Solving stochastic optimization problems using exascale-ready algorithms

Deepak Rajan
Center for Applied and Scientific Computing, Lawrence Livermore National Laboratory
rajan3@llnl.gov

1 Background and Motivation

The DOE mission abounds with complex systems that are modeled as large and difficult-to-solve stochastic optimization problems, where samples (or scenarios) represent the possible realization of uncertainty in the model. Usually modeled as coarse-grained sample-based Mixed-Integer Programs (MIPs) [6], these problems are solved on workstations using generic MIP solvers. Often, the MIP solvers are unable to find the optimal solution in a reasonable amount of time, and the implementation of lower-quality solutions results in increased cost and/or increased risk. Consider the electrical power grid, where the solution to the stochastic version of the unit-commitment problem is used to determine which generators to schedule and at what production levels. In an example from [5], the stochastic nature of the problem derives from the unpredictability in demand and generation from renewable energy sources such as solar and wind; usually a set of plausible samples is forecasted. The time to optimal solution when using CPLEX, a state-of-the-art generic MIP solver, grows exponentially as function of the number of samples considered. Even with just 20 scenarios, this problem takes unacceptably long (close to a day) to solve. A scalable, resilient, extreme-scale sample-based MIP solver will achieve a significant breakthrough in solving difficult stochastic optimization problems of interest to DOE.

There exist many state-of-the-art generic MIP solvers, the best of which are commercial (CPLEX, ExpressMP [13], and Gurobi [4]). These codes are highly optimized in many strategies critical to the performance of MIP solution schemes, representing years of manpower to make these strategies work synergistically. There are also some parallel implementations of non-commercial generic MIP solvers, such as ParaSCIP [11] and PICO [8], which scale to around 10k cores. However many recent studies have shown that, for many problem instances, the additional parallelism does not help much, if at all [7]. As a result, the MIP community has not leveraged the computing power of supercomputing resources. Traditionally, supercomputers have been used to model and simulate physical systems and not to solve stochastic optimization problems, typically solved on workstations.

We believe there is an enormous opportunity in leveraging the full potential of supercomputing resources to revolutionize the field of stochastic optimization. This is because generic MIP solvers do not solve sample-based MIPs efficiently, and because the unique structure of sample-based MIPs can be leveraged to develop specialized parallel algorithms for stochastic optimization problems. An approach that holds promise (see further discussion in Section 2) relies on solving surrogate optimization problems, the Dantzig-Wolfe and Lagrangian relaxations [2], which naturally lend themselves to parallel solution schemes since they are easily decomposed into subproblems that are solved iteratively. In this approach, the available parallelism is limited by the amount of computational resources and not by the number of samples. This is because many relaxations are being solved simultaneously in the Branch and Bound (B&B) scheme used to solve MIPs, with as many decomposed subproblems for a particular Dantzig-Wolfe relaxation as the number of samples.

A decomposition-based scheme that successfully leverages massively parallel infrastructures to solve sample-based MIPs will require innovative advances on two fronts. (1) Theoretical: Many optimization problems will need to be solved to develop new effective strategies in the B&B scheme for sample-based MIPs, while ensuring that these strategies maintain the decomposable problem structure. (2) Algorithmic: Many challenges posed by extending the algorithms to be exascale-ready will need to be addressed. This must include solving problems with large sample sets that are too big to fit on a single core, intelligently interleaving the work from potentially many levels of parallelism, and resiliency from soft non-catastrophic errors that will become more common [10].

2 Related Work

Generic MIP solvers rely on the strength of a surrogate optimization problem, the Linear Programming (LP) relaxation, which tends to be very weak for sample-based MIPs. Furthermore, the size of the mathematical formulation of sample-based MIPs grows rapidly as a function of the number of samples. This often results in problems too big to fit on a single core, which will be exacerbated on exascale infrastructures, with projections of decreasing memory per core. There are a few MIP solvers using decomposition-based relaxations, such as GCG [3], BapCod [12] and Dip [9]. However, none of these are built with parallel implementations in mind, and they do not contain any of the optimized strategies that are critical to the success of generic MIP solvers. Nevertheless, recent experiments [1,3] are promising, since it leads us to believe that an optimized, parallel implementation may be quite effective in solving hard sample-based MIPs in a reasonable time frame.

3 Assessment

Challenges Addressed: An effective decomposition-based parallel algorithm that leverages the computational power of supercomputers to solve sample-based MIPs will require the use of special techniques and novel ideas in many areas of optimization, ranging from MIP theory to task scheduling, and in many areas of parallel programming.

Maturity and Novelty: See discussion in Section 2. We believe that due to their unique structure, stochastic optimization problems can be efficiently solved using the HPC architectures of the future. There is much work to be done to make it happen — a naive implementation of decomposition schemes is known to not work well in practice, and no one has yet built an optimized decomposition scheme geared towards solving the wide class of stochastic optimization problems.

Uniqueness: The challenges of building an effective decomposition based scheme for sample-based MIPs are many, and definitely not unique to exascale systems. However, exascale architectures will bring ever-larger challenges in solving sample-based MIPs using generic MIP schemes. The structure of sample-based MIPs presents a unique opportunity in building an exascale-ready parallel framework for solving stochastic optimization problems.

Applicability: As mentioned before, a scalable sample-based MIP solver in not unique to exascale systems — even current systems will yield immediate benefits. By leveraging HPC architectures intelligently, we believe that stochastic optimization problems can be solved in an order-of-magnitude less wall clock time. In the long term, by designing for exascale and developing algorithms that work best when run in parallel, we will be able to leverage the computational resources that will come online in the next few years.

Effort: By focusing on fundamental research on algorithmic and theoretical issues related to solving sample-based MIPs using parallel algorithms, we can leverage any enhancements (scaling or otherwise) in generic MIP solvers — every future enhancement improves the effectiveness of a sample-based MIP solver. On the other hand, building an exascale-ready algorithm for generic MIPs is much larger in scope and has been the target of a multi-year multiple person effort at Sandia National Labs (PICO), even at much smaller scale.

4 References

[1] Martin Bergner, Marco E. Lübbecke, Enrico Malaguti, and Emiliano Traversi. Partial convexification of general mips by dantzig-wolfe reformulation. In *Integer Programming and Combinatorial Optimization, volume 6655 of Lect. Notes Comput. Sci*, pages 39–51. Springer, 2011.

[2] Claus C. Carøe and Rüdiger Schultz. Dual decomposition in stochastic integer programming. *Operations Research Letters*, 24(12):37 – 45, 1999.

[3] Gerald Gamrath and Marco E. Lübbecke. Experiments with a generic dantzig-wolfe decomposition for integer programs. In Paola Festa, editor, *Experimental Algorithms*, volume 6049 of *Lecture Notes in Computer Science*, pages 239–252. Springer Berlin Heidelberg, 2010.

[4] Gurobi optimizer. `http://www.gurobi.com/`.

[5] California ISO. Integration of renewable resources: transmission and operating issues and recommendations for integrating renewable resources on the california iso-controlled grid, 2007.

[6] Willem K. Klein Haneveld and Maarten H. van der Vlerk. Stochastic integer programming: General models and algorithms. *Ann. Oper. Res.*, 85:39–57, 1999. Stochastic programming. State of the art, 1998 (Vancouver, BC).

[7] Thorsten Koch, Ted Ralphs, and Yuji Shinano. Could we use a million cores to solve an integer program? *Mathematical Methods of Operations Research*, 76:67–93, 2012.

[8] Cynthia A. Phillips, Jonathan Eckstein, and William Hart. *Massively Parallel Mixed-Integer Programming: Algorithms and Applications*, chapter 17, pages 323–340. SIAM Books, 2006.

[9] T. Ralphs and M. Galati. Dip – Decomposition for integer programming, 2009. `https://projects.coin-or.org/Dip`.

[10] Bianca Schroeder, Eduardo Pinheiro, and Wolf-Dietrich Weber. DRAM errors in the wild: a large-scale field study. In *SIGMETRICS '09: Proceedings of the eleventh international joint conference on Measurement and modeling of computer systems*, pages 193–204, New York, NY, USA, 2009. ACM.

[11] Yuji Shinano, Tobias Achterberg, Timo Berthold, Stefan Heinz, and Thorsten Koch. ParaSCIP: A parallel extension of SCIP. In Christian Bischof, Heinz-Gerd Hegering, Wolfgang E. Nagel, and Gabriel Wittum, editors, *Competence in High Performance Computing 2010*, pages 135–148. Springer Berlin Heidelberg, 2012.

[12] F. Vanderbeck. BaPCod – A generic branch-and-price code, 2005. `https://wiki.bordeaux.inria.fr/realopt/pmwiki.php/Project/BaPCod`.

[13] FICO Xpress Optimization Suite. `http://www.fico.com/en/Products/DMTools/Pages/FICO-Xpress-Optimization-Suite.aspx`.