spinrad. *Graph Classes: A Survey*. SIAM, 1999.

S. Chaplick, P. Hell, Y. Otachi, T. Saitoh, and R. Uehara. Intersection Dimension of Bipartite Graphs. in preparation, 2013.

C. Colbourn. On Testing Isomorphism of Permutation Graphs. *Networks*, 11:13–21, 1981.

A. R. Curtis, M. C. Lin, R. M. McConnell, Y. Nussbaum, F. J. Soulignac, J. P. Spinrad, and J. L. Szwarcfiter. Isomorphism of graph classes related to the circular-ones property. Technical Report 1203.4822v1, arXiv, March 2012.

S. Datta, N. Limaye, P. Nimbhorkar, T. Thierauf, and F. Wagner. Planar Graph Isomorphism is in Log-Space. arXiv:0809.2319v2, Jan. 2009.

M. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Annals of Discrete Mathematics 57. Elsevier, 2nd edition, 2004.

M. Grohe. Structural and logical approaches to the graph isomorphism problem. In *Proc. 23rd Ann. ACM-SIAM Symp. on Discrete Algorithms*, page 188. ACM, 2012.

M. Grohe. Logical and Structural Approaches to the Graph Isomorphism Problem. In *Proc. 38th International Symposium on Mathematical Foundations of Computer Science*, page 42. Lecture Notes in Computer Science Vol. 8087, Springer-Verlag, 2013.

M. Grohe and D. Marx. Structure theorem and isomorphism test for graphs with excluded topological subgraphs. In *Proc. 44th ACM Symp. on the Theory of Computing*, pages 173–192. ACM, 2012.

A. Hoffman, A. Kolen, and M. Sakarovitch. Totally-Balanced and Greedy Matrices. *SIAM J. Alg. Disc. Meth.*, 6(4):721–730, 1985.

J. Hopcroft and R. Tarjan. Efficient planarity testing. *Journal of the ACM*, 21:449–568, 1974.

W.-L. Hsu. $O(M \cdot N)$ Algorithms for the Recognition and Isomorphism Problem on Circular-Arc Graphs. *SIAM Journal on Computing*, 24(3):411–439, 1995.

S. i. Nakano, R. Uehara, and T. Uno. A new approach to graph recognition and applications to distance-hereditary graphs. *Journal of Computer Science and Technology*, 24(3):517–533, May 2009.

J. Köbler, U. Schöning, and J. Torán. *The Graph Isomorphism Problem: Its Structural Complexity*. Birkhäuser, 1993.

J. Köbler, S. Kuhnert, B. Laubner, and O. Verbitsky. Interval Graphs: Canonical Representations in Logspace. *SIAM Journal on Computing*, 40(5):1292–1315, 2011.

J. Kratochvil. A special planar satisfiability problem and a consequence of its NP-completeness. *Discrete Applied Mathematics*, 52:233–252, 1994.

D. Kratsch, R. McConnell, K. Mehlhorn, and J. Spinrad. Certifying Algorithms for Recognizing Interval Graphs and Permutation Graphs. In *Proc. 14th Ann. ACM-SIAM Symp. on Discrete Algorithms*, pages 158–167. ACM, 2003.

A. Lubiw. Doubly Lexical Orderings of Matrices. *SIAM Journal on Computing*, 16(5):854–879, 1987.

T. McKee and F. McMorris. *Topics in Intersection Graph Theory*. SIAM, 1999.

G. B. Mertzios. The recognition of triangle graphs. *Theoretical Computer Science*, 438:34–47, 2012.

G. B. Mertzios. The Recognition of Simple-Triangle Graphs and of Linear-Interval Orders Is Polynomial. In *ESA 2013*, pages 719–730. Lecture Notes in Computer Science Vol. 8125, Springer-Verlag, 2013.

G. B. Mertzios and D. G. Corneil. Vertex splitting and the recognition of trapezoid graphs. Technical Report AIB-2009-16, RWTH Aachen University, 2009.

I. Mustaţă and M. Pergel. Unit Grid Intersection Graphs: Recognition and Properties. Technical report, arXiv:1306.1855, 2013.

W. Myrvold and W. Kocay. Errors in graph embedding algorithms. *Journal of Computer and System Sciences*, 7(2):430–438, 2011.

Y. Otachi, Y. Okamoto, and K. Yamazaki. Relationships between the class of unit grid intersection graphs and other classes of bipartite graphs. *Discrete Applied Mathematics*, 155:2383–2390, 2007.

R. Paige and R. Tarjan. Three Partition Refinement Algorithms. *SIAM Journal on Computing*, 16(6): 973–989, 1987.

O. Reingold. Undirected ST-connectivity in log-space. In *Proc. 37th ACM Symp. on the Theory of Computing*, pages 376–385. ACM, 2005.

O. Reingold. Undirected connectivity in log-space. *Journal of the ACM*, 55(4):17:1–17:24, 2008.

A. M. S. Shrestha, S. Tayu, and S. Ueno. On orthogonal ray graphs. *Discrete Applied Mathematics*, 158: 1650–1659, 2010.

J. Spinrad. Doubly Lexical Ordering of Dense 0-1 matrices. *Information Processing Letters*, 45:229–235, 1993.

J. Spinrad. *Efficient Graph Representations*. American Mathematical Society, 2003.

J. Spinrad, A. Brandstädt, and L. Stewart. Bipartite Permutation Graphs. *Discrete Applied Mathematics*, 18:279–292, 1987.

J. Torán. On the hardness of graph isomorphism. *SIAM Journal on Computing*, 33(5):1093–1108, 2004.

R. Uehara. Simple Geometrical Intersection Graphs. In *Second International Workshop on Algorithms and Computation (WALCOM 2008)*, pages 25–33. Lecture Notes in Computer Science Vol. 4921, Springer-Verlag, 2008.

R. Uehara. Tractabilities and Intractabilities on Geometric Intersection Graphs. *Algorithms*, 6(1):60–83, 2013. URL `doi:10.3390/a6010060`. DOI:10.3390/a6010060.

R. Uehara, S. Toda, and T. Nagoya. Graph Isomorphism Completeness for Chordal Bipartite Graphs and Strongly Chordal Graphs. *Discrete Applied Mathematics*, 145(3):479–482, 2004. doi: 10.1016/j.dam.2004.06.008.

V. Zemlyachenko, N. Korneenko, and R. Tyshkevich. Graph Isomorphism Problem. *Journal of Soviet Mathematics*, 29(4):1426–1481, 1985.