Agnostic MPI-SPPY and Consensus ADMM Under Uncertainty

David L Woodruff [1]
(with coauthors noted as we go along)

INFORMS 2024

[1] Graduate School of Management,
University of California, Davis

Optimization Under Uncertainty

Today we will work with abstract problems such as:

$$\min_{x} h(x, \Xi)$$

- Ξ is a random variable
- The function h captures constraints as well as any data modeled as known with certainty.

But the random variable necessitates some additional specification such as requiring a form of robustness or perhaps...

Optimization Under Uncertainty

Today we will work with abstract problems such as:

$$\min_x h(x, \Xi)$$

- Ξ is a random variable
- The function h captures constraints as well as any data modeled as known with certainty.

But the random variable necessitates some additional specification such as requiring a form of robustness or perhaps...

$$\min_x E_{\xi \sim F} h(x, \xi) \tag{1}$$

Where the distribution F is unknown and, of course, unknownable (exactly).

MPI-sppy: The paper and the software

- B. Knueven, D Mildebrath, C. Muir, JD Siirola, J-P Watson, DL Woodruff, "A parallel hub-and-spoke system for large-scale scenario-based optimization under uncertainty," *MPC*
- Find $\hat{x}$ with bounds and/or confidence intervals on the objective function for a scenario-based T-stage expected value problem with scenario set Ξ.

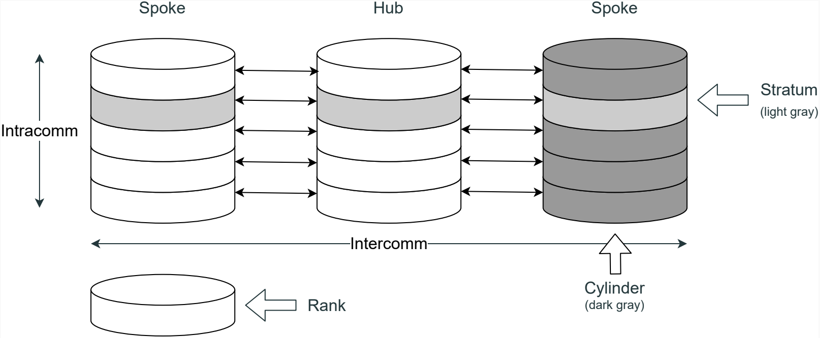
MPI-sppy: The paper and the software

- B. Knueven, D Mildebrath, C. Muir, JD Siirola, J-P Watson, DL Woodruff, "A parallel hub-and-spoke system for large-scale scenario-based optimization under uncertainty," *MPC*
- Find $\hat{x}$ with bounds on the objective function (and confidence intervals) for the problem solved to do that; generally scenario-based, e.g.:

$$\min_x \frac{1}{N} \sum_{i=1}^{N} h(x, D_i).$$

  for some sample D of size N.
- I want to talk mainly about the architecture, but first a few words about the software
  - Available at https://github.com/Pyomo/mpi-sppy
  - It is a library, but we also have a generic program (coming back toward PySP)
  - It is designed for HPC, but does run on a laptop.

The Architecture



But things are changing

Agnostic to the AML

For scenario based decomposition...

- Loose:
  - You code your AML to write an MPS (or maybe lp) file for each scenario along with a json file for each scenario that lists the nonanticipative variabes for each node in the scenario tree traversed by the scenario.
  - You do this once and let mpi-sppy take over.
  - No Python programming required (unless, of course, thats how interact with your AML)

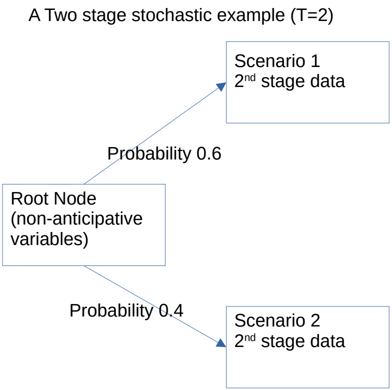
- Tight: If your "AML" is callable in the sense that an outside caller can modify the objective, then
  - You hope we already have added support for your "AML" (we now have support for AMPL, GAMS, and GurobiPy)
  - You need to write a thin wrapper in Python for your model

Consensus ADMM Under Uncertainty

- There is a paper with Aymeric Legros on OOL, but write to me for a somewhat better version.
- Today, I will give a brief overview, with almost no notation.
  - You might want to do scenario decomposition for stochastics and you might want to do consensus ADMM decomposition because you have a huge problem (or you might be decomposing just to get parallel speed-up or for security reasons).
  - We combine the two. Under the hood, the trick is the tree.
  - But the interface is that you tell the software about your scenario tree for stochastics and about your consensus variables and subproblems for ADMM using wrappers for your model.
- The software is available on github as part of MPI-SPPY.

A Simple Example

- Consider a batch production/distribution problem with uncertain production yields
- Batch sizes must be non-anticipative, while shipping quantities, inventory etc. can depend on realized yields.
- Suppose the ADMM subproblems are regions with a few arcs between them.
- So there must be a consensus for flow on the arcs between two regions.

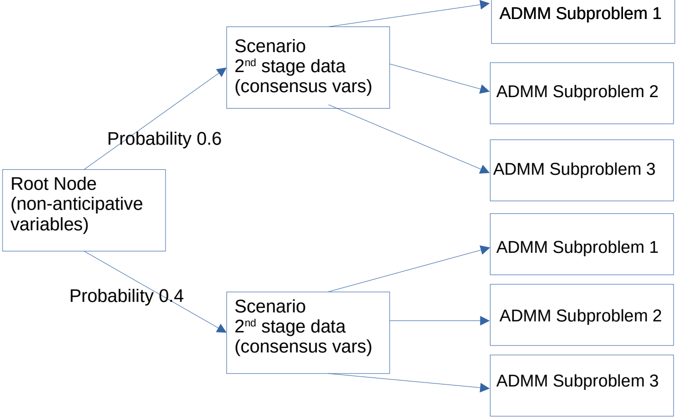

A Two stage stochastic example (T=2)
Scenario 1
2nd stage data
Probability 0.6
Root Node
(non-anticipative
variables)
Probability 0.4
Scenario 2
2nd stage data

Under the hood: Extended Scenarios and Tree

- The collection of ADMM subproblems, A, are considered to emanate from a scenario tree node that is replicated for addition to the original scenario tree at every original leaf node.
- So now we have a tree with $T+1$ stages and $|\Xi||A|$ *extended scenarios*.
- There's going to need to be some funny business with non-anticipative variables and with probabilities if we are going to use standard stochastic scenario decomposition algorithms.

Combined “Scenario” Tree

Suppose we have 3 consensus ADMM subproblems (T+1=3)



Conclusions about Stochastic consensus ADMM

- Our paper describes methods and software for using a stochastic programming decomposition algorithm for stochastic consensus ADMM.
- You could use similar thinking to adapt an ADMM algorithm for stochastic ADMM.
- Aside: decomposition seems to be needed for only a fraction of "pure" stochastic problems, so ADMM problems seem like a good place to hawk our wares.