Heuristic Picker for Book Drawings

Jonathan Klawitter and Tamara Mchedlidze

Institut für Theoretische Informatik, Karlsruhe Institute of Technology, Germany
`jo.klawitter@gmail.com`, `mched@iti.uka.de`

Introduction. A *book* with k pages consists of a line (the *spine*) and k half-planes (the *pages*), each with the spine as boundary. In a *k-page book drawing* of a graph the vertices lie on the spine, and each edge is drawn as a circular arc in one of the k pages. The minimum number of edge crossings in a k-page book drawing of a graph is called its *k-page crossing number*, which, in general, is $\mathcal{NP}$-hard to determine [1]. Multiple heuristic approaches to compute a k-page drawing with a small number of crossings are available in the literature. On a very high level, they can be categorized as *simple* heuristics, those that consist of a single run, and *complex* ones, based on neural networks [8, 13, 16, 6], simulated annealing and evolutionary techniques [5, 9, 14, 2, 15]. Notice that a book drawing consists of two ingredients, an order of the vertices on the spine, and a distribution of the edges to the pages. Simple heuristics, given in literature, create vertex order and edge distribution independently. A complete book drawing is constructed by either applying a combination of a vertex order and an edge distribution heuristic, or by applying the mentioned above complex approaches that use simple heuristics as basis. As a result, the performance of the complex approaches depends on the performance of the applied simple heuristics. Up to our knowledge, every attempt to compare the performance of the existing simple heuristics is limited in some sense. These experiments are either limited to very few of them [15], or use very specific graph classes as benchmarks, or limit the experiments to one or two pages [11, 10, 6]. One of the goals of this work is to extend these experiments. Observe that complex heuristics, using advanced search patterns, almost always outperform the simple heuristics [15, 6]. But, since they use simple heuristics in their base, we believe that it is necessary to understand the relevant performance of the simple heuristics first. Thus, this work focuses on simple heuristics only. In particular the content of the poster is as follows: 1. We present several new heuristics and among them several *full drawing* heuristics, that create vertex order and edge distribution at the same time. 2. We present results of our extensive experimental study. The general target of the experiment was to provide an easy way to access the following information: given a graph class and the number of pages, which is the best combination of simple vertex order and edge distribution heuristics? Based on the experiments in the literature, our experimentation and intuition, we have chosen the most promising 7 heuristics from the literature, implemented them, as well as the new heuristics, and compared their performance based on the number of crossings they produce and the running time (complete experiment can be found in [12]).

Heuristics. We start with the vertex order heuristics. The heuristics `randDFS`

[2] and `smlDgrDFS` [7] compute a vertex order based on a DFS traversal choosing the next vertex randomly and the one with the smallest degree, respectively. We introduce the heuristic `treeBFS`, which orders the vertices based on a crossing-free 1-page book drawing of a computed BFS spanning tree. The heuristic `conCro` [3] at each step selects the vertex with the most already placed neighbors and places it on one of the two ends of the current spine where it introduces the fewest new crossings. As an extension of this heuristic, we introduce `conGreedy` which considers not only the two ends of the spine but any position on it.

The edge distribution heuristics `eLen` [4], `ceilFloor` and `circ` [15] sort edges in some particular order and distribute them greedily to the page where they create fewest crossings. `eLen` and `ceilFloor` sort them by decreasing length in linear and circular spine, respectively. `circ` considers the order inspired by the construction of the book embeddings of complete graphs on their pagenumber. `slope` [11] considers a circular drawing and places the edges with similar slope to the same page. We introduce the heuristic `earDecomp`, which constructs the conflict graph of the edges in a circular drawing, and an ear decomposition of the conflict graph, and then alternates the vertices of each ear (edges of the original graph) between two or three pages.

Following the idea by He at al. [10], we extended the vertex order heuristics `randDFS`, `smlDgrDFS`, `conGreedy` to full heuristics `randDFS+`, `smlDgrDFS+`, `conGreedy+`, respectively, which distribute an edge to the best page greedily as soon as it gets closed, i.e. at the moment its second end-vertex appears on the spine. In contrast to `smlDgrDFS+` and `randDFS+`, `conGreedy+` decides for the position of a vertex based on the number of new crossings, and thus the order it computes is different from `conGreedy`. Thus, `conGreedy+` can also be used as an improved vertex order heuristic by discarding the edge distribution.

Experiment and Discussion. We tested the heuristics on graphs of different classes, size n and number of pages p. Among others our test suite includes random graphs of different densities, planar and 1-planar graphs, k-trees, cycle products and hypercubes. In each case we used 200 instances and measured the average number of crossings. A digest is given in the poster. The maximal used number of pages was determined either by the pagenumber of the graph, or when the best heuristic produced no more than 1 crossing on average.

From our experiments we concluded that the best heuristic combination depends not only on the density of the graphs, but remarkably also on the structural properties of the graphs. For example, the combination `conGreedy+-ceilFloor` performs best on planar and 1-planar graphs, while `conGreedy+` , as full drawing heuristic, performs best on random graphs with the same density.

In general, we observe that the extension of `conCro` to `conGreedy` as well as the full drawing heuristic `conGreedy+` often construct book drawings with fewer crossings, however, with the cost of higher running time, which was also clearly noticeable in the experiments.

Furthermore, we could observe that `conGreedy+-ceilFloor`/ `eLen` achieved crossing-free book drawings of hypercubes Q_d when $p =$ pagenumber (tested up to $d = 10$).

References

1. Bannister, M.J., Eppstein, D.: Crossing Minimization for 1-page and 2-page Drawings of Graphs with Bounded Treewidth, pp. 210–221. Springer Berlin Heidelberg, Berlin, Heidelberg (2014), http://dx.doi.org/10.1007/978-3-662-45803-7_18
2. Bansal, R., Srivastava, K., Varshney, K., Sharma, N.: An evolutionary algorithm for the 2-page crossing number problem. In: 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence). pp. 1095–1102 (June 2008)
3. Baur, M., Brandes, U.: Crossing Reduction in Circular Layouts, pp. 332–343. Springer Berlin Heidelberg, Berlin, Heidelberg (2005), http://dx.doi.org/10.1007/978-3-540-30559-0_28
4. Cimikowski, R.: Algorithms for the fixed linear crossing number problem. Discrete Applied Mathematics 122(1), 93–115 (2002)
5. He, H., Newton, M., Sýkora, O.: Genetic algorithms for bipartite and outerplanar graph drawings are best! (2005)
6. He, H., Sălăgean, A., Mäkinen, E., Vrt'o, I.: Various heuristic algorithms to minimise the two-page crossing numbers of graphs. Open Computer Science 5(1) (August 2015)
7. He, H., Sýkora, O.: New circular drawing algorithms. Proceedings of the Workshop on Information Technologies - Applications and Theory (ITAT) (2004)
8. He, H., Sýkora, O., Mäkinen, E.: An improved neural network model for the two-page crossing number problem. IEEE Transactions on Neural Networks 17(6), 1642–1646 (Nov 2006)
9. He, H., Sýkora, O., Mäkinen, E.: Genetic algorithms for the 2-page book drawing problem of graphs. Journal of Heuristics 13(1), 77–93 (2007), http://dx.doi.org/10.1007/s10732-006-9000-4
10. He, H., Sýkora, O., Salagean, A., Vrt'o, I.: Heuristic crossing minimisation algorithms for the two-page drawing problem (2006)
11. He, H., Sýkora, O., Vrt'o, I.: Crossing minimisation heuristics for 2-page drawings. Electronic Notes in Discrete Mathematics 22, 527 – 534 (2005), http://www.sciencedirect.com/science/article/pii/S1571065305052637, 7th International Colloquium on Graph Theory
12. Klawitter, J.: Algorithms for crossing minimisation in book drawings. Master's thesis, Karlsruhe Institute of Technology (2016), https://i11www.iti.uni-karlsruhe.de/en/members/tamara_mchedlidze/supervised_theses#bachelormaster_theses
13. López-Rodríguez, D., Mérida-Casermeiro, E., Ortíz-de Lazcano-Lobato, J.M., Galán-Marín, G.: K-Pages Graph Drawing with Multivalued Neural Networks, pp. 816–825. Springer Berlin Heidelberg, Berlin, Heidelberg (2007), http://dx.doi.org/10.1007/978-3-540-74695-9_84
14. Poranen, T., Mäkinen, E., He, H.: A simulated annealing algorithm for the 2-page crossing number problem. In: Proceedings of International Network Optimization Conference (INOC) (2007)
15. Satsangi, D., Srivastava, K., Srivastava, G.: $$k$$-page crossing number minimization problem: An evaluation of heuristics and its solution using gesakp. Memetic Computing 5(4), 255–274 (2013), http://dx.doi.org/10.1007/s12293-013-0115-5
16. Wang, J.: Hopfield neural network based on estimation of distribution for two-page crossing number problem. Circuits and Systems II: Express Briefs, IEEE Transactions on 55(8), 797–801 (Aug 2008)