Parallelization Challenges for Scenario-Based Decomposition of Stochastic Programs

Project: Higher Confidence Integer Programming
Project: Risk Management and Planning in Complex Networks

J.P. Watson[1], R.J.B. Wets[2], D.L. Woodruff[3]
[1] Discrete Algorithms and Math Department, Sandia National Laboratories
Mail Stop 1318, P.O. Box 5800, Albuquerque, NM 87185-1318 USA
[2] Department of Mathematics, University of California, Davis
Davis, CA 95616-8633 USA
[3] Graduate School of Management, University of California, Davis
Davis, CA 95616-8609 USA

Abstract

Stochastic programming is a foundational approach to modeling optimization problems with uncertain parameters. Real-world applications abound, in domains including electric grid operations and planning, finance, critical infrastructure protection, and natural resource management. Large-scale, real-world stochastic programs are notoriously difficult to solve, particularly in the presence of discrete decision variables. Typically, direct solution of the full problem – known as the *extensive form* – is not practical in terms of either memory or time requirements. To achieve practically useful results, decomposition strategies have been developed, which have led to dramatic reductions in the required compute times. Yet, given the number of scenarios required to accurately represent uncertainty in large-scale problems, compute times often remain prohibitive.

Intuitively, decomposition strategies such as the L-shaped method and Progressive Hedging are amenable to parallelization; sub-problems can be iteratively solved in parallel, interleaved with a master barrier synchronization. However, there exist a number of unresolved practical issues in developing effective decomposition strategies for large-scale parallel computing environments.

In this talk, we focus on two key issues involved in the parallelization of Rockafellar and Wets' Progressive Hedging scenario-based decomposition algorithm, which has been successfully used to solve stochastic linear, non-linear, and linear mixed-integer stochastic programs.

The first issue relates to the presence of the barrier synchronization in the master algorithm, following scenario sub-problem solves in each iteration of the algorithm. Given high sub-problem solution time variability, particularly in the mixed-integer and non-linear cases, parallel efficiency degrades dramatically as the number of scenarios (or equivalently processors) is increased. We show that relaxation of this barrier synchronization fails to impact the theoretical convergence properties of Progressive Hedging, under mild conditions. We discuss our implementation of and computational experimentation with this asynchronous version of Progressive Hedging.

The second issue relates to the relative balance of communication and computation involved in the solution of scenario sub-problems. Often, individual sub-problems solve quickly, degrading parallel efficiency by inflating the communication-to-computation ratio. We discuss how scenario bundling – the aggregation of multiple sub-problems into a single, new sub-problem – both mitigates this issue and leads to a dramatic reduction in the number of iterations of the master algorithm required for convergence. We discuss alternative strategies for forming bundles (e.g., maximally diverse versus maximally homogeneous), and detail our computational experience.

All of our algorithms and test models are available via Sandia's open-source PySP stochastic programming software library [1].

[1]https://software.sandia.gov/trac/coopr/wiki/PySP