Planarizing Graphs and their Drawings by Vertex Splitting^{*}

Soeren Nickel¹, Martin Nöllenburg¹, Manuel Sorge¹, Anaïs Villedieu¹, Hsiang-Yun Wu², and Jules Wulms¹

- 1 Algorithms and Complexity Group, TU Wien, Vienna, Austria {soeren.nickel|noellenburg|manuel.sorge|avilledieu|jwulms}@ac.tuwien.ac.at
- 2 St. Pölten University of Applied Sciences, St. Pölten, Austria and Research Unit of Computer Graphics, TU Wien, Vienna, Austria hsiang.yun.wu@acm.org

— Abstract

The splitting number of a graph G = (V, E) is the minimum number of vertex splits required to turn G into a planar graph, where a vertex split removes a vertex $v \in V$, introduces two new vertices v_1, v_2 , and distributes the edges formerly incident to v among its two split copies v_1, v_2 . The splitting number problem is known to be NP-complete. In this paper we shift focus to the splitting number of graph drawings in \mathbb{R}^2 , where the new vertices resulting from vertex splits must be re-embedded into the existing drawing of the remaining graph. We show the NP-completeness of the splitting number problem for graph drawings, even for its two subproblems of (1) selecting a minimum subset of vertices to split and (2) for re-embedding a minimum number of copies of a given set of vertices, which does not need to be a solution to (1). We present an FPT algorithm for the latter subproblem, parameterized by the number of vertex splits, which reduces the instance to bounded outerplanarity and then uses dynamic programming on its sphere-cut decomposition.

Related Version arXiv:2202.12293

1 Introduction

Visualizing dense graphs is a challenging task due to the potentially large number of edge crossings, which make tracing of individual edges harder and create clutter that negatively impacts readability [31]. Several approaches have been proposed to mitigate this issue [20], many aim to achieve readability properties similar to those of crossing-free drawings of planar graphs [30, 32, 34]. One such technique is to apply a sequence of vertex splitting operations. This approach has been studied from a theoretical perspective [8, 11, 23, 26], and is used in practice, e.g., by biologists and social scientists [18, 19, 29, 35, 36]. For a given graph G = (V, E) and a vertex $v \in V$, a vertex split of v replaces v by two non-adjacent copies v_1, v_2 and distributes the edges formerly incident to v to v_1 and v_2 . The minimum number of splits needed to obtain planarity is known as the splitting number of a graph and computing it is NP-hard [13]. The splitting numbers of complete graphs, complete bipartite graphs and the 4-cube [12, 16, 17, 22] are known. Similarly, the planar split thickness of a graph G is the minimum k such that G can be turned into a planar graph by applying a k-split (which creates k copies v_1, \ldots, v_k) to each vertex v of G. Deciding whether a graph has split thickness k is NP-complete [11].

^{*} Manuel Sorge acknowledges funding by the Alexander von Humboldt Foundation, Anaïs Villedieu acknowledges funding by the Austrian Science Fund (FWF) under grant P31119 and Jules Wulms acknowledges funding partially by the Austrian Science Fund (FWF) under grant P31119 and partially by the Vienna Science and Technology Fund (WWTF) under grant ICT19-035.

³⁸th European Workshop on Computational Geometry, Perugia, Italy, March 14–16, 2022.

This is an extended abstract of a presentation given at EuroCG'22. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear eventually in more final form at a conference with formal proceedings and/or in a journal.

14:2 Planarizing Graphs and their Drawings by Vertex Splitting

Contributions. Our focus in this paper is on vertex splitting for topological graph drawings in the plane \mathbb{R}^2 , where the subgraph induced by the non-split vertices retains its drawing. Similarities can be found with simultaneous embedding problems [5, 14, 15], and planar drawing extension problems [1, 2, 6, 7, 9, 10]. The underlying algorithmic problem for vertex splitting in drawings of graphs is two-fold: firstly, a suitable (minimum) subset of vertices to be split must be selected, and secondly the newly created copies of these vertices must be re-embedded in a crossing-free way together with a partition of the original edges of each split vertex into a subset for each copy. We show that both problems are NP-complete, and present an FPT algorithm for the re-embedding subproblem of the splitting number problem for graph drawings parameterized by the number of splits. We note that the smallest set of vertices as computed for the first subproblem is not necessarily the correct set of vertices to split when solving the complete problem.

Preliminaries. Let G = (V, E) be a graph. We write G[V'] to denote the subgraph of G induced by $V' \subseteq V$ and $N_G(v)$ to denote the neighborhood of a vertex v in G.

Let Γ be a topological drawing (for simplicity, from now on called a drawing) of G, which maps each vertex to a point in \mathbb{R}^2 and each edge to a simple curve (a Jordan arc) connecting the points corresponding to the incident vertices of that edge. We still refer to the points and curves as vertices and edges, respectively, in such a drawing. We assume Γ is a simple drawing, meaning no two edges intersect more than once, no three edges intersect in one point (except common endpoints), and adjacent edges do not cross. A split operation of a vertex $v \in V$ into two copies $\dot{v}^{(1)}, \dot{v}^{(2)}$ results in a drawing of the graph G' = (V', E') where $V' = V \setminus \{v\} \cup \{\dot{v}^{(1)}, \dot{v}^{(2)}\}$ and E' is obtained from E by distributing the edges incident to vamong $\dot{v}^{(1)}, \dot{v}^{(2)}$ such that $N_G(v) = N_{G'}(\dot{v}^{(1)}) \cup N_{G'}(\dot{v}^{(2)})$. It assigns new coordinates $\Gamma(\dot{v}^{(i)})$ to $\dot{v}^{(1)}, \dot{v}^{(2)}$ as well as new curves $\Gamma(e)$ to all edges e incident to any of the split vertices. If a copy \dot{v} of a vertex v is split again, then any copy of \dot{v} is also called a copy of the original vertex v and we use the notation $\dot{v}^{(i)}$ for $i = 1, 2, \ldots$ to denote the different copies of v.

▶ **Problem 1** (EMBEDDED SPLITTING NUMBER). Given a graph G = (V, E), a drawing Γ of G and an integer k, can G be transformed into a graph G' by applying at most k splits to G such that G' has a planar drawing that coincides with Γ when restricted to $G'[V(G) \cap V(G')]$?

Problem 1 includes two interesting subproblems, namely the *candidate selection problem* and the *re-embedding problem*. The candidate selection problem is related to the NP-complete problem of deleting at most k vertices from a non-embedded graph to make it planar [25, 28]. However, here we deal with a given drawing of a graph (with crossings).

▶ Problem 2 (CANDIDATE SELECTION). Given a graph G = (V, E), a drawing Γ of G and an integer k, can we find a candidate set $S_{cdt} \subset V$ of at most k vertices such that the drawing Γ restricted to $G[V \setminus S_{cdt}]$ is planar?

The vertices split in a solution of Problem 1 necessarily form such a candidate set, however, a minimum cardinality candidate set might not be the set that requires the least amount of splits to solve Problem 1, as vertices can be split multiple times and we might have to additionally split vertices whose incident edges are not involved in crossings.

Once a candidate set has been obtained we want to solve the second subproblem:

▶ **Problem 3** (SPLIT SET RE-EMBEDDING). Given a graph G = (V, E), a candidate set $S_{\text{cdt}} \subset V$, a drawing Γ of the subgraph $G[V \setminus S_{\text{cdt}}]$, and an integer $k \geq |S_{\text{cdt}}|$, can we perform at most k splits, splitting only vertices in S_{cdt} and splitting each vertex in S_{cdt} at least once, such that the resulting graph G' has a planar drawing that coincides with Γ when restricted to $G[V \setminus S_{\text{cdt}}]$?



Figure 1 (a) An example graph G, **(b)** a planar drawing Γ of G where S_{cdt} has been removed, and **(c)** a solution drawing Γ^* . Pistils are squares, copies are circles and vertices in S_{cdt} are disks.

While we find that SPLIT SET RE-EMBEDDING is FPT, the parameterized complexity of CANDIDATE SELECTION remains open.

2 Embedded Splitting Number Subproblems are NP-Complete

The reduction showing SPLITTING NUMBER to be NP-complete [13] does not seem to extend to EMBEDDED SPLITTING NUMBER. Here we show that CANDIDATE SELECTION is NP-complete using a reduction from planar 3-SAT inspired by Hummel et al. [21].¹

▶ **Theorem 2.1.** CANDIDATE SELECTION is NP-complete.

We then show that SPLIT SET RE-EMBEDDING also is NP-complete. We reduce from FACE COVER [4], where we are given a planar graph and a vertex subset S and we ask for the smallest set of faces F such that each vertex in S is incident to a face in F. We construct an instance of SPLIT SET RE-EMBEDDING with the same graph and an extra vertex v, where the candidate set is the vertex v and its neighborhood is N(v) = S. A re-embedding of k copies of v uses faces that induce a face cover and vice-versa, a face cover of size k gives the faces in which we can re-embed k copies of v.

▶ **Theorem 2.2.** SPLIT SET RE-EMBEDDING is NP-complete.

3 Split Set Re-Embedding is Fixed-Parameter Tractable

In this section we propose an FPT algorithm for Problem 3 (SPLIT SET RE-EMBEDDING) and prove the following theorem.

▶ **Theorem 3.1.** SPLIT SET RE-EMBEDDING can be solved in $2^{O(k^2)} \cdot n^{O(1)}$ time, where k is the number of allowed splits and n is the number of vertices in the input graph G.

Algorithm outline. We aim to re-embed copies of our candidate vertices with the following setup (Fig. 1). First, from the given set S_{cdt} of candidate vertices (disks in Fig. 1a) we choose how many copies of each vertex we will make. We initialize a set S_{λ} with one copy of every candidate vertex, then loop over every possibility of splitting vertices in $S_{cdt} k - |S_{cdt}|$ times. Note that $|S_{cdt}| \leq k$, and thus every computed set S_{λ} is obtained from k split operations. This creates $2^{O(k^2)}$ different sets. For each such computed set S_{λ} of copies we determine the connections among them. Next, we transform our input to be able to compute

¹ Alternatively, one can reduce from INDEPENDENT SET on segment intersection graphs [24] as suggested by a reviewer.

14:4 Planarizing Graphs and their Drawings by Vertex Splitting



Figure 2 (a) A graph and **(b)** its sphere-cut decomposition. Each labeled leaf corresponds to the same labeled edge of the graph. The middle set of each colored edge in the tree corresponds to the vertices of the corresponding colored dashed noose in the graph.

a sphere-cut decomposition of the new drawing, as explained in Section 3.1. We then use dynamic programming on the tree defined by this decomposition as sketched in Section 3.2. If this algorithm finds that our instance is a yes instance then a solution exists (see Fig. 1c).

We introduce the following terminology. Any vertex v that has a neighbor in S_{cdt} is called a *pistil*. Each face that is incident to a pistil is called a *petal*. Let p be a pistil in the input graph G with neighbors N(p). Let \dot{v} be a copy of some $v \in S_{\text{cdt}}$, where $v \in N(p)$. Given a drawing $\tilde{\Gamma}$ of a subgraph of G, we say \dot{v} covers p if \dot{v} is adjacent to p in $\tilde{\Gamma}$.

3.1 Finding a Sphere-Cut Decomposition

Given an instance of SPLIT SET RE-EMBEDDING (SSRE), we transform the induced graph $G[V \setminus S_{cdt}]$ in the following manner: any vertex $v \in V \setminus S_{cdt}$ that is not incident to a petal is removed. Then, any bridge in that new drawing is transformed into a multi-edge to obtain G' and its drawing Γ' . We can show that the instance obtained is a yes-instance if and only if the original instance is a yes-instance and the graph G' is 6k-outerplanar. A graph is ℓ -outerplanar if after ℓ times removing all vertices on the outer face the graph becomes empty. This can be exploited algorithmically in the following.

A branch decomposition of a (multi-)graph G is a pair (T, λ) where T is an unrooted binary tree, and λ is a bijection between the leaves of T and E(G). Every edge $e \in E(T)$ defines a bipartition of E(G) into A_e and B_e corresponding to the leaves in the two connected components of T - e. We define the *middle set* mid(e) of an edge $e \in E(T)$ to be the set of vertices incident to an edge in both sets A_e and B_e . The *width* of a branch decomposition is the size of the biggest middle set in that decomposition. The *branchwidth* of G is the minimum width over all branch decompositions of G.

A sphere-cut decomposition of a planar (multi-)graph G with a planar embedding Γ on a sphere Σ is a branch decomposition (T, λ) of G such that for each edge $e \in E(T)$ there is a noose $\eta(e)$: a closed curve on Σ such that its intersection with Γ is exactly the vertex set mid(e) (i.e., the curve does not intersect any edge of Γ) and such that the curve visits each face of Γ at most once (see Fig. 2). The removal of e from E(T) partitions T into two subtrees T_1, T_2 whose leaves correspond, respectively, to the noose's partition of Γ into two embedded subgraphs G_1, G_2 . Sphere-cut decompositions were introduced by Seymour and Thomas [33], more details can also be found in [27, Section 4.6]. The *length* of the noose $\eta(e)$ for an edge $e \in E(T)$ is the number of vertices on the noose (or the size of mid(e)) and it is at most the branchwidth of the decomposition. We defined drawings in the plane, whereas we need drawings on the sphere for sphere-cut decompositions. However, if we treat the outer face of a planar drawing just as any other face, then spherical and planar drawings are homeomorphic.

An ℓ -outerplanar graph has branchwidth at most 2ℓ [3] and a connected bridgeless planar graph of branchwidth at most b has a sphere-cut decomposition of width at most b that can be computed in $O(n^3)$ time (see [27, Section 4.6]). Since G' is 6k-outerplanar and bridgeless, we obtain a sphere-cut decomposition of G' of branchwidth 12k.

3.2 Dynamic Programming on a Sphere-Cut Decomposition Tree

Initialization. The dynamic program works bottom-up in the sphere-cut decomposition tree T from the leaves to an arbitrarily chosen root, considering iteratively larger subgraphs of G'. The algorithm determines how partial solutions look like on the interface between subgraphs and the rest of G'. We first transform T by defining a root vertex and move the information of the middle set from each edge to the child vertex (according to the new parentchild relations). For each vertex t of T, its noose $\eta(t)$ splits the graph into two subgraphs. We define the subgraph whose edges correspond to the leaves of the subtree of T rooted at t to be the graph *inside* the noose. A partial solution on a subgraph G'_t inside noose $\eta(t)$ is a planar drawing of that subgraph, with a subset $S'_{\lambda} \subseteq S_{\lambda}$ of copies embedded in it together with edges to the copies' neighborhood in G'_t such that all pistils not on the noose are covered. To describe those partial solutions we build tuples called *signatures* for each possible solution for each noose. A signature holds the following information (see Fig. 3): (i) the set of copies $S_{\rm in}$ used in faces entirely inside $\eta(t)$ to cover pistils, (ii) the set N_{η} of sets X_{v} of neighbors of each vertex $v \in \eta(t)$ that do not cover v, (iii) graphs that represent embeddings of copies for all the faces traversed by the noose, and (iv) for each such graph a pair of pointers p_s, p_e that describe which vertices of that embedding are used to cover pistils in G'_t . We find that the number of signatures is upper bounded by $2^{O(k^2)}$. The embeddings from (iii) are described by a set C_{out} of graphs called *nesting graphs* (see Fig. 4), which are planar graphs C_f , where a set of copies are embedded inside a cycle and each vertex of the cycle has exactly one neighbor that is a copy. The intuition behind a nesting graph is that, when embedded inside a face f, one can simultaneously traverse the cycle of C_f and the face f in the same direction, and draw an edge between each cycle vertex and a corresponding pistil. Then, after removing the cycle edges, and contracting the cycle vertices to pistils edges, we obtain a planar embedding for f, where some pistils strictly inside $\eta(t)$ are covered by the copies in C_f .



Figure 3 The information stored in a signature of the partial solution inside the orange noose: (grey) copies used in these faces are stored in S_{in} , (blue) noose vertices, for which missing neighbors (outgoing edges outside the noose) are stored in N_{η} , (red) an example of a nesting graph for a face traversed by the noose, with four (dotted) edges connecting to the cycle, (green) p_s and p_e pointers.



Figure 4 (a) A face f and the copies inside the orange noose. **(b)** The corresponding nesting graph C_f with its p_s and p_e vertices in orange. The two light blue vertices represent different copies of the same removed vertex.

Traversing the Sphere-Cut Decomposition Tree. To find a solution we perform bottomup dynamic programming on T. For each node $t \in V(T)$ we compute a table of all signatures with a corresponding partial solution. For a leaf $t \in V(T)$, graph G_t is an edge (u_1, u_2) , and we can go over all possible signatures and check whether we can cover all neighbors of u_1 and u_2 not in $N_{\eta} = \{X_{u_1}, X_{u_2}\}$, using for each incident face f the subgraph of $C_f \in C_{\text{out}}$ that lies between p_s, p_e . For internal nodes we merge some pairs of child signatures corresponding to two nooses $\eta(t_1)$ and $\eta(t_2)$. We merge when (1) faces not shared between the nooses do not have copies in common, (2) shared faces use identical nesting graphs and (3) use disjoint subgraphs of those nesting graphs to cover pistils, and (4) noose vertices in mid (t_1) and mid (t_2) do not have remaining missing neighbors. Thus we can find valid signatures for all nodes of T and notably for its root. If we find a valid signature for the root, we also have a partial solution. In Γ' all pistils are covered and it is planar, as the nesting graphs are planar and they represent a combinatorial embedding that allowed to cover pistils. We verify that the remaining pistils in $S_{\lambda} \setminus S'_{\lambda}$ form a planar graph which allows us to embed them in a face of Γ' to obtain a solution Γ^* .

— References

- Patrizio Angelini, Giuseppe Di Battista, Fabrizio Frati, Vít Jelínek, Jan Kratochvíl, Maurizio Patrignani, and Ignaz Rutter. Testing planarity of partially embedded graphs. ACM Transactions on Algorithms, 11(4):32:1–32:42, 2015. doi:10.1145/2629341.
- 2 Alan Arroyo, Fabian Klute, Irene Parada, Raimund Seidel, Birgit Vogtenhuber, and Tilo Wiedera. Inserting one edge into a simple drawing is hard. In Isolde Adler and Haiko Müller, editors, Proc. 46th International Workshop on Graph-Theoretic Concepts in Computer Science (WG), volume 12301 of LNCS, pages 325–338. Springer, 2020. doi:10.1007/978-3-030-60440-0_26.
- 3 Therese Biedl. On triangulating k-outerplanar graphs. Discrete Applied Mathematics, 181:275-279, 2015. doi:10.1016/j.dam.2014.10.017.
- 4 Daniel Bienstock and Clyde L. Monma. On the complexity of covering vertices by faces in a planar graph. *SIAM Journal on Computing*, 17(1):53–76, 1988. doi:10.1137/0217004.
- 5 Peter Braß, Eowyn Cenek, Christian A. Duncan, Alon Efrat, Cesim Erten, Dan Ismailescu, Stephen G. Kobourov, Anna Lubiw, and Joseph S. B. Mitchell. On simultaneous planar

graph embeddings. Computational Geometry: Theory and Applications, 36(2):117–130, 2007. doi:10.1016/j.comgeo.2006.05.006.

- 6 Markus Chimani, Carsten Gutwenger, Petra Mutzel, and Christian Wolf. Inserting a vertex into a planar graph. In Claire Mathieu, editor, Proc. 20th Symposium on Discrete Algorithms (SODA), pages 375–383. SIAM, 2009. doi:10.1137/1.9781611973068.42.
- 7 Markus Chimani and Petr Hlinený. Inserting multiple edges into a planar graph. In Sándor P. Fekete and Anna Lubiw, editors, Proc. 32nd International Symposium on Computational Geometry (SoCG), volume 51 of LIPIcs, pages 30:1–30:15, 2016. doi: 10.4230/LIPIcs.SoCG.2016.30.
- 8 Peter Eades and Candido F. X. de Mendonça N. Vertex splitting and tension-free layout. In Franz-Josef Brandenburg, editor, Proc. 3rd International Symposium on Graph Drawing (GD), volume 1027 of LNCS, pages 202–211. Springer, 1995. doi:10.1007/BFb0021804.
- 9 Eduard Eiben, Robert Ganian, Thekla Hamm, Fabian Klute, and Martin Nöllenburg. Extending nearly complete 1-planar drawings in polynomial time. In Javier Esparza and Daniel Král', editors, Proc. 45th International Symposium on Mathematical Foundations of Computer Science (MFCS), volume 170 of LIPIcs, pages 31:1–31:16. Schloss Dagstuhl Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.MFCS.2020.31.
- 10 Eduard Eiben, Robert Ganian, Thekla Hamm, Fabian Klute, and Martin Nöllenburg. Extending partial 1-planar drawings. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, Proc. 47th International Colloquium on Automata, Languages, and Programming (ICALP), volume 168 of LIPIcs, pages 43:1–43:19. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ICALP.2020.43.
- 11 David Eppstein, Philipp Kindermann, Stephen G. Kobourov, Giuseppe Liotta, Anna Lubiw, Aude Maignan, Debajyoti Mondal, Hamideh Vosoughpour, Sue Whitesides, and Stephen K. Wismath. On the planar split thickness of graphs. *Algorithmica*, 80(3):977–994, 2018. doi:10.1007/s00453-017-0328-y.
- 12 Luérbio Faria, Celina M. H. de Figueiredo, and Candido F. X. de Mendonça N. The splitting number of the 4-cube. In Claudio L. Lucchesi and Arnaldo V. Moura, editors, *Proc. 3rd Latin American Symposium on Theoretical Informatics (LATIN)*, volume 1380 of *LNCS*, pages 141–150. Springer, 1998. doi:10.1007/BFb0054317.
- 13 Luérbio Faria, Celina M. H. de Figueiredo, and Candido F. X. de Mendonça N. Splitting number is NP-complete. *Discrete Applied Mathematics*, 108(1):65–83, 2001. doi:10.1016/ S0166-218X(00)00220-1.
- 14 Fabrizio Frati, Michael Kaufmann, and Stephen G. Kobourov. Constrained simultaneous and near-simultaneous embeddings. Journal of Graph Algorithms and Applications, 13(3):447-465, 2009. doi:10.7155/jgaa.00194.
- 15 Emilio Di Giacomo, Walter Didimo, Marc J. van Kreveld, Giuseppe Liotta, and Bettina Speckmann. Matched drawings of planar graphs. *Journal of Graph Algorithms and Applications*, 13(3):423-445, 2009. doi:10.7155/jgaa.00193.
- 16 Nora Hartsfield. The toroidal splitting number of the complete graph k_n. Discrete Mathematics, 62(1):35–47, 1986. doi:10.1016/0012-365X(86)90039-7.
- 17 Nora Hartsfield, Brad Jackson, and Gerhard Ringel. The splitting number of the complete graph. Graphs and Combinatorics, 1(1):311–329, 1985. doi:10.1007/BF02582960.
- 18 Nathalie Henry, Anastasia Bezerianos, and Jean-Daniel Fekete. Improving the readability of clustered social networks using node duplication. *IEEE Transactions on Visualization* and Computer Graphics, 14(6):1317–1324, 2008. doi:10.1109/TVCG.2008.141.
- Nathalie Henry Riche and Tim Dwyer. Untangling euler diagrams. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1090–1099, 2010. doi:10.1109/TVCG.2010.
 210.

14:8 Planarizing Graphs and their Drawings by Vertex Splitting

- 20 Ivan Herman, Guy Melançon, and M. Scott Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43, 2000. doi:10.1109/2945.841119.
- 21 Matthias Hummel, Fabian Klute, Soeren Nickel, and Martin Nöllenburg. Maximizing ink in partial edge drawings of k-plane graphs. In Daniel Archambault and Csaba D. Tóth, editors, Proc. 27th International Symposium on Graph Drawing and Network Visualization (GD), volume 11904 of LNCS, pages 323–336. Springer, 2019. doi:10.1007/ 978-3-030-35802-0_25.
- 22 Brad Jackson and Gerhard Ringel. The splitting number of complete bipartite graphs. Archiv der Mathematik, 42(2):178–184, 1984. doi:10.1007/BF01772941.
- 23 Kolja Knauer and Torsten Ueckerdt. Three ways to cover a graph. *Discrete Mathematics*, 339(2):745–758, 2016. doi:10.1016/j.disc.2015.10.023.
- 24 Jan Kratochvíl and Jaroslav Nešetřil. Independent set and clique problems in intersectiondefined classes of graphs. *Commentationes Mathematicae Universitatis Carolinae*, 31(1):85– 93, 1990.
- 25 John M. Lewis and Mihalis Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980. doi:10.1016/0022-0000(80)90060-4.
- 26 Annegret Liebers. Planarizing graphs A survey and annotated bibliography. Journal of Graph Algorithms and Applications, 5(1):1-74, 2001. doi:10.7155/jgaa.00032.
- 27 Dániel Marx and Michal Pilipczuk. Optimal parameterized algorithms for planar facility location problems using voronoi diagrams. CoRR, abs/1504.05476, 2015. arXiv:1504. 05476.
- 28 Dániel Marx and Ildikó Schlotter. Obtaining a planar graph by vertex deletion. Algorithmica, 62(3-4):807-822, 2012. doi:10.1007/s00453-010-9484-z.
- 29 Sune S. Nielsen, Marek Ostaszewski, Fintan McGee, David Hoksza, and Simone Zorzan. Machine learning to support the presentation of complex pathway graphs. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 18(3):1130–1141, 2019. doi: 10.1109/TCBB.2019.2938501.
- 30 Takao Nishizeki and Md. Saidur Rahman. Planar Graph Drawing, volume 12 of Lecture Notes Series on Computing. World Scientific, 2004. doi:10.1142/5648.
- 31 Helen C. Purchase. Which aesthetic has the greatest effect on human understanding? In Giuseppe Di Battista, editor, *Proc. 5th International Symposium on Graph Drawing (GD)*, volume 1353 of *LNCS*, pages 248–261. Springer, 1997. doi:10.1007/3-540-63938-1_67.
- 32 Helen C. Purchase, Christopher Pilcher, and Beryl Plimmer. Graph drawing aesthetics—created by users, not algorithms. *IEEE Transactions on Visualization and Computer Graphics*, 18(1):81–92, 2012. doi:10.1109/TVCG.2010.269.
- 33 Paul D. Seymour and Robin Thomas. Call routing and the ratcatcher. Combinatorica, 14(2):217-241, 1994. doi:10.1007/BF01215352.
- 34 Luca Vismara. Planar straight-line drawing algorithms. In Roberto Tamassia, editor, Handbook on Graph Drawing and Visualization, pages 193–222. Chapman and Hall/CRC, 2013. doi:10.1007/3-540-62495-3_42.
- 35 Hsiang-Yun Wu, Martin Nöllenburg, Filipa L. Sousa, and Ivan Viola. Metabopolis: Scalable network layout for biological pathway diagrams in urban map style. BMC Bioinformatics, 20(1):1–20, 2019. doi:10.1186/s12859-019-2779-4.
- 36 Hsiang-Yun Wu, Martin Nöllenburg, and Ivan Viola. Multi-level area balancing of clustered graphs. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–15, 2020. doi:10.1109/TVCG.2020.3038154.