QCQP Project Plan

Chuwen Zhang

April 5, 2021

1 QCQP Project Plan

1.1 Scope of our work

Develop a QCQP solver that uses SDP-relaxation-and-refinement approach. The QCQP solver should be problem-independent that works for any QCQP instance.

1.1.1 Modeling interface

The modeling interface is domain specific language, a simple tool for user to define a **QCQP** problem, then the solver translates into canonical form of QCQP. For example, for a HQCQP, canonical form includes parameters Q, $A_i, b_i, \forall i$

To-dos:

- The modeling part does the canonicalization, works like cvxpy, yalmip, etc.
- Directly use COPT.

See Agrawal et al. (2018), Diamond and Boyd (2016), Dunning, Huchette, and Lubin (2017), Lofberg (2004)

1.1.2 SDP interface

The SDP interface should be solver **independent**. SDP interface starts with canonical form to create a SDP-relaxation. So the users do not have to derive SDP by themselves. The interface should output a SDP problem in a standard format, e.g., SDPA format, that can be accepted by any SDP solver.

To-dos including:

- starts with canonical form.
- interface with solver: create problems, extract solutions, status, etc.

1.1.3 Local Refinement

Local Refinement from SDP solution to QP seems to be problem dependent, whereas we can start with:

- Use Gurobi to do the refinement
- use existing methods, including residual minimization (SNL), randomization (for BQP), see Luo et al. (2010) and papers for SNL.
- add an **option** for user to choose a refinement method.

1.2 Computational tests

Test on SNL, kissing problem, etc.

1.3 Development plan

- start with Pure Python or Julia interface as a fast prototype.
  - in Python one can use cvxpy or other AMLs; in Julia one may use JuMP.
  - computational tests on kissing problem, SNL using different SDP solver can be handled at the same time
- add and move to C/C++ interface. does same thing as Python, then this backend with replace the pure Python one.
- Add Python, Matlab support for C interface.

Reference

Agrawal, Akshay, Robin Verschueren, Steven Diamond, and Stephen Boyd. 2018. "A Rewriting System for Convex Optimization Problems." *Journal of Control and Decision* 5 (1): 42–60.

Diamond, Steven, and Stephen Boyd. 2016. "CVXPY: A Python-Embedded Modeling Language for Convex Optimization." *The Journal of Machine Learning Research* 17 (1): 2909–13.

Dunning, Iain, Joey Huchette, and Miles Lubin. 2017. "JuMP: A Modeling Language for Mathematical Optimization." *SIAM Review* 59 (2): 295–320.

Lofberg, Johan. 2004. "YALMIP: A Toolbox for Modeling and Optimization in MATLAB." In *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No. 04CH37508)*, 284–89. IEEE.

Luo, Zhi-Quan, Wing-Kin Ma, Anthony Man-Cho So, Yinyu Ye, and Shuzhong Zhang. 2010. "Semidefinite Relaxation of Quadratic Optimization Problems." *IEEE Signal Processing Magazine* 27 (3): 20–34.