High Performance Computing in Semidefinite Programming

Dmitriy (Tim) Kunisky [dk3105]

Semidefinite programming (SDP) [11] refers to optimization with respect to matrix variables $\boldsymbol{X}$, which may be subject not only to linear constraints $\langle \boldsymbol{X}, \boldsymbol{A} \rangle = b$ (as in linear programming) but also to positive semidefiniteness constraints $\boldsymbol{X} \succeq \boldsymbol{0}$. This seemingly small change makes it possible to address many new problems. For instance, SDPs can capture stability criteria for dynamical systems leading to applications in control theory, can describe useful regularization techniques in machine learning like *nuclear norm minimization*, and can be used to form convex relaxations of complicated non-convex feasible sets in combinatorial optimization problems.

On the other hand, solving generic SDPs is computationally challenging. The most common family of methods currently, so-called *primal-dual interior point methods*, are iterative methods requiring a linear system to be solved in each step whose matrix is positive definite and whose size is the number of linear constraints in the SDP, which in turn requires a Cholesky decomposition of this size. (The basic idea is to replace the various constraints with "softened" versions by penalizing the objective function to some extent that varies over the course of the algorithm, and then take a step of Newton's method for this proxy problem, which is what gives rise to the large linear systems.) This often turns out to be the bottleneck in practical applications, where there are many linear constraints corresponding to highly structured feasible matrices $\boldsymbol{X}$ having various entrywise symmetries. Thus, the number of linear constraints is often used as a heuristic for the computational cost of an SDP. A sequence of works [12, 2, 1, 3] describes recent progress in designing large-scale parallelized implementations of SDP solvers able to solve problems with millions of linear constraints. This progress is in turn driven by progress in such algorithms for large-scale Cholesky decompositions [9, 8], which are able to take advantage of GPU capabilities.

Most of the software in the work referenced above was run on the TSUBAME series of supercomputers. This series of computers typically ranks highly on the Green500 list (the most recent generation, TSUBAME3.0, is currently in 5th place), possibly due to a focus on cooling technology, but less so on the Top500 list (where TSUBAME3.0 is in 22nd place). On the other hand, its ranking improves for the HPCG (sparse linear algebra) benchmark, under which TSUBAME3.0 is in 15th place. This is compatible with applications like the above, which gain significantly from any speedups in linear algebra operations with sparse data.

One important research problem in this area is to explore whether highly structured problems from specific domains might admit specialized algorithms or optimizations in the standard interior point methods. For instance, some work has used the alternating direction method of multipliers (ADMM) for SDPs with a particular sparsity structure arising in power grid optimization problems [5], and others have used dynamic ways of specializing message-passing implementations of interior point methods to the sparsity in a given problem [6]. One application that is important in practice but where such specializations have not been explored thoroughly is SDPs for polynomial optimization [7]. Small-scale software for this purpose exists (e.g. [10, 4]), but since the size of the SDPs involved grows quickly with the problem size, these tools are only practical for small problems. Specialized distributed SDP solvers in this domain would therefore be very useful and would allow more detailed study of these techniques for problems with many variables or in high dimension.

References

[1] Katsuki Fujisawa, Toshio Endo, Yuichiro Yasui, Hitoshi Sato, Naoki Matsuzawa, Satoshi Matsuoka, and Hayato Waki. Petascale general solver for semidefinite programming problems with over two million constraints. In *Parallel and Distributed Processing Symposium, 2014 IEEE 28th International*, pages 1171–1180. IEEE, 2014.

[2] Katsuki Fujisawa, Hitoshi Sato, Satoshi Matsuoka, Toshio Endo, Makoto Yamashita, and Maho Nakata. High-performance general solver for extremely large-scale semidefinite programming problems. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, page 93. IEEE Computer Society Press, 2012.

[3] Katsuki Fujisawa, Toyotaro Suzumura, Hitoshi Sato, Koji Ueno, Satoshi Imamura, Ryo Mizote, Akira Tanaka, Nozomi Hata, and Toshio Endo. Advanced computing and optimization infrastructure for extremely large-scale graphs on post-peta-scale supercomputers. In *Advanced Software Technologies for Post-Peta Scale Computing*, pages 207–226. Springer, 2019.

[4] Didier Henrion, Jean-Bernard Lasserre, and Johan Löfberg. Gloptipoly 3: moments, optimization and semidefinite programming. *Optimization Methods & Software*, 24(4-5):761–779, 2009.

[5] Ramtin Madani, Abdulrahman Kalbat, and Javad Lavaei. Distributed computation for sparse semidefinite programming with applications to power optimization problems.

[6] Sina Khoshfetrat Pakazad, Anders Hansson, Martin S Andersen, and Anders Rantzer. Distributed semidefinite programming with application to large-scale system analysis. *IEEE Transactions on Automatic Control*, 63(4):1045–1058, 2018.

[7] Pablo A Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical programming*, 96(2):293–320, 2003.

[8] Yuki Tsujita and Toshio Endo. Data driven scheduling approach for the multi-node multi-GPU Cholesky decomposition. In *Job Scheduling Strategies for Parallel Processing*, pages 69–82. Springer, 2015.

[9] Yuki Tsujita, Toshio Endo, and Katsuki Fujisawa. The scalable petascale data-driven approach for the Cholesky factorization with multiple GPUs. In *Proceedings of the First International Workshop on Extreme Scale Programming Models and Middleware*, pages 38–45. ACM, 2015.

[10] Hayato Waki, Sunyoung Kim, Masakazu Kojima, Masakazu Muramatsu, and Hiroshi Sugimoto. Algorithm 883: Sparsepop—a sparse semidefinite programming relaxation of polynomial optimization problems. *ACM Transactions on Mathematical Software (TOMS)*, 35(2):15, 2008.

[11] Henry Wolkowicz, Romesh Saigal, and Lieven Vandenberghe. *Handbook of semidefinite programming: theory, algorithms, and applications*, volume 27. Springer Science & Business Media, 2012.

[12] Makoto Yamashita, Katsuki Fujisawa, and Masakazu Kojima. SDPARA: Semidefinite programming algorithm parallel version. *Parallel Computing*, 29(8):1053–1067, 2003.