Posudek bakalářské práce

Matematicko-fyzikální fakulta Univerzity Karlovy

Thesis author	Jakub Komárek
Thesis title	Experimental Analysis of Scaling Methods for LP
Submitted	2023
Program	Computer Science **Specialization** Discrete Models and Algorithms

Review author	László Végh **Role** reviewer
Position	London School of Economics and Political Science

Review text:

The dissertation gives a thorough computational study of the effects of a certain preprocessing step for solving linear programs.

The preprocessing step amounts to finding a rescaling of the columns of an LP in standard equality form to approximately optimize the 'circuit imbalance measure' κ. This was proposed in a 2020 paper by Dadush, Huiberts, Natura, and myself in the context of layered least squares interior point methods (LLS IPMs). LLS IPMs are relevant in the context of strongly polynomial computability: the running time of such algorithms only depends on $\log \kappa$.

The preprocessing step enables to find a rescaling such that $\log \kappa$ becomes within a constant factor of the best achievable value, yielding asymptotically to the best performance of this algorithm. It comprises two main parts: finding circuits containing every two columns if one exists, and then performing a minimum mean cycle computation. While the theoretical guarantees only apply for the particular LLS IPMs, it is conceivable that the rescaling yields to better conditioning of the matrices. and thus also improves on other IPMs.

The thesis gives a clear and concise presentation of the theoretical background on LPs, matroids, and circuit imbalances. and gives a thorough and mature computational study using standard LP test instances and experiments with a range of standard LP solvers. While the experiments do not contain any LLS IPM (only standard IPM implementations), this is understandable as these are very complicated algorithms and I am not aware of any implementation since the first such algorithm was published by Vavasis and Ye in 1996. The conclusion is that the even though the rescaling yields improved running times in some instances. the performance deteriorates in other cases; overall, the effect is neutral at best.

The main contribution of the thesis is the computational part. The algorithm description in our paper is high level and theoretical, and it is quite challenging to translate it into a working implementation. In particular, numerical instability becomes a major issue and one cannot rely on floating point arithmetics. This is discussed in detail, and exact arithmetics is used after considering also other options. There were also implementational challenges such as being able to appropriately read the test instances; the bug fix was added to the original codebase.

The linear algebra operations also constituted nontrivial challenges. Implementations were also given that exploit the sparse input structure.

While the main contribution of the thesis is computational, the theoretical parts are also presented clearly and in a rigorous mathematical style. There are proofs included of some technical statements. Some of these are somewhat overcomplicated, such as the lemmas in Sec 3.1.2 on the effect of two simple transformations.

The presentation is excellent. It is difficult to give a good high level presentation and explanation of implementation work, and the thesis does a very good job on this account. The presentation of the results is also clear and transparent. The thesis is also very well structured. One small presentational issue: the bibliography should give the full list of authors of the papers rather than using "et al.".

Overall, this is an excellent thesis and I recommend the highest mark (1).

Questions

1. The tables compare the running times of the LP algorithms before and after rescaling. Could you please also indicate the running times of the preprocessing? I expect this will be much larger than the LP running time. (But perhaps could be improved using a lower level implementation and paralellization).

2. Standard primal-dual interior point methods used e.g. in GLPK are scaling invariant: their performance should remain the same under any column scaling. (The Vavasis–Ye and other LLS IPMs are exceptions here.) So, I am surprised why the scaling changes the running time. Do you have an explanation for this? Could this be due to some other preprocessing done in the solver?

3. As a benchmark, one could also compare the running time with this rescaling to the running time of a random rescaling; I expect a random rescaling could perform often better.

4. Even though κ is very hard to approximate, it would be interesting to see how much the rescaling algorithm changes it. It would be good to list the largest changes in the ratios of the columns, or e.g., show by how much the max. ratios between elements of circuits found by the algorithm change under the rescaling.

I recommend the thesis for defense.

I suggest to not consider the thesis for the annual award.

date

Signature: