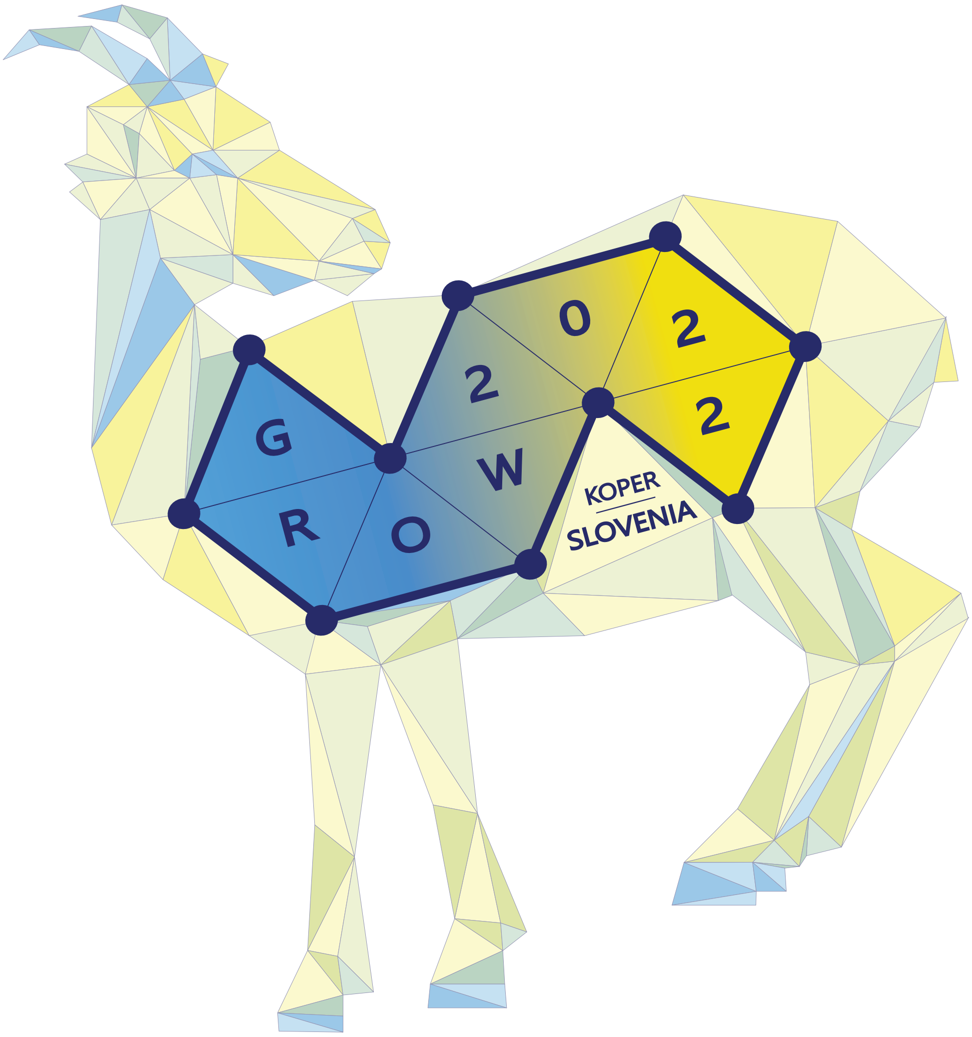
10th Workshop on Graph Classes, Optimization, and Width Parameters

GROW 2022

Book of Open Problems



19–22 September 2022
Koper, Slovenia

Kataložni zapis o publikaciji (CIP) pripravili v Narodni in univerzitetni knjižnici v Ljubljani

COBISS.SI-ID 124271875
ISBN 978-961-07-1311-1 (PDF)

10th Workshop on Graph Classes, Optimization, and Width Parameters
GROW 2022: Book of Abstracts

Editors:	Sergio Cabello, Matjaž Krnc, Martin Milanič
Web access (URL):	`https://grow.famnit.upr.si/GROW-BOP.pdf`
Frontpage image:	Katja Berčič
Logo:	Angelica Fallavier
DOI:	10.5281/zenodo.7143192
ISBN:	978-961-07-1311-1
COBISS-ID:	124271875

Koper, 19–22 September, 2022

Preface

The *Workshop on Graph Classes, Optimization, and Width Parameters* (GROW) series of workshops brings together experts in both theoretical and practical issues to design new strategies for dealing with intractable graph problems. Its 10th edition, GROW 2022, was hosted by the University of Primorska, Faculty of Mathematics, Natural Sciences and Information Technologies, Koper, Slovenia, from September 19–22, 2022. The event was originally planned for September 2021, but it was decided to postpone it for a year because of the COVID-19 pandemic. This decision allowed to reduce the risks of an outbreak that would reduce dramatically the participation and, eventually, lead to a fully in-person event, as in all previous editions.

In this edition we enjoyed three distinguished keynote speakers: Marthe Bonamy, Bart M. P. Jansen and Bojan Mohar. We also had the pleasure to have a session to celebrate the work of Vadim Lozin because of his 60th birthday.

This booklet contains descriptions of open problems presented at GROW 2022. Together with the *Book of Abstracts*, they provide a glimpse of the breadth and excellent scientific quality of the research presented during the event. Nevertheless, they do not show the enjoyment of taking part of GROW, a unique event to meet colleagues and discuss research in an enjoyable, fruitful atmosphere.

Sergio Cabello and Martin Milanič
Chairs of GROW 2022

Tradition of the Workshop on Graph Classes, Optimization, and Width Parameters:
GROW 2022 – Koper (Slovenia)
GROW 2019 – Wien (Austria)
GROW 2017 – Toronto (Canada)
GROW 2015 – Aussois (France)
GROW 2013 – Santorini Island (Greece)
GROW 2011 – Daejeon (South Korea)
GROW 2009 – Bergen (Norway)
GROW 2007 – Eugene (USA)
GROW 2005 – Prague (Czech Republic)
GROW 2001 – Barcelona (Spain)

List of authors:

Bounds for the twin-width of graphs

Jungho Ahn
KAIST & IBS DIMAG

Given a graph G, let $\operatorname{tww}(G)$ denote its twin-width (see, e.g., [1]).

Problem. *Is there an n-vertex graph G such that* $\operatorname{tww}(G) > (n-1)/2$*?*

It was shown in [2] that

$$1.088 < \frac{4\sqrt{6}}{9} \leq \limsup_{m\to\infty} \max_{|E(G)|=m} \frac{\operatorname{tww}(G)}{\sqrt{E(G)}} \leq \sqrt{3} < 1.733.$$

Problem. *What is the correct value of*

$$\limsup_{m\to\infty} \max_{|E(G)|=m} \frac{\operatorname{tww}(G)}{\sqrt{E(G)}}?$$

References

[1] Édouard Bonnet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. 2021. Twin-width I: Tractable FO Model Checking. J. ACM 69, 1, Article 3 (February 2022), 46 pages. https://doi.org/10.1145/3486655

[2] Jungho Ahn, Kevin Hendrey, Donggyu Kim, and Sang-il Oum. "Bounds for the Twin-width of Graphs." arXiv preprint arXiv:2110.03957 (2021).

Recognizing weighted cocomparability graphs

Jesse Beisegel
Brandenburg University of Technology

A *k-weighted graph* (G, ω) consists of a graph G and a surjective edge weight function $\omega : E(G) \to \{1, \ldots, k\}$. For any $1 \leq i \leq k+1$, the *i-th level graph* of (G, ω) is the spanning subgraph of G containing all edges $e \in E(G)$ with $\omega(e) \geq i$. Thus, the first level graph is G and the $(k+1)$-th level graph is the edgeless graph with $|V(G)|$ vertices.

For a given graph class $\mathcal{G}$, we call a k-weighted graph (G, ω) a *level-$\mathcal{G}$ weighted graph* if every level graph of (G, ω) is an element of the class $\mathcal{G}$. These concepts where considered by Laurent and Seminaroti [2] as well as Laurent and Tanigawa [3].

Another way to define a weighted graph class are characterizing linear vertex orders, a well known example of which are *perfect elimination orders* that characterize chordal graphs. Laurent and Tanigawa [3], define a *weighted perfect elimination ordering* as a linear ordering σ of the vertex set $V(G)$ such that for $x, y, z \in V$ with $x \prec_\sigma y \prec_\sigma z$ it holds that $w(x, y) \geq \min\{w(x, z), w(y, z)\}$. They also show that the resulting graph class is properly contained in the class of level-chordal weighted graphs and ask for a recognition algorithm.

For unit interval graphs this definition is equivalent to Robinsonian Similarities, for which several recognition algorithms are known (see Laurent and Seminaroti [2]). However, these concepts can be transferred to many other graph classes which can be characterized by linear vertex orderings. One natural example is the class of cocomparability graphs which can be characterized by so-called *umbrella-free orderings*. A *weighted umbrella-free* ordering is then defined as a linear ordering σ of the vertex set $V(G)$ such that $x, y, z \in V$ with $x \prec_\sigma y \prec_\sigma z$ it holds that $w(x, z) \leq \max(w\{xy), w(y, z)\}$. For the resulting class it is again easy to se that it is properly contained in the class of level-cocomparability weighted graphs.

As for the weighted chordal graphs, it remains open how weighted cocomparability graphs defined in this way can be recognized.

References

[1] Jesse Beisegel, Nina Chiarelli, Ekkehard Köhler, Matjaž Krnc, Martin Milanič, Nevena Pivač, Robert Scheffler, and Martin Strehler. Edge elimination and weighted graph classes. In Isolde Adler and Haiko Müller, editors, *Graph-Theoretic Concepts in Computer Science*, volume 12301 of *LNCS*, pages 134–147, Cham, 2020. Springer.

[2] Monique Laurent and Matteo Seminaroti. A Lex-BFS-based recognition algorithm for Robinsonian matrices. *Discrete Applied Mathematics*, 222:151–165, 2017.

[3] Monique Laurent and Shin-ichi Tanigawa. Perfect elimination orderings for symmetric matrices. *Optimization Letters*, 14:339–353, 2017.

Simultaneous edge colorings

Sergio Cabello
University of Ljubljana and IMFM

Let $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ be two graphs on the same vertex set. Let Δ be an upper bound on the maximum degree of G_1 and on the maximum degree of G_2.

Conjecture. *There exists an edge coloring* $c : E_1 \cup E_2 \to \{1, \ldots, \Delta + 2\}$ *that is a proper edge coloring when restricted to* E_1 *and a proper edge coloring when restricted to* E_2*.*

$\Delta + 2$ is sort of a natural candidate because when the graph $G' = (V, E_1, \cap E_2)$ is regular, then it is easy to see that it is possible: if G' is r-regular, color G' using $r + 1$ colors, and then reuse the same $\Delta + 1 - r$ colors to color $(V, E_1 \setminus E_2)$ and $(V, E_2 \setminus E_1)$ separetely.

Actually, I do not know any case where $\Delta + 2$ colors are needed. I made some non-systematic computer search around 2016 and could not find any single case where $\Delta + 1$ colors would not suffice.

Bousquet and Durain [1] have shown in a short, cute note, that one can color $E_1 \cup E_2$ with $3\Delta/2 + 4$ colors. Thus, showing, say, that $\Delta + 100$ colors suffice, would be already a great improvement.

References

[1] N. Bousquet, B. Durain: A note on the simultaneous edge coloring. Discret. Math. 343(5): 111781 (2020)

Crossing number of near-planar graphs with small degree

Sergio Cabello
University of Ljubljana and IMFM

A graph is *near-planar* if there exists an edge whose removal gives a planar graph. Together with Bojan Mohar [1] we showed that it is NP-hard to compute the crossing number of near-planar graphs. The construction to show NP-hardness uses vertices of very high degree. On the other hand, it is known that the crossing number of near-planar graphs with maximum degree 3 can be computed in polynomial time [2, 3]. What about degree 4? More generally, could this be parameterized by the maximum degree of the graph when it is near-planar.

References

[1] S. Cabello, B. Mohar: Adding One Edge to Planar Graphs Makes Crossing Number and 1-Planarity Hard. SIAM J. Comput. 42(5): 1803-1829 (2013)

[2] S. Cabello, B. Mohar: Crossing Number and Weighted Crossing Number of Near-Planar Graphs. Algorithmica 60(3): 484-504 (2011)

[3] A. Riskin: The crossing number of a cubic plane polyhedral map plus an edge. Stud. Sci. Math. Hung. 31, 405–413 (1996)

1-Planar Drawing Extension

Robert Ganian
TU Wien

1-plane drawings are generalizations of classical plane drawings in which we allow each edge to cross at most one other edge, and to do so at most once.

In the 1-Planar Vertex Deletion Drawing Extension problem, we are given a graph G, a subset X of the vertices of G, and a 1-plane drawing H of $G - X$. The problem asks whether H can be extended to a 1-plane drawing of G, i.e., if it is possible to add the vertices of X into H in a way which keeps the drawing 1-plane.

1-Planar Vertex Deletion Drawing Extension is known to be NP-complete in general, and has been shown to be XP parameterized by $|X|$ [1]. However, it is open whether 1-Planar Vertex Deletion Drawing Extension is FPT when parameterized by $|X|$, or whether it is W[1]-hard.

Remark. *In case a fixed-parameter algorithm is provided, it is likely that one can obtain fixed-parameter tractability for a more natural (and more general) variant of the problem where we allow for the deletion of edges and vertices (and parameterize by the vertex plus edge deletion distance between the drawing H and the original graph G). A similar result has been shown for the related notion of IC-planarity [2].*

References

[1] Eduard Eiben, Robert Ganian, Thekla Hamm, Fabian Klute, Martin Nöllenburg: Extending Nearly Complete 1-Planar Drawings in Polynomial Time. MFCS 2020: 31:1-31:16

[2] Eduard Eiben, Robert Ganian, Thekla Hamm, Fabian Klute, Martin Nöllenburg: Extending Partial 1-Planar Drawings. ICALP 2020: 43:1-43:19

Cyclability for cocomparability graphs

Petr Golovach
Department of Informatics, Bergen University

A graph G is said to be *k-cyclable* for a positive integer k if for every set S of vertices of size k, there is a cycle C containing every vertex of S. The Cyclability problem takes as its input a graph G and a positive integer k, and the task is to decide whether G is k-cyclable. In [1], it was shown that Cyclability can be solved in polynomial time for interval graphs and bipartite permutation graphs.

Problem. *Is it possible to extend this result on superclasses of these graph classes for which Hamiltonian Cycle is polynomial? In particular, is Cyclability polynomial-time solvable on cocomparability graphs?*

References

[1] C. Crespelle, P. A. Golovach, Cyclability in graph classes. Discret. Appl. Math. 313: 147-178 (2022)

Simplicial Bump Number Problem

Michel Habib
IRIF

Definition. *Given a chordal graph G and $\sigma = x_1, \dots, x_n$ a simplicial elimination ordering also called PEO of G, let us define the Simplicial Bump number as follows:*

$$S.\mathrm{bump}(G,\sigma) = |\{i \textit{ with } 1 \leq i \leq n-1 \textit{ such that } x_i x_{i+1} \notin E(G)\}|,$$
$$S.\mathrm{bump}(G) = \min_{\sigma} S.\mathrm{bump}(G,\sigma)$$

In his GROW 2022 presentation, R. Scheffler characterized the graphs with $S.\mathrm{bump}(G) = 0$ as semi-proper interval graphs, together with an algorithmic recognition in linear time.

Problem. *What is the complexity of the computation of S.bump for chordal graphs?*

Generalized Crowns

Bart M. P. Jansen
Eindhoven University of Technology

The GENERALIZED CROWN task is described as follows.

Input: A graph G and an integer k.

Output: If there exists an independent set I such that the set $C := N_G(I)$ is a minimum vertex cover for $G[C \cup I]$ of size at most k, then output a vertex v that belongs to a minimum vertex cover of G.

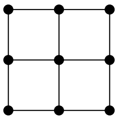
If there is no such independent set I, the algorithm should output "error", or a vertex v as above.

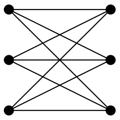
Problem. *Can* GENERALIZED CROWN *be solved in time $f(k) \cdot n^{\mathcal{O}(1)}$?*

Induced Minor Obstructions of Tree-Independence Number

Tuukka Korhonen
University of Bergen

Tree-independence number was introduced by Dallard, Milanič, and Štorgel [1]. The independence number of a tree decomposition is the maximum of the independence numbers of its bags, and the tree-independence number $\mathsf{tree}\text{-}\alpha(G)$ of a graph G is the minimum independence number of a tree decomposition of it. A graph H is an induced minor of a graph G if H can be obtained from G by vertex deletions and edge contractions. If H is an induced minor of G, then $\mathsf{tree}\text{-}\alpha(H) \leq \mathsf{tree}\text{-}\alpha(G)$.





The graphs $t \times t$-grid and the complete bipartite graph $K_{t,t}$ are induced minor obstructions of bounded tree-independence number in the sense that a graph with bounded tree-independence number must not contain either of them as an induced minor for some constant t. We ask whether also the converse holds.

Problem. *Is there a function $f(t)$ so that all graphs G that exclude both the $t \times t$-grid and $K_{t,t}$ as induced minors have $\mathsf{tree}\text{-}\alpha(G) \leq f(t)$?*

Some participants conjectured that the answer would be negative, and in particular that the layered wheels of Sintiari and Trotignon [2] could be a counterexample.

References

[1] Clément Dallard, Martin Milanič, and Kenny Štorgel. Treewidth versus clique number. II. Tree-independence number. *arXiv:2111.04543*, 2022.

[2] Ni Luh Dewi Sintiari and Nicolas Trotignon. (Theta, triangle)-free and (even hole, K$_4$)-free graphs - Part 1: Layered wheels. *J. Graph Theory*, 97(4):475–509, 2021.

Towards Strong Dichotomy of Graph Covers

Jan Kratochvil
Charles University, Prague

We consider undirected graphs with multiple edges, loops and semi-edges (a *semi-edge* is a single-vertex edge that contributes 1 to the degree of its vertex, as opposed to a loop, which contributes 2). A graph is called *simple* if it contains no loops, no semi-edges and no multiple edges. A *graph covering projection* from a graph G to a graph H is a pair of incidence preserving mappings between the vertices and between the edges of G and H which is locally bijective on the neighborhoods of all vertices (in other words, the preimage of a normal edge is a matching spanning the preimage of its vertex set, the preimage of a loop is a disjoint union of cycles spanning the preimage of its vertex, and the preimage of a semi-edge is a disjoint union of semi-edges and a matching spanning the preimage of its vertex, see [2] for a formal definition). We write $G \longrightarrow H$ if G allows a covering projection onto H. The Strong Dichotomy Conjecture states that for every graph H, deciding if an input graph G allows a covering projection onto H is polynomial time solvable for arbitrary inputs, or NP-hard for simple input graphs G. In order to better understand the role of simple graphs on input, we introduced the following notion in [1]:

Definition. *Given graphs A, B, we say that A is* stronger than *B, if for every simple graph G, it is true that G covers B whenever G covers A. We write $A \triangleright B$ in such a case.*

Observation 1. *If A covers B, then A is stronger than B.* (This follows from transitivity of covering projections.)

Observation 2. *If A is a simple graph, then A is stronger than B, if and only if A covers B.*

Example 1. Let $F(a, b)$ denote the one-vertex graph with a semi-edges and b loops. Then neither $F(1, 1)$ covers $F(3, 0)$, nor $F(3, 0)$ covers $F(1, 1)$. Every simple graph that covers $F(3, 0)$ is a cubic 3-edge-colorable graph, thus it has a perfect matching, and therefore covers $F(1, 1)$. Hence $F(3, 0) \triangleright F(1, 1)$.

Example 2. Let $\widetilde{P}_k$ be the path on k vertices with one semi-edge pending on each of its end-vertices. Then $\widetilde{P}_k \triangleright C_{2k}$.

All other examples we know are based on variations of the above two. We would like to understand the relation $\triangleright$. Of course, this relation is at least as difficult as graph covering itself, but perhaps pairs A, B such that $A \triangleright B$ and $A \not\longrightarrow B$ can be described. In this direction we propose the following conjectures:

Conjecture 1. If A has no semi-edges, then $A \triangleright B$ if and only if $A \longrightarrow B$.

Conjecture 2. If A has no semi-edges and no loops, then $A \triangleright B$ if and only if $A \longrightarrow B$.

References

[1] Bok, J., Fiala, J., Jedličková, N., Kratochvíl, J., Seifrtová, M. (2021). Computational complexity of covering disconnected multigraphs. In *International Symposium on Fundamentals of Computation Theory* (pp. 85-99).

[2] Bok, J., Fiala, J., Hliněný, P., Jedličková, N., Kratochvíl, J. (2021). Computational complexity of covering multigraphs with semi-edges: small cases. In *46th International Symposium on Mathematical Foundations of Computer Science* (MFCS 2021).

A Graph Coloring Problem: Stack Allocation for Structured Programs

Philipp K. Krause

Stack Allocation for Structured Programs.
This problem has practical applications in compiler construction, for putting local variables of functions that have not been allocated to registers onto the stack efficiently.

Input: A graph G of bounded treewidth, a set V of connected subgraphs of G, a function $s\colon V \to \mathbb{N}$.

Output: A coloring of the nodes of the intersection graph of V, such that each $v \in V$ gets assigned $s(v)$ consecutive colors. Minimize the highest color used.

In general, this is an NP-hard problem. So I wonder how well it can be approximated. For the special case of $s = 1$, there is a a polynomial time $(\lfloor \frac{tw(G)}{2} \rfloor + 1)$-approximation [2]. The decision problem asking if a fixed number of colors is sufficent can be decided in polynomial time (exponential in the fixed number and in $tw(G)$) [1].

Variants.
Any combination of the following yields an interesting variant.

- The values of s are powers of 2.
- There is a small bound, such as 4 or 8, on s.
- Alignment: there is an $a\colon V \to \mathbb{N}$ the values of which are powers of 2, such that $a \leq s$. The lowest color assigned to v has to be a multiple of $a(v)$.

References

[1] Philipp K. Krause. Optimal Register Allocation in Polynomial Time. In *Compiler Construction - 22nd International Conference, CC 2013, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2013. Proceedings*, volume 7791 of *LNCS*, pages 1–20. Springer, 2013.

[2] Mikkel Thorup. All Structured Programs Have Small Tree Width and Good Register Allocation. *Information and Computation*, 142(2):159–181, 1998.

Non-preemptive tree packing

Stefan Lendl
Institute of Operations and Information Systems, University of Graz, Austria

The *tree packing problem* of Nash-Williams is a classical problem. Here, one is given a graph with nonnegative integral edge weights $w : E \to \mathbb{N}_0$. The goal is to pack a maximum number of spanning trees into the graph so that for every edge, the number of spanning trees using e does not exceed $w(e)$. The problem can be solved in strongly polynomial time.

We interpret the tree-packing problem as a scheduling problem: Every edge is a resource that can be scheduled for a total of $w(e)$ time units, where preemption of edges is allowed. (See Fig. 1 as an illustration, here edge e_3 gets preempted at time 1 and resumed at time 2.) The goal is that the scheduled edges connect the graph for a maximal amount of time. The optimal objective value for a graph G and weight function w is denoted by $\mathsf{tp}(G, w)$.

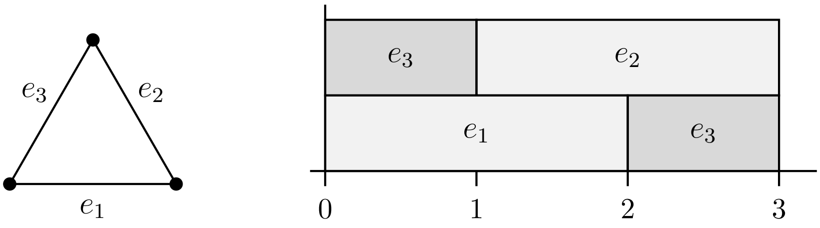


Figure 1: The three edges in the graph on the left hand side have weights $w(e_1) = w(e_2) = w(e_3) = 2$. The schedule on the right hand side keeps the graph connected for a total of three time units, i.e., $\mathsf{tp}(G, w) = 3$. Observe, that in the non-preemptive case the graph can only be kept connected for two time units, i.e., $\mathsf{ntp}(G, w) = 2$.

The non-preemptive version of tree packing. We consider a non-preemptive variant of the above tree packing problem, where the execution of edges must not be preempted: Every edge e is activated at some time point $\tau(e)$ chosen by the scheduler, and then remains active without interruption during the full time interval $[\tau(e),\, \tau(e) + w(e)]$. The objective is again to activate the edges in such a way that the graph remains connected for the longest possible overall time. The resulting combinatorial optimization problem is called *non-preemptive tree packing* (N-TreePack for short), and the optimal objective value for a graph $G = (V, E)$ with edge weights $w : E \to \mathbb{N}_0$ will be denoted $\mathsf{ntp}(G, w)$.

In the example in Figure 1, one sees that one can pack three spanning trees, but $\mathsf{ntp}(G, w) = 2$.

Several hardness results are known for this problem [1]:

- Computing $\mathsf{ntp}(G, w)$ is strongly NP-hard, even on complete bipartite graphs $K_{2,n}$.
- Computing $\mathsf{ntp}(G, w)$ is strongly NP-hard, even on graphs of bandwidth 2.
- Deciding whether $\mathsf{ntp}(G, w) \geq 7$ is NP-complete.
- There exists no 7/6-approximation for $\mathsf{ntp}(G, w)$, unless P=NP.

On the positive side, the following results are known [1]:

- A simple greedy algorithm (always connect using maximum weight edges) is an $(n-1)$-approximation. On cactus graphs this greedy algorithm always finds an optimal solution, whereas for every non-cactus graph G there exist edge weights w so that on the input (G, w) the greedy algorithm fails to find an optimal solution.
- Computing $\mathsf{ntp}(G, w)$ is fixed parameter tractable with respect to the size k of a feedback edge set. There is a kernel with $\mathcal{O}(k)$ vertices and edges.

Open questions.

Question 1. *Is the gap between* $\mathsf{ntp}(G, w)$ *and* $\mathsf{tp}(G, w)$ *constant?*

- The gap of an instance is the quotient of the solution of the spanning tree packing problem, and the non-preemptive spanning tree packing problem. So far, we do not even have examples where the gap is larger than 2!

Question 2. *Is there a constant-factor approximation for* N-TreePack*?*

- It is known that there exists a $\mathcal{O}(|V|)$-approximation.
- A currently unpublished $\mathcal{O}(\sqrt{|V|})$-approximation is known.

Question 3. *Is computing* $\mathsf{ntp}(G, w)$ *fixed parameter tractable with respect to the size k, the distance to cactus graphs?*

References

[1] Stefan Lendl, Gerhard Woeginger, and Lasse Wulf. Non-preemptive tree packing. *Algorithmica*, 2022.

Deciding atomicity of hereditary graph classes

Vadim Lozin
University of Warwick

A hereditary class X of graphs is said to be *atomic* if X cannot be expressed as a union of two non-empty hereditary classes different from X. It is well-known that atomicity is equivalent to the *joint embedding property* (JEP), which can be defined as follows: for any two graphs $G \in X$ and $H \in X$ there is a graph $F \in X$ containing G and H as induced subgraphs.

The question of deciding atomicity, or JEP, was addressed in various contexts. In particular, in [2] Braunfeld has shown that this question is undecidable for hereditary classes of graphs defined by finitely many forbidden induced subgraphs. On the other hand, atomicity is decidable for hereditary subclasses of threshold graphs. This is because, first, there is a bijection between n-vertex threshold graphs and binary words of length $n-1$, and second, atomicity is decidable for subword-closed languages over finite alphabets [1].

The class of threshold graphs is precisely the class of $(P_4, C_4, 2K_2)$-free graphs. Since the class of P_4-free graphs (cographs) is well-quasi-ordered under the induced subgraph relation [3], each hereditary subclass of cographs is defined by finitely many forbidden induced subgraphs.

Problem. *Can decidability of atomicity of hereditary subclasses of threshold graphs be extended to all hereditary subclasses of P_4-free graphs? In other words, is there an algorithm, which decides atomicity of a subclass of P_4-free graphs given by a finite collection of minimal forbidden induced subgraphs?*

References

[1] A. Atminas, V. Lozin, Deciding atomicity of subword-closed languages, *Lecture Notes in Computer Science*, 13257 (2022) 69–77.

[2] S. Braunfeld, The undecidability of joint embedding and joint homomorphism for hereditary graph classes. *Discrete Math. Theor. Comput. Sci.* 21 (2019), no. 2, Paper No. 9, 17 pp.

[3] P. Damaschke, Induced subgraphs and well-quasi-ordering, *J. Graph Theory*, 14 (1990) 427–435.

Inherence of (s, t, v)

Martin Milanič
FAMNIT and IAM, University of Primorska

We describe a problem posed recently by Gurvich et al. in [1]. We restrict ourselves to the description of the problem and refer the interested reader to [1] for motivation and further background.

Let the *fork* be the tree with 5 vertices that contains exactly 3 leaves. Let G be a graph and let (s, t, v) be an induced 3-vertex path in G. We say that an ordered pair of vertices (s', t') is an *extension* of (s, t, v) if s' is adjacent to s, t' is adjacent to t, and $\{s, t, v, s', t'\}$ induces a fork in G. We say that (s, t, v) is *avoidable* in G if for every extension (s', t') of (s, t, v) there exists an s', t'-path in G all of whose inner vertices belong to $V(G) \setminus N[\{s, t, v\}]$. Here, as usual, we denote by $N[\{s, t, v\}]$ the set of all vertices in G that are either equal or adjacent to a vertex in $\{s, t, v\}$.

Consider now the 3-vertex path (s, t, v) as a labeled graph on its own. We say that (s, t, v) is *inherent* if every graph G that contains an induced P_3 admits an avoidable (s, t, v).

Conjecture (Gurvich et al. [1])**.** *The path (s, t, v) is inherent.*

References

[1] V. Gurvich, M. Krnc, M. Milanič, and M. Vyalyi. Avoidability beyond paths. arXiv:2208.12803 [math.CO]. 2022.

Hamiltonian Cycle in Proper Chordal graphs

Evangelos Protopapas
LIRMM, Université Montpellier, CNRS

We first present the necessary definitions to introduce Proper Chordal graphs.

Definition (Tree-layout)**.** *Let $G = (V, E)$ be a graph on n vertices. A triple (T, r, ρ), where T is a rooted tree defined on a set V_T of n nodes, $r \in V_T$ is the root of T, and $\rho : V \to V_T$ is a bijection, is a* tree-layout *if:*

- *for every edge $xy \in E$, either $\rho(x)$ is an ancestor of $\rho(y)$ or vice-versa.*

Let $\mathbf{T} = (T, r, \rho)$ be a tree-layout of the graph $G = (V, E)$. Let x and y be two vertices of G. We note $x \prec_{\mathbf{T}} y$ if $\rho(x)$ is an ancestor of $\rho(y)$ in $\mathbf{T}$.

Definition (Indifference)**.** *A tree-layout $\mathbf{T} = (T, r, \rho)$ of a graph G is an* indifference tree-layout *if for every triple x, y, z of vertices such that $x \prec_{\mathbf{T}} y \prec_{\mathbf{T}} z$, if $xz \in E$ then $xy \in E$ and $yz \in E$.*

Definition (Proper Chordal)**.** *A graph is a* proper chordal graph *if it has an indifference tree-layout.*

Question. *Is* Hamiltonian Cycle *in P when restricted to Proper Chordal graphs?*

By definition the class is a superclass of Proper Interval graphs for which we know we can do it in linear time [1].

References

[1] Louis Ibarra. A simple algorithm to find Hamiltonian cycles in proper interval graphs, 2009. *Information Processing Letters*, Volume 109, Issue 18, Pages 1105-1108. `doi: 10.1016/j.ipl.2009.07.010`.

Recognizing level-$\mathcal{G}$ weighted graphs

Robert Scheffler
Brandenburg University of Technology

A *k-weighted graph* (G, ω) consists of a graph G and a surjective edge weight function $\omega : E(G) \to \{1, \ldots, k\}$. For any $1 \leq i \leq k+1$, the *i-th level graph* of (G, ω) is the spanning subgraph of G containing all edges $e \in E(G)$ with $\omega(e) \geq i$. Thus, the first level graph is G and the $(k+1)$-th level graph is the edgeless graph with $|V(G)|$ vertices. For a given graph class $\mathcal{G}$, we call a k-weighted graph (G, ω) a *level-$\mathcal{G}$ weighted graph* if every level graph of (G, ω) is an element of the class $\mathcal{G}$. These concepts where considered by Laurent and Seminaroti [2] as well as Laurent and Tanigawa [3].

If for a given graph G with n vertices and m edges we can decide in time $p(n, m)$ if $G \in \mathcal{G}$, then we can recognize a level-$\mathcal{G}$ weighted graph in time $\mathcal{O}(m \cdot p(n, m))$ by simply applying the recognition algorithm of $\mathcal{G}$ to all level graphs of (G, ω). As was shown by Beisegel et al. [1], we can improve the running time of a recognition algorithm of level-$\mathcal{G}$ weighted graphs to linear time if $\mathcal{G}$ is one of the following graph classes: split graphs, threshold graphs, or chain graphs.

Problem. *Are there other graph classes $\mathcal{G}$ for which level-$\mathcal{G}$ weighted graphs can be recognized faster than the trivial algorithm?*

References

[1] Jesse Beisegel, Nina Chiarelli, Ekkehard Köhler, Matjaž Krnc, Martin Milanič, Nevena Pivač, Robert Scheffler, and Martin Strehler. Edge elimination and weighted graph classes. In Isolde Adler and Haiko Müller, editors, *Graph-Theoretic Concepts in Computer Science*, volume 12301 of *LNCS*, pages 134–147, Cham, 2020. Springer.

[2] Monique Laurent and Matteo Seminaroti. A Lex-BFS-based recognition algorithm for Robinsonian matrices. *Discrete Applied Mathematics*, 222:151–165, 2017.

[3] Monique Laurent and Shin-ichi Tanigawa. Perfect elimination orderings for symmetric matrices. *Optimization Letters*, 14:339–353, 2017.

χ-bounded hereditary classes

Nicolas Trotignon
CNRS, LIP, ENS de Lyon

A graph class is *hereditary* if it is closed under vertex removal.

Problem. *Is it true that every hereditary class of graph for which there exists a polynomial time algorithm for computing the maximum stable set is χ-bounded?*

As pointed out by Viktor Zamaraev (and communicated to us by Konrad Dabrowski), the answer is known to be affirmative for classes of graphs defined by a finite number of forbidden induced subgraphs, assuming $\mathsf{P} \neq \mathsf{NP}$. If the stable set problem can be solved in polynomial time on a hereditary graph class defined by finitely many forbidden induced subgraphs, then one of the forbidden graphs must be a disjoint union of subdivided claws and paths, assuming $\mathsf{P} \neq \mathsf{NP}$. If H is a subdivided star, then the class of H-free graphs is χ-bounded, and if the classes of H_1-free and H_2-free graphs are both χ-bounded, then so is the class of $(H_1 + H_2)$-free graphs.

Obstructions to cycle rank

Sebastian Wiederrecht
Discrete Mathematics Group, Institute for Basic Science, Daejeon, South Korea

Definition (Cycle rank)**.** *Let D be a digraph. The* cycle rank *of D, denoted by $\mathsf{cr}(D)$, is recursively defined as follows:*

$$\begin{aligned} \textit{If } |V(D)| = 1, \textit{ then } \mathsf{cr}(D) &= 0, \\ \textit{if } D \textit{ is strongly connected, then } \mathsf{cr}(D) &= 1 + \min_{v \in V(D)} \mathsf{cr}(D - v), \textit{ and} \\ \mathsf{cr}(D) &= \max_{\substack{C \subseteq D \\ \textit{strong component}}} \mathsf{cr}(C) \textit{ otherwise.} \end{aligned}$$

Cycle rank is a generalisation of treedepth to the setting of digraphs and strong connectivity. For undirected graphs, the parameter treedepth is parametrically equivalent to the length of a longest path. For the cycle rank, however (as far as I know) no obstruction is known.

Since we are dealing with strong connectivity and therefore with directed cycles, it seems reasonable to allow a slight generalisation of the subgraph relation for the obstruction: butterfly minors.

An edge (u, v) is *butterfly contractible* in a digraph D if it is the only edge with head v or the only edge with tail u in D. A digraph H is a *butterfly minor* of some digraph D, if there exists a subgraph $D' \subseteq D$ such that H can be obtained from D' by a iteratively contracting butterfly contractible edges.

Problem. *What are the obstructions for cycle rank?*

In Fig. 2 you can find illustrations of two suggestions for candidates: The first is the family of *undirected paths*, that is the family of digraphs obtained from paths (as undirected graphs) by replacing every undirected edge by a digon (that is a pair of antiparallel edges). The second family is obtained from two directed paths of the same length,

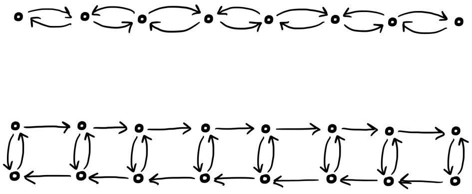


Figure 2: The two families of obstructions.

say n, by joining the ith vertex of the first path to the $n-i+1$st vertex of the second path with a digon for every $i \in [n]$.

Notice that these two families both have large cycle rank but no long enough undirected path contains a long member of the second family as a butterfly minor and vice versa. Please note that it is also possible that there are more obstruction families than these two!

Is the class of induced subgraphs of hypercubes small?

Viktor Zamaraev
University of Liverpool

Let $\mathcal{Q}$ be the hereditary closure of hypercube graphs, and let $\mathcal{Q}_n$ be the set of n-vertex graphs in $\mathcal{Q}$ with vertex set $\{1, 2, \dots, n\}$. It was shown recently that $|\mathcal{Q}_n| \leq n^{2n}$ [1]. We conjecture that a stronger bound holds.

Conjecture. *Class $\mathcal{Q}$ is* small*, i.e., there exists a constant c such that $|\mathcal{Q}_n| \leq c^n n!$.*

If this conjecture is true, class $\mathcal{Q}$ would be an explicit counterexample to the small conjecture [2]. The small conjecture says that every small class has bounded twin-width. It is known to be false, but I am not aware of explicit counter-examples.

References

[1] Bogdan Alecu, Vladimir E Alekseev, Aistis Atminas, Vadim Lozin, and Viktor Zamaraev. Graph parameters, implicit representations and factorial properties. In *International Workshop on Combinatorial Algorithms*, pages 60–72. Springer, 2022.

[2] Édouard Bonnet, Colin Geniet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width II: small classes. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1977–1996. SIAM, 2021.