Split-and-Merge Method for Accelerating Convergence of Stochastic Linear Programs

Akhil Langer and Udatta Palekar*

Department of Computer Science, *Department of Business

University of Illinois at Urbana-Champaign

PARALLEL PROGRAMMING LABORATORY

COLLEGE *of* BUSINESS *at* ILLINOIS

MOTIVATION

US Air Mobility Command Mission:

Optimally schedule 1300 aircraft for

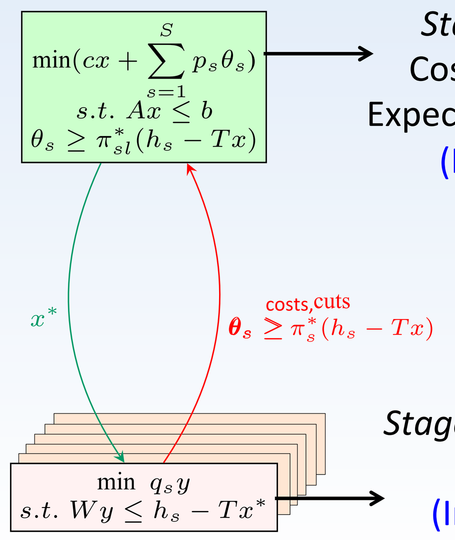
- ❑ Cargo shipment
- ❑ Distinguished visitor support
- ❑ Personnel movement
- ❑ Air refueling

CHALLENGE

Mission requirements subject to enormous uncertainty, even in peacetime

- ❑ Demand delays
- ❑ Weather events
- ❑ Demand changes
- ❑ Natural disaster
- ❑ Aircraft breakdown
- ❑ Conflicts

STOCHASTIC OPTIMIZATION OVERVIEW



Stage 1 (strategic decision)
Cost of resource allocation + Expected cost of stage 2 decisions
(Integer) Linear Program

Planning Stage: Makes decision about known parameters

Stage 2 (operational decisions)
Execution cost
(Integer) Linear Program

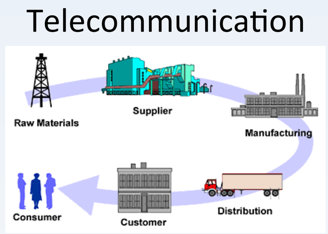
Execution stage: Assumes probabilistic distribution of uncertain parameters is known and evaluates them for the stage 1 decision

APPLICATIONS

Agriculture

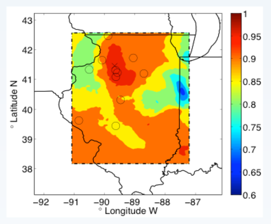
Telecommunication

Finance

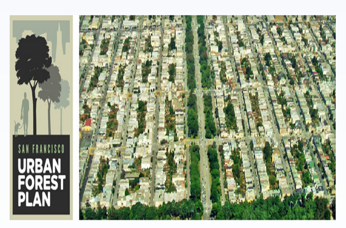


Supply-chain Management

Production planning



Energy-grids



Forestry

Transportation

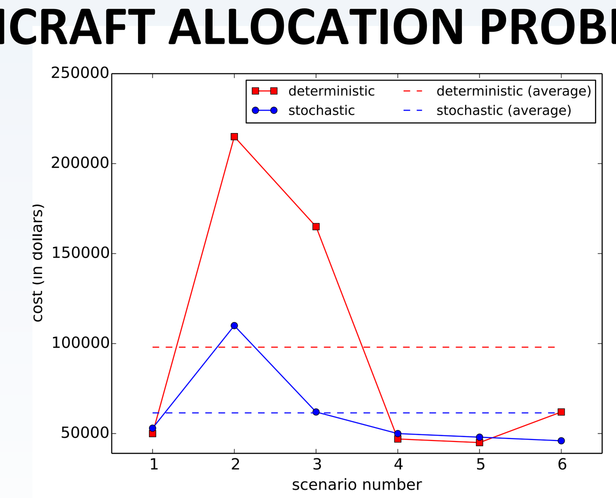
STATE-OF-THE-ART CANNOT SOLVE LARGE PROBLEMS TO OPTIMALITY

What is new that we are doing?

Combine stochastic programming with high performance computing

- Facilitates reconciliation of myriad possible outcomes in a timely manner
- Makes it feasible to solve integer programs

CASE STUDY: US MILITARY AICRAFT ALLOCATION PROBLEM

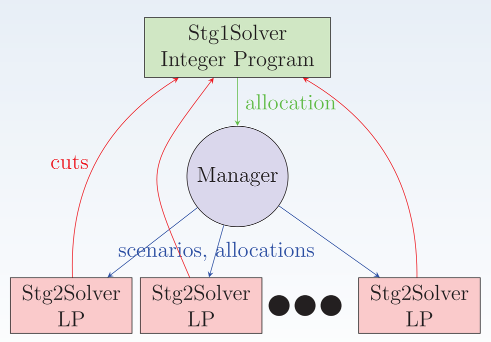


US AMC yearly expenses: ~USD 4 Billion

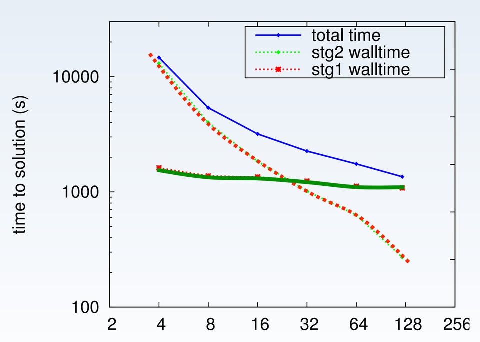
❑Average cost benefit: 35%

❑Robustness: 66% reduction in variance

LIMITATIONS OF NAÏVE PARALLELIZATION



Parallel Design

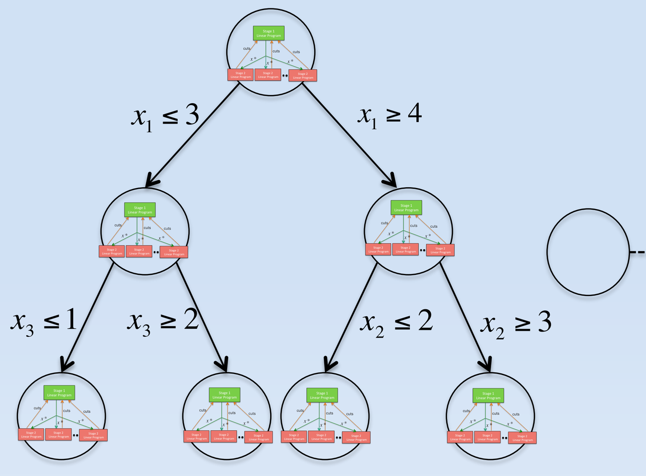


Scalability: 120 scenarios

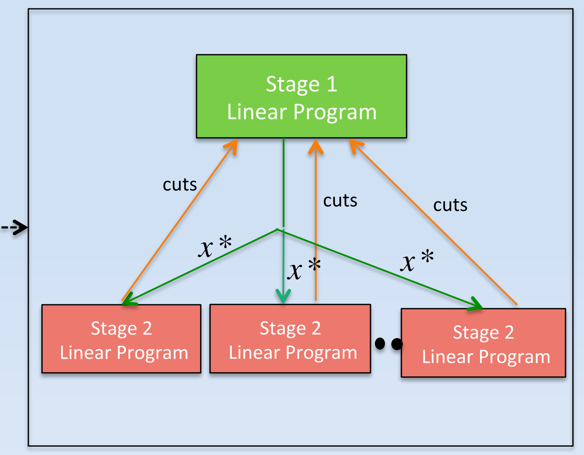
- Scalability limited by #scenarios
- Stage 1 bottleneck
  - Parallel efficiency decreases with increase in Stage 1 size

NESTED BRANCH-AND-BOUND PARALLELISM

Relax integer program to linear program → Each vertex is a stochastic linear program



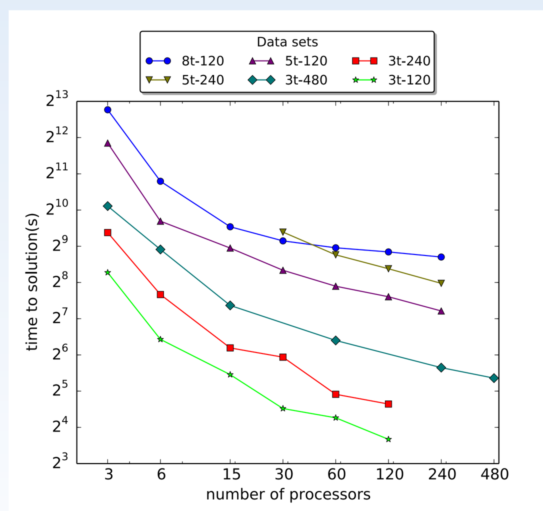
Nested Parallelism



- ❑Parallel exploration of tree vertices
- ❑Simultaneous evaluation of scenarios

PERFORMANCE COMPARISON WITH COMMERCIAL SOLVERS

Scaling



SIPS Performance

1. 96x speedup from 3 to 480 cores

VS

Gurobi Optimization — Gurobi IP Solver Performance

Typical parallel efficiency of just 0.1 at 32 cores

Challenge

Root vertex solve is a bottleneck, when #scenarios are large

RELATED WORK

- ❑ Magnanti and Wong, 1981
  - ➢ *Add only dominating cuts*
  - ➢ *Requires solving additional optimization problems*
- ❑ Linderoth et al, 2003
  - ➢ *Requires solving additional optimization problems to determine usability of cuts*
- ❑ Trust Region, Ruszczynski, 1886 and Linderoth et al, 2003
  - ➢ *Add objective term to minimize movement of candidate solution*
  - ➢ *Requires doing several minor iterations between major iterations*
- ❑ Progressive Hedging Algorithm, 1991
  - ➢ *Requires search for optimal Lagrangean multiplier which can be prohibitive*

PROPOSED SPLIT-AND-MERGE (SAM) METHOD

Split original problem into many small subproblems each with a subset of scenarios

Perform stochastic linear optimization of subproblems (in parallel) until converged or until a specified number of iterations

Merge cuts from subproblems

Solve original problem with collected cuts

Input: S (**set** of scenarios), Original Stochastic Program (P)

Divide S into n clusters, $S_1, S_2, ..., S_n$
Generate n stochastic programs, $P_1, P_2, ..., P_n$, with scenarios from $S_1, S_2, ..., S_n$, respectively
Scale scenario probabilities **in** each of these subproblems such that they sum up to 1

```
#pragma omp parallel for
for i in range(1,n):
  scosts_i = [] #scenario costs
  cuts_i = [] #scenarios cut constraints
  while r_i < r or hasConverged(i):
    x_i = solveStage1(P_i, scosts_i, cuts_i)
    scosts_i, cuts_i = solveStage2(x_i)
    r_i=r_i + 1
  end while
```

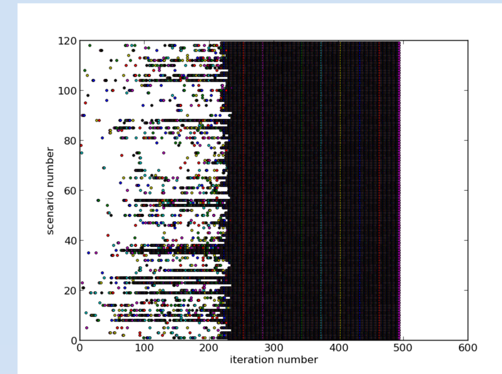
```
#wait until all the subproblems have returned
cuts = []
scosts = []
for i in range(1,n):
  cuts.add(getCutConstraints(P_i))
```

```
#now solve the original problem
while not hasConverged(P):
  x = solveStage1(P, scosts, cuts)
  scosts, cuts = solveStage2(x)
end while
```

RESULTS

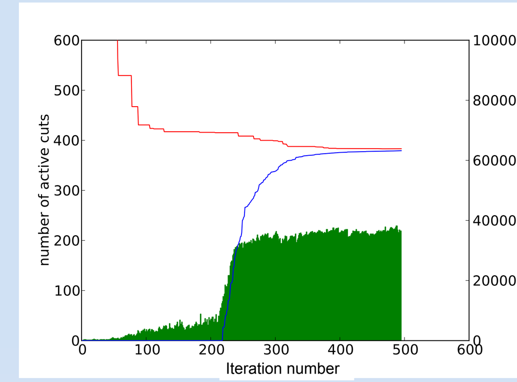
Multicut Benders method

Scenario Activity



Total Iterations = 495

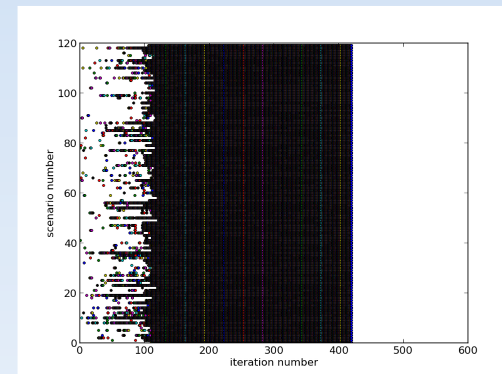
Cut Activity



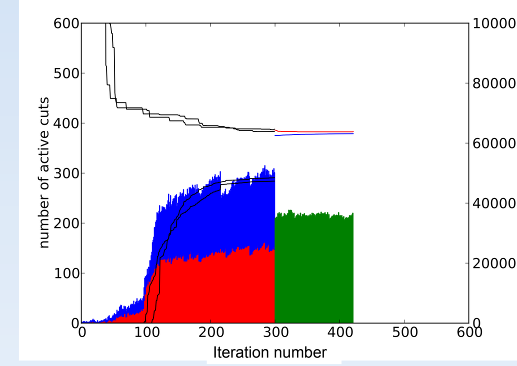
Time to Solution = 1190s

SAM method

34% improvement in time



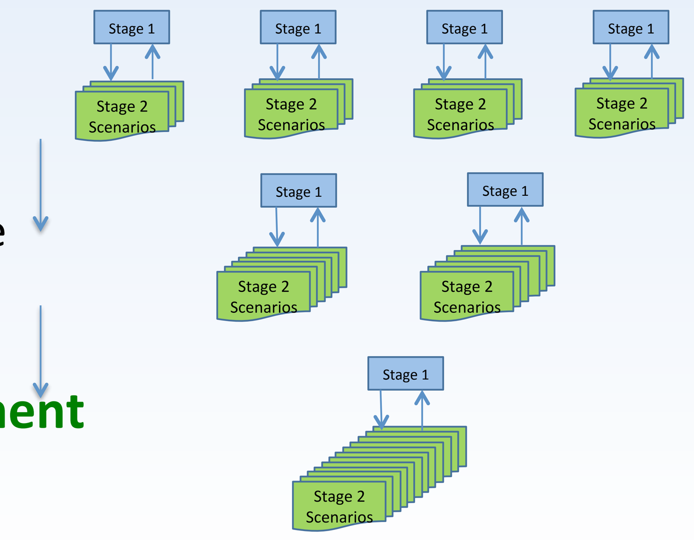
Total Iterations = 415

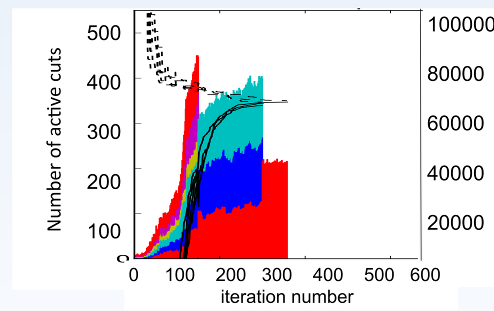


Time to Solution = 784s

Hierarchical Split-and-Merge Method

58% improvement in time





Total Iterations = 360

Time to Solution = 507s

SAM BENEFITS

- ❑ Higher cut activity from initial iterations of Benders method
- ❑ Reduced Stage 1 bottleneck size
- ❑ Increased Parallelism in Stage 1
- ❑ Reduced total iterations and time to solution
- ❑ 58% improvement in time to solution compared with Benders method

FUTURE WORK

- ❑ Further exploration of HSAM method
- ❑ Automated determination of split-phase duration
- ❑ Determining optimal subproblem size

TAKEAWAYS

- ❑ Accelerated convergence by problem decomposition
- ❑ Enabled large-scale stochastic optimizations leading to robust planning of US AMC operations
- ❑ Reduced time to solution
- ❑ Asynchronous parallel programming model for maximum productivity and performance

REFERENCES

- Akhil Langer, Ramprasad Venkataraman, Udatta Palekar, and Laxmikant V. Kale. "Parallel branch-and-bound for two-stage stochastic integer optimization." In *High Performance Computing (HiPC), 2013 20th International Conference on*, pp. 266-275. IEEE, 2013. **Best Paper Award.**
- Langer, Akhil, Ramprasad Venkataraman, Udatta Palekar, Laxmikant Kale, and Steven Baker. "Performance Optimization of a Parallel, Two Stage Stochastic Linear Program." In *2012 IEEE 18th International Conference on Parallel and Distributed Systems*, pp. 676-683. IEEE, 2012.
- Laxmikant Kale, Anshu Arya, Nikhil Jain, Akhil Langer, Jonathan Liander, Harshitha Menon, Xiang Ni, Yanhua Sun, Ehsan Totoni, Ramprasad Venkataraman, and Lukasz Wesolowski. Migratable Objects + Active Messages + Adaptive Runtime = Productivity + Performance A Submission to 2012 HPC Class II Challenge. Technical Report 12-47, Parallel Programming Laboratory, November 2012. **HPCC 2012 Finalist.**
- Langer, Akhil, Ramprasad Venkataraman, Gagan Gupta, Laxmikant Kale, Udatta Palekar, Steven Baker, and Mark Surina. "Poster: enabling massive parallelism for stochastic optimization." In *Proceedings of the 2011 companion on High Performance Computing Networking, Storage and Analysis Companion*, pp. 89-90. ACM, 2011.