HIPPO – HPC Solver for Security Constrained Unit Commitment Problem

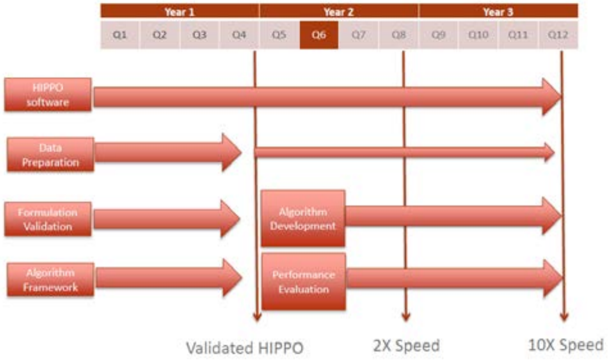
FENG PAN (PRESENTER)

PNNL, MISO, GE, GUROBI

FERC 2018

HIPPO Overview

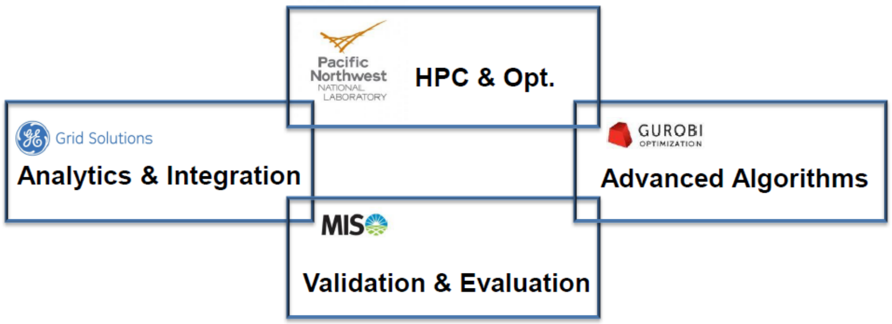
Pacific Northwest
NATIONAL LABORATORY
Proudly Operated by **Battelle** *Since 1965*



- HIPPO – a three-year project
  - Funded by ARPA-E.
  - FY16 – FY19.
  - Goal is to speed up solving security constrained unit commitment (SCUC) by a factor of 10.
- HIPPO Optimizer
  - Python package for solving SCUC.
  - Can be run on workstations and HPC.
  - Gurobi as the basic MIP solver.
- HIPPO Status
  - SCUC model is validated with MISO production tool.
  - Currently about 2-3 time speed-up (upper bound methods)

Team

Pacific Northwest
NATIONAL LABORATORY
Proudly Operated by Battelle Since 1965

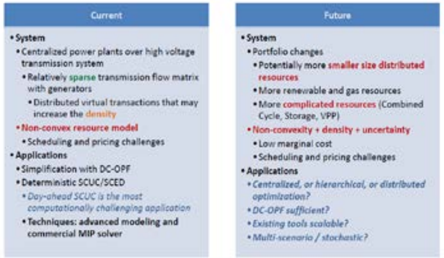


- PNNL – MIP, algorithm development, HPC, implementation and testing
- GUROBI – MIP, Gurobi solver and parallel/distributed computing
- GE – market simulator, benchmark, domain knowledge, MIP and OPF
- MISO – domain knowledge, algorithm development, data, model validation, market operations, and MIP.
- UF – Optimization, cutting planes, and integer programming
- LNNL – parallel MIP

- PNNL
  - Feng Pan (PI, Optimization)
  - Steve Elbert (Co-PI, HPC, Optimization)
  - Jesse Holzer (Optimization)
  - Arun Veeramany (Applied Math, Machine Learning)
- GUROBI
  - Ed Rothberg (Optimization)
  - Daniel Espinoza (Optimization)
- GE
  - Jie Wan (Optimization, Power System Application)
  - Xiaofeng Yu (Market Application)
  - Sandeep Lakshmichandjain (Software)
- MISO
  - Yonghong Chen (Optimization, analytics, Electricity Market)
  - Yaming Ma (Electricity Market)
- UF – FY18
  - Yongpie Guan (Optimization, SCUC)
  - Yanna Yu (Optimization)
- LNNL – FY18
  - Deepak Rajan (Optimization, HPC)

MISO SCUC

- Motivation for HIPPO – forwarding planning, SCUC is getting larger and more complex
  - More resources – gas, renewable, …
  - More complicated – CC, storage, …
  - More market products – virtuals, dispatchable demands
  - Uncertainties, subhourly, decentralization, AC vs DC…
- Speed for solving deterministic SCUC is critical for improving future market operation.
- MISO SCUC
  - Large size (# of resources, security constraints) among ISOs/RTOs.
  - Time variant generation bounds, ramp rates.
  - Commitment for reserve.
  - Many virtuals and security constraints!!!



Chen, FERC 2018

HIPPO Optimizer – Approaches

To reduce runtime

- Solve smaller subproblems then recover solutions.
- Make local improvement of a solution.
- Improve structure of the original problem (convex hull).
- Find oracle to tell us what the solutions are.
- Add computing resource.

Runtiem
vs QOS

Continuously improve upper bounds and lower bounds independently and in parallel

HIPPO Optimizer – Individual Algorithms

Pacific Northwest NATIONAL LABORATORY
Proudly Operated by Battelle Since 1965

Improving solving SCUC as MIP (exact method)

- Make MIP closer to LP - cutting planes, tighter formulation
- Solve reduced problems and gradually restore to the original – delay constraint and delay variable generation. E.g., Benders
- Branch and bound – branching rules, parallel implementation, bounds…

Improving solving SCUC via heuristics

- Separate the original problem into small subproblems and pretend SCUC as a convex problem. E.g. ADMM.
- Reduce size of problem by fixing/removing variables and constraints.
- Focus on a part of SCUC. E.g., polishing

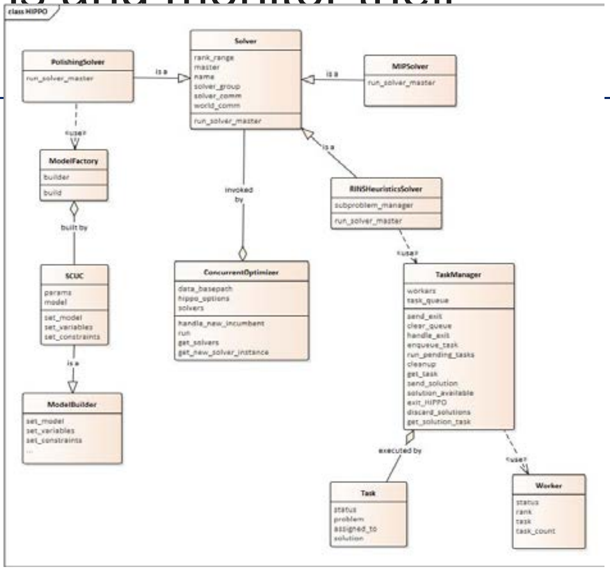
Improving solving SCUC via oracles

- Use domain specific knowledge to make decisions.

HIPPO Optimizer – Concurrent Optimizer

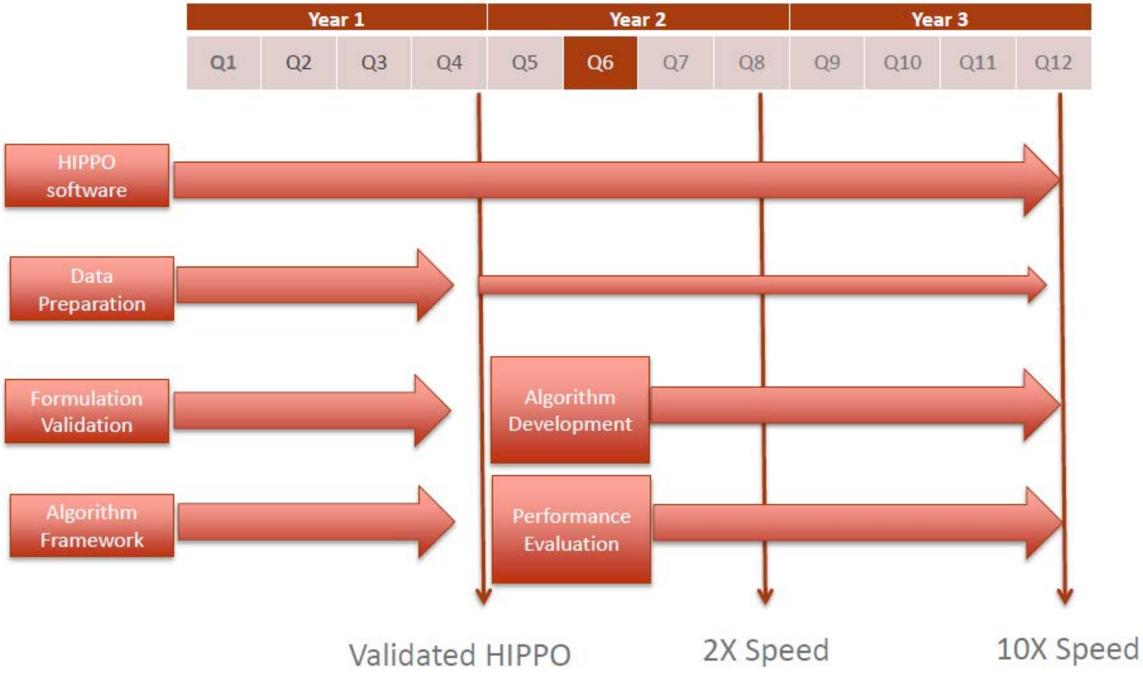
HIPPO concurrent optimizer

- The mastermind of all algorithms.
- The concurrent optimizer is run as a single solver which calls many instance of individual algorithms
- Concurrent optimizer supports the communication among individual algorithms with sharing bounds and solutions and monitor their behaviors.



HIPPO – Year 1 and Year 2

- Model validation.
  - Discuss validation results
- Implementation of Individual algorithms and performance testing
  - Show preliminary performance testing on formulation and algorith



Validation

HIPPO formulation is validated with MISO production model!!!

- Validate HIPPO SCUC formulation with GE SCUC solver.
- Criteria – feasibility, optimality.
- Cross check solutions to validate
  - Solution from one formulation is feasible in the other formulation.
  - Objective values of different formulations agree for a solution.
- There are a good amount of multiple solutions within 0.1% gap.

Number of generators different in two models

Case	# Gen
MSS_92201201709093649_0X	31
MSS_92301201709092353_0X	25
MSS_91501201709092501_0X	23
MSS_110101201611105417_0X	22
MSS_92501201709093020_0X	17

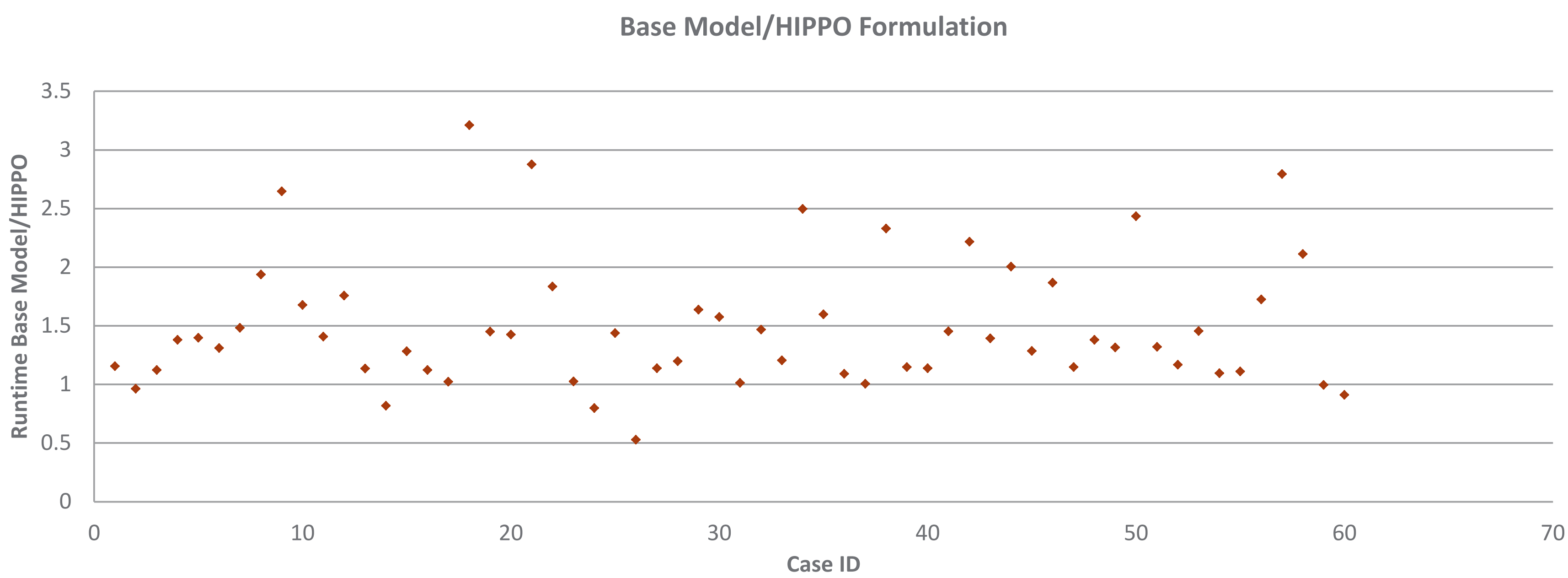
Difference in schedules

MSS_120501201712102650_0X	Hydro_1	000000000000000000000000000000hhhhhh
MSS_110101201611105417_0X	Hydro_1	00000000000000a1h0100000000000000111
MSS_91401201709092680_0X	Hydro_1	0000000000000111111h1000000000111111
MSS_71801201707092232_0X	CC_1	0000000a11111111111111h000000000aaaa
MSS_82201201708091245_1X	CC_1	0000000000aaaaa1111111h0000000000000

Performance Testing on Formulations – 60 More Mixed Cases

Pacific Northwest NATIONAL LABORATORY

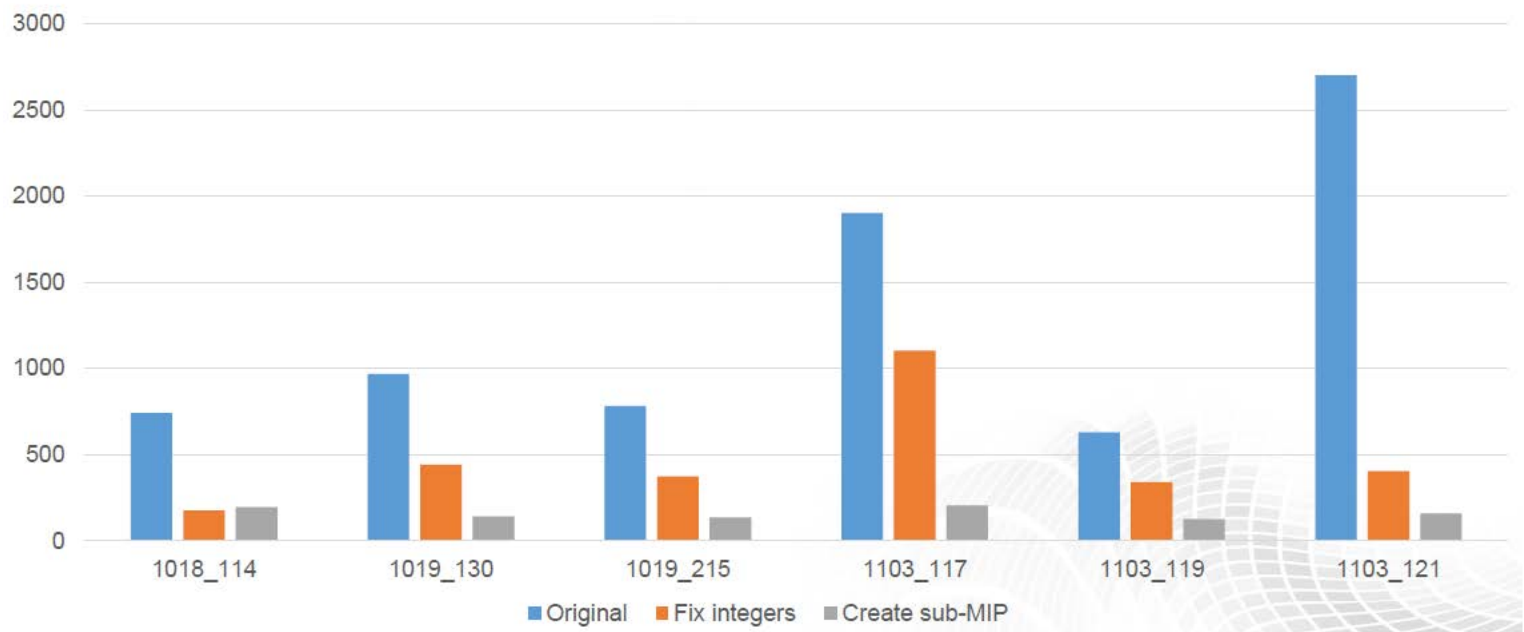
Proudly Operated by **Battelle** *Since 1965*



- Test the performance of HIPPO formulation on 60 cases against base model.
- Overall, we see time saving in HIPPO formulations.

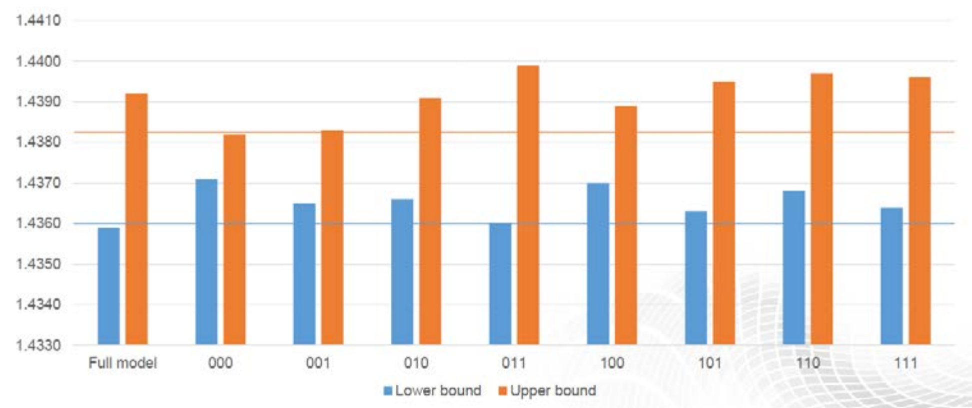
Heuristic – Variable Fixing with Reduced Cost

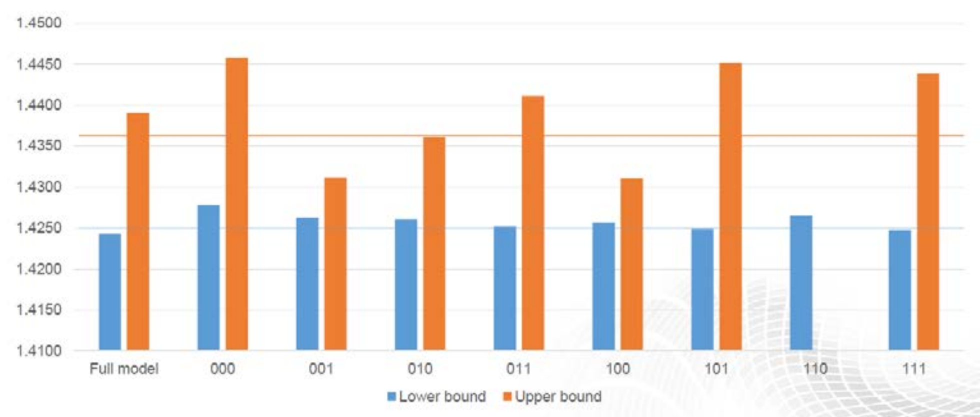
- Solve LPR, fix binary variables based on reduced costs
- Time to find a 0.1% gap solution
- Find high quality incumbent solutions



Heuristic – Variable Fixing with Reduced Cost

- Solve LPR, divide binary variables to blocks (3 blocks)
- fix binary variables based on reduced costs
- Create disjuction in each block
- Find high quality incumbent solutions and better lower bounds

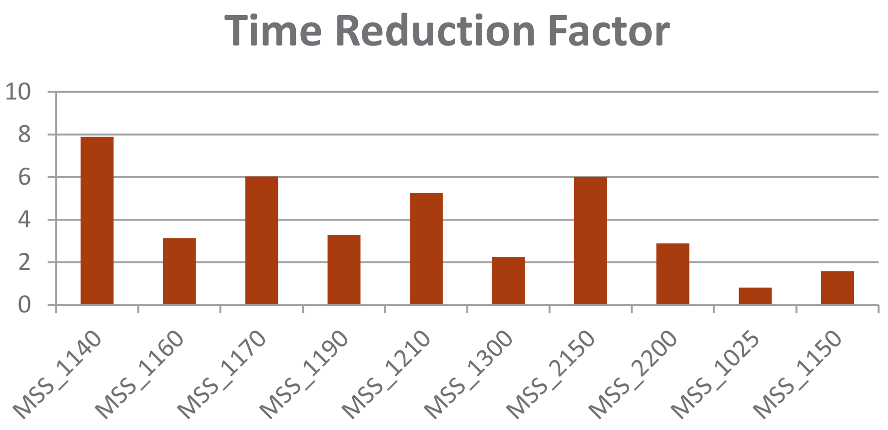


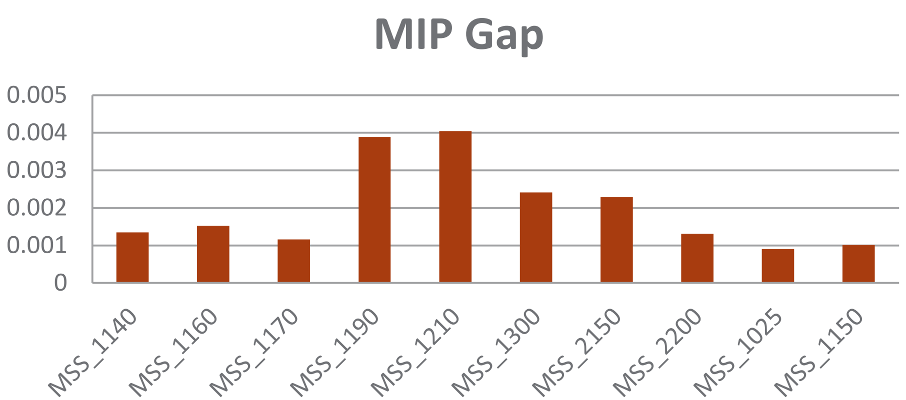


Heuristic – Variable Fixing with Rounding

Pacific Northwest NATIONAL LABORATORY
Proudly Operated by **Battelle** *Since 1965*

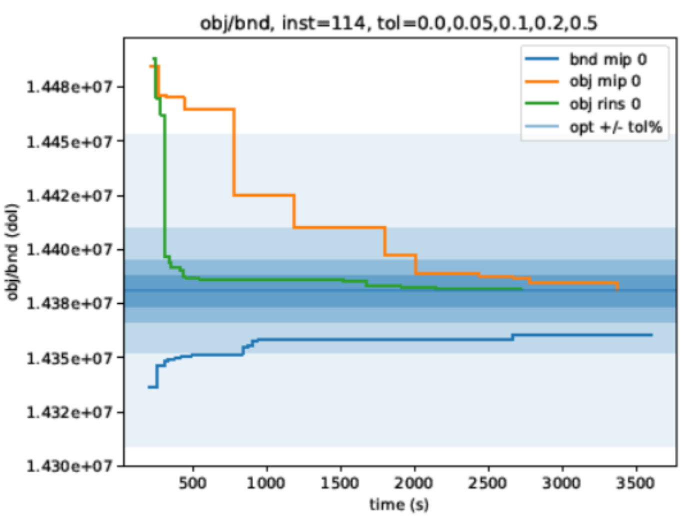
- Solve LPR, fixing binary variables by rounding.
- MIP gaps are based on true lower bounds
- Find high quality incumbent solutions
  - Accuracy of rounding depends on generator characteristics.
  - Formulation.
- Runtime includes solving LP and solving reduced MIP

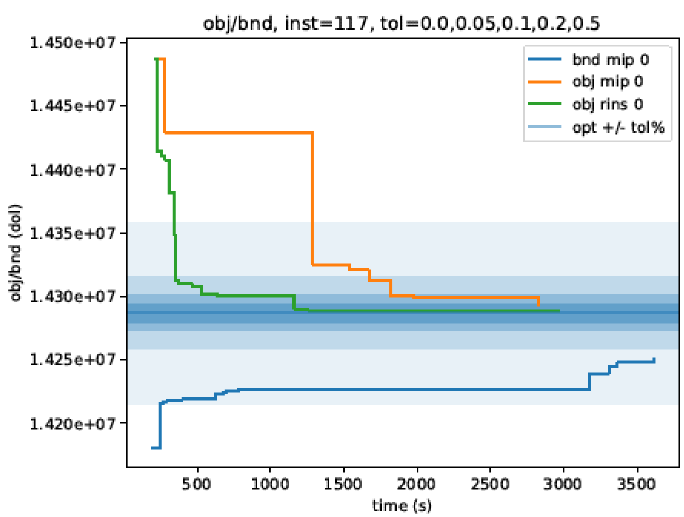




RINS-E

- Relaxation induced neighborhood search. Local optimizer for improving an incumbent solution.
- Implemented outside of GRUOBI branch-and-bound with callback to utilize hardware resource and add flexibility of defining neighborhood.
- Have good runtime for finding incumbent solution. Overall runtime is dominated by improving lower bounds.





Other Algorithms

- Polishing method
  - Identify generators which can potentially lead to more saving.
  - Local optimizer for improving solutions.
- ADMM
  - Generate feasible solutions and improving solutions.
  - Solve subproblems generated from other methods
- Cutting planes and strong branching
  - Getting tighter lower bound fast.

Concurrent Optimizer – Work in Progress

- The objective of Concurrent Optimizer is to be able to
  - Run individual algorithms simultaneously towards optimal solution
  - Orchestrate the execution of algorithms and monitor the overall progress.
  - Communicate among solvers to reach optimality quicker
    - Incumbent solutions
    - Variables to be fixed
    - Lower bound
    - Binding constraints

- Sounds counter-intuitive, but has advantages
  - Every algorithm is trying to optimally use its resources, does concurrency interfere with the execution?
  - Algorithms have deficiencies, these get corrected through concurrency
  - For example, polishing gets good upper bounds, but MIP can complement it with a good lower bound
  - When applied judiciously, concurrent optimizer adds more benefit than harm

Summary

Pacific Northwest NATIONAL LABORATORY
Proudly Operated by **Battelle** *Since 1965*

- Validation of HIPPO formulation
  - GE put tremendous amount of efforts into comparing solutions and tracking down causes.
- Performance evaluation
  - HIPPO formulation – comparable to base model and better in some cases.
  - Preliminary algorithm testing – most performance gains are by heuristics
  - MISO shared their past experiences, domain knowledge and developed algorithms.
  - GUROBI shared extensive knowledge how MIP behaves and ideas for improving performance and scoping exercises.