A novel optimization approach for sub-hourly unit commitment with large numbers of generators and virtuals

Jianghua Wu[1], Bing Yan[1], Mikhail Bragin[1],
Yonghong Chen[2], Peter B. Luh[1]

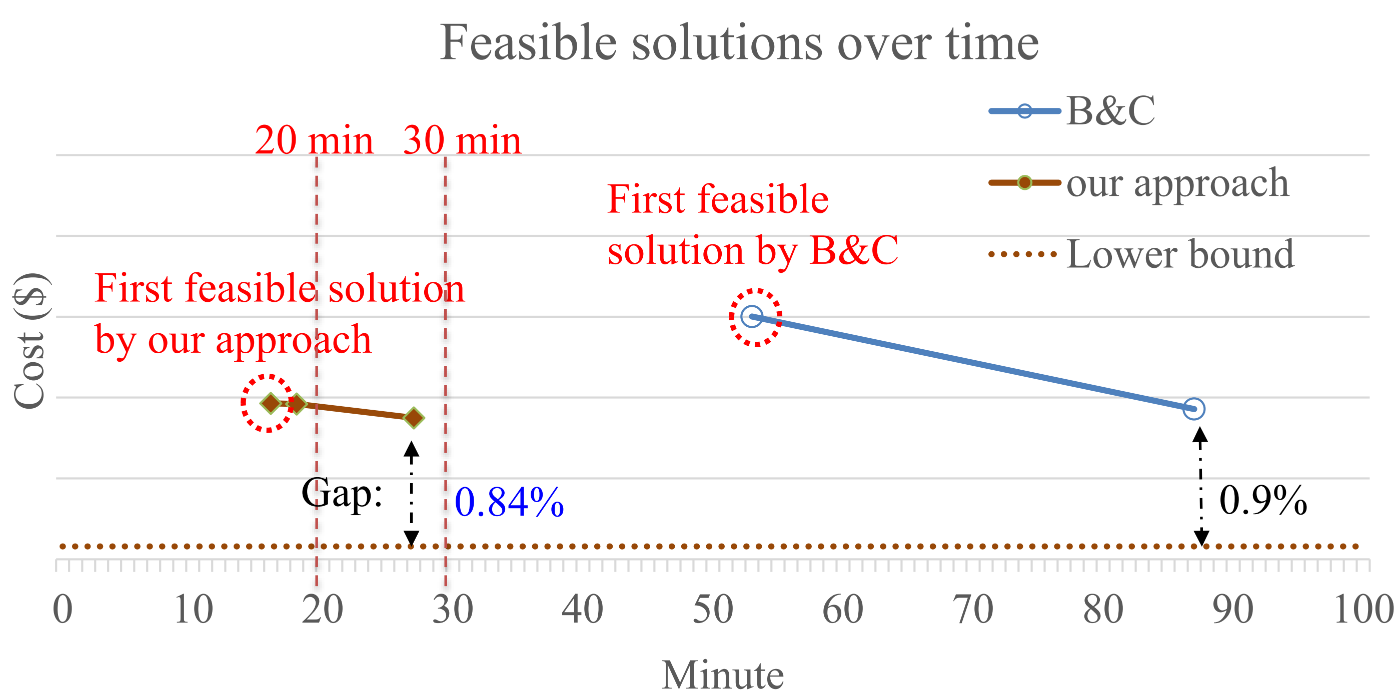
[1]Electrical & Computer Engineering, University of Connecticut
[2]Midcontinent Independent System Operator (MISO)

Sub-hourly UC and virtuals

- Hourly unit commitment is becoming insufficient for operators to manage power systems efficiently [1]
- Sub-hourly UC is a way to improve the reliability of power systems. Comparing to hourly UC,
  - Increased number of periods $\Rightarrow$ larger size
  - Reduced unit ramping capabilities per period $\Rightarrow$ increased complexity
- For certain systems, e.g., MISO's, there are many virtuals:
  - Related to continuous variables only and with linear costs
  - Coupled with transmission constraints $\Rightarrow$ increased computation

[1] "FERC Order 764," Available: https://www.ferc.gov/whats-new/comm-meet/2012/062112/E-3.pdf

- Consider a MISO's problem with 1,100 generators and 15,000 virtuals over 36 hours with 15 minutes as time intervals
- State-of-art branch-and-cut (B&C) does not find a quality solution within 50 minutes because of the difficulties mentioned previously
- Our novel decomposition and coordination approach finds a near-optimal solution in 20 minutes



Overview of the presentation

- Problem formulation
- A novel decomposition and coordination approach
  - Obtain subproblems' "good enough" solutions without formally pursuing optimality
- Virtuals included in each subproblem
- Parallelization
- Numerical testing results

Problem formulation

- Sub-hourly UC: the same formulation as hourly UC

$$\min_{\substack{p,u,\delta,\\x,\ldots}} \sum_{t\in T}\left[\sum_{i\in G}\boxed{(C_{it}^{gen}p_{it}+C_{it}^{NL}u_{it}+C_i^{start}\delta_{it})}+\sum_{i\in VT}\boxed{C_{it}^{virtual}x_{it}}+\boxed{\ldots}\right] \qquad (1)$$

Generation cost, no-load cost and start-up cost

Cost of virtuals

Other costs, e.g., reserve cost

- Subject to standard
  - unit-level constraints, e.g., minimum up/down time, ramp rate and so on
  - system-wide constraints, e.g., demand, reserve and transmission capacity

A novel decomposition and coordination approach

- Decomposition and coordination - Lagrangian Relaxation (LR)
  - A price-based "dual" approach ⇒ reduce complexity exponentially
  - Major difficulties:
    - Solve all subproblems ⇒ time-consuming
    - Suffer from major zigzagging of multipliers
    - Need to guess the optimal dual value q^*

⇒Death to LR?

- Surrogate LR (SLR): Overcame all above difficulties
  - Surrogate optimality condition ⇒ guarantee convergence
    - Satisfied after solving one subproblem to update multipliers
    - Obtain smoother directions with much less effort
  - Stepsizing rule without requiring q^*
- Surrogate Absolute-Value Lagrangian Relaxation (SAVLR)
  - Accelerate convergence by adding absolute value penalties
  - A subproblem may require a long time to solve

- The original problem is decomposed into subproblems by
  - Dividing generators into subproblems by areas
  - Including all virtuals in each subproblem (discuss later)
- The objective function of subproblem j at iteration k :

$$\min_{\substack{\lambda^{D,k},p^k,u^k\\ \delta^k,x^k}} \sum_{t\in T}\left[\sum_{i\in G_j}(C_{it}^{gen}p_{it}^k + C_{it}^{NL}u_{it}^k + C_i^{start}\delta_{it}^k) + \sum_{i\in VT} C_{it}^{virtual}x_{it}^k\right] \quad (3)$$

$$\boxed{+\lambda_t^{D,k}\Big(\sum_{i\in G_j}(p_{it}^k)+\sum_{i\notin G_j}(p_{it}^{k-1})+\sum_{i\in VT}x_{it}^k - D_t\Big)} + \boxed{\frac{c}{2}\left|\sum_{i\in G_j}(p_{it}^k)+\sum_{i\notin G_j}(p_{it}^{k-1})+\sum_{i\in VT}x_{it}^k - D_t\right|}$$

Demand constraints relaxed by multipliers $\lambda_t^{D,k}$

Absolute value penalties for demand constraints violations

  - For each iteration, the simple surrogate optimality condition is checked to guarantee algorithm convergence

$$\boxed{L(\lambda_J^{D,k}, p_J^k, p_{-J}^{k-1}, u_J^k, \delta_J^k, x_J^k)} < L(\lambda_J^{D,k}, p_J^{k-1}, p_{-J}^{k-1}, u_J^{k-1}, \delta_J^{k-1}, x_J^{k-1}) \quad (4)$$

Surrogate dual value

- Subproblems are normally solved by B&C in SAVLR

- However, SAVLR may still require a long CPU time
  - Long subproblem solving time (≈160s)
- Can this be improved? Yes. How?
  - A specific property of SAVLR:
    - Subproblem solutions only need to be "good enough" to satisfy the surrogate optimality condition

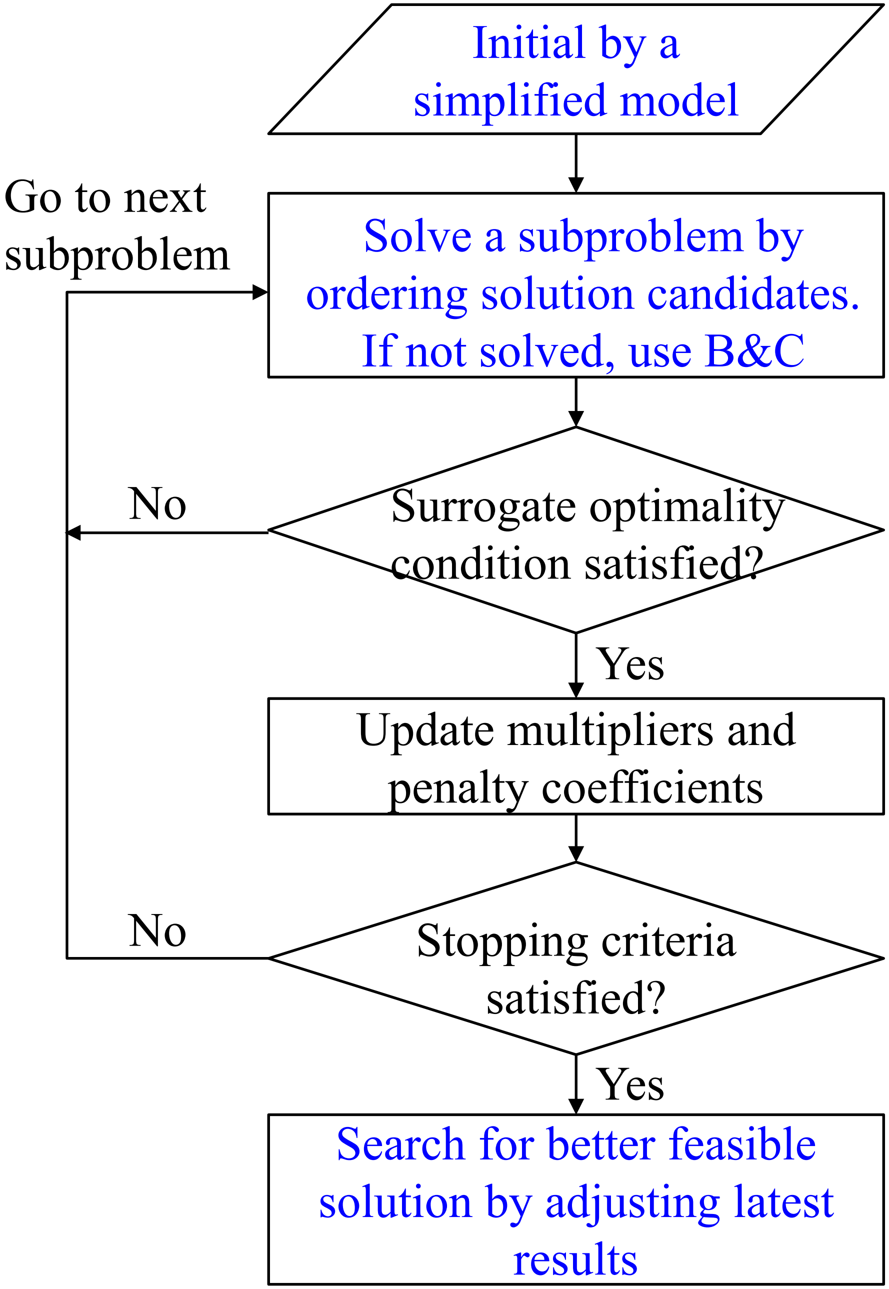
⇒ A new optimization approach inspired by Ordinal Optimization

- Obtain subproblems' "good enough" solutions without formally pursuing optimality
  - Solve simplified models or modify previous solutions ⇒ solution candidates ⇒ obtain subproblem solutions quickly by "ordering" candidates based on surrogate dual values
  - Use B&C to solve subproblems when a good enough solution cannot be obtained (rarely)

⇒ Reduce the subproblem solving time significantly (160s → 50s)

- When the surrogate optimality condition is satisfied:
  - Even though the quality of subproblem solutions may not be as good as those obtained by B&C, the overall solution quality is not affected
  - The total number of iterations needed is about the same as that in SAVLR+B&C

Flow chart



- Subproblems are generally solved by ordering solution candidates

$\Rightarrow$ Reduce the solving time significantly

- Can this be further improved? How?

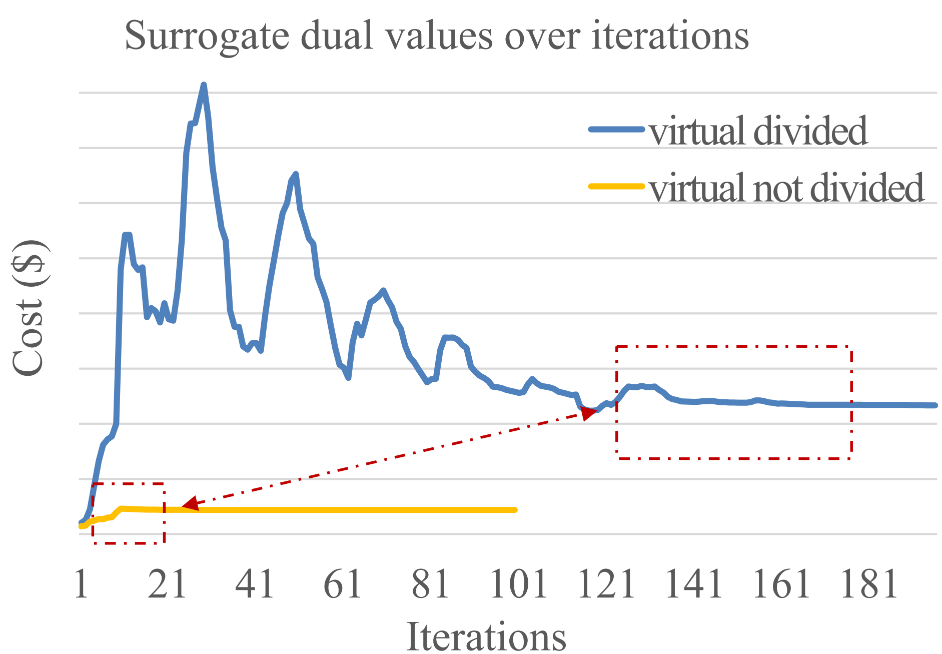
Virtuals included in each subproblem

- To obtain fast convergence, virtuals are not divided into subproblems

virtuals are related to continuous variables only and with linear costs ⇒ virtuals and multipliers are very sensitive to each other ⇒

virtuals may deviate drastically across iterations

Multiplier values change significantly



- Divide virtuals into subproblems ⇒ slow convergence
- Include virtuals in each subproblem ⇒ fast convergence

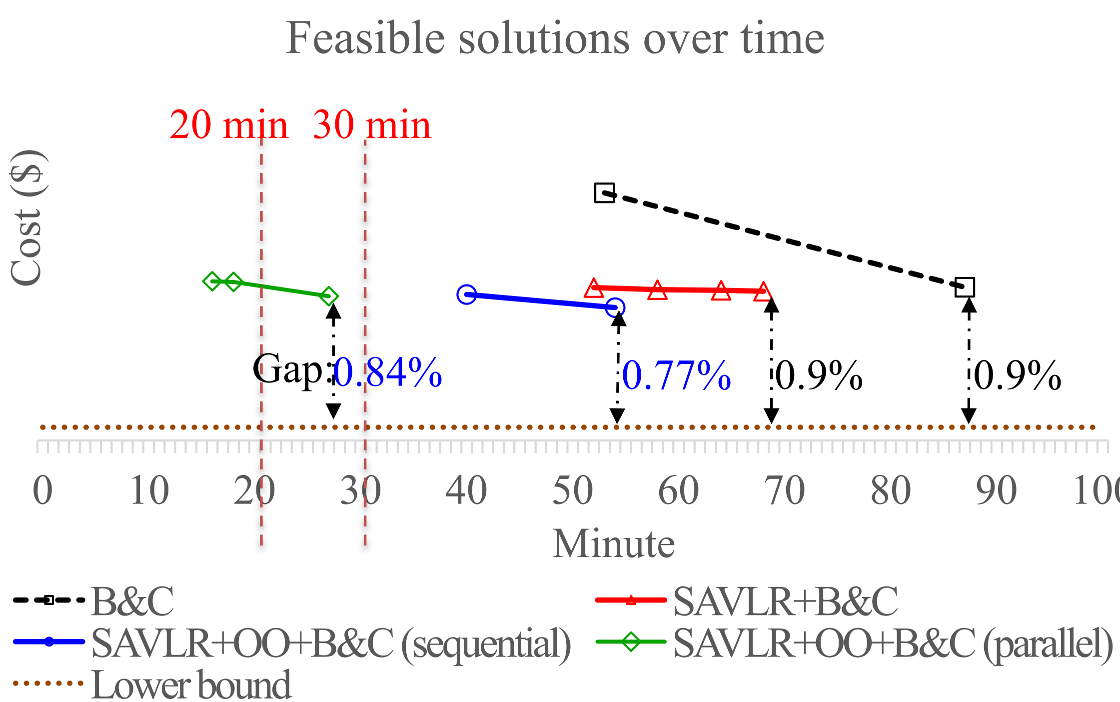
Parallelization

- Parallelization is used to further accelerate the approach
  - Solve a subset of subproblems at a time $\Rightarrow$ avoid major zigzagging of multipliers as observed in LR
  - Use the combination of subproblem solutions that meets the surrogate optimality condition $\Rightarrow$ guarantee convergence
  - Adjust values of virtuals based on the merged results $\Rightarrow$ reduce constraint violations

Numerical testing results

- Testing on multiple MISO's UC problems, which are hard even for hourly UC (> 20 min)
  - With about 1,100 generators and 15,000 virtuals
  - Over 36 hours with 15 minute time intervals
- Implementation
  - Python 2.7 and Gurobi 7.5.0 on HIPPO platform
  - HP EliteBook with MISO's server, Intel® Core (TM) i7-7600U CPU @ 2.80 GHz RAM 16 GB
- To measure solution quality, the best known lower bounds are obtained by B&C in advance

Case 1: 1105 generators; 15843 virtuals; 227 transmission lines				
	Solving Time (s)	Total Time (s)	Iteration #	Gap (%)
B&C	5211	5443	N/A	0.9
SAVLR+B&C	2985	4086	20	0.9
SAVLR+OO +B&C (sequential)	1484	3237	20	0.77
SAVLR+OO +B&C (parallel)	979	1639	20	0.84



- Our approach never uses B&C to obtain the subproblem solution in this case
- Our approach uses the same number of iterations to obtain the feasible solution and reduces the subproblem solving time from 162s to 53s
- Our approach outperforms pure B&C and SAVLR+B&C

Case 2: 1109 generators; 16504 virtuals; 220 transmission lines			
	Solving Time (s)	Total Time (s)	Gap (%)
B&C	2548	3000	2
Our approach (parallel)	990	1409	1.1
Case 3: 1118 generators; 14955 virtuals; 226 transmission lines			
	Solving Time (s)	Total Time (s)	Gap (%)
B&C	2787	3120	4.31
Our approach (parallel)	638	993	3.09
Case 4: 1102 generators; 14482 virtuals; 235 transmission lines			
	Solving Time (s)	Total Time (s)	Gap (%)
B&C	3089	3600	76
Our approach (parallel)	619	1016	1.6

- The parallel version of our approach obtains near-optimal solutions efficiently in multiple hard cases, and the results demonstrate that the approach is robust

Conclusion

- SAVLR+OO+B&C is a vast improvement over traditional LR
  - New optimization concept!
    - **Without formally pursuing optimality of subproblems** ⇒ drastically reduces subproblems' solving time
  - Virtuals included in each subproblem ⇒ fast convergence
  - Efficient parallelization ⇒ with further speedup
- Exciting results
  - Our approach takes the same number of iterations as those of SAVLR to obtain feasible solutions of the same quality but with much reduced subproblem solving time
  - Our approach obtains near-optimal solutions within 20 minutes while B&C could not find a quality solution within 50 minutes
- The approach has potential to solve other complex problems in power systems and beyond

Thank you!