Amortised Analysis of Dynamic Data Structures

Rotenberg, Eva

Published in:
Proceedings of 40th International Symposium on Theoretical Aspects of Computer Science

Link to article, DOI:
10.4230/LIPIcs.STACS.2023.2

Publication date:
2023

Document Version
Publisher's PDF, also known as Version of record

Amortised Analysis of Dynamic Data Structures

Eva Rotenberg
Technical University of Denmark, Lyngby, Denmark

Abstract

In dynamic data structures, one is interested in efficiently facilitating queries to a data set, while being able to efficiently perform updates as the data set undergoes changes. Often, relaxing the efficiency measure to the amortised setting allows for simpler algorithms. A well-known example of a data structure with amortised guarantees is the splay tree by Sleator and Tarjan [14].

Similarly, in data structures for dynamic graphs, one is interested in efficiently maintaining some information about the graph, or facilitating queries, as the graph undergoes changes in the form of insertion and deletion of edges. Examples of such information include connectivity, planarity, and approximate sparsity of the graph: is the graph presently connected? Is it planar? Has its arboricity grossly exceeded some specified number $\widetilde{\alpha}$? The related queries could be: is a connected to b? Are the edges uv and uw consecutive in the ordering around u in its current planar embedding? Or, report the $O(\alpha)$ out-edges of vertex x.

In this talk, we will see Brodal and Fagerberg's amortised algorithm for orienting sparse graphs (i.e. of arboricity $\leq \alpha$), so that each vertex has $O(\alpha)$ out-edges [2]. The algorithm itself is extremely simple, and uses an elegant amortised argument in its analysis. Then, we will visit the problem of dynamic planarity testing: is the graph presently planar? Here, we will see an elegant amortised reduction to the seemingly easier problem, where planarity-violating edges may be detected and rejected [5]. We will see a sketch of how the current state-of-the-art algorithm for efficient planarity testing [8] uses ideas similar to those in [2] to analyse the behaviour of a greedy algorithm via a possibly inefficient algorithm with provably low recourse [9]. If time permits, we will touch upon a recent simple amortised data structure for maintaining information in dynamic forests [10], which builds on ideas from splay trees.

The talk concludes with some open questions in the area.

2012 ACM Subject Classification Theory of computation → Data structures design and analysis

Keywords and phrases Amortised analysis, splaying, dynamic graphs, planarity testing

Digital Object Identifier 10.4230/LIPIcs.STACS.2023.2

Category Invited Talk

Funding *Eva Rotenberg*: Partially supported by the Independent Research Fund Denmark grant 2020-2023 (9131-00044B) "Dynamic Network Analysis", the VILLUM Foundation grant 37507 "Efficient Recomputations for Changeful Problems", and the Carlsberg Young Researcher Award CF21-0302 "Graph Algorithms with Geometric Applications".

References

1. Stephen Alstrup, Jacob Holm, Kristian De Lichtenberg, and Mikkel Thorup. Maintaining information in fully dynamic trees with top trees. *ACM Trans. Alg.*, 1(2):243–264, 2005.
2. Gerth Stølting Brodal and Rolf Fagerberg. Dynamic representation of sparse graphs. In *WADS '99*, volume 1663 of *Lecture Notes in Computer Science*, pages 342–351. Springer, 1999. doi:10.1007/3-540-48447-7_34.
3. Richard Cole. On the dynamic finger conjecture for splay trees. part II: the proof. *SIAM J. Comput.*, 30(1):44–85, 2000. doi:10.1137/S009753979732699X.
4. Richard Cole, Bud Mishra, Jeanette P. Schmidt, and Alan Siegel. On the dynamic finger conjecture for splay trees. part I: splay sorting log n-block sequences. *SIAM J. Comput.*, 30(1):1–43, 2000. doi:10.1137/S0097539797326988.

5 David Eppstein, Zvi Galil, Giuseppe F. Italiano, and Thomas H. Spencer. Separator based sparsification: I. planarity testing and minimum spanning trees. *Journal of Computer and Systems Sciences*, 52(1):3–27, February 1996. doi:10.1006/jcss.1996.0002.

6 Zvi Galil, Giuseppe F. Italiano, and Neil Sarnak. Fully dynamic planarity testing with applications. *Journal of the ACM*, 46(1):28–91, 1999. doi:10.1145/300515.300517.

7 Jacob Holm, Kristian de Lichtenberg, and Mikkel Thorup. Poly-logarithmic deterministic fully-dynamic algorithms for connectivity, minimum spanning tree, 2-edge, and biconnectivity. *J. ACM*, 48(4):723–760, 2001. doi:10.1145/502090.502095.

8 Jacob Holm and Eva Rotenberg. Fully-Dynamic Planarity Testing in Polylogarithmic Time. In *STOC 2020*, pages 167–180. ACM, 2020. doi:10.1145/3357713.3384249.

9 Jacob Holm and Eva Rotenberg. Worst-case polylog incremental spqr-trees: Embeddings, planarity, and triconnectivity. In *SODA 2020*, pages 2378–2397. SIAM, 2020. doi:10.1137/1.9781611975994.146.

10 Jacob Holm, Eva Rotenberg, and Alice Ryhl. Splay Top Trees. In *SOSA '23*. SIAM, 2023.

11 John E. Hopcroft and Robert Endre Tarjan. Efficient planarity testing. *Journal of the ACM*, 21(4):549–568, 1974. doi:10.1145/321850.321852.

12 Johannes A. La Poutré. Alpha-algorithms for incremental planarity testing (preliminary version). In *STOC '94*, pages 706–715, 1994. doi:10.1145/195058.195439.

13 Daniel D. Sleator and Robert Endre Tarjan. A data structure for dynamic trees. *Journal of Computer and System Sciences*, 26(3):362–391, 1983. doi:10.1016/0022-0000(83)90006-5.

14 Daniel D. Sleator and Robert E. Tarjan. Self-Adjusting Binary Search Trees. *J. ACM*, 32(3):652–686, 1985. doi:10.1145/3828.3835.