HIPPO – Solving Large Security Constrained Unit Commitment Problem

Feng Pan, Jesse Holzer, Yonghong Chen

FERC Technical Conference

June 23, 2020

HIPPO Background

- Funded by ARPA-E, 11/2016 – 1/2020.
- Problem – Day-ahead security constrained unit commitment problem
- Challenge - Slow solution times lead to inefficient cost, reduced reliability and slow adaptation of new market designs.
- Solution – A solution framework based on parallel and concurrent optimization.
- Goal – 10+ speedup.

Team - PNNL

- PNNL
  - Feng Pan, Jesse Holzer, Arun Veeremany
- MISO
  - Yonghong Chen, Joanna Wu, Yamin Ma, Jessica Harrison
- GE
  - Jie Wan, Xiaofeng Yu
- GUROBI Optimization
  - Ed Rothberg
- University of Florida
  - Yongpei Gu, Yanan Yu
- Cognitive Analytics
  - Jim Ostrowski, Jonathan Schrock

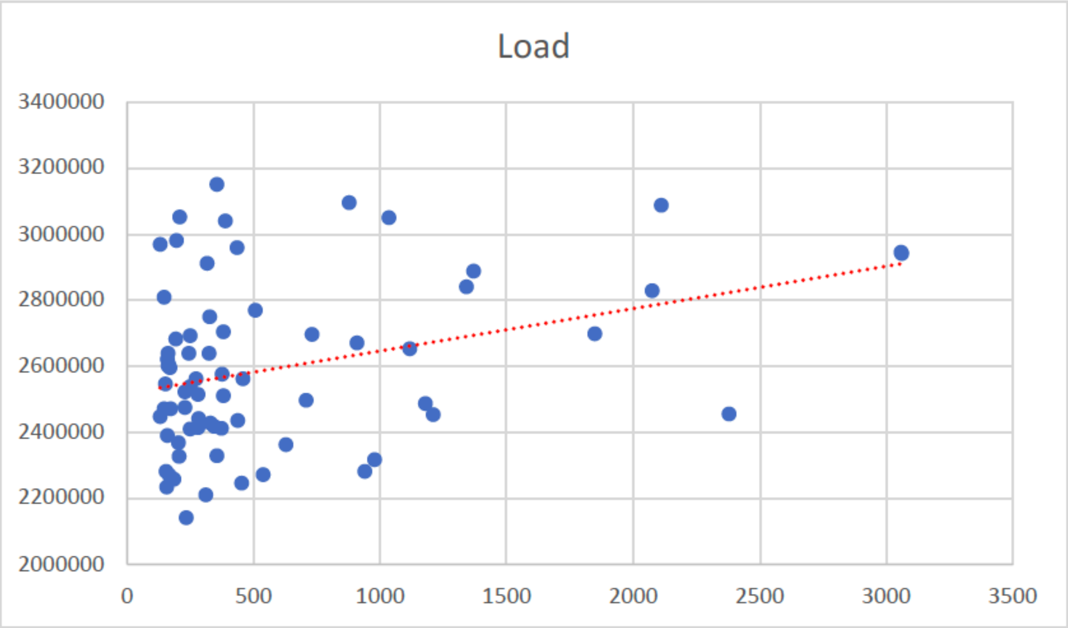
Key Statistics

Market Participants	408
MWs of Generating Capacity (Mkt)	178,140
Peak Load (MW)	132,893
Generating Units (Market)	1390
Network Buses	43,962
Miles of Transmission Lines	65,800
Square Miles of Territory	900,000
States Served	15 Plus Manitoba Province, Canada
Millions of People Served	42

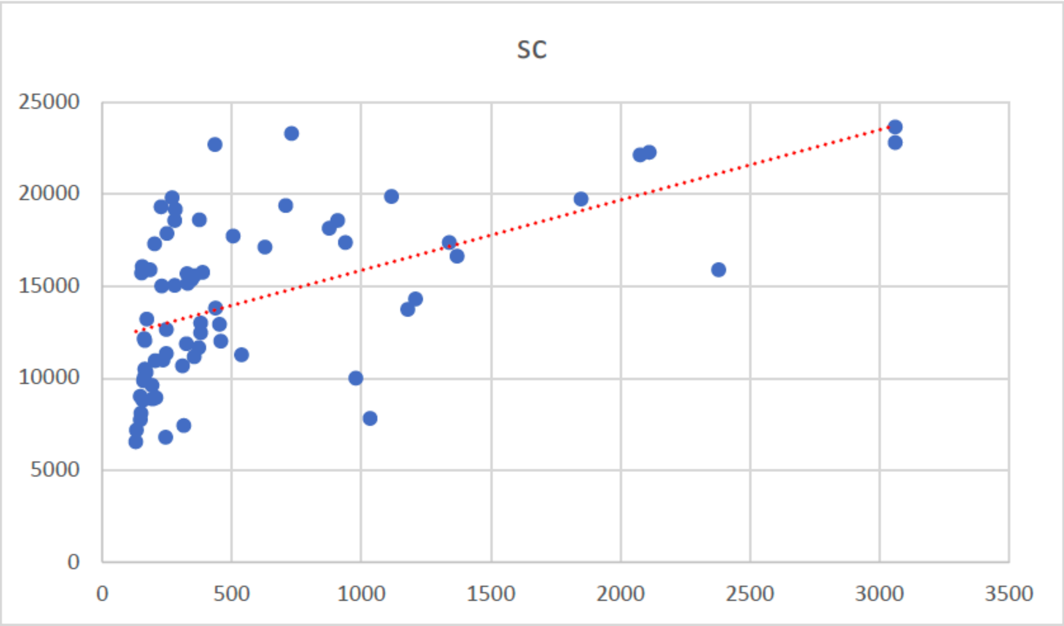
Characteristics of SCUC at MISO

- MISO has a large footprint.
- System-level constraints
  - Watch-list includes about 7000 security constraints
  - Three system-level constraints for reserve products in each period
- Generator-level
  - Additional binary variables for committing regulating reserve.
  - Generators can have two sets of bounds
  - Limits on total daily energy use and startups
- Many virtual bids ~ 16,000+

SCUC Runtime vs Individual Elements

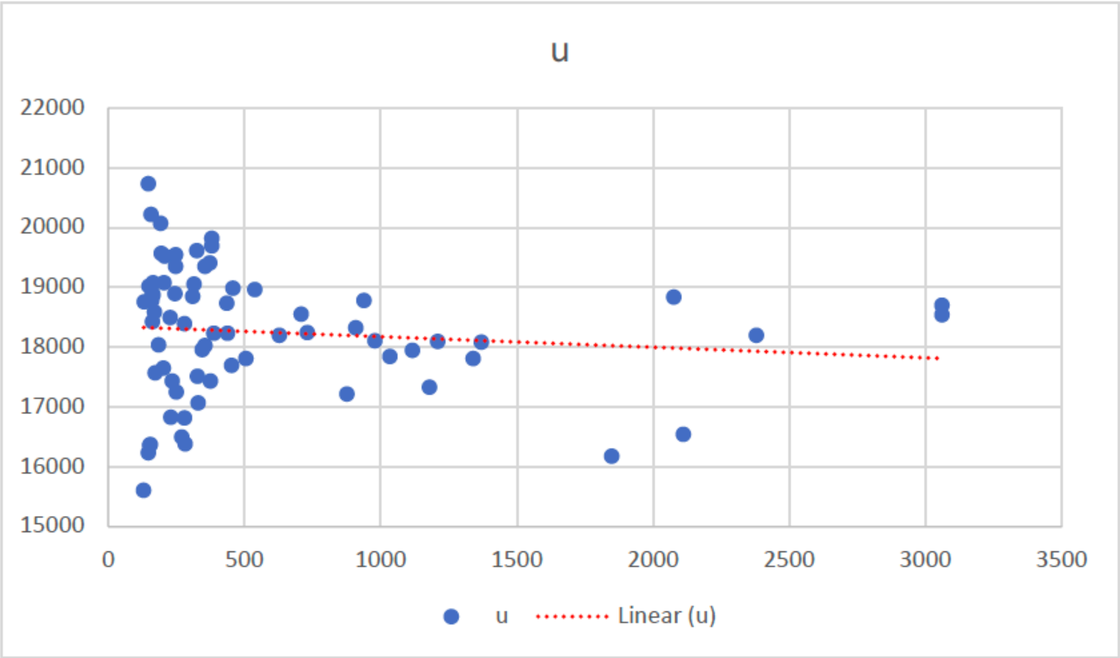


Runtime vs load

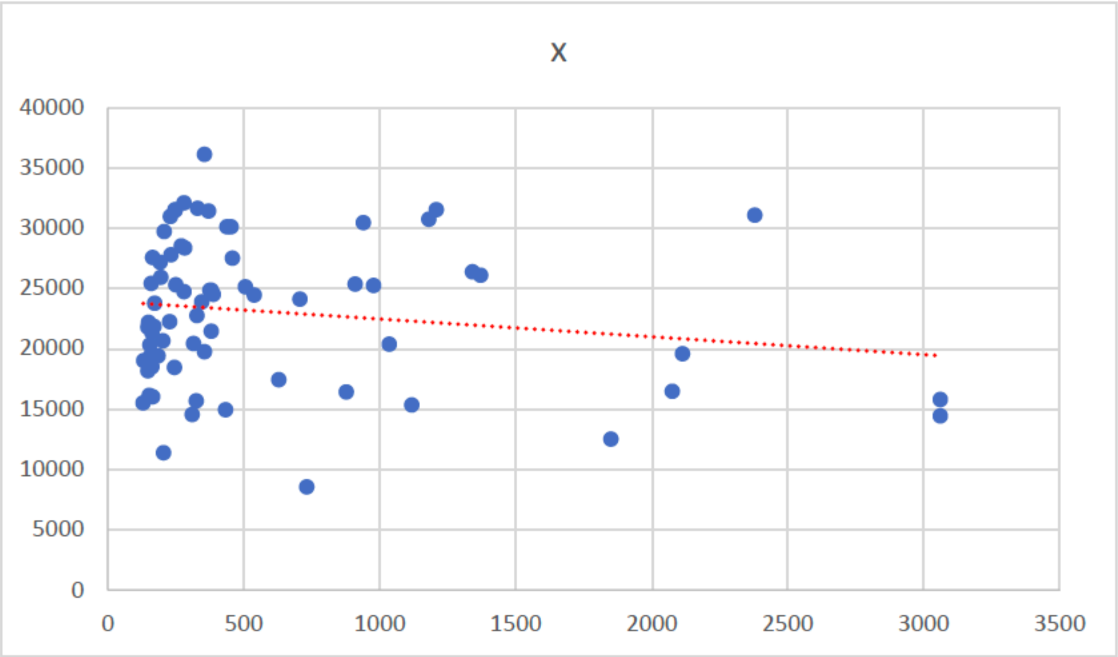


Runtime vs number of Security Constraints

SCUC Runtime vs Individual Elements



Runtime vs number of commitment variables



Runtime vs number of virtual bids

Solving Unit Commitment in Day-ahead Market

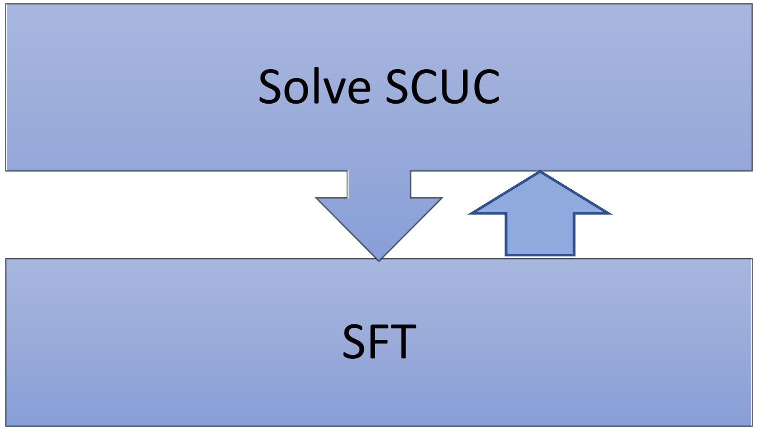
Cost Minimization

Solve security constrained unit commitment (SCUC)

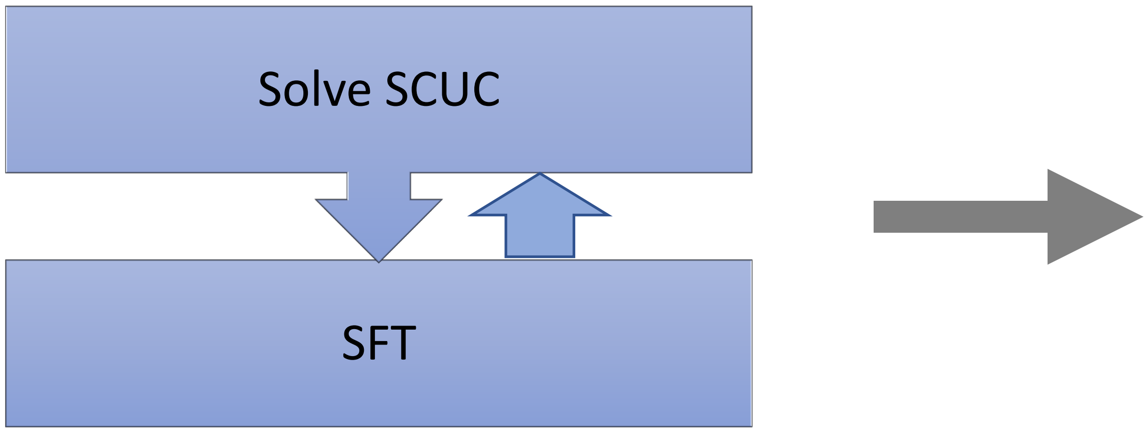
↔

Security Check

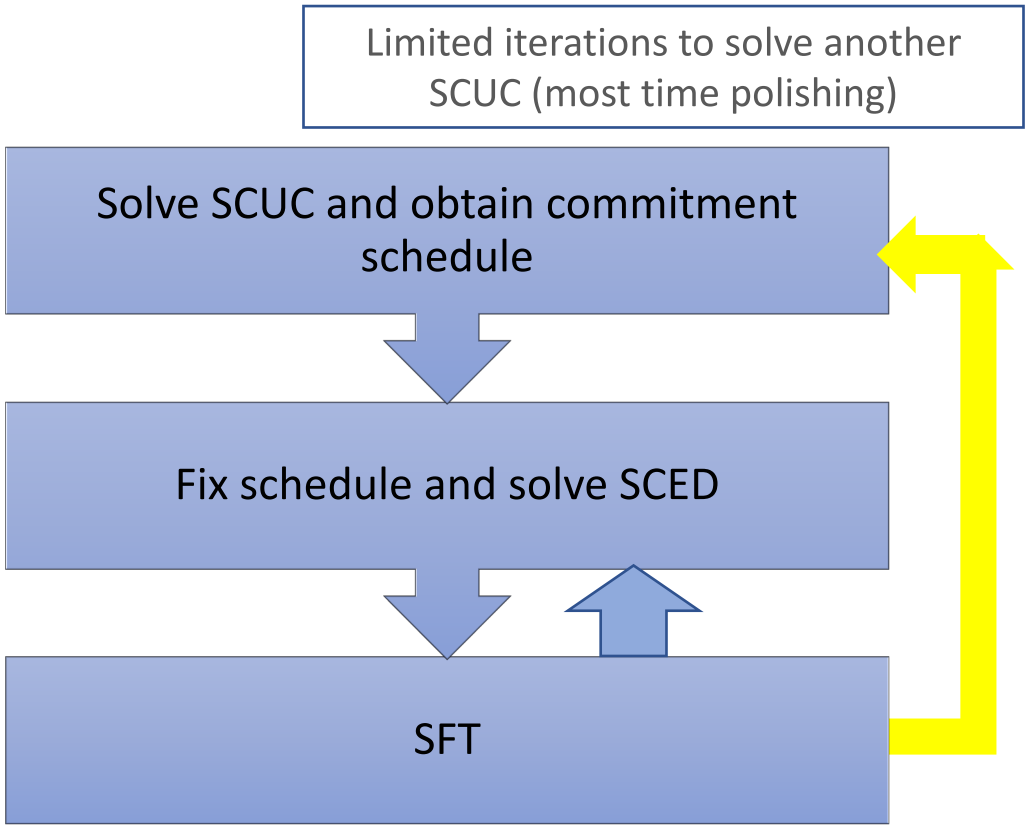
simultaneous feasibility test (SFT)



Solving Day-ahead Unit Commitment – Approach Used in Practice



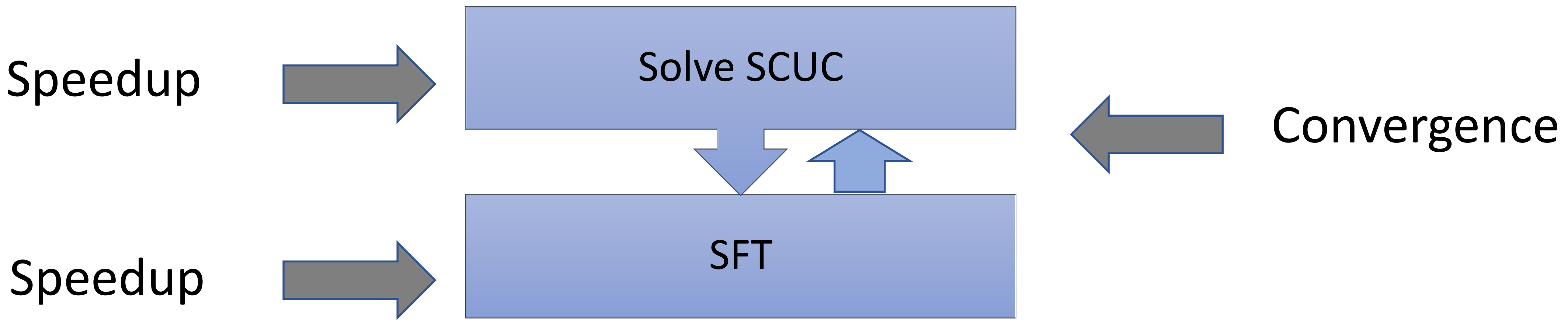
Long Computation Time
Optimal



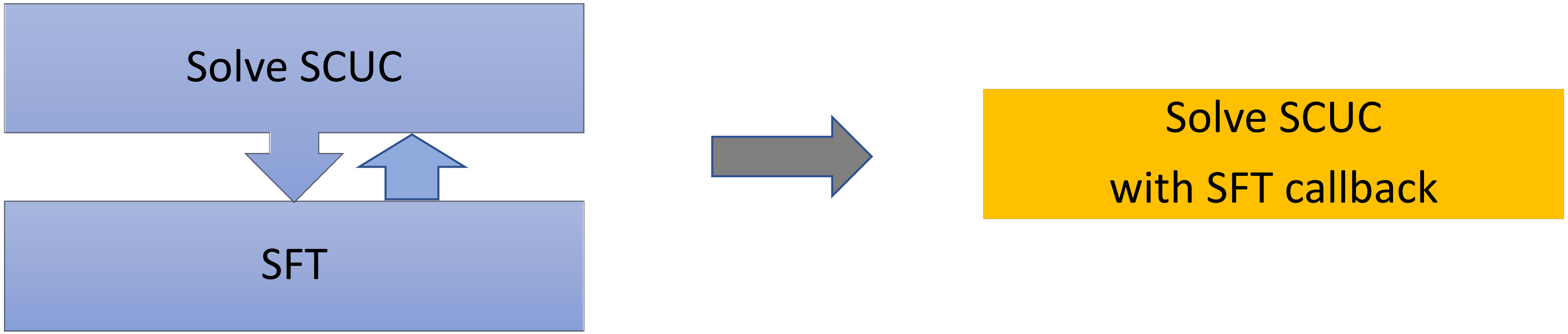
Shorter Computation Time
Suboptimal

Goal of the HIPPO Project

Achieve Optimality with full iteration between SCUC and SFT at fast speed



Sequential vs Callback



- SCUC (MIP) is solved multiple times
- SFT is called 2-5 times

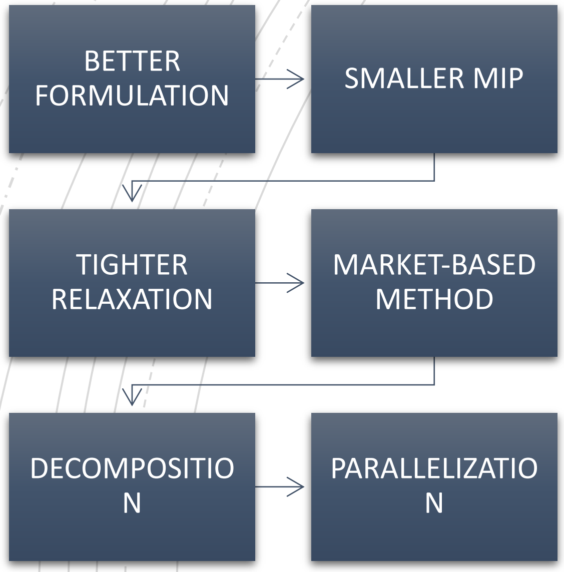
- SCUC (MIP) is solved once
- SFT is called many times (for every incumbent solution)

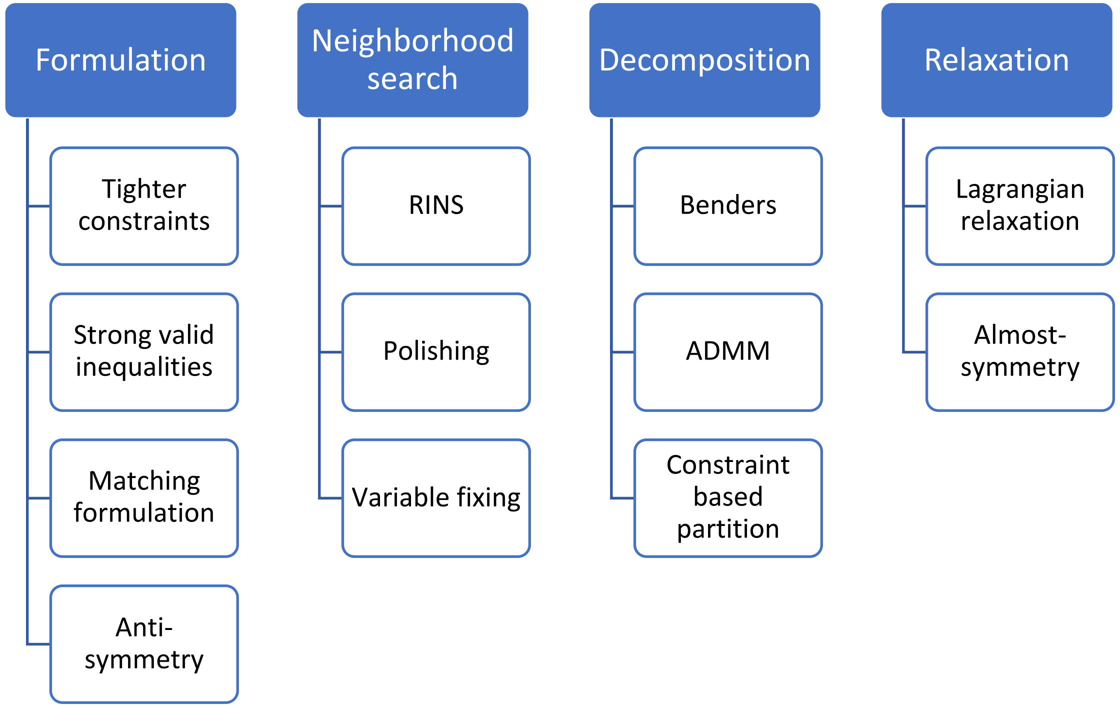
Improve Computation

Solve SCUC

Solve SFT

HIPPO - Solving SCUC

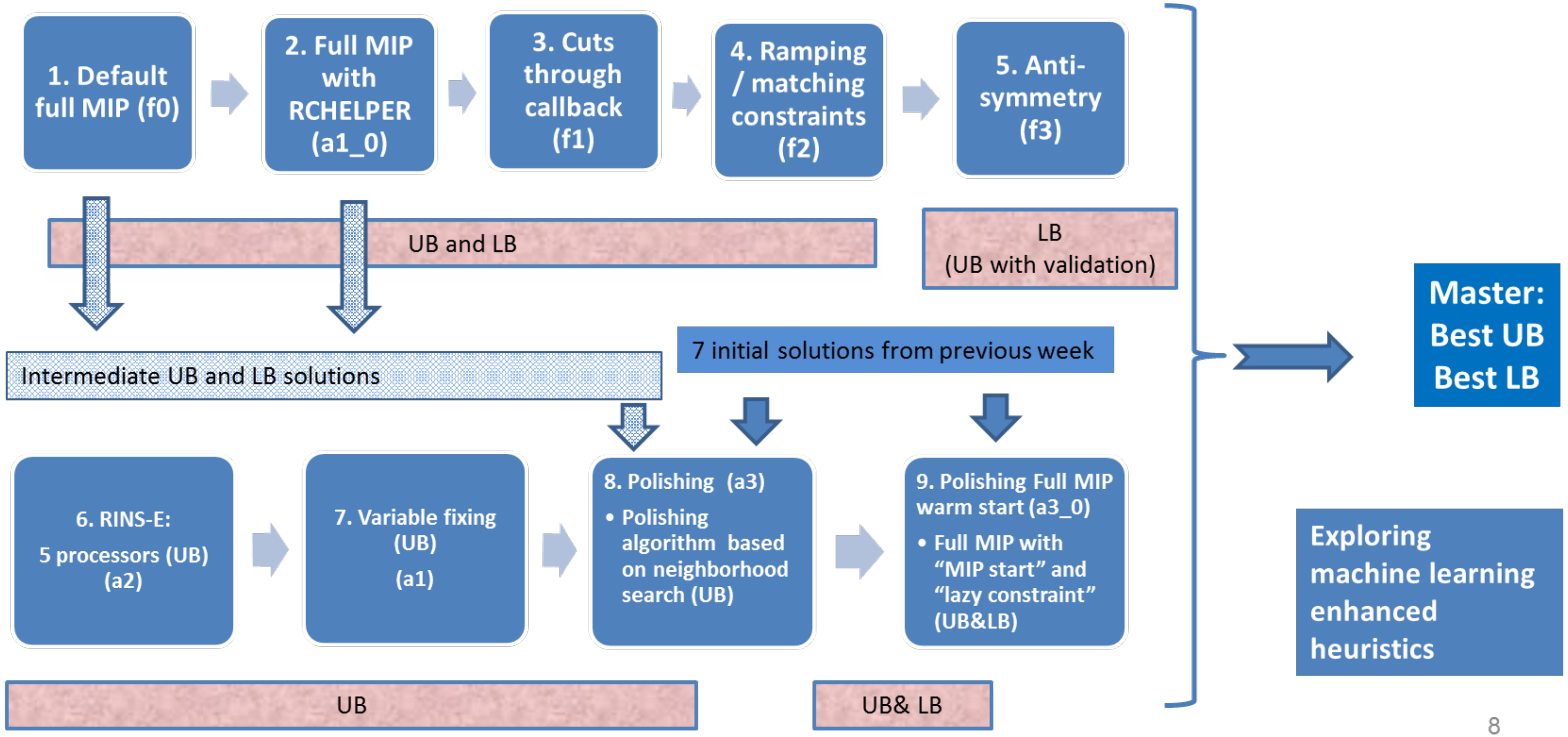




Concurrent Optimizer launches multiple algorithms simultaneously and enables them to communicate

Sample Configuration - HIPPO Concurrent Optimizer

Gurobi full MIP with different settings:
Using customized **Gurobi8.1.0** with variable fixing fork-off



Performance – Solving SCUC

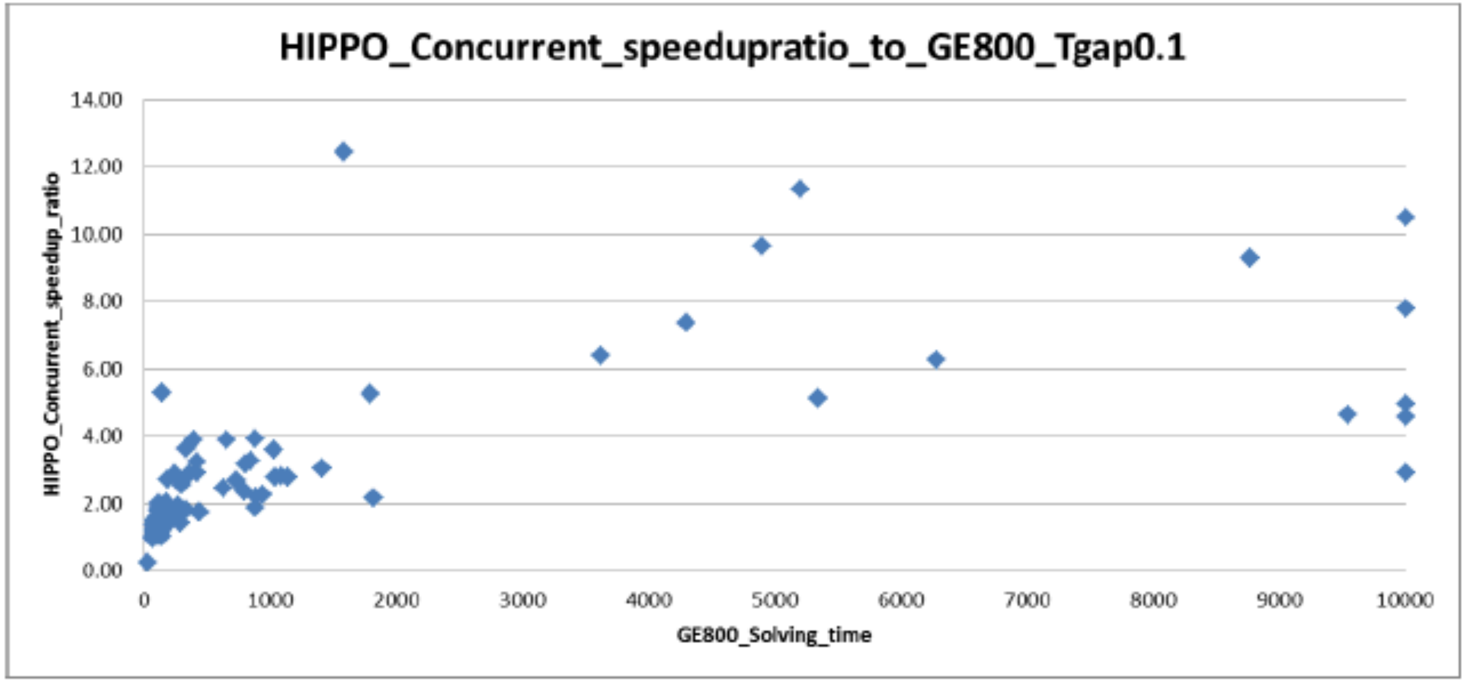
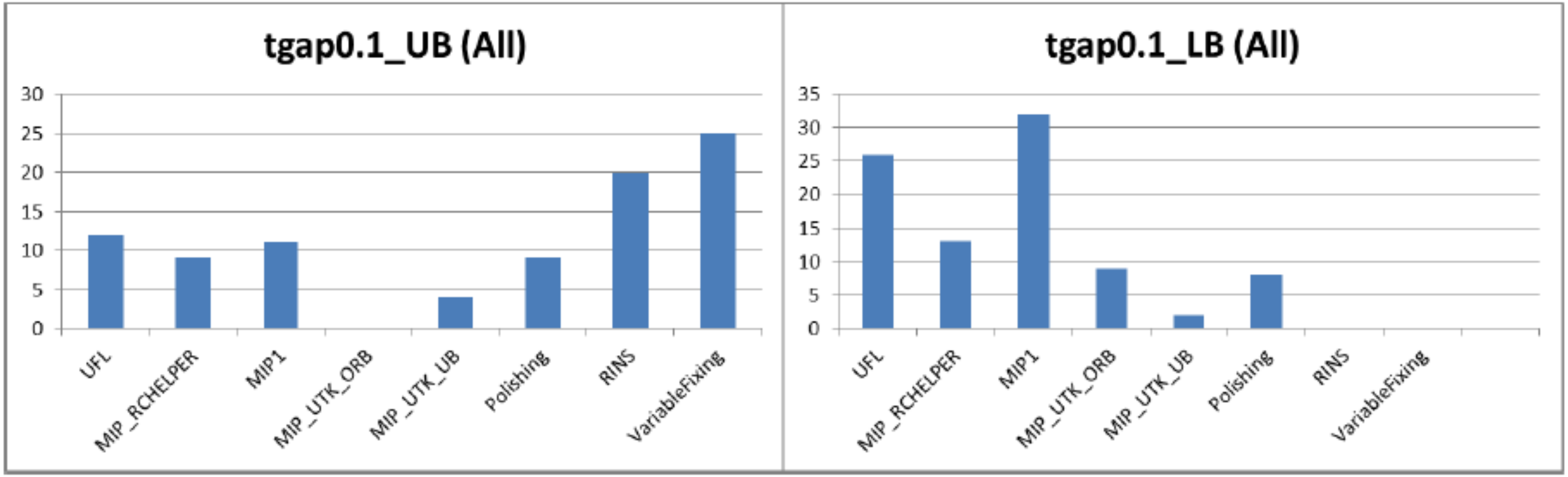


Figure 13.2: Speedup ratio by HIPPO CO comparing to GE solver for solving 90 MISO market cases.

Performance – Solving SCUC



HIPPO - Solving SFT

Solving Ax = b with many similar A matrices

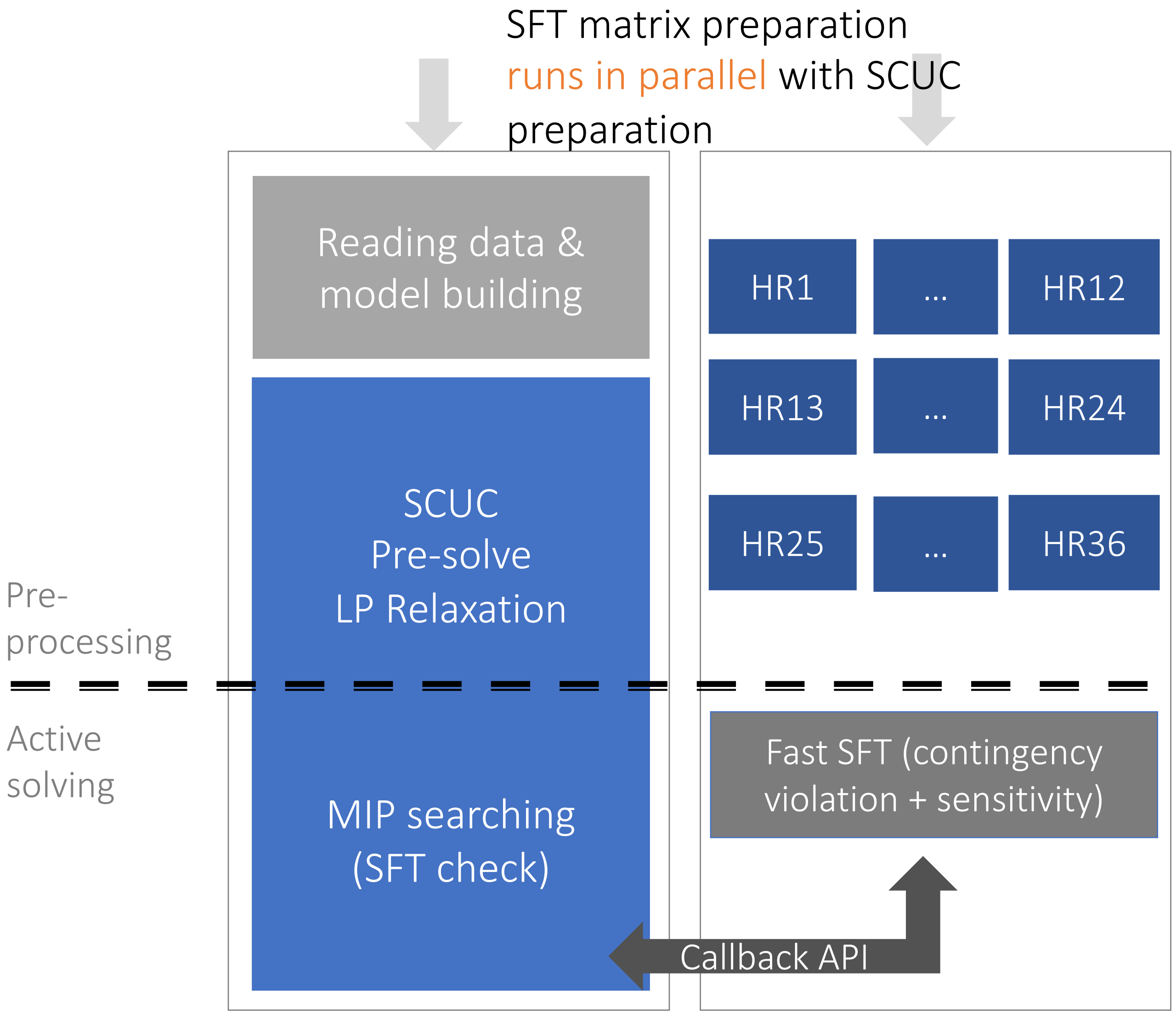
- CHOLESKY FACTORIZATION
- LOWER RANK SMW UPDATE
- ORDER OF COMPUTATION
- PRE-PROCESSING
- AGGREGATION OF BUSES
- PARALLELIZATION

Performance – Solving SFT

SFT configuration	3node*12processor	1node *12 processor	1node*36processor	6node*6processor
Pre-processing #Matrix/Node	12	12	36	6
#nodes	3	1	1	6
#Matrix	36	12	36	36
SFT check time | end time | #violation	49.83 | 212.91 | 261	33.76 | 217.23 | 261	441.19 | 608.93 | 261	18.88 | 185.56 | 261
	3.61 | 236.38 | 6	6.04 | 362.93 | 5	5.16 | 752.99 | 5	3.97 | 212.07 | 6
	3.45 | 262.06 | 0	5.89 | 703.35 | 2	5.06 | 1089.43 | 2	3.53 | 361.49 | 1
	3.55 | 309.53 | 4	5.59 | 709.1 | 2	5.17 | 1094.76 | 2	3.54 | 780.8 | 2
	3.17 | 332.59 | 0	5.61 | 714.89 | 1	5.04 | 1099.99 | 1	3.4 | 790.79 | 0
	3.36 | 514.85 | 2	5.68 | 720.76 | 1	4.98 | 1105.15 | 1	3.62 | 794.65 | 1
	3.15 | 541.2 | 0	5.77 | 726.79 | 0	5.07 | 1110.47 | 0	3.45 | 798.28 | 0
	3.06 | 875.21 | 1			
	3.01 | 878.38 | 0			
Total Time	879	727	1111	798

- 1K ctgs, 10K monitored branches, 36 time periods,
- Startup time for SFT is with SCUC construction and presolve time
- Very fast solve time. 3 to 9 seconds, compared to 800 seconds with old method

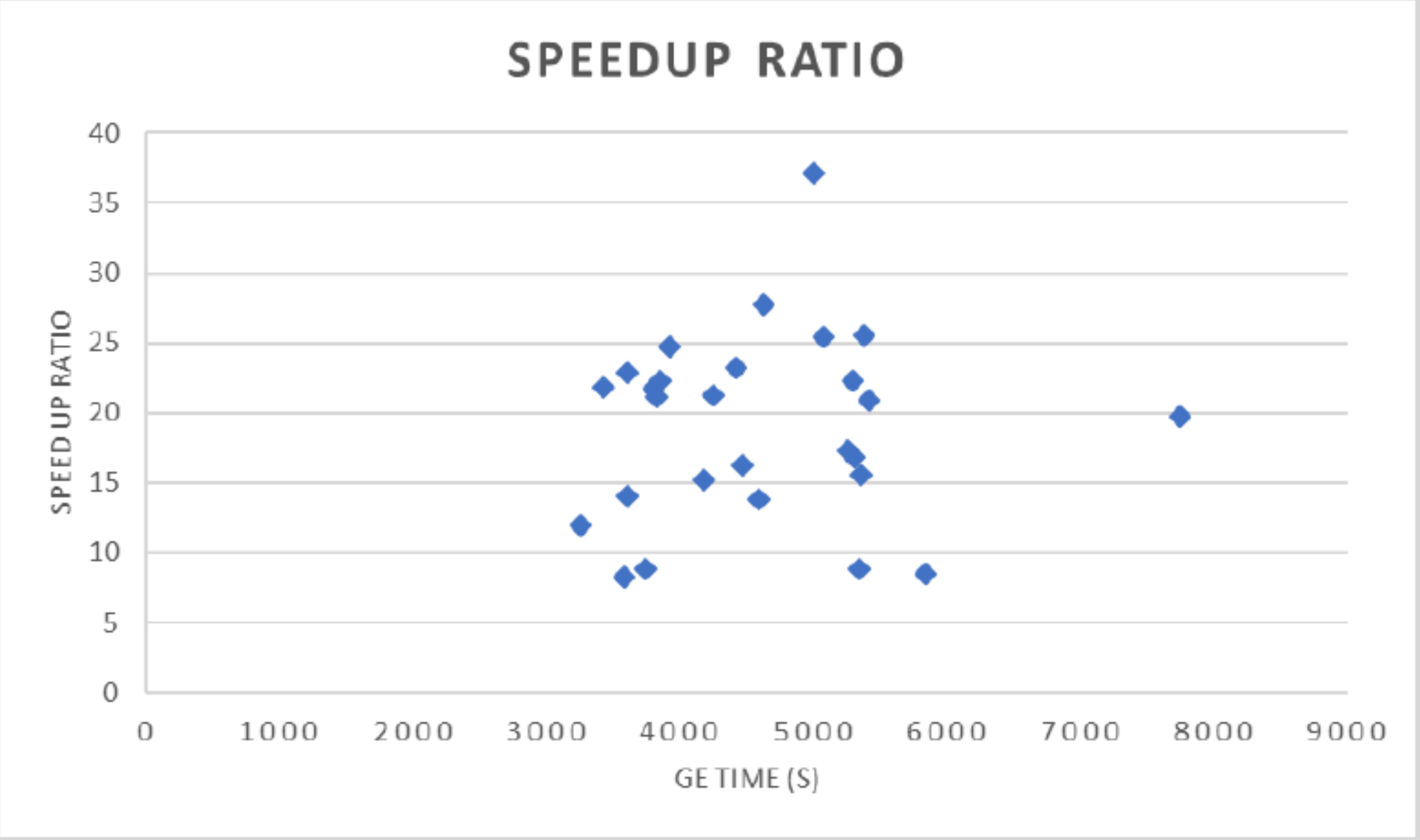
New SFT design uses parallel processing, is easily configurable across server nodes & uses efficient communication between SFT & MIP.



- SFT preparation for 36 intervals can be a bottleneck and require 3 nodes. (1interval1node and 36interval3node are similar)
- 1,000 contingencies are embedded in one matrix: all solved at once
- MIP callback API checks SFT and adds new constraints for each new incumbent solution

Performance – SCUC + SFT

- Production solver used sequential iterations
- Production solver took 2-5 iterations to solve SCUC+SFT to convergence.
- HIPPO launched about ~10 different algorithms



HIPPO Performance in Future Market Designs

Will HIPPO Concurrent Optimizer be scalable in future SCUC instances?

Preliminary Results for SCUC with 15-min Interval

Cases	HIPPO concurrent polishing + MIP solving time (s)	Default MIP time (s)	Speedup ratio
Hard 1	919	24,625	26.81
Hard 2	1,055	12,243	11.60
Hard 3	1,244	63,014	50.65
Hard 4	1,349	21,095	15.63
Hard 5	1,660	43,728	26.35
Normal 1	781	1,055	1.35
Normal 2	1,392	1,369	0.98
Normal 3	734	2,454	3.35
Normal 4	780	1,588	2.04
Normal 5	783	1,623	2.07

- Launch hourly interval and 15-min interval in HIPPO Concurrent Optimizer
- Use solution from hourly model with 1% MIP gap as an initial solution to Polishing Method.
- Polishing method iteratively make improvements until 0.1% gap or $24K absolute gap

Y. Chen, F. Pan, J. Holzer, E. Rothberg, Y. Ma, A. Veeramany, High Performance Computing Based Market Economics Driven Neighborhood Search & Polishing Algorithm for Security Constrained Unit Commitment, IEEE Tran. on Power Systems, Accepted.

Future Work

1

Move HIPPO to MISO cloud environment for further testing and evaluation.

2

Develop HIPPO as a software platform for market design and prototyping.

3

Integration to future market clearing system.