Journal of Scientific and Practical Computing
Vol. 2, No. 2 (2008) 51–51

SPIELMAN-TENG'S SMOOTHED ALGORITHM ANALYSIS AND LINEAR PROGRAMMING

ACM's Special Interest Group on Algorithms and Computing Theory (SIGACT) has presented the 2008 Gödel Prize to Daniel A. Spielman and Shang-Hua Teng for their outstanding contribution in developing "a rigorous framework to explain the practical success of algorithms on real data and real computers that could not be clearly understood through traditional techniques." This technique is called Smoothed Analysis.

When an algorithm is designed for a specific problem, how good the algorithm is depends on both its worst-case performance and also its average performance. For example, Quick-sort is a good algorithm whose worst-case time cost in computing is $O(n^2)$, but its average time cost is $O(n \log n)$. However, understanding a good algorithm with the worst-case exponential time cost is not easy. The simplex algorithm for linear programming discovered and developed by George Dantzig in 1947 is such an example. Over the years, the simplex method is proven as a very important and practical algorithm, but its time complexity is not polynomial. Polynomial time algorithms for linear programming were found by LG Khachiian in 1979 and N. Karmarkar in 1984, respectively. Khachiian's breakthrough answered the question of whether linear programming is an NP-hard problem. Karmarkar's algorithm reduced the time complexity from $O(n^6)$ to $O(n^{3+})$. $O(n^3)$ is still not fast enough for a problem with a large number of input variables. Therefore, the simplex method seems still to be the best in the industry.

Spielman-Teng's paper "Smoothed Analysis of Algorithms: Why the Simplex Algorithm Usually Takes Polynomial Time" explains why the Simplex Algorithm "works effectively in many practical areas especially in business." Spielman-Teng's paper introduces a method that analyzes the "local" average time spent on a problem. The word "local" refers to the structure of local instances or the inputs in a neighborhood of a problem. Some researchers are working on using the Smoothed Algorithm Analysis method to tread an NP-hard optimization problem.

References

Spielman, Daniel and Teng, Shang-Hua (2001), "Smoothed analysis of algorithms: why the simplex algorithm usually takes polynomial time", Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing, ACM, pp. 296–305.