21, rue d'Artois, F-75008 PARIS
http://www.cigre.org

CIGRE US National Committee
2019 Grid of the Future Symposium

On Improving Efficiency of Electricity Market Clearing Software

Y. CHEN
Midcontinent ISO (MISO)
USA

F. PAN, J. HOLZER
Pacific Northwest National Lab
USA

SUMMARY

Wholesale electricity markets in the U.S. have brought significant benefits to society by maximizing social welfare while ensuring system security. Optimization models and algorithms are at the core of the software tools used to make this happen. In recent years, the Midcontinent Independent System Operator (MISO) has focused its attention on improving its market clearing software performance to enable further developments in its market products. This has largely entailed reducing solve time to ensure the timely commitment of resources and maintaining high optimality in the solutions. With rapid evolution in the power sector, MISO believes software performance will become increasingly important. The diversity of resources offering their services into MISO's markets is growing and the portfolio of existing resources and fuel types has shifted rapidly in recent years. In turn, resource modeling requirements are becoming more complicated, system constraints more intricate, and the volume of information greater. This paper introduces the research and development of the next generation market clearing software under the Department of Energy (DOE) Advanced Research Projects Agency–Energy (ARPA-E) project to develop high performance computer based optimization engines. The goal is to position Regional Transmission Organization or Independent System Operator (RTO/ISO) for future industry evolution.

KEYWORDS

Electricity market, mixed integer programming, security constrained unit commitment, security constrained economic dispatch, high performance computer, distributed computing, parallel computing

ychen@misoenergy.org

1. INTRODUCTION

The Midcontinent Independent System Operator (MISO) manages one of the largest electricity markets in the world. Its footprint includes 15 U.S. states and 65,800 miles of transmission, thousands of generators and over 130 GW of peak load. The size and complexity of MISO's system and markets create unique challenges for computational efficiency.

Since MISO launched its market in 2005, MISO has continued to make enhancements that require greater performance from the market clearing software. In tandem, MISO has continually worked to improve the performance of the market clearing software. The multi-stage market clearing process ranges from seven-day forward reliability assessment to four-second automatic generation control. The day-ahead (DA) market is the most computationally challenging process with about 98% resources being committed in this process. MISO has steadily reduced the DA market clearing window over the years even with the increased complexity and size of the market clearing model. MISO started with a six-hour DA clearing window in 2005 for an energy only market. The clearing window was reduced to five hours in 2007. In 2009, MISO started a co-optimized energy and ancillary service market with more an even more intricate SCUC and SCED model. MISO was able to further reduce DA clearing window to four hours and maintain the same clearing window with the south region integrated in 2013. Following the expansion of its market footprint and an uptick in virtual trading volumes, in early 2014, MISO again encountered increased computational performance needs for solving its day-ahead security constrained unit commitment (SCUC). MISO then identified bottlenecks in the DA clearing process and collaborated with GE Grid Solution to develop solutions to address these bottlenecks. The DA clearing window was successfully reduced to three hours in November 2016 [1][2]. It has greatly improved the coordination between electricity and gas markets and therefore enhanced the reliable operation of the market systems.

With the changing industry landscape, MISO believes RTOs/ISOs computational system performance will continue to be challenged. To prepare, MISO has continued to committed resources to explore further enhancements. For the near-term, MISO continues the research on improving resource modeling and mathematic formulations [3]. Methods on improving existing commercial solver performance through warm start and distributed solution process have shown very promising results [4]. With large transmission network, pre-screening on transmission constraints combined with warm start can further speed up the performance [5]. These enhancements allow more robust computational performance that enable the implementation of complicated market models such as configuration based combined cycle [3][6].

Besides finding ways to further enhance the usability and efficiency of existing optimization commercial solver, MISO has also partnered with Pacific Northwest National Laboratory (PNNL), Gurobi Optimization, GE Grid Solution, University of Tennessee and University of Florida to develop high performance distributed and parallel computing based SCUC and security analysis Simultaneous Feasibility Test (SFT) software. The project, entitled, High-Performance Power Grid Optimization (HIPPO), is funded in part by the Department of Energy's ARPA-E program and aims to achieve ten times speed up of the day ahead market clearing software.

The HIPPO's prototype concurrent solver is benchmarked with MISO production DA market clearing software. The technology developed under HIPPO can potentially be implemented in production market clearing engine to speed up the performance and enable future market enhancement. MISO also uses it as a research tool to evaluate future resource impact and market design options.

2. High performance computing based HIPPO concurrent optimizer for solving SCUC mixed integer programming (SCUC-MIP)

HIPPO's prototype concurrent optimizer includes fast concurrent SCUC-MIP and fast SFT solvers. MISO has a large footprint and needs to monitor large number of transmission constraints. Today, the MISO network model includes about 45,000 buses. The real world market clearing models usually do not include the full explicit network model due to the computation issues. Instead, the iterative approach between SCUC-MIP and SFT is used to check line capacity violation in base case and contingency with SFT and add those violations as constraints to SCUC-MIP. To avoid too many SCUC-MIP and SFT iterations, a list of transmission constraints based on historical loading and operational study are added into initial SCUC-MIP. They are referred as “watchlist” constraints. The number of “watchlist” constraint is around 200 per interval with total 36 intervals in MISO day-ahead market (DA) cases. The watchlist constraints can be for base case or contingency cases. The sensitivity of flow to bus injections are calculated ahead of time for the watchlist constraints. The left hand side of the watchlist constraints is modeled as the sum of the product of net injection at bus and the bus-to-line sensitivity. Watchlist transmission constraints coupled with large number of virtuals can introduce additional great performance challenge [1].

The concurrent optimizer in the HIPPO software includes three main components: model factory, algorithm factory and execution management.

Model factory. The model factory includes the most advanced formulations developed by the team and additional university partners.

The base SCUC generator is formulated with three binary variables (i.e., commitment, startup and shut down). Compared to SCUC model in the literature [7]-[9], MISO real world cases include the following special features and the HIPPO prototype MIP is formulated to reflect these needs.

1) Time varying parameters such as limits and ramp rates.
2) Resource minimum run time and minimum down time may be disabled for certain intervals to prevent gaming opportunities.
3) Resource maximum daily energy and maximum daily start constraints.
4) Resource regulating limits can be different from economic limits. Hence, regulation commitment variables may be required to choose the proper limits.
5) Large numbers of virtuals coupled with large numbers of transmission constraints can increase the solving time significantly [1]. The pricing node (PNode) aggregation formulation on transmission constraints [4] is implemented in HIPPO as the default transmission constraint formulation.
6) The convex envelope piece wise linear formulation in [13] is implemented to tighten the cost function.

Other than the default formulation (f0), several different formulations are implemented that may be configured to run concurrently:

- Formulation 1 (f1): two-period and three-period convex hull ramping constraints [10] implemented through callback cut generation.
- Formulation 2 (f2): ramping polytope and matching formulation in [11]-[12].
- Formulation 3 (f3): symmetry identification and anti-symmetry formulation [14]. There is on-going work to identify and handle approximate symmetry.

Algorithm factory

The HIPPO team explored many algorithms to take advantage of parallel computing with high performance computers. A set of neighborhood search methods turns out to be very successful in speeding up the solution performance. These methods identify the set of variables to be fixed and constraints to be excluded or set as lazy. By fixing variables or setting lazy constraints, the size of the original MIP optimization problem can be greatly reduced. These methods can reach high quality upper bound optimization solution much faster than MIP solver. These methods developed under HIPPO include:

i) Variable fixing (a1):

Fixing binary variables is based on linear programming (LP) relaxation solution and machine learning approaches were also introduced to improve the accuracy of the variable fixing. This method can only provide valid upper bound.

The solver vendor and team member, Gurobi, implemented a special version of variable fixing as a heuristic method inside the solver with an option parameter to turn it on or off. HIPPO can include this version to solve concurrently with other methods (a1_0).

ii) RINS-E (a2)
Fixing binary variables based on the difference between the incumbent solution and the linear programming (LP) relaxation solution. Compared to the general RINS in MIP solver, RINS-E in the HIPPO prototype is separated from the branch-and-bound (B&B) process in MIP solver and can be solved in parallel to the main B&B procedure. In additional, RINS-E has the flexibility to applied domain specific strategies to improve the performance such as handling small virtuals and transmission constraints. This method can only provide valid upper bound.

iii) Polishing method (a3)
This method can start from any repaired previous commitment solutions (i.e., initial commitment) to identify "out-of-money" resources and lazy transmission constraints based on the initial commitment. This information can be used as hints to fix binary variables and set lazy constraints for transmission. A method is also developed to fix large percentage of virtuals based on the LP solution around the neighborhood. This method can only provide valid upper bound solution. The hints can also be applied to the full MIP problems for warm start (a3_0). This approach can provide both valid upper bound and lower bound.

The team also tested decomposition methods such as alternating direction method of multipliers (ADMM) and Bender's Decomposition. Our existing test results showed lacking of performance and difficulty to converge for the MISO SCUC. For example, in the HIPPO prototype version of ADMM, the sub-problems can be solved in a few seconds with multi-threading. But it is difficult to reach a solution with 0.1% optimality gap for the Mixed Integer Program (MIP). Overall for the set of MISO production DA SCUC cases, the team has not identified decomposition methods which can show significant advantages over Gurobi MIP solver.

Execution management

The HIPPO prototype concurrent optimizer is implemented to manage the execution and communication among individual algorithms through Message Passing Interface (MPI). The software is written with Python. It can be configured to run on single server or on high performance computer.

HIPPO can be configured to run with any combination of the formulations or algorithms in the library. A master session manages the execution and collects the best upper bound and the best lower bound from the concurrent solvers. It terminates the solvers as soon as the time limit or the MIP gap tolerance is reached.

Besides reporting solution to the master session, the solution can also be exchanged between different concurrent solvers. With the current implementation, we observe the performance improvement with 29 nodes used for HIPPO, without SFT iterations. Further scalability testing is needed to investigate the benefit of additional computing nodes.

HIPPO is built on a special version of Gurobi8.1.0 with variable fixing a1_0. The configuration with 29 nodes (without SFT) is set as shown on Fig. 1:

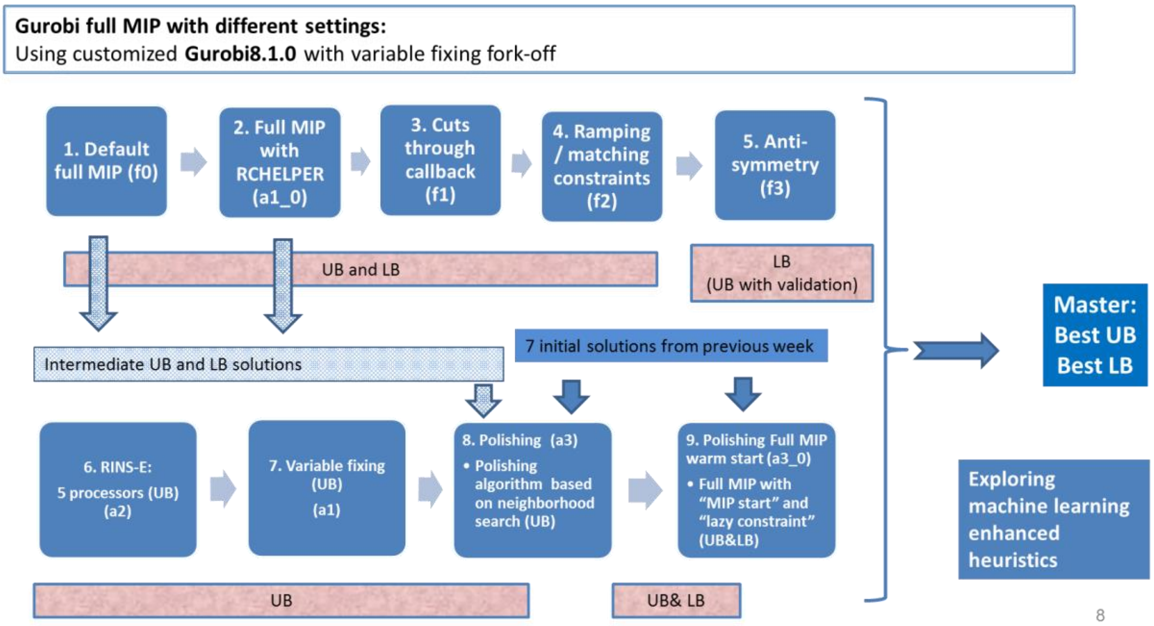

Figure 1 HIPPO concurrent solver (29 nodes to solve MIP without SFT)

The HIPPO SCUC-MIP performance improvement (without SFT) is significant. Using production MIP stopping criterion of 0.1% relative MIP gap, HIPPO concurrent SCUC-MIP is compared to production SCUC-MIP. The results is shown in Fig. 2. The median speedup ratio is 2.63 times and the average speedup ratio is 3.45 times. The speedup ratio is mostly over five times for the set of hard cases that require over 2,000 seconds with production SCUC-MIP.

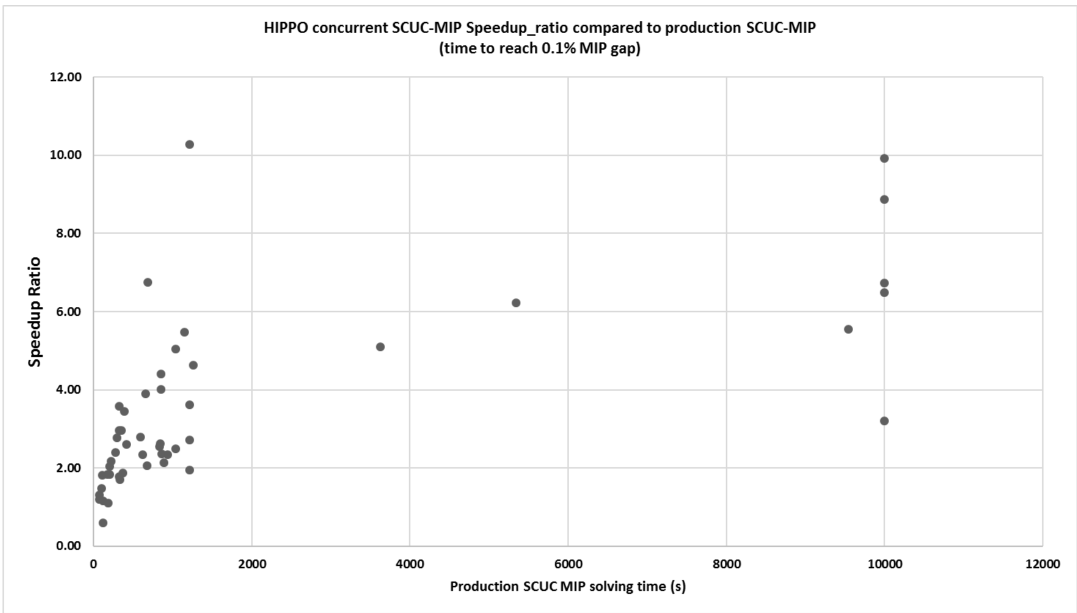

Figure 2 HIPPO SCUC-MIP speedup ratio without SFT

3. HIPPO security analysis software SFT

SCUC-MIP includes a set of "watchlist" constraints as the initial representation of the power grid flow limits. SCUC-MIP solution needs to be checked to make sure other base case or N-1 contingency flows are not violated. SFT is a procedure to check flow violations by solving DC power flow in base case and contingency cases. SFT also computes sensitivities between bus injection and line flow for the violated transmission constraints. The violated constraints are added to SCUC-MIP similar to watchlist constraints.

MISO day ahead SFT includes about 1,000 pre-screened contingencies with about 10,000 monitored branches. The network topology can vary by intervals due to outages. Current production SFT takes about 10 minutes to solve for each given SCUC solution (i.e., 36 interval injections). Under HIPPO, a much faster SFT is developed to solve in ten to approximately 20s for 36 interval injections with 1,000 contingencies and 10,000 monitored branches.

The HIPPO prototype SFT is coded in Python with open source linear algebra libraries. It uses Sherman-Morrison-Woodbury formula to treat contingencies, instead of partial re-factorization used by current method. The much faster SFT computation time enables network security evaluation within SCUC algorithms.

Using default full MIP (f0) as an example, SFT matrix preparation can be run in parallel with SCUC MIP pre-solve and root relaxation. Current production software creates up to 36 base matrices, one for each interval with outage changes from previous interval. Pre-processing 36 large matrices can take very long. With HIPPO, the 36 matrices can be allocated to multiple nodes.

After pre-processing, the HIPPO prototype SFT can run in ten to 40 seconds with three nodes, five seconds with six nodes (1,000 contingencies times 36 intervals). Such fast SFT make it possible to run through the Gurobi solver callback API. After the Gurobi solver finds each new incumbent solution, the MIPSolution callback can send the solution to SFT to check violations and send back sensitivities and limits. The new transmission constraints can be added to the Gurobi solver as "cb_lazy" constraints. The Gurobi solver can incorporate these constraints into the searching process afterwards. It'll only report valid upper bound solution when there is no new SFT constraints. With this approach, the Gurobi solver can finish SCUC-MIP and SFT in one pass. There is no need to run SFT after SCUC-MIP finishes.

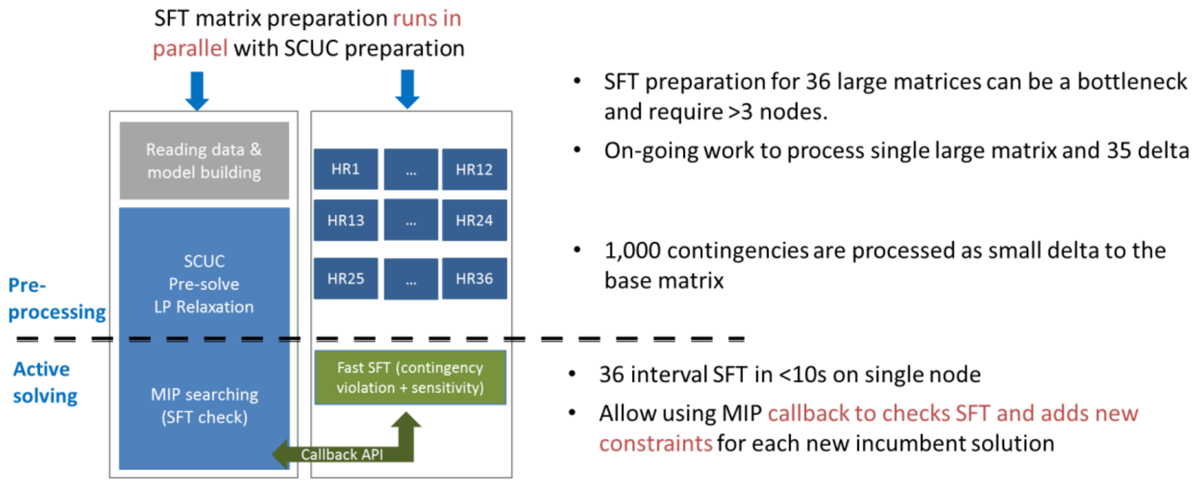

Figure 3 HIPPO SCUC-MIP and SFT Configuration

Table 1 compares full MIP with SFT callback run for one case under different number of nodes and processors per node configuration. With six SFT nodes and six parallel processors per node, each node pre-processes six matrices. SFT pre-processing finishes before the first incumbent solution from MIP solver. It takes 18.88s to solve SFT for the first incumbent solution, which includes:

- Solving DC power flow and contingency analysis for 36 intervals
- Computing sensitivities for 261 violated constraints
- Adding 261 constraints to MIP solver through callback

The MIP solver continues and check SFT whenever there is a new incumbent solution. After the first one, the number of violations from SFT is usually only a few. MIP with SFT finishes in 798 seconds.

However, if we configure HIPPO with only one SFT node, SFT pro-processes 36 matrices in one node. The pre-processing time increases significantly. With one node and 36 parallel processors per node, after MIP solver calls SFT for the first incumbent solution, it takes 441.19s to get SFT solution back, mainly waiting for SFT pre-processing to finishing. Afterwards, solving 36 interval SFT one node is only four to five seconds. The total time is 1,111s.

With three SFT nodes and 12 parallel processors per node, the time for the first SFT is reduced to 49.83 seconds. The total time is 879 seconds.

In reality, the differences between the 36 intervals are driven by less than a few hundred outages. There is no need to process 36 large matrices. We are developing improved approach to handle outages similar to contingencies with a small delta change to the full matrix.

Table 1 MIP solver with SFT callback comparison

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

For the same case, current production software takes 5,185s to solve with 3 MIP-SFT iterations. Each SFT run takes about ten to approximately 20 minutes.

MIP1 (s)	SFT1 (s)	SFT_AddConstr_1	MIP2 (s)	SFT2 (s)	SFT_AddConstr_2	MIP3 (s)	SFT3 (s)	SFT_AddConstr_3	MIP4 (s)
398	1212	211	623	764	10	731	768	5	689
Total Time (s)	5185								

Currently we have fully integrated HIPPO SFT with methods f0, a1/a1_0, and a3/a3_0. We ran the same set of cases with these methods, each paired with its own SFT (1 node and 12 parallel processors per node. The speed up ratio between HIPPO and the production software is shown in Figure 4. All except one case (2.7 times) reaches greater than 4 times speedup. The median speedup is 9 times and the maximum speedup is 20 times.

4. R&D prototype for future market clearing problems

MISO is in the process of design future market clearing system. HIPPO has shown great potential in meeting future computational needs. However, the industry may not move to use high performance computer (HPC) immediately. HIPPO is built with the flexibility to run on single server or HPC. The team plans to work on the most cost effective hardware configuration in between and the path to bring HIPPO technology to production type of environment.

Meanwhile, MISO R&D team is using HIPPO as a prototype tool to study new market rule and market system design options. A case library with over 120 historical cases has been built and used for these studies. The list of research projects includes:

a. Preparing for future resources types such as hybrid plants and distributed energy resources (DER)

b. Evaluation of market rules and software performance, such as smaller day ahead market intervals, enhanced combined cycle and pumped storage optimization
c. Watchlist constraint pre-screening
d. Pricing study
e. Historical data / machine learning

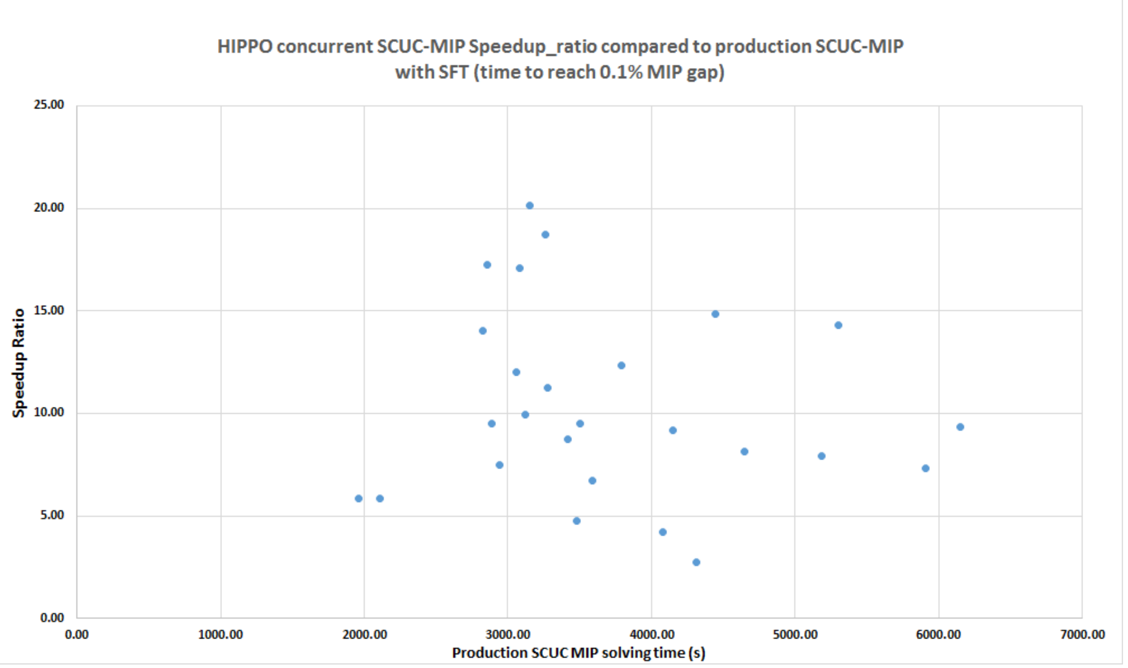

Figure 4 HIPPO SCUC-MIP speedup ratio with SFT

5. Conclusion

This paper introduces the HPC based HIPPO concurrent SCUC-MIP with SFT solver sponsored by DOE ARPA-E to address future computational challenges for electricity market clearing. With distributed concurrent method, HIPPO SCUC-MIP has multiple advanced formulations and fast heuristic methods to speed up MIP solution time. With parallel computing, HIPPO SFT can be configured to run on multiple nodes and multiple processors within each node. Extremely fast SFT allows efficient integration between MIP solver and SFT. Computational results on MISO cases shows 2.7 to approximately 20 times with a median speedup of nine times.

BIBLIOGRAPHY

[1] Y. Chen; A. Casto; F. Wang; Q. Wang; X. Wang; J. Wan, "Improving Large Scale Day-Ahead Security Constrained Unit Commitment Performance," *IEEE Trans. Power Syst.*, vol. 31, no. 6, pp. 4732-4743, Nov. 2016
[2] Y. Chen; F. Wang; J. Wan; F. Pan, “Developing Next Generation Electricity Market Clearing Optimization Software, IEEE PES General Meeting, Portland, OR, 2018
[3] Y. Chen, F. Wang, “MIP Formulation Improvement for Large Scale Security Constrained Unit Commitment with Configuration based Combined Cycle Modeling”, Electric Power Systems Research, Vol. 148, pp.147-154, July 2017.
[4] Yonghong Chen, Fengyu Wang, Yaming Ma, Yiyun Yao, “A Distributed Framework for Solving and Benchmarking Security Constrained Unit Commitment with Warm Start”, IEEE Transactions on Power Systems, under review.

[5] S. Zhang, H. Ye, F. Wang, Y. Chen, S. Rose, Y. Ma, "A Data-aided Security Constraint Prescreening Technique and Application to Real-world System", North American Power Symposium (NAPS), 2019, under review
[6] C. Dai, Y. Chen, F. Wang, J. Wan, L. Wu, "A Tight Configuration-Component Based Hybrid Model for Combined-Cycle Units in MISO Day-Ahead Market", IEEE Trans. Power Syst., Vol. 34, no. 2, pp. 883-896, March 2019
[7] D. Rajan and S. Takriti, "Minimum up/down polytopes of the unit commitment problem with start-up costs", IBM Res. Rep, 2005.
[8] C. Gentile, G. Morales-Espana, and A. Ramos, "A tight MIP formulation of the unit commitment problem with start-up and shut-down constraints", EURO Journal on Computational Optimization, 2016.
[9] J. Ostrowski, M. F. Anjos, and A. Vannelli, "Tight mixed integer linear programming formulations for the unit commitment problem", IEEE Trans-actions on Power Systems, vol. 27, no. 1, p. 39, 2012.
[10] K. Pan and Y. Guan, "A polyhedral study of the integrated minimum-up/-down time and ramping polytope", 2015, https://arxiv.org/abs/1604.02184
[11]B. Knueven, J. Ostrowski, and J.Wang, "The ramping polytope and cut generation for the unit commitment problem", INFORMS Journal on Computing, Vol. 30, 2018.
[12] Bernard Knueven, James Ostrowski, and Jean-Paul Watson. A novel matching formulation for startup costs in unit commitment, 2018. http://www.optimization-online.org/DB_FILE/2017/03/5897.pdf
[13] Bowen Hua and Ross Baldick. A convex primal formulation for convex hull pricing. IEEE Transactions on Power Systems, 32(5):3814–3823, 2017.
[14] J. Ostrowski, M. Anjos, A. Vannelli, Modified orbital branching for structured symmetry with an application to unit commitment, Mathematical Programming, vol. 150, 2015.
[15] J. Holzer, Y. Chen, F. Pan, E. Rothberg, A. Veeramany, "Fast evaluation of security constraints in a security constrained unit commitment algorithm", FERC Technical Conference June 25-27, 2019, available online: https://www.ferc.gov/CalendarFiles/20190626080612-4%20-%20FERC%202019%20Holzer%20fast%20evaluation%20of%20security%20constraints.pdf