Publications of Thomas Bläsius

This document lists all peer-reviewed publications of Thomas Bläsius, Chair for Algorithm Engineering, Hasso Plattner Institute, Potsdam, Germany. This listing was automatically generated on May 26, 2019. An up-to-date version is available online at hpi.de/friedrich/docs/publist/blaesius.pdf.

Journal articles

[1] Bläsius, T., Friedrich, T., Krohmer, A., Laue, S., Efficient Embedding of Scale-Free Graphs in the Hyperbolic Plane. In: *IEEE/ACM Transactions on Networking* 26, pp. 920–933, 2018.

Hyperbolic geometry appears to be intrinsic in many large real networks. We construct and implement a new maximum likelihood estimation algorithm that embeds scale-free graphs in the hyperbolic space. All previous approaches of similar embedding algorithms require at least a quadratic runtime. Our algorithm achieves quasilinear runtime, which makes it the first algorithm that can embed networks with hundreds of thousands of nodes in less than one hour. We demonstrate the performance of our algorithm on artificial and real networks. In all typical metrics, like Log-likelihood and greedy routing, our algorithm discovers embeddings that are very close to the ground truth.

[2] Bläsius, T., Karrer, A., Rutter, I., Simultaneous Embedding: Edge Orderings, Relative Positions, Cutvertices. In: *Algorithmica* 80, pp. 1214–1277, 2018.

A simultaneous embedding (with fixed edges) of two graphs G^1 and G^2 with common graph $G = G^1 \cap G^2$ is a pair of planar drawings of G^1 and G^2 that coincide on G. It is an open question whether there is a polynomial-time algorithm that decides whether two graphs admit a simultaneous embedding (problem SEFE). In this paper, we present two results. First, a set of three linear-time preprocessing algorithms that remove certain substructures from a given SEFE instance, producing a set of equivalent Sefe instances without such substructures. The structures we can remove are (1) cutvertices of the union graph $G^{\cup} = G^1 \cup G^2$, (2) most separating pairs of $G^{\cup}$, and (3) connected components of G that are biconnected but not a cycle. Second, we give an $O(n^3)$-time algorithm solving Sefe for instances with the following restriction. Let u be a pole of a P-node μ in the SPQR-tree of a block of G^1 or G^2. Then at most three virtual edges of μ may contain common edges incident to u. All algorithms extend to the sunflower case, i.e., to the case of more than two graphs pairwise intersecting in the same common graph.

[3] Baum, M., Bläsius, T., Gemsa, A., Rutter, I., Wegner, F., Scalable Exact Visualization of Isocontours in Road Networks via Minimum-Link Paths. In: *Journal of Computational Geometry* 9, pp. 24–70, 2018.

Isocontours in road networks represent the area that is reachable from a source within a given resource limit. We study the problem of computing accurate isocontours in realistic, large-scale networks. We propose isocontours represented by polygons with minimum number of segments that separate reachable and unreachable components of the network. Since the resulting problem is not known to be solvable in polynomial time, we introduce several heuristics that run in (almost) linear time and are simple enough to be implemented in practice. A key ingredient is a new practical linear-time algorithm for minimum-link paths in simple polygons. Experiments in a challenging realistic setting show excellent performance of our algorithms in practice, computing near-optimal solutions in a few milliseconds on average, even for long ranges.

[4] Bläsius, T., Stumpf, P., Ueckerdt, T., Local and Union Boxicity. In: *Discrete Mathematics* 341, pp. 1307–1315, 2018.

The boxicity $box(H)$ of a graph H is the smallest integer d such that H is the intersection of d interval graphs, or equivalently, that H is the intersection graph of axis-aligned boxes in R^d. These intersection representations can be interpreted as covering representations of the complement H^c of H with co-interval graphs, that is, complements of interval graphs. We follow the recent framework of global, local and folded covering numbers (Knauer and Ueckerdt, 2016) to define two new parameters: the local boxicity $box_l(H)$ and the union boxicity $box_u(H)$ of H. The union boxicity of H is the smallest d such that H^c can be covered with d vertex-disjoint unions of co-interval graphs, while the local boxicity of H is the smallest d such that H^c can be covered with co-interval graphs, at most d at every vertex. We show that for every graph H we have $box_l(H) \leq box_u(H) \leq box(H)$ and that each of these inequalities can be arbitrarily far apart. Moreover, we show that local and union boxicity are also characterized by intersection representations of appropriate axis-aligned boxes in R^d . We demonstrate with a few striking examples, that in a sense, the local boxicity is a better indication for the complexity of a graph, than the classical boxicity.

[5] Bläsius, T., Friedrich, T., Krohmer, A., Cliques in Hyperbolic Random Graphs. In: *Algorithmica* 80, pp. 2324–2344, 2018.

Most complex real world networks display scale-free features. This characteristic motivated the study of numerous random graph models with a power-law degree distribution. There is, however, no established and simple model which also has a high clustering of vertices as typically observed in real data. Hyperbolic random graphs bridge this gap. This natural model has recently been introduced by Krioukov et al. and has shown theoretically and empirically to fulfill all typical properties of real world networks, including power-law degree distribution and high clustering. We study cliques in hyperbolic random graphs G and present new results on the expected number of k-cliques $E[K_k]$ and the size of the largest clique $\omega(G)$. We observe that there is a phase transition at power-law exponent $\beta = 3$. More precisely, for $\beta \in (2, 3)$ we prove $E[K_k] = n^{k(3-\beta)/2}\Theta(k)^{-k}$ and $\omega(G) = \Theta(n^{(3-\beta)/2})$, while for $\beta \geq 3$ we prove $E[K_k] = n\Theta(k)^{-k}$ and $\omega(G) = \Theta(\log(n)/\log\log n)$. Furthermore, we show that for $\beta \geq 3$, cliques in hyperbolic random graphs can be computed in time $O(n)$. If the underlying geometry is known, cliques can be found with worst-case runtime $O(mn^{2.5})$ for all values of β.

[6] Bläsius, T., Rutter, I., A new perspective on clustered planarity as a combinatorial embedding problem. In: *Theoretical Computer Science* 609, pp. 306–315, 2016.

The clustered planarity problem (c-planarity) asks whether a hierarchically clustered graph admits a planar drawing such that the clusters can be nicely represented by regions. We introduce the cd-tree data structure and give a new characterization of c-planarity. It leads to efficient algorithms for c-planarity testing in the following cases. (i) Every cluster and every co-cluster (complement of a cluster) has at most two connected components. (ii) Every cluster has at most five outgoing edges. Moreover, the cd-tree reveals interesting connections between c-planarity and planarity with constraints on the order of edges around vertices. On one hand, this gives rise to a bunch of new open problems related to c-planarity, on the other hand it provides a new perspective on previous results.

[7] Bläsius, T., Rutter, I., Wagner, D., Optimal Orthogonal Graph Drawing with Convex Bend Costs. In: *Transactions on Algorithms* 12, pp. 33, 2016.

Traditionally, the quality of orthogonal planar drawings is quantified by the total number off bends or the maximum number of bends per edge. However, this neglects that, in typical applications, edges have varying importance. We consider the problem OptimalFlexDraw that is defined as follows. Given a planar graph G on n vertices with maximum degree 4 (4-planar graph) and for each edge e a cost function $cost_e \colon N_0 \to R$ defining costs depending on the number of bends e has, compute a planar orthogonal drawing of G of minimum cost. In this generality OptimalFlexDraw is NP-hard. We show that it can be solved efficiently if (1) the cost function of each edge is convex and (2) the first bend on each edge does not cause any cost. Our algorithm takes time $O(n, \cdot, T_{flow}(n)$ and $O(n^2, \cdot, T_{flow}(n))$ for biconnected and connected graphs, respectively, where $T_{flow}(n)$ denotes the time to compute a minimum-cost flow in a planar network with multiple sources and sinks. Our result is the first polynomial-time bend-optimization algorithm for general 4-planar graphs optimizing over all embeddings. Previous work considers restricted graph classes and unit costs.

[8] Bläsius, T., Lehmann, S., Rutter, I., Orthogonal Graph Drawing with Inflexible Edges. In: *Computational Geometry* 55, pp. 26–40, 2016.

We consider the problem of creating plane orthogonal drawings of 4-planar graphs (planar graphs with maximum degree 4) with constraints on the number of bends per edge. More precisely, we have a flexibility function assigning to each edge e a natural number $flex(e)$, its flexibility. The problem FlexDraw asks whether there exists an orthogonal drawing such that each edge e has at most $flex(e)$ bends. It is known that FlexDraw is NP-hard if $flex(e) = 0$ for every edge e. On the other hand, FlexDraw can be solved efficiently if $flex(e) \geq 1$ and is trivial if $flex(e) \geq 2$ for every edge e. To close the gap between the NP-hardness for $flex(e) = 0$ and the efficient algorithm for $flex(e) \geq 1$, we investigate the computational complexity of FlexDraw in case only few edges are inflexible (i.e., have flexibility 0). We show that for any $\epsilon > 0$ FlexDraw is NP-complete for instances with $O(n^\epsilon)$ inflexible edges with pairwise distance $\Omega(n^{1-\epsilon})$ (including the case where they induce a matching). On the other hand, we give an FPT-algorithm with running time $O(2^k \cdot n \cdot T_{flow}(n))$, where $T_{flow}(n)$ is the time necessary to compute a maximum flow in a planar flow network with multiple sources and sinks, and k is the number of inflexible edges having at least one endpoint of degree 4.

[9] Bläsius, T., Rutter, I., Simultaneous PQ-Ordering with Applications to Constrained Embedding Problems. In: *Transactions on Algorithms* 12, pp. 16, 2016.

In this article, we define and study the new problem of SIMULTANEOUS PQ-ORDERING. Its input consists of a set of PQ-trees, which represent sets of circular orders of their leaves, together with a set of child-parent relations between these PQ-trees, such that the leaves of the child form a subset of the leaves of the parent. SIMULTANEOUS PQ-ORDERING asks whether orders of the leaves of each of the trees can be chosen simultaneously; that is, for every child-parent relation, the order chosen for the parent is an extension of the order chosen for the child. We show that SIMULTANEOUS PQ-ORDERING is NP-complete in general, and we identify a family of instances that can be solved efficiently, the 2-fixed instances. We show that this result serves as a framework for several other problems that can be formulated as instances of SIMULTANEOUS PQ-ORDERING. In particular, we give linear-time algorithms for recognizing simultaneous interval graphs and extending partial interval representations. Moreover, we obtain a linear-time algorithm for PARTIALLY PQ-CONSTRAINED PLANARITY for biconnected graphs, which asks for a planar embedding in the presence of 16 PQ-trees that restrict the possible orderings of edges around vertices, and a quadratic-time algorithm for SIMULTANEOUS EMBEDDING WITH FIXED EDGES for biconnected graphs with a connected intersection. Both results can be extended to the case where the input graphs are not necessarily biconnected but have the property that each cutvertex is contained in at most two nontrivial blocks. This includes, for example, the case where both graphs have a maximum degree of 5.

[10] Bläsius, T., Rutter, I., Disconnectivity and relative positions in simultaneous embeddings. In: *Computational Geometry* 48, pp. 459–478, 2015.

For two planar graphs $G^1 = (V^1, E^1)$ and $G^2 = (V^2, E^2)$ sharing a common subgraph $G = G^1 \cap G^2$ the problem Simultaneous Embedding with Fixed Edges (SEFE) asks whether they admit planar drawings such that the common graph is drawn the same. Previous algorithms only work for cases where G is connected, and hence do not need to handle relative positions of connected components. We consider the problem where G, G^1 and G^2 are not necessarily connected.First, we show that a general instance of SEFE can be reduced in linear time to an equivalent instance where $V^1 = V^2$ and G^1 and G^2 are connected. Second, for the case where G consists of disjoint cycles, we introduce the CC-tree which represents all embeddings of G that extend to planar embeddings of G^1. We show that CC-trees can be computed in linear time, and that their intersection is again a CC-tree. This yields a linear-time algorithm for SEFE if all k input graphs (possibly $k > 2$) pairwise share the same set of disjoint cycles. These results, including the CC-tree, extend to the case where G consists of arbitrary connected components, each with a fixed planar embedding on the sphere. Then the running time is $O(n^2)$.

[11] Angelini, P., Bläsius, T., Rutter, I., Testing Mutual duality of Planar graphs. In: *Computational Geometry and Applications* 24, pp. 325–346, 2014.

We introduce and study the problem Mutual Planar Duality, which asks for planar graphs G_1 and G_2 whether G_1 can be embedded such that its dual is isomorphic to G_2. We show NP-completeness for general graphs and give a linear-time algorithm for biconnected graphs. We consider the common dual relation , where G_1 G_2 if and only they admit embeddings that result in the same dual graph. We show that is an equivalence relation on the set of biconnected graphs and devise a succinct, SPQR-tree-like representation of its equivalence classes. To solve Mutual Planar Duality for biconnected graphs, we show how to do isomorphism testing for two such representations in linear time. A special case of Mutual Planar Duality is testing whether a graph is self-dual. Our algorithm can handle the case of biconnected graphs in linear time and our NP-hardness proof extends to self-duality and also to map self-duality testing (which additionally requires to preserve the embedding).

[12] Bläsius, T., Krug, M., Rutter, I., Wagner, D., Orthogonal Graph Drawing with Flexibility Constraints. In: *Algorithmica* 68, pp. 859–885, 2014.

Traditionally, the quality of orthogonal planar drawings is quantified by either the total number of bends, or the maximum number of bends per edge. However, this neglects that in typical applications, edges have varying importance. In this work, we investigate an approach that allows to specify the maximum number of bends for each edge individually, depending on its importance. We consider a new problem called FlexDraw that is defined as follows. Given a planar graph $G = (V, E)$ on n vertices with maximum degree 4 and a function $flex\colon E \to N_0$ that assigns a flexibility to each edge, does G admit a planar embedding on the grid such that each edge e has at most $flex(e)$ bends? Note that in our setting the combinatorial embedding of G is not fixed. FlexDraw directly extends the problem β-embeddability asking whether G can be embedded with at most β bends per edge. We give an algorithm with running-time $O(n^2)$ solving FlexDraw when the flexibility of each edge is positive. This includes 1-embeddability as a special case and thus closes the complexity gap between 0-embeddability, which is NP-hard to decide, and 2-embeddability, which is efficiently solvable since every planar graph with maximum degree 4 admits a 2-embedding except for the octahedron. In addition to the polynomial-time algorithm we show that FlexDraw is NP-hard even if the edges with flexibility 0 induce a tree or a union of disjoint stars.

Conference papers

[13] Bläsius, T., Friedrich, T., Lischeid, J., Meeks, K., Schirneck, M., Efficiently Enumerating Hitting Sets of Hypergraphs Arising in Data Profiling. In: *Algorithm Engineering and Experiments (ALENEX)*, pp. 130–143, 2019.

We devise an enumeration method for inclusion-wise minimal hitting sets in hypergraphs. It has delay $O(m^{k^*+1} \cdot n^2)$ and uses linear space. Hereby, n is the number of vertices, m the number of hyperedges, and k^* the rank of the transversal hypergraph. In particular, on classes of hypergraphs for which the cardinality k^* of the largest minimal hitting set is bounded, the delay is polynomial. The algorithm solves the extension problem for minimal hitting sets as a subroutine. We show that the extension problem is W[3]-complete when parameterised by the cardinality of the set which is to be extended. For the subroutine, we give an algorithm that is optimal under the exponential time hypothesis. Despite these lower bounds, we provide empirical evidence showing that the enumeration outperforms the theoretical worst-case guarantee on hypergraphs arising in the profiling of relational databases, namely, in the detection of unique column combinations.

[14] Bläsius, T., Friedrich, T., Sutton, A. M., On the Empirical Time Complexity of Scale-Free 3-SAT at the Phase Transition. In: *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, 2019.

The hardness of formulas at the solubility phase transition of random propositional satisfiability (SAT) has been intensely studied for decades both empirically and theoretically. Solvers based on stochastic local search (SLS) appear to scale very well at the critical threshold, while complete backtracking solvers exhibit exponential scaling. On industrial SAT instances, this phenomenon is inverted: backtracking solvers can tackle large industrial problems, where SLS-based solvers appear to stall. Industrial instances exhibit sharply different structure than uniform random instances. Among many other properties, they are often heterogeneous in the sense that some variables appear in many while others appear in only few clauses. We conjecture that the heterogeneity of SAT formulas alone already contributes to the trade-off in performance between SLS solvers and complete backtracking solvers. We empirically determine how the run time of SLS vs. backtracking solvers depends on the heterogeneity of the input, which is controlled by drawing variables according to a scale-free distribution. Our experiments reveal that the efficiency of complete solvers at the phase transition is strongly related to the heterogeneity of the degree distribution. We report results that suggest the depth of satisfying assignments in complete search trees is influenced by the level of heterogeneity as measured by a power-law exponent. We also find that incomplete SLS solvers, which scale well on uniform instances, are not affected by heterogeneity. The main contribution of this paper utilizes the scale-free random 3-SAT model to isolate heterogeneity as an important factor in the scaling discrepancy between complete and SLS solvers at the uniform phase transition found in previous works.

[15] Bläsius, T., Fischbeck, P., Friedrich, T., Schirneck, M., Understanding the Effectiveness of Data Reduction in Public Transportation Networks. In: *Workshop on Algorithms and Models for the Web Graph (WAW)*, 2019.

Given a public transportation network of stations and connections, we want to find a minimum subset of stations such that each connection runs through a selected station. Although this problem is NP-hard in general, real-world instances are regularly solved almost completely by a set of simple reduction rules. To explain this behavior, we view transportation networks as hitting set instances and identify two characteristic properties, locality and heterogeneity. We then devise a randomized model to generate hitting set instances with adjustable properties. While the heterogeneity does influence the effectiveness of the reduction rules, the generated instances show that locality is the significant factor. Beyond that, we prove that the effectiveness of the reduction rules is independent of the underlying graph structure. Finally, we show that high locality is also prevalent in instances from other domains, facilitating a fast computation of minimum hitting sets.

[16] Bläsius, T., Friedrich, T., Katzmann, M., Krohmer, A., Hyperbolic Embeddings for Near-Optimal Greedy Routing. In: *Algorithm Engineering and Experiments (ALENEX)*, pp. 199–208, 2018.

Greedy routing computes paths between nodes in a network by successively moving to the neighbor closest to the target with respect to coordinates given by an embedding into some metric space. Its advantage is that only local information is used for routing decisions. We present different algorithms for generating graph embeddings into the hyperbolic plane that are well suited for greedy routing. In particular our embeddings guarantee that greedy routing always succeeds in reaching the target and we try to minimize the lengths of the resulting greedy paths. We evaluate our algorithm on multiple generated and real wold networks. For networks that are generally assumed to have a hidden underlying hyperbolic geometry, such as the Internet graph, we achieve near-optimal results, i.e., the resulting greedy paths are only slightly longer than the corresponding shortest paths. In the case of the Internet graph, they are only 6% longer when using our best algorithm, which greatly improves upon the previous best known embedding, whose creation required substantial manual intervention.

[17] Bläsius, T., Eube, J., Feldtkeller, T., Friedrich, T., Krejca, M. S., Lagodzinski, J. A. G., Rothenberger, R., Severin, J., Sommer, F., Trautmann, J., Memory-restricted Routing With Tiled Map Data. In: *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 3347–3354, 2018.

Modern routing algorithms reduce query time by depending heavily on preprocessed data. The recently developed Navigation Data Standard (NDS) enforces a separation between algorithms and map data, rendering preprocessing inapplicable. Furthermore, map data is partitioned into tiles with respect to their geographic coordinates. With the limited memory found in portable devices, the number of tiles loaded becomes the major factor for run time. We study routing under these restrictions and present new algorithms as well as empirical evaluations. Our results show that, on average, the most efficient algorithm presented uses more than 20 times fewer tile loads than a normal A*.

[18] Bläsius, T., Freiberger, C., Friedrich, T., Katzmann, M., Montenegro-Retana, F., Thieffry, M., Efficient Shortest Paths in Scale-Free Networks with Underlying Hyperbolic Geometry. In: *International Colloquium on Automata, Languages, and Programming (ICALP)*, pp. 20:1–20:14, 2018.

A common way to accelerate shortest path algorithms on graphs is the use of a bidirectional search, which simultaneously explores the graph from the start and the destination. It has been observed recently that this strategy performs particularly well on scale-free real-world networks. Such networks typically have a heterogeneous degree distribution (e.g., a power-law distribution) and high clustering (i.e., vertices with a common neighbor are likely to be connected themselves). These two properties can be obtained by assuming an underlying hyperbolic geometry. To explain the observed behavior of the bidirectional search, we analyze its running time on hyperbolic random graphs and prove that it is $\tilde{O}(n^{2-1/\alpha}+n^{1/(2\alpha)}+\delta_{\max})$ with high probability, where $\alpha\in(0.5,1)$ controls the power-law exponent of the degree distribution, and $\delta_{\max}$ is the maximum degree. This bound is sublinear, improving the obvious worst-case linear bound. Although our analysis depends on the underlying geometry, the algorithm itself is oblivious to it.

[19] Bläsius, T., Friedrich, T., Katzmann, M., Krohmer, A., Striebel, J., Towards a Systematic Evaluation of Generative Network Models. In: *Workshop on Algorithms and Models for the Web Graph (WAW)*, pp. 99–114, 2018.

Generative graph models play an important role in network science. Unlike real-world networks, they are accessible for mathematical analysis and the number of available networks is not limited. The explanatory power of results on generative models, however, heavily depends on how realistic they are. We present a framework that allows for a systematic evaluation of generative network models. It is based on the question whether real-world networks can be distinguished from generated graphs with respect to certain graph parameters. As a proof of concept, we apply our framework to four popular random graph models (Erdős-Rényi, Barabási-Albert, Chung-Lu, and hyperbolic random graphs). Our experiments for example show that all four models are bad representations for Facebook's social networks, while Chung-Lu and hyperbolic random graphs are good representations for other networks, with different strengths and weaknesses.

[20] Bläsius, T., Radermacher, M., Rutter, I., How to Draw a Planarization. In: *Current Trends in Theory and Practice of Computer Science (SOFSEM)*, pp. 295–308, 2017.

We study the problem of computing straight-line drawings of non-planar graphs with few crossings. We assume that a crossingminimization algorithm is applied first, yielding a planarization, i.e., a planar graph with a dummy vertex for each crossing, that fixes the topology of the resulting drawing. We present and evaluate two different approaches for drawing a planarization in such a way that the edges of the input graph are as straight as possible. The first approach is based on the planarity-preserving force-directed algorithm ImPrEd, the second approach, which we call Geometric Planarization Drawing, iteratively moves vertices to their locally optimal positions in the given initial drawing.

[21] Kovacs, R., Seufert, A., Wall, L., Chen, H.-T., Meinel, F., Müller, W., You, S., Brehm, M., Striebel, J., Kommana, Y., Popiak, A., Bläsius, T., Baudisch, P., TrussFab: Fabricating Sturdy Large-Scale Structures on Desktop 3D Printers. In: *Human Factors in Computing Systems (CHI)*, pp. 2606–2616, 2017.

We present TrussFab, an integrated end-to-end system that allows users to fabricate large scale structures that are sturdy enough to carry human weight. TrussFab achieves the large scale by complementing 3D print with plastic bottles. It does not use these bottles as "bricks" though, but as beams that form structurally sound node-link structures, also known as trusses, allowing it to handle the forces resulting from scale and load. TrussFab embodies the required engineering knowledge, allowing non-engineers to design such structures and to validate their design using integrated structural analysis. We have used TrussFab to design and fabricate tables and chairs, a 2.5 m long bridge strong enough to carry a human, a functional boat that seats two, and a 5 m diameter dome.

[22] Baum, M., Bläsius, T., Gemsa, A., Rutter, I., Wegner, F., Scalable Exact Visualization of Isocontours in Road Networks via Minimum-Link Paths. In: *European Symposium on Algorithms (ESA)*, pp. 7:1–7:18, 2016.

Isocontours in road networks represent the area that is reachable from a source within a given resource limit. We study the problem of computing accurate isocontours in realistic, large-scale networks. We propose isocontours represented by polygons with minimum number of segments that separate reachable and unreachable components of the network. Since the resulting problem is not known to be solvable in polynomial time, we introduce several heuristics that run in (almost) linear time and are simple enough to be implemented in practice. A key ingredient is a new practical linear-time algorithm for minimum-link paths in simple polygons. Experiments in a challenging realistic setting show excellent performance of our algorithms in practice, computing near-optimal solutions in a few milliseconds on average, even for long ranges.

[23] Bläsius, T., Friedrich, T., Krohmer, A., Hyperbolic Random Graphs: Separators and Treewidth. In: *European Symposium on Algorithms (ESA)*, pp. 15:1–15:16, 2016.

When designing and analyzing algorithms, one can obtain better and more realistic results for practical instances by assuming a certain probability distribution on the input. The worst-case run-time is then replaced by the expected run-time or by bounds that hold with high probability (whp), i.e., with probability $1 - O(1/n)$, on the random input. Hyperbolic random graphs can be used to model complex real-world networks as they share many important properties such as a small diameter, a large clustering coefficient, and a power-law degree-distribution. Divide and conquer is an important algorithmic design principle that works particularly well if the instance admits small separators. We show that hyperbolic random graphs in fact have comparatively small separators. More precisely, we show that a hyperbolic random graph can be expected to have a balanced separator hierarchy with separators of size $O(\sqrt{n^{(3-\beta)}})$, $O(\log n)$, and $O(1)$ if $2 < \beta < 3$, $\beta = 3$ and $3 < \beta$, respectively (β is the power-law exponent). We infer that these graphs have whp a treewidth of $O(\sqrt{n^{(3-\beta)}})$, $O(\log^2 n)$, and $O(\log n)$, respectively. For $2 < \beta < 3$, this matches a known lower bound. For the more realistic (but harder to analyze) binomial model, we still prove a sublinear bound on the treewidth. To demonstrate the usefulness of our results, we apply them to obtain fast matching algorithms and an approximation scheme for Independent Set.

[24] Bläsius, T., Friedrich, T., Krohmer, A., Laue, S., Efficient Embedding of Scale-Free Graphs in the Hyperbolic Plane. In: *European Symposium on Algorithms (ESA)*, pp. 16:1–16:18, 2016. EATCS Best Paper Award.

Hyperbolic geometry appears to be intrinsic in many large real networks. We construct and implement a new maximum likelihood estimation algorithm that embeds scale-free graphs in the hyperbolic space. All previous approaches of similar embedding algorithms require a runtime of $\Omega(n^2)$. Our algorithm achieves quasilinear runtime, which makes it the first algorithm that can embed networks with hundreds of thousands of nodes in less than one hour. We demonstrate the performance of our algorithm on artificial and real networks. In all typical metrics like Log-likelihood and greedy routing our algorithm discovers embeddings that are very close to the ground truth.

[25] Bläsius, T., Friedrich, T., Schirneck, M., The Parameterized Complexity of Dependency Detection in Relational Databases. In: *International Symposium on Parameterized and Exact Computation (IPEC)*, pp. 6:1–6:13, 2016.

We study the parameterized complexity of classical problems that arise in the profiling of relational data. Namely, we characterize the complexity of detecting unique column combinations (candidate keys), functional dependencies, and inclusion dependencies with the solution size as parameter. While the discovery of uniques and functional dependencies, respectively, turns out to be W[2]-complete, the detection of inclusion dependencies is one of the first natural problems proven to be complete for the class W[3]. As a side effect, our reductions give insights into the complexity of enumerating all minimal unique column combinations or functional dependencies.

[26] Bläsius, T., Lehmann, S., Rutter, I., Orthogonal Graph Drawing with Inflexible Edges. In: *Conference on Algorithms and Complexity (CIAC)*, pp. 61–73, 2015.

We consider the problem of creating plane orthogonal drawings of 4-planar graphs (planar graphs with maximum degree 4) with constraints on the number of bends per edge. More precisely, we have a flexibility function assigning to each edge e a natural number $flex(e)$, its flexibility. The problem FlexDraw asks whether there exists an orthogonal drawing such that each edge e has at most $flex(e)$ bends. It is known that FlexDraw is NP-hard if $flex(e) = 0$ for every edge e [7]. On the other hand, FlexDraw can be solved efficiently if $flex(e) \geq 1$ [2] and is trivial if $flex(e) \geq 2$ [1] for every edge e. To close the gap between the NP-hardness for $flex(e) = 0$ and the efficient algorithm for $flex(e) \geq 1$, we investigate the computational complexity of FlexDraw in case only few edges are inflexible (i.e., have flexibility 0). We show that for any $\epsilon > 0$ FlexDraw is NP-complete for instances with $O(n^\epsilon)$ inflexible edges with pairwise distance $\Omega(n^{1-\epsilon})$ (including the case where they induce a matching). On the other hand, we give an FPT-algorithm with running time $O(2^k \cdot n \cdot T_{flow}(n))$, where $T_{flow}(n)$ is the time necessary to compute a maximum flow in a planar flow network with multiple sources and sinks, and k is the number of inflexible edges having at least one endpoint of degree 4.

[27] Alam, M. J., Bläsius, T., Rutter, I., Ueckerdt, T., Wolff, A., Pixel and Voxel Representations of Graphs. In: *Graph Drawing (GD)*, pp. 472–486, 2015.

We study contact representations for graphs, which we call pixel representations in 2D and voxel representations in 3D. Our representations are based on the unit square grid whose cells we call pixels in 2D and voxels in 3D. Two pixels are adjacent if they share an edge, two voxels if they share a face. We call a connected set of pixels or voxels a blob. Given a graph, we represent its vertices by disjoint blobs such that two blobs contain adjacent pixels or voxels if and only if the corresponding vertices are adjacent. We are interested in the size of a representation, which is the number of pixels or voxels it consists of. We first show that finding minimum-size representations is NP-complete. Then, we bound representation sizes needed for certain graph classes. In 2D, we show that, for k-outerplanar graphs with n vertices, $\Theta(kn)$ pixels are always sufficient and sometimes necessary. In particular, outerplanar graphs can be represented with a linear number of pixels, whereas general planar graphs sometimes need a quadratic number. In 3D, $\Theta(n^2)$ voxels are always sufficient and sometimes necessary for any n-vertex graph. We improve this bound to $\Theta(n \cdot \tau)$ for graphs of treewidth τ and to $O((g + 1)^2 n \log^2 n)$ for graphs of genus g. In particular, planar graphs admit representations with $O(n \log^2 n)$ voxels.

[28] Bläsius, T., Brückner, G., Rutter, I., Complexity of Higher-Degree Orthogonal Graph Embedding in the Kandinsky Model. In: *European Symposium on Algorithms (ESA)*, pp. 161–172, 2014.

We show that finding orthogonal grid-embeddings of plane graphs (planar with fixed combinatorial embedding) with the minimum number of bends in the so-called Kandinsky model (which allows vertices of degree >4) is NP-complete, thus solving a long-standing open problem. On the positive side, we give an efficient algorithm for several restricted variants, such as graphs of bounded branch width and a subexponential exact algorithm for general plane graphs.

[29] Bläsius, T., Rutter, I., A New Perspective on Clustered Planarity as a Combinatorial Embedding Problem. In: *Graph Drawing (GD)*, pp. 440–451, 2014.

The clustered planarity problem (c-planarity) asks whether a hierarchically clustered graph admits a planar drawing such that the clusters can be nicely represented by regions. We introduce the cd-tree data structure and give a new characterization of c-planarity. It leads to efficient algorithms for c-planarity testing in the following cases. (i) Every cluster and every co-cluster has at most two connected components. (ii) Every cluster has at most five outgoing edges. Moreover, the cd-tree reveals interesting connections between c-planarity and planarity with constraints on the order of edges around vertices. On one hand, this gives rise to a bunch of new open problems related to c-planarity, on the other hand it provides a new perspective on previous results.

[30] Biedl, T. C., Bläsius, T., Niedermann, B., Nöllenburg, M., Prutkin, R., Rutter, I., Using ILP/SAT to Determine Pathwidth, Visibility Representations, and other Grid-Based Graph Drawings. In: *Graph Drawing (GD)*, pp. 460–471, 2013.

We present a simple and versatile formulation of grid-based graph representation problems as an integer linear program (ILP) and a corresponding SAT instance. In a grid-based representation vertices and edges correspond to axis-parallel boxes on an underlying integer grid; boxes can be further constrained in their shapes and interactions by additional problem-specific constraints. We describe a general d-dimensional model for grid representation problems. This model can be used to solve a variety of NP-hard graph problems, including pathwidth, bandwidth, optimum st-orientation, area-minimal (bar-k) visibility representation, boxicity-k graphs and others. We implemented SAT-models for all of the above problems and evaluated them on the Rome graphs collection. The experiments show that our model successfully solves NP-hard problems within few minutes on small to medium-size Rome graphs.

[31] Bläsius, T., Karrer, A., Rutter, I., Simultaneous Embedding: Edge Orderings, Relative Positions, Cutvertices. In: *Graph Drawing (GD)*, pp. 220–231, 2013.

A simultaneous embedding (with fixed edges) of two graphs G^1 and G^2 with common graph $G = G^1 \cap G^2$ is a pair of planar drawings of G^1 and G^2 that coincide on G. It is an open question whether there is a polynomial-time algorithm that decides whether two graphs admit a simultaneous embedding (problem Sefe). In this paper, we present two results. First, a set of three linear-time preprocessing algorithms that remove certain substructures from a given Sefe instance, producing a set of equivalent Sefe instances without such substructures. The structures we can remove are (1) cutvertices of the union graph $G^\cup = G^1 \cup G^2$, (2) most separating pairs of $G^\cup$, and (3) connected components of G that are biconnected but not a cycle. Second, we give an $O(n^3)$-time algorithm solving Sefe for instances with the following restriction. Let u be a pole of a P-node μ in the SPQR-tree of a block of G^1 or G^2 . Then at most three virtual edges of μ may contain common edges incident to u. All algorithms extend to the sunflower case, i.e., to the case of more than three graphs pairwise intersecting in the same common graph.

[32] Bläsius, T., Rutter, I., Wagner, D., Optimal Orthogonal Graph Drawing with Convex Bend Costs. In: *International Colloquium on Automata, Languages, and Programming (ICALP)*, pp. 184–195, 2013.

Traditionally, the quality of orthogonal planar drawings is quantified by the total number of bends or the maximum number of bends per edge. However, this neglects that, in typical applications, edges have varying importance. We consider the problem OptimalFlexDraw that is defined as follows. Given a planar graph G on n vertices with maximum degree 4 (4-planar graph) and for each edge e a cost function $cost_e \colon N_0 \to R$ defining costs depending on the number of bends e has, compute a planar orthogonal drawing of G of minimum cost. In this generality OptimalFlexDraw is NP-hard. We show that it can be solved efficiently if (1) the cost function of each edge is convex and (2) the first bend on each edge does not cause any cost. Our algorithm takes time $O(n, \cdot, T_{flow}(n))$ and $O(n^2, \cdot, T_{flow}(n))$ for biconnected and connected graphs, respectively, where $T_{flow}(n)$ denotes the time to compute a minimum-cost flow in a planar network with multiple sources and sinks. Our result is the first polynomial-time bend-optimization algorithm for general 4-planar graphs optimizing over all embeddings. Previous work considers restricted graph classes and unit costs.

[33] Angelini, P., Bläsius, T., Rutter, I., Testing Mutual Duality of Planar Graphs. In: *International Symposium on Algorithms and Computation (ISAAC)*, pp. 350–360, 2013.

We introduce and study the problem Mutual Planar Duality, which asks for planar graphs G_1 and G_2 whether G_1 can be embedded such that its dual is isomorphic to G_2. We show NP-completeness for general graphs and give a linear-time algorithm for biconnected graphs. We consider the common dual relation $\sim$, where $G_1 \sim G_2$ if and only they admit embeddings that result in the same dual graph. We show that is an equivalence relation on the set of biconnected graphs and devise a succinct, SPQR-tree-like representation of its equivalence classes. To solve Mutual Planar Duality for biconnected graphs, we show how to do isomorphism testing for two such representations in linear time. A special case of Mutual Planar Duality is testing whether a graph is self-dual. Our algorithm can handle the case of biconnected graphs in linear time and our NP-hardness proof extends to self-duality and also to map self-duality testing (which additionally requires to preserve the embedding).

[34] Bläsius, T., Rutter, I., Simultaneous PQ-Ordering with Applications to Constrained Embedding Problems. In: *Symposium on Discrete Algorithms (SODA)*, pp. 1030–1043, 2013.

In this article, we define and study the new problem of SIMULTANEOUS PQ-ORDERING. Its input consists of a set of PQ-trees, which represent sets of circular orders of their leaves, together with a set of child-parent relations between these PQ-trees, such that the leaves of the child form a subset of the leaves of the parent. SIMULTANEOUS PQ-ORDERING asks whether orders of the leaves of each of the trees can be chosen simultaneously; that is, for every child-parent relation, the order chosen for the parent is an extension of the order chosen for the child. We show that SIMULTANEOUS PQ-ORDERING is NP-complete in general, and we identify a family of instances that can be solved efficiently, the 2-fixed instances. We show that this result serves as a framework for several other problems that can be formulated as instances of SIMULTANEOUS PQ-ORDERING. In particular, we give linear-time algorithms for recognizing simultaneous interval graphs and extending partial interval representations. Moreover, we obtain a linear-time algorithm for PARTIALLY PQ-CONSTRAINED PLANARITY for biconnected graphs, which asks for a planar embedding in the presence of 16 PQ-trees that restrict the possible orderings of edges around vertices, and a quadratic-time algorithm for SIMULTANEOUS EMBEDDING WITH FIXED EDGES for biconnected graphs with a connected intersection. Both results can be extended to the case where the input graphs are not necessarily biconnected but have the property that each cutvertex is contained in at most two nontrivial blocks. This includes, for example, the case where both graphs have a maximum degree of 5.

[35] Bläsius, T., Rutter, I., Disconnectivity and Relative Positions in Simultaneous Embeddings. In: *Graph Drawing (GD)*, pp. 31–42, 2012.

For two planar graphs $G^1 = (V^1, E^1)$ and $G^2 = (V^2, E^2)$ sharing a common subgraph $G = G^1 \cap G^2$ the problem Simultaneous Embedding with Fixed Edges (SEFE) asks whether they admit planar drawings such that the common graph is drawn the same. Previous algorithms only work for cases where G is connected, and hence do not need to handle relative positions of connected components. We consider the problem where G, G^1 and G^2 are not necessarily connected.First, we show that a general instance of SEFE can be reduced in linear time to an equivalent instance where $V^1 = V^2$ and G^1 and G^2 are connected. Second, for the case where G consists of disjoint cycles, we introduce the CC-tree which represents all embeddings of G that extend to planar embeddings of G^1. We show that CC-trees can be computed in linear time, and that their intersection is again a CC-tree. This yields a linear-time algorithm for SEFE if all k input graphs (possibly $k > 2$) pairwise share the same set of disjoint cycles. These results, including the CC-tree, extend to the case where G consists of arbitrary connected components, each with a fixed planar embedding on the sphere. Then the running time is $O(n^2)$.

[36] Bläsius, T., Krug, M., Rutter, I., Wagner, D., Orthogonal Graph Drawing with Flexibility Constraints. In: *Graph Drawing (GD)*, pp. 92–104, 2010.