A Critical Review of a FAST MAX FLOW ALGORITHM, James B. Orlin, MIT, jorlin@mit.edu; Xiao-Yue Gong MIT, xygong@mit.edu

Summary

in 2019, James Berger Orlin is an American operations researcher, the Edward Pennell Brooks Professor in Management and Professor of Operations Research at the MIT Sloan School of Management, and Xiao-Yue Gong a PhD student at the MIT Operations Research Center, published the paper of A fast max flow algorithm to compare between the complexity of different types of algorithms and present a new fast max flow algorithm that runs in O ($\frac{nmlogn}{loglogn+log\frac{m}{n}}$).

The paper contributions

1. Present a simple variant of the stack-scaling algorithm in which there are no stacks as the Large-Medium Excess-Scaling (LMES) Algorithm.
2. giving a new and simpler proof of Orlin's Contraction Lemma which was used to develop an O (nm) max flow algorithm.
3. In the modified version of the LMES algorithm (called the Enhanced LMES Algorithm), they permit slightly negative ode excesses. When the negative excess of a node v reaches a threshold value, then node v is added to the flow-return forest," which is a data structure designed for the Enhanced LMES.
4. Phases, flow is sent to node v, after which e (v)>= 0.
5. The new algorithm achieves its improved running time without relying on the dynamic tree data structure.
6. The contraction lemma and its implications were incredibly interesting and I believe can help in future researches.

Strengths

1. the Strengths of the paper are they wrote the Properties of Flow Return Forest with details and pseudocode of each function which are push, pull, add, Reverse_DFS, Delete and Recursive_Delete_and_Merge, their data structure, and their running time.
2. It also goes into great depth with an excellent explanation of how the running time was deduced.
3. The boost in the running time is substantial compared to other algorithms

Weaknesses

1. The pseudocode is very abstract.
2. It was a bit tricky to keep track of all the variables which he introduced in the paper.
3. It didn’t have any real-time performance analysis.
4. The contractions were vaguely described and any pseudocode for it would have helped.

We have learned a lot from this paper on how to compare the complexity of the different algorithms and how to implement a new algorithm that improves the running time.

References

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. Network Flows. Theory, Algorithms, and Applications. Prentice Hall, Englewood Cli_s, NJ, 1993.

[2] R. K. Ahuja and J. B. Orlin. A fast and simple algorithm for the maximum ow problem. Operations Research, 37:748{759, 1989.

[3] R. K. Ahuja, J. B. Orlin, and R. E. Tarjan. Improved time bounds for the maximum ow problem. SIAM Journal on Computing, 18:939{954, 1989.

[4] J. Cheriyan and T. Hagerup. A randomized maximum-ow algorithm. 30th Annual Symposium on Foundations of Computer Science, 118{123, 1989.

[5] E. A. Dinic Algorithm for the solution of a problem of maximum ow in networks with power estimation. Soviet Mathematics Doklady, 11:1277{280, 1970.
[6] J.Edmonds and R. M. Karp. Theoretical improvements in algorithmic e_ciency for network ow problems. JACM, 19:248{264, 1972.
[7] L. R. Ford and D. R. Fulkerson. Maximal ow through a network. Canadian Journal of Mathematics, 8:399{404, 1956.
[8] H. N. Gabow Scaling algorithms for network problems. 24th Annual Symposium on Foundations of Computer Science, 248{258, 1983.
[9] A. V. Goldberg and S. Rao. Beyond the ow decomposition barrier. Journal of the ACM, 45:783{797, 1998.
[10] A. V. Goldberg and R. E. Tarjan. A new approach to the maximum-ow problem. J. ACM, 35:921{940, 1988.
[11] D. S. Hochbaum. The pseudoow algorithm: a new algorithm for the maximum-ow problem. Operations Research, 56:992{1009, 2008.
[12] Y. T. Lee and A. Sidford, Path _nding methods for linear programming: solving linear programs in ~O (
p
rank)
iterations and faster algorithms for maximum ow 55th Annual Symposium on Foundations of Computer Science 424{433, 2014.
[13] A. V. Karzanov. Determining the max ow in a network by the method of preows. Soviet Mathematics Doklady, 15:435{437, 1974.
[14] V. King, S. Rao, and R. Tarjan. A faster deterministic maximum ow algorithm. J. Algorithms, 23:447{474, 1994.
[15] J. B. Orlin. A faster strongly polynomial minimum cost ow algorithm. Operations Research, 41:338{350, 1993.
[16] J. B. Orlin. Max flow in O(nm) Time, or Better Proceedings of the Forty-_fth Annual ACM Symposium on Theory of Computing, 765{774, 2013.
[17] Y.Shiloach and U. Vishkin. An O(n2 log n) parallel max ow algorithm. Journal of Algorithms, 3:128{146, 1982.
[18] D. D. Sleator and R. E. Tarjan. A data structure for dynamic trees. J. Computer and System Sciences, 24:362{391, 1983.
[19] E Tardos. A strongly polynomial algorithm to solve combinatorial linear programs. Operations Research, 34:250{256, 1986.