(Some of) What's new in MPI-SPPY

David L Woodruff [1]
Jean-Paul Watson [2]
Ben Knueven [3]

ICSP 2025

[1]Graduate School of Management,
University of California, Davis
[2]Lawrence Livermore National Laboratory
[3]National Renewable Energy Laboratory

Special issue of CMS

Checkout the special issue of CMS from the last ICSP!

https://link.springer.com/collections/ijeejjabgb

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, almost always unknowable.

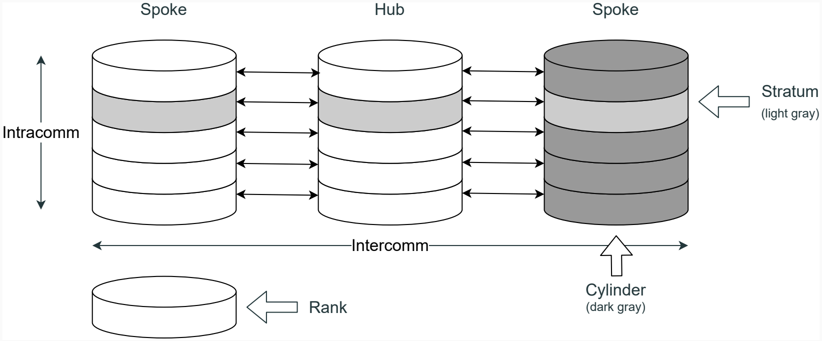
Outline

- What is MPI-SPPY?
- New communication paradigm
- New user interface.
- New ways to interface with AMLs
- If time...

MPI-sppy: The paper and the software

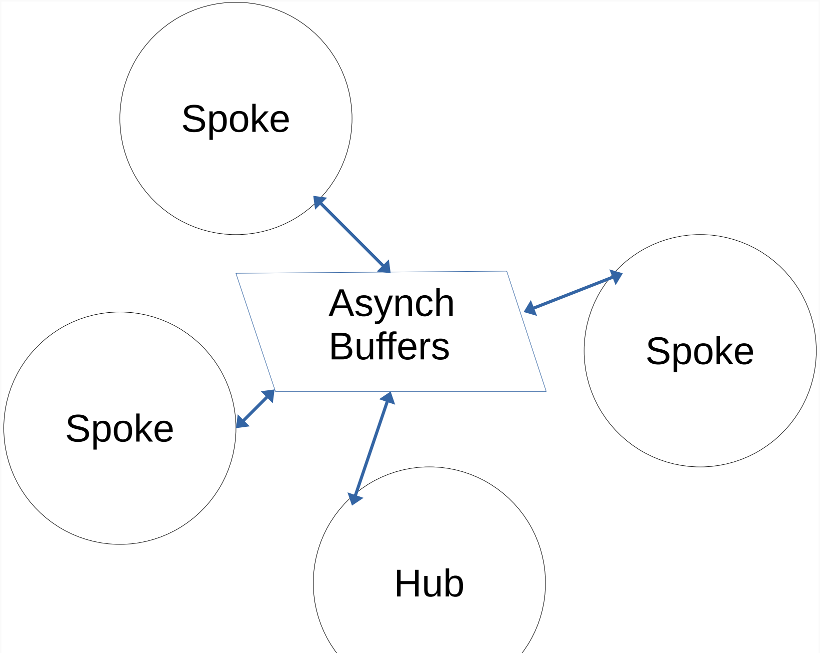
- B. Knueven, D Mildebrath, C. Muir, JD Siirola, J-P Watson, DL Woodruff, “A parallel hub-and-spoke system for large-scale scenario-based optimization under uncertainty,” *MPC*
- Find $\hat{x}$ with bounds and/or confidence intervals on the objective function for a scenario-based T-stage expected value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



But the connections have changed

Top View of Newer Version



Changes

- The cylinder functioning as the hub is no longer the center of communication. Every cylinder can see the asynchronous buffers.
- The hub cylinder now
  - keeps track of bounds,
  - does some output, and
  - decides when to shut down and signals all other cylinders to do so.
- This allows more flexibility to have spokes that tighten variable bounds, find and adjust parameter values (e.g. ρ), and compute upper/lower objective function bounds.

generic_cylinders.py

- Provides access to many cylinders and extensions without needing a user-written driver.
- Allows developers to provide driver maintenance instead of users.
- Gives users easy access to the latest features.
- Has too many command line options to be any fun.
- Supports problems from other AMLs.

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 and a little other data.
    - You do this once and point mpi-sppy to the files.
    - No Python programming required (unless, of course, that's how you interact with your AML)

- Tight: If your "AML" is callable in the sense that an outside caller can modify the objective at runtime, 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

More about loose coupling

You (or your LLM) are probably going to want/need to write a script to use meaningful variable names in the MPS file and then use the same names in the JSON file.

Example of json file

```
{
  "scenarioData": {
    "name": "Scenario1",
    "scenProb": 0.3333333333333333
  },
  "treeData": {
    "ROOT": {
      "condProb": 1.0,
      "nonAnts": [
        "NumProducedFirstStage(1)",
        "NumProducedFirstStage(2)",
        "NumProducedFirstStage(3)",
        "NumProducedFirstStage(4)",
        ...
        "NumUnitsCutFirstStage(10_10)"
      ]
    }
  }
}
```

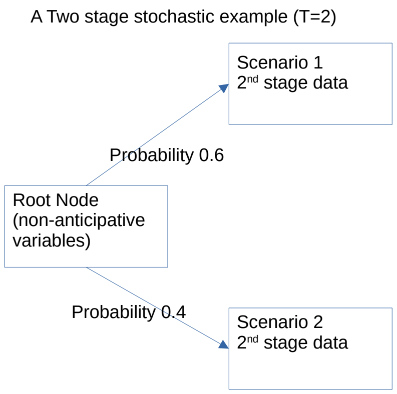
Consensus ADMM Under Uncertainty

- 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.
- Currently, you need to write a driver (not supported by generic_cyinders).

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.

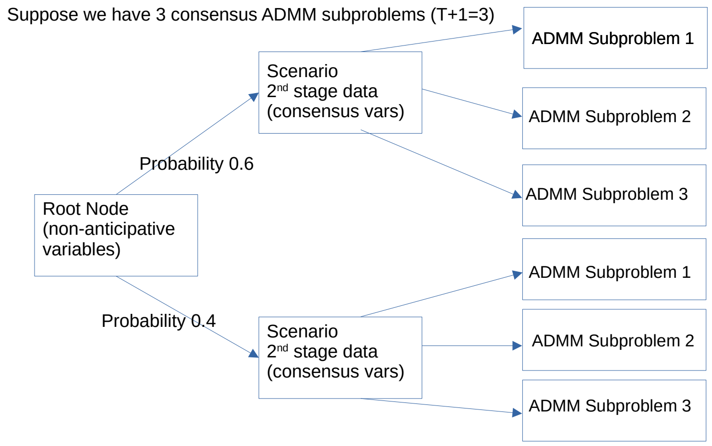
Progressive Hedging and the like



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)
Root Node (non-anticipative variables)
Probability 0.6
Probability 0.4
Scenario 2nd stage data (consensus vars)
Scenario 2nd stage data (consensus vars)
ADMM Subproblem 1
ADMM Subproblem 2
ADMM Subproblem 3
ADMM Subproblem 1
ADMM Subproblem 2
ADMM Subproblem 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.