Graph Harvester

A Website for Extracting Graphs from Illustrations in Scientific Papers

Julius Deynet[1] Tim Hegemann[2] Sebastian Kempf[2] Alexander Wolff[2]

University of Würzburg

[1]jdeynet@googlemail.com [2]firstname.secondname@uni-wuerzburg.de

Problem

the House of Graphs

Database of interesting graphs –

but incomplete

+

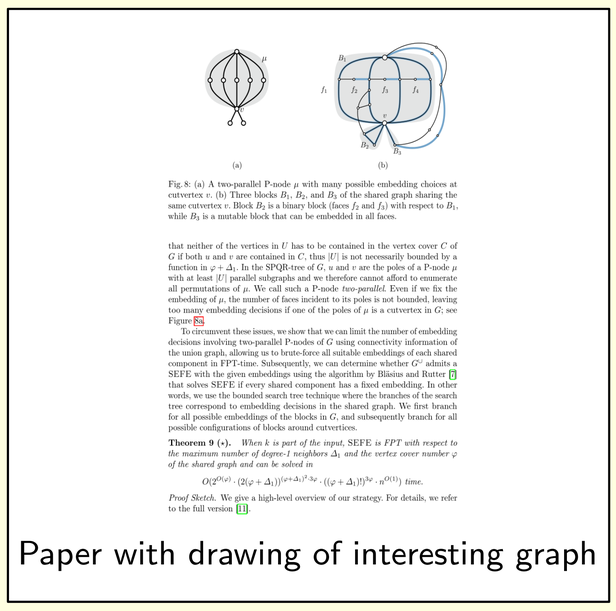


Fig. 8: (a) A two-parallel P-node μ with many possible embedding choices at cutvertex v. (b) Three blocks B_1, B_2, and B_3 of the shared graph sharing the same cutvertex v. Block B_2 is a binary block (faces f_2 and f_3) with respect to B_1, while B_3 is a mutable block that can be embedded in all faces.

that neither of the vertices in U has to be contained in the vertex cover C of G if both u and v are contained in C, thus $|U|$ is not necessarily bounded by a function in $\varphi + \Delta_1$. In the SPQR-tree of G, u and v are the poles of a P-node μ with at least $|U|$ parallel subgraphs and we therefore cannot afford to enumerate all permutations of μ. We call such a P-node *two-parallel*. Even if we fix the embedding of μ, the number of faces incident to its poles is not bounded, leaving too many embedding decisions if one of the poles of μ is a cutvertex in G; see Figure 8a.

To circumvent these issues, we show that we can limit the number of embedding decisions involving two-parallel P-nodes of G using connectivity information of the union graph, allowing us to brute-force all suitable embeddings of each shared component in FPT-time. Subsequently, we can determine whether $G^\cup$ admits a SEFE with the given embeddings using the algorithm by Bläsius and Rutter [7] that solves SEFE if every shared component has a fixed embedding. In other words, we use the bounded search tree technique where the branches of the search tree correspond to embedding decisions in the shared graph. We first branch for all possible embeddings of the blocks in G, and subsequently branch for all possible configurations of blocks around cutvertices.

Theorem 9 (⋆). *When k is part of the input,* SEFE *is FPT with respect to the maximum number of degree-1 neighbors Δ_1 and the vertex cover number φ of the shared graph and can be solved in*

$$O(2^{O(\varphi)} \cdot (2(\varphi+\Delta_1))^{(\varphi+\Delta_1)^2 \cdot 3\varphi} \cdot ((\varphi+\Delta_1)!)^{3\varphi} \cdot n^{O(1)}) \text{ time.}$$

Proof Sketch. We give a high-level overview of our strategy. For details, we refer to the full version [11].

Paper with drawing of interesting graph

=

Graph Harvester

- Extracting illustrations from scientific papers
- Graph6 encoding for extracted graphs is provided
- Easy integration with House of Graphs:
  view/edit/add extracted graphs

go.uniwue.de/graph-harvester

Pipeline

Input

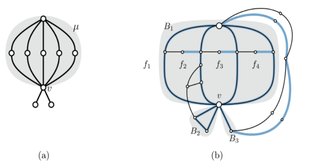


Fig. 8: (a) A two-parallel P-node μ with many possible embedding choices at cutvertex v. (b) Three blocks B_1, B_2, and B_3 of the shared graph sharing the same cutvertex v. Block B_2 is a binary block (faces f_2 and f_3) with respect to B_1, while B_3 is a mutable block that can be embedded in all faces.

that neither of the vertices in U has to be contained in the vertex cover C of G if both u and v are contained in C, thus $|U|$ is not necessarily bounded by a function in $\varphi + \Delta_1$. In the SPQR-tree of G, u and v are the poles of a P-node μ with at least $|U|$ parallel subgraphs and we therefore cannot afford to enumerate all permutations of μ. We call such a P-node *two-parallel*. Even if we fix the embedding of μ, the number of faces incident to its poles is not bounded, leaving too many embedding decisions if one of the poles of μ is a cutvertex in G; see Figure 8a.

To circumvent these issues, we show that we can limit the number of embedding decisions involving two-parallel P-nodes of G using connectivity information of the union graph, allowing us to brute-force all suitable embeddings of each shared component in FPT-time. Subsequently, we can determine whether $G^\cup$ admits a SEFE with the given embeddings using the algorithm by Bläsius and Rutter [7] that solves SEFE if every shared component has a fixed embedding. In other words, we use the bounded search tree technique where the branches of the search tree correspond to embedding decisions in the shared graph. We first branch for all possible embeddings of the blocks in G, and subsequently branch for all possible configurations of blocks around cutvertices.

Theorem 9 (⋆). *When k is part of the input,* SEFE *is FPT with respect to the maximum number of degree-1 neighbors Δ_1 and the vertex cover number φ of the shared graph and can be solved in*

$$O(2^{O(\varphi)} \cdot (2(\varphi+\Delta_1))^{(\varphi+\Delta_1)^2 \cdot 3\varphi} \cdot ((\varphi+\Delta_1)!)^{3\varphi} \cdot n^{O(1)}) \text{ time.}$$

Proof Sketch. We give a high-level overview of our strategy. For details, we refer to the full version [11].

"Parameterized Complexity of Simultaneous Planarity" Fink et al.

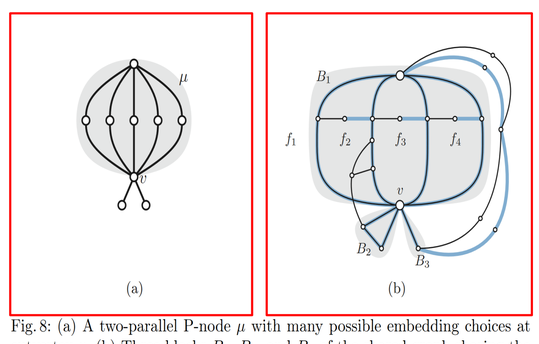
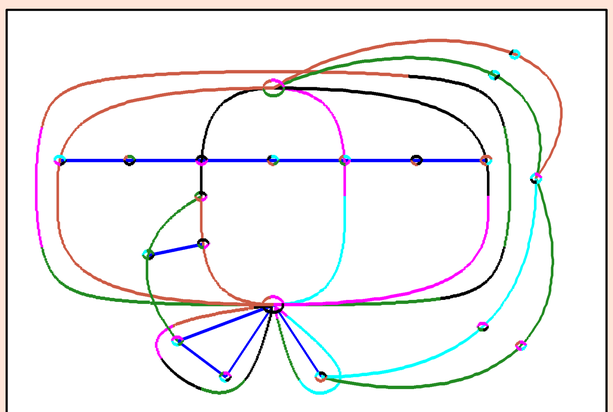


Fig. 8: (a) A two-parallel P-node μ with many possible embedding choices at cutvertex v. (b) Three blocks B_1, B_2, and B_3 of the shared graph sharing the same cutvertex v. Block B_2 is a binary block (faces f_2 and f_3) with respect to B_1, while B_3 is a mutable block that can be embedded in all faces.

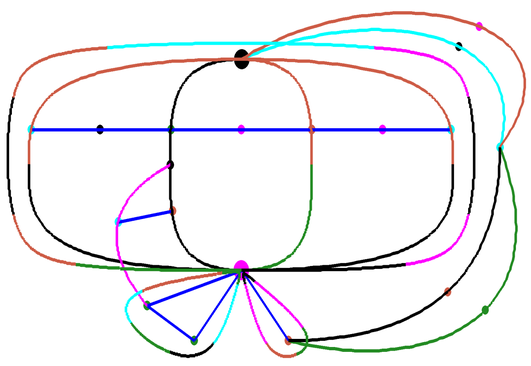
PDF extraction:

- Clustering to seperate figures
- Keyword-based caption detection



Figures consisting of:

- Bézier curves
- Line segments
- Rectangles

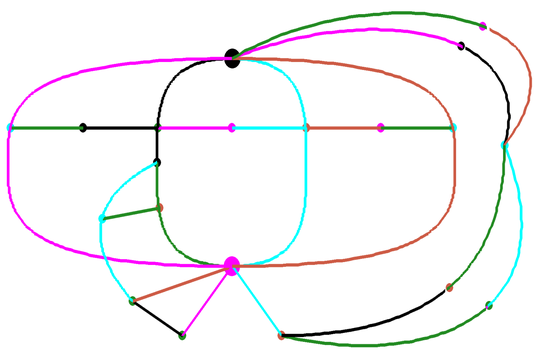


Vertex Candidates:

- Circles
- Rectangles

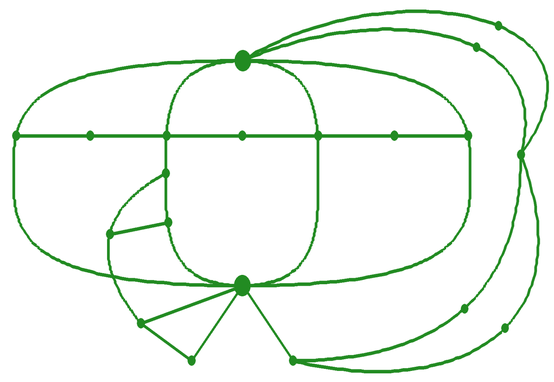
Edge Candidates:

- Bézier curves
- Line segments



Edge detection:

Any path of connected edge candidates that connects two vertex candidates.

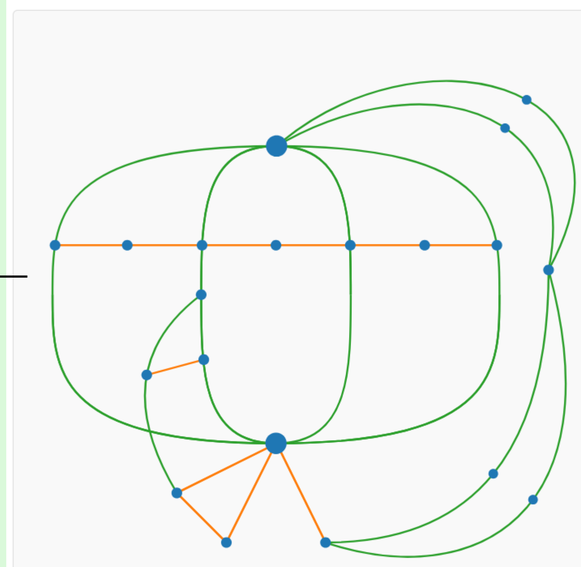


graph6 encoding:

- Separate connected subgraphs
- Compute adjacency matrix
- Compute graph6 encoding

Results

Extracted graph drawing



Original figure for reference and verification

graph6 encoding

Simply add/edit graph via link to House of Graphs