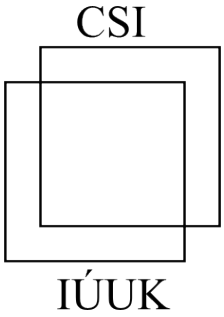
CHARLES UNIVERSITY



Mgr. Martin Koutecký, Ph.D.
Charles University
Computer Science Institute
Malostranské nám. 25, 3rd floor, room #326
118 00 Praha 1, Czech Republic
Telephone: +420 774 853 316
Email: koutecky@iuuk.mff.cuni.cz

Review of the Bachelor Thesis "Experimental Analysis of Scaling Methods for LP" by Jakub Komárek

Linear Programming is a fundamental problem in the theory of optimization, and its solvers play a major role in industry, not only in the solution of actual linear programs, but also a subroutine when solving integer linear programs and other optimization tasks. Dadush et al.[1] has recently designed an algorithm which rescales the columns of the constraint matrix in order to improve a certain condition number of this matrix, which theoretically results in making the problem easier for certain interior point methods. The author has implemented this algorithm and evaluated whether the computed rescaling makes it easier for various LP solvers to solve many real-world benchmark instances.

I wish to point out that reading and understanding the algorithm of Dadush et al. requires significant effort. The student needed to grasp concepts from linear algebra and matroid theory. Moreover, as always, the implementation is not straightforward. There are issues with floating point arithmetics, using sparse operations etc. In fact, the student discovered and fixed several bugs in high-profile open source packages (`scikit-glpk`, `sage`).

Judging the obtained results is tricky. From the start one has to mention that computing the rescaling is absolutely impractically slow at this point, so all of the following discussion should be understood to try to answer the question, "does the rescaling help?"; if the answer to this question is positive, it would mean that perhaps putting in the engineering effort to make the rescaling algorithm practical is worth it. The pessimist view of the results is that there is no single method which consistently and significantly benefits from the rescaling. The optimist view is that for a majority of instances, at least one solver benefited from the rescaling; it is not entirely easy to read Figure 1.1, but my estimate is that for at least a half of instances, the speedup was at least 10%. (Admittedly, this number, too, should be taken with a grain of salt, because the tested instances were relatively small.)

A helpful perspective was given by Bixby in his review of the progress of LP solvers between 1987 and 2002. Bixby categorizes individual improvements into four quadrants, according to two dimensions: how "large" and how "wide" was the speed-up. There were a handful powerful ideas which give speed-ups both significant and working on a majority of instances. But the remainder of progress can be attributed to ideas which are either relatively minor in terms of performance but applicable across a wide range of instances, or have a significant performance benefit but only on relatively narrow instance classes. With this in mind, I am cautiously optimistic when considering the obtained results and believe there may be practical potential in the theoretical

[1] Daniel Dadush, Sophie Huiberts, Bento Natura, László A. Végh. *"A scaling-invariant algorithm for linear pro- gramming whose running time depends only on the constraint matrix"*, STOC 2020.

ideas underpinning the rescaling algorithm.

On the formal side, the thesis is written very well – it introduces the relevant literature in appropriate detail, it is well organized and polished, the writing is overall clear and pleasant. Also, the presentation of the results uses helpful figures and is fairly easy to understand.

In summary, the thesis at hand provides an original contribution at a level (both formal and content-wise) comparable to good international venues, and deserves to be accepted with a grade of 1.

Sincerely,
Martin Koutecký