

A Data-aided Security Constraint Prescreening Technique and Application to Real-world System

Shubo Zhang*, Hongxing Ye*, Fengyu Wang[†], Yonghong Chen[†], Stephen Rose[†] and Yaming Ma[†]

*Department of Electrical Engineering and Computer Science, Cleveland State University

[†]Midcontinent Independent System Operator

Email: s.zhang80@vikes.csuohio.edu; h.ye@csuohio.edu; {fwang, ychen,srose,yma}@misoenergy.org

Abstract—Security constraints are necessary in unit commitment problems to guarantee reliable commitments. However, only a small subset of security constraints can bind in the unit commitment problem. This paper presents a data-aided prescreening approach that could identify and remove non-dominating security constraints. This approach takes advantage of both deterministic and statistical methods, which leverages the historical data and linear programming (LP) solution. Lazy constraints are used in optimization solver to maintain solution integrity. Case studies presented here use Midcontinent Independent System Operator (MISO) data, and simulation results show that the proposed prescreening approach significantly improves computational performance.

Index Terms—Security Constraint, Transmission Congestion, Optimal Power Flow, Lazy Constraint, Unit Commitment.

I. INTRODUCTION

SECURITY constrained unit commitment (SCUC) is discussed extensively in the literature and well implemented in daily power system operation by independent system operators (ISO) and Regional Transmission Organizations (RTO) [1]. It is a challenging task to obtain optimal solution in an acceptable time due to the enormous size of the problem. SCUC's objective is to maximize social welfare subject to a variety of constraints, such as generator physical limits, power balance, reserve requirements, and transmission security constraints [2].

SCUC, by nature, is a nonconvex and large scale mixed integer optimization problem. It belongs to the np-hard problem class. Recent improvements in modeling, optimization solver, and efficient algorithm enable system operators to obtain optimal or near-optimal solutions. SCUC can be generally solved by Lagrangian Relaxation (LR) and Mixed Integer Linear Programming (MILP), performance comparison can be found in [2]. MISO has adopted MILP approach for the Co-optimized Energy and Ancillary Service Market since 2009. Day Ahead Market (DAM) is a financial market, and the market-clearing process requires solving SCUC problem for all planned operation periods. It is crucial to obtain a near optimal solution (MILP gap under 0.1%) while meeting the time requirement (1200 seconds) [3].

In DAM, security constraints are defined as power flow transmission constraint for transmission lines considering N-1

security rule according to North American Electric Reliability Corporation (NERC). There are over 15000 security constraints formulated for MISO's DAM. Incorporating all security constraints in the model could cause computational intractability, which means optimality and solution time requirements cannot be guaranteed. Many research efforts have gone into improving SCUC computational performance. Decomposition approaches, such as Benders decomposition and Lagrangian decomposition, divide the original problem, which is difficult to solve, into smaller pieces, acceleration comes from solving the smaller problems in parallel. However, in general, decomposition approaches may have either bloating problem or convergence problem. Data driven approach, such as warm start, gives the solver a starting node, which may help in some situations. However, good quality of warm start information can help filter out inferior nodes and poor quality of warm start information may hurt more than they help. It is not always optimistic to obtain high quality of warm start information. Based on the observation of computationally challenging cases in MISO, security constraints are the primary driving factor of slow the computational performance of SCUC. Models with fewer security constraints can generally reach acceptable optimality gap with less time.

Reducing non-binding hard constraints often help reduce solution time for MILP problems. The goal of this research is to effectively reduce the size of security constraint set in the SCUC model and improve solution quality, while maintain solution integrity. From experience, only a small fraction of security constraints are binding. MISO has identified over 200 routinely binding and critical security constraints per interval, known as "watchlist". Additional constraints may be identified and added to watchlist through the interaction between Simultaneous Feasibility Test (SFT) and DAM Security Constrained Economic Dispatch (SCED) [3].

The overwhelming number of security constraints have attracted much attention in literature. Umbrella constraint concept is introduced in [4] and [5]. The umbrella constraints are dominating constraints defined as the minimum set of constraints that shapes the feasibility region of the original problem. Binding constraints is a subset of umbrella constraints, which can only be determined after solving the original MILP problem. Authors in [6] propose an efficient method to eliminate non-active constraints in standard SCUC, and [7], [8] further take uncertainty into consideration. Authors

This work is in part supported by U.S. National Science Foundation Project Number ECCS-1711217.

in [9] developed methods to eliminate large numbers of linear security constraints in the high-dimensional mixed-integer SCUC problem. Heuristic approaches are applied to remove potentially non-binding constraints in [3] and [10], which addresses the conservatism due to large number of constraints maybe non-binding but are dominating in the reduced constraint set. It is seen that warm start and setting proper lazy constraint could improve the SCUC solution performance [11], [13]. Our work in this paper is an extension from [6], [7], [8].

In this paper, we aim to reduce SCUC problem size by reducing security constraints set in the SCUC formulation. We propose a two-step prescreening approach to identify and eliminate non-dominating security constraints. Step 1 is a numerical method that employs necessary conditions to identify the potential dominating security constraints. Step 2 solves a LP problem to find the truly dominating constraints that further shrinks the pool of potentially dominating security constraints. Lazy constraint resembles a call back function within the optimization solver that checks constraint violation. If set inaccurately, the performance will degenerate due to the overhead that put back the violated constraints. Motivated by this, we introduce a data-aided lazy constraint setting to further improve the performance. The contributions of this paper are three-fold. First, we propose a data-driven approach to determine the confidence interval of net power injection for the calculation of power flow. Second, we employ lazy constraints to resolve potential inaccuracy issue caused by data driven approach. Third, we carry out simulations and validate the performance using the real data from MISO.

The paper is organized as following. Section II, a hybrid prescreening of the two-step deterministic and statistical approaches are presented. Historical data and lazy constraints are employed to tighten the security constraint pool in the optimization model. In Section III, proposed method is tested on the MISO SCUC model. Section VI concludes the paper.

II. SECURITY CONSTRAINT PRESCREENING PROCEDURE

It is a challenge to deal with the large number of security constraints in real world systems. Some ISOs/RTOs develop special procedures to identify creditable security constraints and add only those constraints into the DAM SCUC model. Iterative approaches are often employed to conduct Simultaneous Feasibility Test (SFT) so that any additional security constraint causing violation can be added to the model eventually. For instance, MISO evaluates outages and other relevant information to build the watchlist, which is a much smaller subset of the full set of security constraints. However, the system operator may still confront situations where optimization engine cannot find an acceptable solution within the time limit. There is an emerging and practical need to find acceleration techniques to improve the optimization performance. In this section, we present a two-step approach to build the watchlist.

A. Two-Step Prescreening

The power flow f_l in line l can be calculated based on equation below

$$f_l = \sum_m \Gamma_{l,m} P_m^{\text{inj}},$$

where $\Gamma_{l,m}$ is the power transfer distribution factor (PTDF) for line l with respect to bus m , and P_m^{inj} is the net power injection at bus m . Given the PTDF and relative accurate load forecasting, net power injection interval solely depends on generator commitment and virtual bid/offer. In order to identify all the potentially dominating constraints, we choose the most conservative approach to determine net power injection interval by taking the physical limits of generator output and max/min of virtual bid/offer.

P_m^{inj} is calculated based on (1)

$$P_m^{\text{inj}} = \sum_{i \in m} P_i + \sum_{k \in m} P_k^v - \sum_{n \in m} D_n - \sum_{f \in m} D_f^{\text{flex}}, \quad (1)$$

where P_i is the generator's dispatch, P_k^v is virtual bid/offer, D_n is the fixed demand, D_f^{flex} is the price-sensitive demand. i, k, n and f are indices for generator, virtual, fixed demand, and price-sensitive demand, respectively. The maximum and minimum net power injections are as the following.

$$\bar{P}_m^{\text{inj}} = \sum_{i \in m} \bar{P}_i + \sum_{k \in m} \bar{P}_k^v - \sum_{n \in m} D_n, \quad (2)$$

$$\underline{P}_m^{\text{inj}} = \sum_{k \in m} \underline{P}_k^v - \sum_{n \in m} D_n - \sum_{f \in m} \bar{D}_f^{\text{flex}}. \quad (3)$$

Net power injection interval is $\Delta P_m^{\text{inj}} = \bar{P}_m^{\text{inj}} - \underline{P}_m^{\text{inj}}$, where overline/underline represent the maximum/minimum physical limits, respectively.

1) *Step One:* With the net power injection, we can apply Theorem 3 in [7] to determine those non-binding constraints. For self-contained, we present the result directly here. Denote $P^v = -\sum_m \underline{P}_m^{\text{inj}}$ and $f_l^v = \sum_{m=1}^{N_d} \Gamma_{l,m} \underline{P}_m^{\text{inj}}$. If there exists an integer $j \in [1, N_d]$ such that

$$\Gamma_{l,m_1} \geq \Gamma_{l,m_2} \geq \dots \geq \Gamma_{l,m_d} \quad (4)$$

$$\sum_{n=1}^{j-1} \Delta P_{m_n}^{\text{inj}} \leq P^v \leq \sum_{n=1}^j \Delta P_{m_n}^{\text{inj}} \quad (5)$$

$$\sum_{n=1}^{j-1} (\Gamma_{l,m_n} - \Gamma_{l,m_j}) \Delta P_{m_n}^{\text{inj}} + \Gamma_{l,m_j} P^v + f_l^v \leq F_l \quad (6)$$

Denote the remaining security constraint set as J_1 . According to our experience in real systems, J_1 are still with considerable size, and many constraints in J_1 are not binding at the optimal point. Hence, we employ a data-driven method to tighten the remaining sets by leveraging historical data. The goal is to determine the 95% confidence interval of generator output, which can be substituted in (2) and (3). The new upper and lower bound of net power injection are denoted as $\bar{P}_m^{\text{inj}*}$, $\underline{P}_m^{\text{inj}*}$, respectively. The new net power injection interval,

$$\Delta P_m^{\text{inj}*} = \bar{P}_m^{\text{inj}*} - \underline{P}_m^{\text{inj}*} \leq \Delta P_m^{\text{inj}}$$

Denote the set of new remaining security constraints as J_1^* . Due to the smaller net power injection,

$$J_1^* \subseteq J_1$$

holds. If the true net power injection is outside of $[\bar{P}_m^{\text{inj}}, \underline{P}_m^{\text{inj}}]$, then J_1^* might miss binding constraints. To address this issue, we employ the lazy constraint set

$$J_1 \setminus J_1^*$$

in the optimization engine. Lazy constraints will be discussed in Section II-B.

2) *Step Two*: J_1 detected by Step 1 are the potentially dominating security constraints, and the redundancy still exists. We apply procedure two to further tighten security constraint set. An LP model is formulated. The objective is to maximize the f_l respecting all other security constraints in J_1 . l is a dominating constraint if and only if $f_l > F_l$. It is formulated as

$$(C) \quad f_l := \max_{\{P_m^{\text{inj}}\}} \sum_m \Gamma_{l,m} P_m^{\text{inj}} \quad (7)$$

$$\text{s.t.} \quad \sum_m P_m^{\text{inj}} = 0, \quad m \in \mathcal{M} \quad (8)$$

$$\underline{P}_m^{\text{inj}} \leq P_m^{\text{inj}} \leq \bar{P}_m^{\text{inj}}, \quad m \in \mathcal{M} \quad (9)$$

$$\sum_m \Gamma_{l',m} P_m^{\text{inj}} \leq F_{l'}, \quad l' \in J_1 \setminus l \quad (10)$$

where equation (8) is the power balance constraint. It is equivalent to $\sum_n P_n = \sum_d D_d$, but in a slightly different format. $\mathcal{M}$ is the set of bus indices. F is the power flow limit. We are able to identify the power flow violation as long as the network topology and lower/upper bound of net power injection are available. This feature enables wide range applications. Theorem 3 in [7] is also based on the net power injection model, although identification rate might not be as high as problem (C). For simplicity, we do not present the power flow in negative direction, which is similar.

It is time consuming to solve thousands of LP like (C). As we are only interested in whether $f_l > F_l$ holds, it is not necessary to completely solve problem (C). Instead, we can solve problem (C*), which is a feasibility check for the optimized model. The model will be infeasible if constraint (15) conflicts with others, which means the maximum power flow on l cannot be greater than its power flow limit. Hence, security constraint l is not a dominating constraint, and vice versa.

$$(C^*) \quad \max \quad 0 \quad (11)$$

$$\text{s.t.} \quad \sum_m P_m^{\text{inj}} = 0, \quad m \in \mathcal{M} \quad (12)$$

$$\underline{P}_m^{\text{inj}} \leq P_m^{\text{inj}} \leq \bar{P}_m^{\text{inj}}, \quad m \in \mathcal{M} \quad (13)$$

$$\sum_m \Gamma_{l',m} P_m^{\text{inj}} \leq F_{l'}, \quad l' \in J_1 \setminus l \quad (14)$$

$$\sum_m \Gamma_{l,m} P_m^{\text{inj}} > F_l \quad (15)$$

B. Identify Lazy Constraint

The proposed two-step prescreening approach can significantly reduce the size of U in the SCUC model. However, only a portion of J_2 is binding at the solution point. Denote set of the active constraints at the solution point as J_B . The

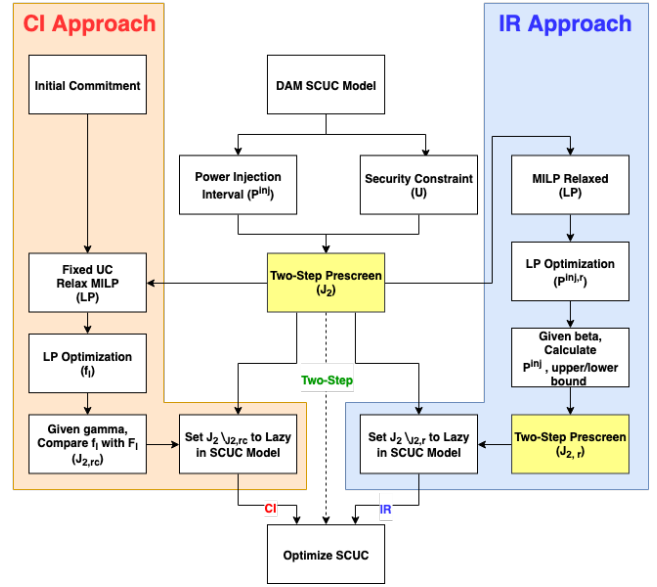


Fig. 1: Data-Aided Prescreening with Lazy Constraints. CI and IR provide a point statistically close to the real solution. Security constraints with values below power flow limits in certain extent are treated as lazy constraints, which reduce the size of MILP and helps solve the problem faster if set properly.

question that naturally arises is how we can effectively shrink the size of J_2 to an approximation of J_B without jeopardizing the security. When data-driven approaches are employed, it becomes challenging to guarantee J_B is all included. Fortunately, the concept of lazy constraint can exactly address this challenge. Lazy constraints are designed to incorporate user confidence in the optimization model to improve solution quality. Constraints that believed to be less probable to bind at the optimal solution can be marked as lazy. The model is initially optimized without any lazy constraints. Once a solution is found, the lazy constraints are checked with the current solution. If any lazy constraints are violated, they will be added back to the reduced model. Optimization process continues till an optimal solution is found without any lazy constraint violations. Users should exercise best judgement to mark lazy constraints, since the intention of lazy constraints is to shrink constraints set and optimize a smaller and easier model. However, if set inappropriately, it may result in more iterations to put back the lazy constraints and solution time may suffer.

It is crucial to separate prescreening result J_2 into the active constraint pool J^{act} , and lazy constraint pool J^{lazy} , where $J^{\text{act}} \cup J^{\text{lazy}} = J_2$ and $J^{\text{act}} \cap J^{\text{lazy}} = \emptyset$. If the active pool is over-conservative, $J^{\text{act}} \approx J_2$, performance improvement would be negligible. If the pool is over-optimistic, $J_B \setminus J^{\text{act}} \subseteq J^{\text{lazy}}$ is large more constraints are added back in the MILP searching process, which may increase solution time.

The key to establish a relatively accurate active constraint pool is to obtain an approximation of SCUC solution. We developed two approaches to achieving such goal. The first is Commitment Initialization (CI) inspired by [11], [13], and the second is based on Integer Relaxation (IR). Procedures are shown in Fig. 1. Both approaches solve a LP problem. The LP

solution offers insight into the security constraint relationship with the SCUC model. We use this information to establish the lazy constraint pool. The basic idea is to find a much smaller active constraint pool J^{act} , and the pool of lazy constraint is

$$J^{\text{lazy}} = J_2 \setminus J^{\text{act}}$$

1) *Integer Relaxation*: Relaxing integer variables in the SCUC model will yield a LP model, which takes very little time to solve. The LP solution is by no means an accurate representation of the MILP solution. However, it provides insights to where the solution point may be. We are solely interested in the net power injection from LP solution $P_m^{\text{inj},r}$. We establish a boundary of net power injection, $\bar{P}_m^{\text{inj},r}$ and $\underline{P}_m^{\text{inj},r}$, to accommodate uncertainties resulting from the the difference between LP and MILP solutions as shown in (16) and (17)

$$\bar{P}_m^{\text{inj},r} = \min \{(1 + \beta) * P_m^{\text{inj},r}, \bar{P}_m^{\text{inj}}\} \quad (16)$$

$$\underline{P}_m^{\text{inj},r} = \max \{(1 - \beta) * P_m^{\text{inj},r}, \underline{P}_m^{\text{inj}}\}, \quad (17)$$

where β is the parameter to be tuned.

To set the lazy constraint, we take the following steps.

- 1) Build DAM SCUC (MILP) model
- 2) Apply the two-step prescreening method, obtain J_2
- 3) Relax integer variables, solve LP problem, extract $P_m^{\text{inj},r}$
- 4) Select β , calculate $\bar{P}_m^{\text{inj},r}$, $\underline{P}_m^{\text{inj},r}$
- 5) Apply the two-step prescreening method on $\bar{P}_m^{\text{inj},r}$, $\underline{P}_m^{\text{inj},r}$, resulting J^{act}
- 6) Obtain lazy constraint pool, $J^{\text{lazy}} = J_2 \setminus J^{\text{act}}$

2) *Commitment Initialization*: According to the persistence studies conducted with MISO cases, 90% of previous day commitment is identical to the current day commitment. We exploit this fact as a warm start to the SCUC model [13]. The basic idea is to get an approximated power flow by solving a LP problem with initialized commitment. The resulting power flow, f_l^c , for each line is often close to the true MILP solution. Given power flow from LP solution f_l^c , power flow limit F_l , and threshold γ , if

$$\frac{f_l^c}{F_l} \geq \gamma, \quad l \in J_2 \quad (18)$$

holds, security constraint for line l is considered as potentially binding, which is put to the active constraint set J^{act} . If equation (18) does not hold, this constraint for line l is set as lazy. We have $J^{\text{lazy}} = J_2 \setminus J^{\text{act}}$.

- 1) Build DAM SCUC (MILP) model
- 2) Apply the two-step prescreening method, obtain J_2
- 3) Fix UCs with initial commitment, solve resulting LP problem
- 4) Calculate power flow, f_l^c , based on LP solution
- 5) Choose threshold γ , use (18) to identify J^{act}
- 6) Obtain lazy constraint pool $J^{\text{lazy}} = J_2 \setminus J^{\text{act}}$

Security constraints are known to be a major cause of SCUC solution deterioration. SCUC solution quality cannot be guaranteed while including all watchlist security constraints. The combined approach of the two step security constraint prescreening and lazy constraint settings, MILP model optimization performance significantly improves by having a smaller

set of active constraints. Model integrity is also maintained while having the potential optimal solution check against lazy constraints.

III. CASE STUDY

In this section, we carry out numerical simulations for MISO cases and analyze the performance of prescreening in MISO market-clearing optimization engine. By using historical data and knowledge, such as Initial Commitment and uncertainty of net power injection, we integrate the prescreening with lazy constraint in the optimization model. The statistical approach helps tighten net power injection range with a given confidence interval, thus building a smaller watchlist with fewer security constraints. Section III-A illustrates dominating constraints detection rate and performance of the two step deterministic prescreening method. In Section III-B, dominating constraints detected by the two-step deterministic prescreening are enforced in the SCUC model. Lazy constraints are applied based on the procedure discussed in II-B. Numerical simulations are carried out on MISO HIPPO project engine [14], using Gurobi 7.5 in Centos with Intel Xeon E5@3.50GHz, 64 GB RAM [12].

A. Dominating Constraint Detection

We apply the two-step deterministic prescreening to all security constraints in the MISO DAM SCUC model. Dominating constraint detection rate and prescreening time is shown in Table I. Case 17 has the fastest processing time of 13.95 seconds (step-1: 4.88s, step-2: 9.07s), 49% of the security constraints are dominating (original: 6604, step-1: 3451, step-2: 3152). In Case 1, prescreening time is 18.52 seconds (step-1: 6.03s, step-2: 12.49s), 43% of the security constraints are dominating (original: 9895, step-1: 4806, step-2: 4251). Step-2 needs to screen 3451 constraints in Case 17 and 4806 constraints in Case 1, which is 40% increase in constraint count. Step-2 processing time is increased by 38% ($\frac{12.49-9.07}{9.07}$). The two step method is consistent through all cases in the study, and processing time scales linearly with quantity of security constraints in the model. The two-step deterministic prescreening method is applied to over 80 MISO DAM cases. Result shows that 8075 security constraints are initially modeled on average. Prescreening determines 45% of the original constraints are dominating ($3634 = 8075 * 45\%$), which will remain in the model. 4421 redundant constraints are safely removed from SCUC model prior to optimization. Average prescreening processing time is 16.65 seconds by solving problem (C*) in step-2, comparing with 53.28 seconds by solving problem (C).

B. Application of Prescreening to MISO SCUC

In the case study, MISO DAM SCUC optimization engine terminates if MILP Gap reaches 0.1% or optimization time reaches 1200 seconds [3]. In this section, we consider case that has base solution time longer than 1000 seconds as hard case. Otherwise, it is considered as easy cases. Following solution approaches are applied to the DAM SCUC model:

TABLE I: MISO Case Prescreening Result

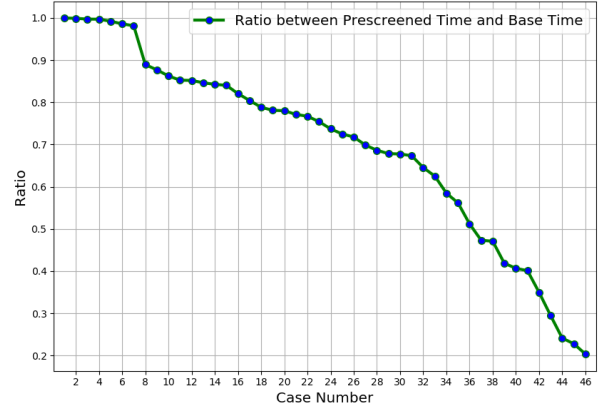
	Case#	Original ¹	Constr Cnt ²	Dom% ³	Time ⁴
Easy	3	6948	3602	51.84%	14.32
	8	9431	4168	44.19%	18.29
	12	7454	3221	43.21%	19.86
	14	6260	3130	50.00%	14.85
Hard	1	9895	4251	42.96%	18.52
	5	6950	3681	52.96%	15.70
	16	10400	4198	40.37%	18.17
	17	6604	3152	47.73%	13.95
	21	9540	5186	54.36%	23.06
	22	8300	4569	55.05%	20.72
Average ⁵	N/A	8075	3654	45.25%	16.65

¹ Security constraint count in DAM SCUC model² Dominating constraint count after prescreen³ Percentage of security constraint remained in the model⁴ Prescreening time, step 2 solving (C*), in seconds⁵ Average value over 80 cases

- **Base solution:** Optimize DAM SCUC directly
- **Two-step solution:** Prescreen security constraints in DAM SCUC model, remove redundant constraints. Optimize the MILP model, no lazy constraint settings are applied.
- **IR solution:** Prescreening DAM SCUC model, remove redundant constraints. Apply IR approach to set lazy constraints in MILP model, optimize. β is set to 0.6.
- **CI solution:** Prescreen DAM SCUC model, remove redundant constraints. Apply CI Approach to set lazy constraints, then optimize. γ is set to 0.8.

There are totally 46 cases where base solution time is over 300 seconds. Fig. 2 illustrates the SCUC optimization time comparison between base solution and best of various prescreening solution approaches (two-step, IR, CI). Each blue dot represents the ratio between prescreened SCUC solution time and base case solution time for a given case. With prescreening techniques, 63% of the cases (26/46) can be solved with 80% or less time than base case. 22% of the cases (10/46) are solved with 50% or less time. Applications of CI and IR approach are restricted by the availability of initial commitment and generator dispatch statistics, tuning hyper parameter β and γ . Lazy constraint setting is a heuristic approach to best guess the binding constraints. The results show that good lazy constraint pool could significantly improve the solution performance in terms of solution time and quality.

Since easy cases take fewer than 1000 seconds to reach MILP gap below 0.1%, we are mainly interested in solution time by aforementioned solution approaches. Results are shown in Table II. Prescreening in Case 8 improves SCUC solution time by 58 (298 - 240) seconds. CI Approach determines 3138 out of 4368 constraints are lazy constraints, and at the end of optimization, 191 constraints are deemed to be non-lazy and put back to the MILP model. CI Approach further decreased solution time by 51 (240-189) seconds. With IR, more constraints are set as lazy, and results in more putting backs, which take a toll on the solution time, 209 seconds. On the other hand, in Case 14, IR is able to achieve better solution time, 442 seconds, than CI Approach, 520 seconds. The solution time resulting from the two lazy constraint setting

**Fig. 2: Ratio between Prescreened SCUC Solution Time and Base Case SCUC Solution Time in Descending Order****TABLE II: Easy Case SCUC Solution Summary**

Case#	Approach	Time ¹	Gap ²	Lz_ct ³	Lz_pb ⁴
3	Base	680	0.08%	0	0
	Two-step	622	0.09%	0	0
	CI	456	0.10%	2952	191
	IR	469	0.10%	3939	860
8	Base	298	0.08%	0	0
	Two-step	240	0.08%	0	0
	CI	189	0.08%	3138	191
	IR	209	0.10%	4137	1069
14	Base	831	0.07%	0	0
	Two-step	648	0.09%	0	0
	CI	520	0.09%	2419	410
	IR	442	0.10%	3062	806

¹ SCUC optimization time, unit in seconds² SCUC MILP gap (%) when optimization has completed³ lazy constraint count in MILP model⁴ lazy constraint put back count after solving MILP

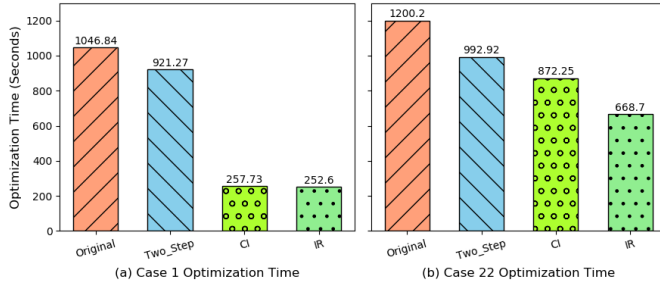
approaches are highly dependent on availability of initial commitment, and hyper parameters β , γ . Over-optimistic values can result in large number of constraints being set as lazy inaccurately. From our studies on MISO SCUC easy cases, the two-step deterministic prescreening method combined with lazy constraint settings shows significantly reduction in SCUC solution time. On average, the aforementioned prescreening approach saves an average of 292 seconds (30%) in SCUC optimization time compared with base solution time.

Hard cases take longer than 1000 seconds to reach MILP gap of 0.1%. Results are shown in Table III. Incorporating prescreening methods helps many cases to reach 0.1% MILP gap under 1200 seconds, whereas previously they could not. In Case 17, the base solution cannot reach 0.1% MILP gap within 1200 seconds, the optimization process is terminated, and the resulting MILP gap is 0.11%. The two-step approach allows MILP gap to reach 0.1% at 1188 seconds. The CI approach is able to reach MILP gap 0.09% at 1063 seconds. Prescreening method can also dramatically reduce MILP solution time in some hard cases. In Case 1, the two-step approach decreases SCUC optimization time by 126 seconds from base case solution. CI and IR are able to reach 0.1% MILP GAP in 270 seconds. Total solution time is reduced by 777 seconds.

Case 1 and 22 simulation results in Fig. 3 illustrate two typical performance improvements when prescreening is applied.

TABLE III: Hard Case SCUC Solution Summary

Case#	Approach	Time ¹	Gap ²	Lz_ct ³	Lz_pb ⁴
1	Base	1047	0.07%	0	0
	two-step	921	0.06%	0	0
	CI	271	0.10%	1592	10
	IR	270	0.09%	3203	1042
5	Base	1103	0.10%	0	0
	two-step	658	0.07%	0	0
	CI	899	0.09%	2366	166
	IR	802	0.09%	3409	1170
17	Base	1200	0.11%	0	0
	two-step	1188	0.10%	0	0
	CI	1063	0.09%	1458	64
	IR	1200	0.08%	2087	429
21	Base	1200	0.26%	0	0
	two-step	1200	0.19%	0	0
	CI	1200	0.10%	3549	8
	IR	1201	0.16%	5110	1752
22	Base	1200	0.23%	0	0
	two-step	993	0.09%	0	0
	CI	872	0.09%	2742	11
	IR	669	0.09%	4557	1504

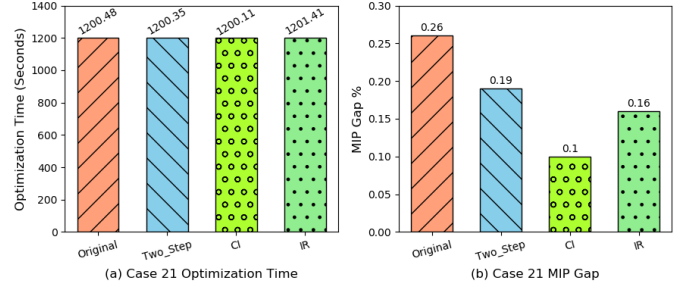
¹ SCUC optimization time, unit in seconds² SCUC MILP gap (%) when optimization has completed³ lazy constraint count in MILP model⁴ lazy constraint put back count after solving MILP**Fig. 3: Optimization Time Comparison.** Figure (a) presents result for Case 1 where optimality gap is reached within time limit without prescreening. Figure (b) presents results for Case 22 where optimality gap is not reached within time limit without prescreening

As shown in Fig. 3(a), optimization engine can get the solution within 1200 seconds. When IR or CI based prescreening is employed, it only takes about 25% of the original time to find the solution with similar optimality gap. In contrast, Fig. 3(b) shows how prescreening could improve the performance when the original model could not be solved within 1200 seconds. It is observed all two-step, IR, and CI based prescreening help the optimization engine solve the problem within time limit.

In a few hard cases, the solver could not get a solution with required gap even if the prescreening is applied. For instance, Case 21 in Fig. 4(a) shows the solver terminates after 1200 seconds. According to Fig. 4(b), prescreening helps reduce the gap significantly. IR based approach decreases the gap to 0.16% from 0.26% while CI approach almost get to 0.1% gap.

IV. CONCLUSION

In this paper, we aim to build a better security constraint pool for DAM clearing tool. We propose a data-aided two-step prescreening approach to identify dominating security constraints and eliminate inactive ones. It is a hybrid of deterministic and statistical methods, which leverages the his-

**Fig. 4: Gap Comparison.** Optimality gap cannot be reached with or without prescreening. Figure (b) shows the gap differences when prescreening is applied.

torical data and solution to a relaxed LP. Lazy constraints are applied in optimization solver to maintain solution integrity. Studies conducted on MISO DAM SCUC model has shown that prescreening approaches effectively identify inactiveness of over 55% security constraints with little computation effort and help improve SCUC solution quality significantly.

REFERENCES

- [1] M. Shahidehpour, H. Yamin, and Z. Li, *Market Operations in Electric Power Systems: Forecasting, Scheduling, and Risk Management*. John Wiley & Sons, 2003.
- [2] Y. Fu, Z. Li, and L. Wu, "Modeling and Solution of the Large-Scale Security-Constrained Unit Commitment," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 3524–3533, Nov. 2013.
- [3] Y. Chen, A. Casto, F. Wang, Q. Wang, X. Wang, and J. Wan, "Improving Large Scale Day-Ahead Security Constrained Unit Commitment Performance," *IEEE Transactions on Power Systems*, vol. 31, no. 6, pp. 4732–4743, Nov. 2016.
- [4] A. J. Ardakani and F. Bouffard, "Identification of Umbrella Constraints in DC-Based Security-Constrained Optimal Power Flow," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 3924–3934, Nov. 2013.
- [5] A. J. Ardakani and F. Bouffard, "Acceleration of Umbrella Constraint Discovery in Generation Scheduling Problems," *IEEE Transactions on Power Systems*, vol. 30, no. 4, pp. 2100–2109, Jul. 2015.
- [6] Q. Zhai, X. Guan, J. Cheng, and H. Wu, "Fast Identification of Inactive Security Constraints in SCUC Problems," *IEEE Transactions on Power Systems*, vol. 25, no. 4, pp. 1946–1954, Nov. 2010.
- [7] H. Ye, J. Wang, and Z. Li, "MILP Reformulation for Max-Min Problems in Two-Stage Robust SCUC," *IEEE Transactions on Power Systems*, vol. 32, no. 2, pp. 1237–1247, Mar. 2017.
- [8] H. Ye and Z. Li, "Necessary Conditions of Line Congestions in Uncertainty Accommodation," *IEEE Transactions on Power Systems*, vol. 31, no. 5, pp. 4165–4166, Sep. 2016.
- [9] R. Madani, J. Lavaei, and R. Baldick, "Constraint Screening for Security Analysis of Power Networks," *IEEE Transactions on Power Systems*, vol. 32, no. 3, pp. 1828–1838, May 2017.
- [10] A. S. Xavier, F. Qiu, F. Wang, and P. R. Thimmapuram, "Transmission Constraint Filtering in Large-Scale Security-Constrained Unit Commitment," *IEEE Transactions on Power Systems*, vol. 34, no. 3, pp. 2457–2460, May 2019.
- [11] Y. Chen, F. Wang, Y. Ma and Y. Yao, "A Distributed Framework for Solving and Benchmarking Security Constrained Unit Commitment with Warm Start," in *IEEE Transactions on Power Systems*. doi: 10.1109/TPWRS.2019.2930706
- [12] Gurobi Optimization, Inc., "Gurobi Optimizer Reference Manual," 2019. [Online]. Available: <http://www.gurobi.com/>.
- [13] Y. Chen, "MISO R&D on Improving the Efficiency of Market Clearing Software," FERC Technical Conference, 2019. Accessed:06/20/19, https://www.ferc.gov/CalendarFiles/20190625100440-3%20-%20T1_Chen_wide.pdf.
- [14] Y. Chen, F. Wang, J. Wan, and F. Pan, "Developing Next Generation Electricity Market Clearing Optimization Software," *IEEE Power & Energy Society General Meeting*, pp 1-5, 2018.