Temporal Decomposition for Security-Constrained Unit Commitment

Farnaz Safdarian, *Student Member,* Ali Mohammadi, *Student Member, IEEE,* and Amin Kargarian, *Member, IEEE*

***Abstract*-This paper proposes a temporal decomposition strategy to reduce the computation time of security-constrained unit commitment (SCUC). The novelty of this paper is twofold. The scheduling horizon is decomposed into multiple subhorizons. The concept of coupling intervals is introduced, and a set of auxiliary counting variables and logical expressions along with their equivalent linear models are formulated to handle intertemporal ramp constraints and minimum on/off times between consecutive subhorizons. An accelerated analytical target cascading (A-ATC) algorithm is developed to coordinate SCUC subproblems and find the optimal solution for the whole operation horizon in a distributed manner. An initialization strategy is presented to enhance the convergence performance of A-ATC. The proposed algorithm is tested on four test systems.**

***Index Terms*—Temporal decomposition, security-constrained unit commitment, accelerated analytical target cascading.**

NOMENCLATURE

Indices, Sets, and Parameters:

c	Index for contingencies.
k	Index for iterations.
t	Index for time intervals.
t_o	Index for coupling time intervals.
u	Index for units.
s	Index for subhorizon (subproblem) s.
s_-	Subhorizon (subproblem) before subhorizon s.
s_+	Subhorizon (subproblem) after subhorizon s.
T_u^{on}, T_u^{off}	Minimum on/off time of unit u.
ρ	Penalty factor.

Variables:

I_{ut}	On/off status of unit u at interval t.
$I_{ut_o(s,s_-)}$	On/off status of unit u at coupling time interval t_o between subhorizons s_- and s determined by subproblem s and sent to subproblem s_-.
p_{ut}	Power generated by unit u at interval t.
$p_{ut_o(s,s_-)}$	Power generated by unit u at coupling time interval t_o between subhorizons s_- and s determined by subproblem s and sent to subproblem s_-.
$r_{s_-,s}^k$	Response shared variables between subhorizons s_- and s calculated by subproblem s_- at iteration k.
x_{s_-}	Set of variables in time intervals belonging to subhorizon s_-.
y_{ut}	Startup indicator of unit u at time t.
z_{ut}	Shutdown indicator of unit u at time t.
$\chi_{s_-,s}^k$	Target shared variables at iteration k.
λ	Vector of Lagrange multipliers.

Auxiliary Counting Variables and Functions:

h_{us-}^{on}	An auxiliary variable to calculate $h_{u(s_-,s)}^{on}$.
h_{us-}^{off}	An auxiliary variable to calculate $h_{u(s_-,s)}^{off}$.
$h_{u(s_-,s)}^{on}$	Number of hours that SP_{s_-} asks SP_s to keep unit u on.
$h_{u(s,s_-)}^{on}$	Number of hours that SP_s asks SP_{s_-} to keep unit u on.
$h_{u(s_-,s)}^{off}$	Number of hours that SP_{s_-} asks SP_s to keep unit u off.
$h_{u(s,s_-)}^{off}$	Number of hours that SP_s asks SP_{s_-} to keep unit u off.
H_u^{on}	An auxiliary variable to linearize max of Φ_{ut}^{on}.
$\mathcal{F}_{us}^{sd}$	First shutdown time of SP_s.
$\mathcal{F}_{us}^{su}$	First startup time of SP_s.
ϕ_{us}^{on}, ϕ_{us}^{off}	A vector whose tth element is nonzero if status of unit u changes in interval t from off/on to on/off.

I. INTRODUCTION

THE computational burden of security-constrained unit commitment (SCUC) increases with growing the size of the system and scheduling intervals [1, 2]. Conventional centralized methods may face difficulties to handle such large optimization problems within an acceptable time. Distributed algorithms are proposed as alternatives to decompose large problems into smaller subproblems and distribute computational burden on multiple computing machines [3-5]. References [4, 6, 7] review distributed algorithms and their applications on power systems. Algorithms, such as alternating direction method of multipliers [8], optimality condition decomposition [9], auxiliary problem principle [10], and analytical target cascading [11, 12], are applied to coordinate subproblems. Such algorithms are widely used for optimal power flow, demand response, and economic dispatch [13-15].

The literature on distributed SCUC is more limited than optimal power flow and economic dispatch. To the best of our knowledge, all existing distributed SCUC approaches are based on decomposition over geographical areas, uncertainty scenarios, and contingency scenarios. In [16], progressive hedging is presented to decompose problems over scenarios. Reference [17] decomposes hourly day-ahead unit commitment over stochastic scenarios. In [18], a scenario-based decomposition is presented for solving two-stage stochastic unit commitment. In [19], Benders decomposition is applied to decompose SCUC into a master problem and several subproblems that handle contingency constraints. Majority of recent publications focus on area decomposition. In [20-22], the system is decomposed into several smaller zones, a SCUC subproblem is formulated for each zone, and distributed algorithms are developed to coordinate subproblems.

Although decompositions over geographical areas, uncertainty scenarios, and contingency scenarios may reduce the computation time of SCUC, they are not capable of handling the

The authors are with the department of Electrical and Computer Engineering, Louisiana State University, Baton Rouge, LA 70803 USA (e-mail: fsafda1@lsu.edu, amoha39@lsu.edu, kargarian@lsu.edu).

computational complexity originated from temporal constraints. Such complexity increases as the number of scheduling intervals grows. Considering contingencies and interdependencies between scheduling intervals increase the computation time of SCUC. To reduce the computation time, decomposing SCUC over the considered scheduling time horizon is potentially a more promising strategy than a geographical decomposition. However, the most challenging part of time decomposition is dealing with ramping and minimum on/off time constraints of units that interconnect decisions made in consecutive intervals. In our previous work [23], we presented the idea of time decomposition for a simple economic dispatch problem without modeling minimum on/off time constraints.

This paper presents a temporal decomposition and coordination strategy to solve SCUC distributedly with the aim of computation time reduction. To the best of our knowledge, this is the first paper proposing a time decomposition strategy for SCUC. The overall scheduling horizon is decomposed into multiple smaller subhorizons, and a SCUC subproblem is formulated for each subhorizon. The concept of *coupling time intervals* is introduced to model ramping limitations of generating units between two consecutive subhorizons. Each subproblem has one or two coupling intervals. Moreover, several auxiliary counting variables and logical expressions along with their linear equivalent models are presented to model minimum on/off time constraints between subhorizons. In addition, to coordinate SCUC subproblems and ensure feasibility and optimality of results, accelerated analytical target cascading (A-ATC) is developed to solve subproblems distributedly. An initialization technique is presented to enhance the convergence performance of A-ATC. The main contributions of the paper are summarized as follows:

- A novel temporal decomposition strategy is proposed to decompose the SCUC problem into multiple time-interdependent SCUC subproblems.
- Modeling approaches are developed to handle generating units' ramp and minimum on/off restrictions for the transition between subhorizons.
- Based on the concept of Nesterov momentum for gradient descent methods, an *accelerated* ATC is presented to coordinated SCUC subproblems distributedly.
- An initialization strategy is presented to enhance the performance of the proposed distributed SCUC algorithm.

II. Proposed Time Decomposition Strategy

The main philosophy of the proposed algorithm is based on Augmented Lagrangian relaxation [4, 24-26]. To take advantage of distributed computing, we divide the scheduling horizon into several smaller, consecutive subhorizons. A SCUC subproblem is formulated for each subhorizon. SCUC subproblems are formulated exactly the same as the centralized SCUC for the whole scheduling horizon, with no need for modifying the intertemporal and non-intertemporal constraints. Intertemporal interdependencies between every two consecutive subproblems are converted into a set of shared variables. Consistency constraints are formed to ensure that each pair of shared variables reaches the same values from the perspective of the two subproblems. The concept of augmented Lagrangian relaxation is used to penalize the consistency constraints into local objective functions and make mismatches between each pair of the shared variables zero iteratively.

A. SCUC Formulations and Notations

For brevity, we show compact forms of SCUC subproblems. We have adopted the SCUC formulation presented in Chapters 3 and 4 of [1] except for modeling startup and shutdown indices and minimum on/off time constraints that are adopted from [27]. s_-, s, and s_+ refer to three consecutive subhorizons. We use subscript (s_-, s) to distinguish coupling variables for modeling transition between subproblems s_- and s. A coupling variable indicated by subscript $(\cdot,\cdot)$ is a decision variable determined by the left side subproblem. For instance, $m_{(s_-,s)}$ are coupling variables m between subproblems s_- and s that are among decision variables in subproblem s_-.

B. Decomposing SCUC

Without loss of generality, we derive equations for two consecutive subhorizons s_- and s and decompose the SCUC problem into SCUC subproblems (1) and (2). Assume that the scheduling time horizon is T. We assign intervals one to n_- to the SCUC subproblem (SP) of subhorizon s_- (named SP_{s_-}).

$$x_{s_-} = \text{argmin} \sum_{t=1}^{n_-} \sum_{u} f(p_{ut}, I_{ut}) \tag{1a}$$

$$s.t. \qquad h_{s_-}(x_{s_-}) = 0 \quad \& \quad g_{s_-}(x_{s_-}) \le 0 \tag{1b}$$

$$\text{with } t = \{1, \dots, n_-\}$$

where x_{s_-} are variables corresponding to intervals one to n_-. h_{s_-} includes power balance constraints under normal and contingency conditions at intervals $\{1, \dots, n_-\}$, and g_{s_-} refers to generations upper and lower bounds, line flow limits, units ramping up/down restrictions, minimum on/off time restrictions under normal and contingency conditions, and expressions coupling power generated by units before and after contingencies. We also assign intervals $n_- + 1$ to T to SP_s.

$$x_s = \text{argmin} \sum_{t=n_-+1}^{n} \sum_{u} f(p_{ut}, I_{ut}) \tag{2a}$$

$$s.t. \qquad h_s(x_s) = 0 \quad \& \quad g_s(x_s) \le 0 \tag{2b}$$

$$\text{with } t = \{n_- + 1, \dots, n\}.$$

x_s are variables corresponding to intervals $n_- + 1$ to T. Subproblems (1) and (2) are smaller than the original centralized SCUC and hence are computationally less expensive. However, the solution to (1) and (2) may be physically infeasible as interdependencies, i.e., temporal constraints, between subhorizons s_- and s are ignored.

C. Modeling Ramp Limits with Coupling Time Intervals

Generation ramp up and down constraints pertaining to power generated in the last interval of subhorizon s_- ($t = n_-$) and the first interval of subhorizon s ($t = n_- + 1$) link decisions made in these subhorizons. Decomposing the centralized SCUC into (1) and (2), as shown in Fig. 1a, does not take into account these ramp constraints. To enable modeling the boundary ramp

constraints, we represent the whole operation horizon as depicted in Fig. 1a. Without loss of generality and to better explain decomposition with a middle SP, we divide the scheduling horizon it into three subhorizons s_-, s, and s_+.

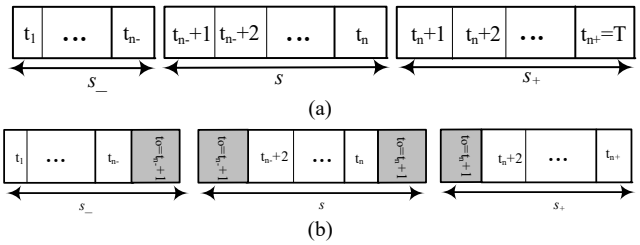


Fig. 1. a) Three consecutive subhorizons and b) subhorizons decomposition with coupling intervals.

As shown in Fig. 1b, the first interval of each subhorizon, except for subhorizon one, is introduced as a coupling time interval, indicated by t_o, between two consecutive subhorizons. We duplicated each coupling interval and assign a copy to a subhorizon to separate the subhorizons. Each copy contains variables and constraints of the corresponding coupling interval. The coupling intervals not only facilitate modeling ramp constraints for transition between subproblems but also allow the separate solution of subproblems. We found this method more efficient for time decomposition rather than duplicating variables that is popular for geographical decompositions. The SP (1) for subhorizon s_- is now reformulated as:

$$x_{s_-} = \text{argmin} \sum_{t=1}^{n_-+1} \sum_{u} f(p_{ut}, I_{ut}) \qquad (3a)$$

$$s.t. \qquad h_{s_-}(x_{s_-}) = 0 \;\; \& \;\; g_{s_-}(x_{s_-}) \le 0 \qquad (3b)$$

$$\text{with } t = \{1, \dots, n_-, \underbrace{n_- + 1}_{\text{coupling interval}} \}$$

where x_{s_-} contains variables corresponding to intervals one to n_- as well as interval $n_- + 1$ that is interval one of subhorizon s. The SP (2) for subhorizon s is reformulated by (4). Since subhorizon s has two neighboring subhorizons, it contains two coupling intervals, i.e., $t = n_- + 1$ between SP_s and SP_{s_-} and $t = n + 1$ between SP_s and SP_{s_+}.

$$x_s = \text{argmin} \sum_{t=n_-+1}^{n} \sum_{u} f(p_{ut}, I_{ut}) \qquad (4a)$$

$$s.t. \qquad h_s(x_s) = 0 \quad \& \quad g_s(x_s) \le 0 \qquad (4b)$$

$$\text{with } t = \{ \underbrace{n_- + 1}_{\text{coupling interval}}, n_- + 2, \dots, n, \underbrace{n+1}_{\text{coupling interval}} \}$$

Power produced by units at coupling intervals $n_- + 1$ appears in both SP_{s_-} and SP_s. Similarly, $p_{u,n+1}$ appears in SP_s and SP_{s_+}. For brevity of notations, we name the coupling intervals $n_- + 1$ and $n + 1$ as t_o and add a subscript to power produced by units at coupling intervals to show the shared information (i.e., coupling variables) between subproblems. For example, $p_{ut_o(s_-,s)}$ is a coupling variable between SP_{s_-} and SP_s that is handled by SP_{s_-}, and $p_{ut_o(s,s_-)}$ is a coupling variable between SP_{s_-} and SP_s being handled by SP_s. To make the SCUC solutions feasible from the perspective of the whole operating horizon, we convert each pair of coupling variables into a consistency constraint as:

$$CC_{s_-,s}: p_{ut_o(s_-,s)} - p_{ut_o(s,s_-)} = 0 \qquad \forall u, t_o = n_- + 1 \qquad (5)$$

$$CC_{s,s_+}: p_{ut_o(s,s_+)} - p_{ut_o(s_+,s)} = 0 \qquad \forall u, t_o = n + 1 \qquad (6)$$

and enforce (5) in SP_{s_-} and SP_s and (6) in SP_s and SP_{s_+}.

D. Minimum On/Off Time Limits

The concept of coupling intervals is introduced for modeling ramping between subproblems; however, they cannot solve challenges for decomposing minimum on/off time constraints between subproblems. These constraints must be decomposed and modeled appropriately to ensure feasibility of the proposed time decomposition strategy's solution. Modeling boundary minimum on/off time constraints is more complex than that ramping as the number of intervals that connect neighboring subhorizons depends on minimum on/off time that is different for different units. That is, two consecutive subhorizons are coupled through several ending/beginning non-coupling intervals at the boundary of subhorizons.

The main idea for solving this challenge is to count the number of on/off times at boundary intervals between consecutive subproblems and try to reach a consistency between these numbers iteratively by penalizing their mismatches in local objective functions. As depicted in Fig. 2, we introduce two sets of auxiliary variables in each SP that count the number of on/off times at boundary intervals. These variables are shared between subproblems to measure inconsistencies between minimum on/off times of an SP with its neighbors. These variables are only counters and have no impact on the form of minimum on/off time constraints. To formulate each SP, the minimum on/off time constraints for all time intervals, including coupling intervals, are formulated using models developed in [27], with no modifications. The counting variables for SP_{s_-}, for instance, are:

1. $h^{on}_{u(s_-,s)}$: Remaining minimum on-time of unit u in SP_{s_-} that should be satisfied in SP_s.
2. $h^{off}_{u(s_-,s)}$: Remaining minimum off-time of unit u in SP_{s_-} that should be satisfied in SP_s.

We define several functions to model these counting shared variables.

Minimum on-time counter for SP_{s_-}: Using startup indicator y [27], in SP_{s_-}, we define a vector $\Phi^{on}_{us_-}$ whose tth element is equal to the time interval index t if the status of unit u has changed from off to on in that interval, otherwise zero.

$$\Phi^{on}_{uts_-} = t \times y_{ut} \qquad \forall u, \forall t \qquad (7)$$

The last interval in which an off to on status change occurs for unit u is equal to the maximum value of the vector $\Phi^{on}_{us_-}$. Hence, the number of intervals that unit u have been on in final intervals of SP_{s_-}, i.e., $h^{on}_{us_-}$, is:

$$h^{on}_{us_-} = n_- - \max\left(\Phi^{on}_{us_-}\right) + 1 \qquad \forall u \qquad (8)$$

For instance, for a subhorizon with 24 intervals, if the last interval in which unit u is turned on is 22, $\max\left(\Phi^{on}_{us_-}\right) = 22$ and expression (8) returns $h^{on}_{us_-} = 24 - 22 + 1 = 3$. Thus, the number of intervals that SP_{s_-} wants SP_s to keep unit u on is:

$$h^{on}_{u(s_-,s)} = \max\left(0, T^{on}_u - h^{on}_{us_-}\right) \qquad \forall u \qquad (9)$$

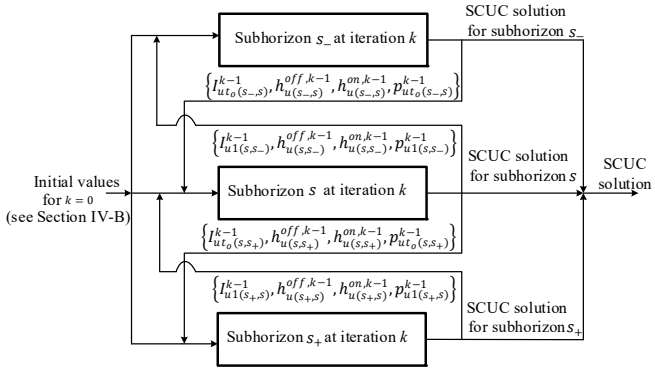


Fig. 2. Block diagram of information sharing for coordinating subproblems.

If $h^{on}_{us_-}$ is smaller than the minimum on-time, $T^{on}_u - h^{on}_{us_-}$ is positive and SP_{s_-} wants SP_s to keep unit u on for at least $T^{on}_u - h^{on}_{us_-}$ intervals at the beginning of its subhorizon; otherwise SP_{s_-} sends zero to SP_s. For instance, for a subhorizon with 24 intervals, if $h^{on}_{us_-} = 3$ and $T^{on}_u = 5$, then $h^{on}_{u(s_-,s)} = 2$; and if $h^{on}_{us_-} = 3$ and $T^{on}_u = 2$, then $h^{on}_{u(s_-,s)} = 0$. Also, if $T^{on}_u = 2$ and unit u is turned on at hour 20 and off at hour 23 and was kept off for the rest of intervals in SP_{s_-}, then (7) produces 20, (8) gives 5, and (9) returns 0.

Minimum on-time counter for SP_s: To count the number of intervals that unit u is on at the beginning of subhorizon s, we should detect the first interval in which the unit is turned off. Using shutdown indicator z [27], we define a vector Φ^{off}_{us} whose tth element is equal to time index t if the status of unit u has changed from on to off in that interval, otherwise zero.

$$\Phi^{off}_{uts} = t \times z_{ut} \qquad \forall u, \forall t \qquad (10)$$

The first shutdown time of SP_s is then the minimum nonzero element of Φ^{off}_{us}.

$$\mathcal{F}^{sd}_{us} = \text{minnonzero}\left(\Phi^{off}_{us}\right) \qquad \forall u \qquad (11)$$

Three possible situations may happen.

1. $h^{on}_{u(s_-,s)}$ calculated by (9) is nonzero, and it is also more beneficial for SP_s to keep unit u on for at least $h^{on}_{u(s_-,s)}$ hours. In this situation, coupling variables between the two subproblems are the same.
2. $h^{on}_{u(s_-,s)}$ is nonzero but SP_s prefers to keep unit u on for less than $h^{on}_{u(s_-,s)}$ hours. In this situation, coupling variables are different and subproblems must come to a trade-off about the number of on-hours.
3. $h^{on}_{u(s_-,s)}$ calculated by (9) is zero, and hence SP_{s_-} does not send any request to SP_s.

To cover these possibilities, we determine $h^{on}_{u(s,s_-)}$ that must take the minimum of $h^{on}_{u(s_-,s)}$, given from SP_{s_-}, and the number of intervals at the beginning of subhorizon s that unit u is on.

$$h^{on}_{u(s,s_-)} = \min\left(h^{on}_{u(s_-,s)}, \mathcal{F}^{sd}_{us} - 1, M \times I_{ut_o(s,s_-)}\right) \qquad \forall u \qquad (12)$$

If unit u is on at the first interval of SP_s, the number of initial on-hours at this subhorizon is counted by the first hour at which a shutdown happens minus one (i.e., $\mathcal{F}^{sd}_{us} - 1$). The term $M \times I_{ut_o(s,s_-)}$, in which a big-M is used, accounts for situations that unit u is off at the first interval of SP_s.

Minimum off-time counter: An analogous strategy as of that for minimum on-time is used for minimum off-time counters.

In SP_{s_-}:

$$h^{off}_{us_-} = n_- - \max\left(\Phi^{off}_{us_-}\right) + 1 \qquad \forall u \qquad (13)$$

$$h^{off}_{u(s_-,s)} = \max\left(0, T^{off}_u - h^{off}_{us_-}\right) \qquad \forall u \qquad (14)$$

In SP_s:

$$\mathcal{F}^{su}_{us} = \text{minnonzero}(\Phi^{on}_{us}) \qquad \forall u \qquad (15)$$

$$h^{off}_{u(s,s_-)} = \min\left(h^{off}_{u(s_-,s)}, \mathcal{F}^{su}_{us} - 1, M \times \left(1 - I_{ut_o(s,s_-)}\right)\right) \quad \forall u (16)$$

Similar expressions can be used to model the counting variables between SP_s and SP_{s_+} and any two consecutive subproblems. The combination of (7)-(16) provides accurate counting variables for all possible on/off situations that might happen for the transition between subproblems.

Illustrative example: Consider the boundary intervals shown in Fig. 3. Assume that $T^{on}_u = 3, T^{off}_u = 1$, and the length of each SP is 24. We have $h^{on}_{u1} = 24 - 23 + 1 = 2$, $h^{on}_{u(1,2)} = \max(0,3-2) = 1$, $\mathcal{F}^{sd}_{u2} = 6$, $h^{on}_{u(2,1)} = \min(1,6-1, M \times 0) = 0$, $h^{off}_{u1} = 24 - 21 + 1 = 4$, $h^{off}_{u(1,2)} = \max(0,1-4) = 0$, $\mathcal{F}^{su}_{u1} = 3$, and $h^{off}_{u(2,1)} = \min(0,3-1, M \times 1) = 0$. Note that this example is not feasible. A detailed tutorial for minimum on/off modeling is provided in Section V.A.

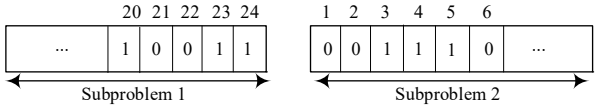


Fig. 3. An illustrative example for on/off status at boundary intervals.

Consistency constraints: $h^{on}_{u(s_-,s)}$ and $h^{on}_{u(s,s_-)}$ must reach the same value to have consistency between the on-counting variables in SP_{s_-} and SP_s. $h^{off}_{u(s_-,s)}$ and $h^{off}_{u(s,s_-)}$ must also reach the same values for consistency between the off-counting variables. Otherwise, the obtained solution is not feasible from the perspective of the whole scheduling horizon. To ensure consistency between SP_{s_-} and SP_s, we formulate three sets of consistency constraints and impose them to each subproblem.

$$CC_{s_-,s}: \begin{cases} I_{ut_o(s_-,s)} - I_{ut_o(s,s_-)} = 0 & \forall u, t_o = n_- + 1 \\ h^{on}_{u(s_-,s)} - h^{on}_{u(s,s_-)} = 0 & \forall u \\ h^{off}_{u(s_-,s)} - h^{off}_{u(s,s_-)} = 0 & \forall u \end{cases} \qquad (17)$$

The consistency constraints between SP_s and SP_{s_+} are:

$$CC_{s,s_+}: \begin{cases} I_{ut_o(s,s_+)} - I_{ut_o(s_+,s)} = 0 & \forall u, t_o = n + 1 \\ h^{on}_{u(s,s_+)} - h^{on}_{u(s_+,s)} = 0 & \forall u \\ h^{off}_{u(s,s_+)} - h^{off}_{u(s_+,s)} = 0 & \forall u \end{cases} \qquad (18)$$

Constraints (17) are enforced in SP_{s_-} and SP_s, and (18) are added in SP_s and SP_{s_+}.

E. Summary of SCUC Subproblems Formulation

In summary, the SCUC subproblem for each subhorizon s is formulated by adding (5)-(18) to the optimization problem (4).

Although (7)-(16) model interdependencies between SP_s and its previous subproblem SP_{s_-}, a set of similar expressions are required to model interdependencies between SP_s and its next subproblem SP_{s_+}. For modeling minimum on/off time constraints, we use models presented in [27] where h_u^{on} and h_u^{off} are known parameter from initial conditions. However, in our developed models, h_u^{on} (indicated by $h_{u(s,s_-)}^{on}$ and $h_{u(s,s_+)}^{on}$) and h_u^{off} (indicated by $h_{u(s,s_-)}^{off}$ and $h_{u(s,s_+)}^{off}$) are shared decision variables that are byproducts of SCUC decision variables. In addition to regular SCUC input parameters (e.g., network topology and capacity of units), SP_s receives $p_{ut_o(s_-,s)}$, $I_{ut_o(s_-,s)}$, $h_{u(s_-,s)}^{on}$, and $h_{u(s_-,s)}^{off}$ from SP_{s_-} and $p_{ut_o(s_+,s)}$, $I_{ut_o(s_+,s)}$, $h_{u(s_+,s)}^{on}$, $h_{u(s_+,s)}^{off}$ from SP_{s_+} as inputs, and determines Ξ_s (see (4)), $p_{ut_o(s,s_-)}$, $I_{ut_o(s,s_-)}$, $p_{ut_o(s,s_+)}$, $I_{ut_o(s,s_+)}$, $h_{u(s,s_-)}^{on}$, $h_{u(s,s_-)}^{off}$, $h_{u(s,s_+)}^{on}$, $h_{u(s,s_+)}^{off}$ as outputs.

F. Discussion on Number of Subhorizons

Increasing the number of subhorizons increases the number of shared variables and consequently the required number of iterations for coordination algorithms to converge. Hence, although smaller subproblems take less time to be solved, increasing the number of subhorizons is not necessarily efficient for time-saving. There should be a tradeoff between the number of subhorizons and iterations. Another factor that should be considered is the power of computational facilities. Although a computing machine takes less time to solve smaller optimization problems, further size reduction beyond a certain level may not yield significant time-saving.

Determining the optimal number of subhorizons is a challenging, but valuable research question. A possible approach is to solve a given SCUC problem considering different numbers of subhorizons and select the one that yields the best time-saving. We suggest determining the length of subhorizons in a way to be larger than the largest minimum on/off times of generating units. Another suggestion is to have subhorizons with a similar size to take advantage of parallel computing with the least CPUs idle times. Since the behavior of the problem may change depending on the loading condition, different numbers of subhorizons may be more appropriate for different load pattern. The yearly load can be classified into different groups (e.g., a group for each season of a year), and a suitable number of subhorizons can be found for each group using a trial and error approach. We are working on this research question as future work.

III. Converting Model Into MIP

Closed-form linear models minnonzero, min, and max functions in (10)-(18) are needed to formulate the problem in the standard MIP format.

A. Minnonzero Functions

We model the minnonzero function in (11) using a Big-M value and replace it by:

$$\mathcal{F}_{us}^{sd} = \min\left((1 - z_u) \times M + \Phi_{us}^{off}\right) \quad \forall u \quad (19)$$

where z_u is a vector containing the shutdown indicator for each interval t. This constraint assigns the first time index of SP_s in which a shutdown happens (i.e., $z_{ut} = 1$) to $\mathcal{F}_{us}^{sd}$. To model the minnonzero function for minimum off-time constraint (15), we replace (15) by (20) to determine the first startup time.

$$\mathcal{F}_{us}^{su} = \min\left((1 - y_u) \times M + \Phi_{us}^{on}\right) \quad \forall u \quad (20)$$

B. Min Function

We follow the concepts presented in [28] for linearizing min functions. In SP_s, instead of equation (11), we use:

$$\mathcal{F}_{us}^{sd} \le \left((1 - z_u) \times M + \Phi_u^{off}\right) \quad \forall u \quad (21)$$

The right-hand-side of (21) is a vector and its left-hand-side is a variable. Therefore, $\mathcal{F}_{uN}^{sd}$ is smaller than the minimum value of the right-hand-side vector. We also replace (12) as follows:

$$h_{u(s,s_-)}^{on} \le h_{u(s_-,s)}^{on} \quad \forall u \quad (22)$$

$$h_{u(s,s_-)}^{on} \le \mathcal{F}_{us}^{sd} - 1 \quad \forall u \quad (23)$$

$$h_{u(s,s_-)}^{on} \le M \times I_{ut_o(s,s_-)} \quad \forall u, t_o = n_- + 1 \quad (24)$$

As explained in Section IV, we formulate augmented Lagrangian penalty functions, whose goal is to vanish mismatches between coupling variables, in a way for $h_{u(s,s_-)}^{on}$ to appear with negative terms in the objective function of SP_s. Since SCUC is a minimization problem, (22)-(24) along with including $h_{u(s,s_-)}^{on}$ in the objective function result in $h_{u(s,s_-)}^{on}$ taking the minimum of $h_{u(s_-,s)}^{on}$, $M \times I_{ut_o(s,s_-)}$, and $\mathcal{F}_{us}^{sd} - 1$. Other min functions can be converted analogously.

C. Max Function

We follow the concepts presented in [28] for linearizing max functions. For calculating $h_{us_-}^{on}$ in (8), we introduce an auxiliary variable H_u^{on} for each unit u to formulate (25)-(26) and use (27)-(28) instead of (9) for calculating $h_{u(s_-,s)}^{on}$.

$$H_u^{on} \ge \Phi_{ut}^{on} \quad \forall u, \forall t \quad (25)$$

$$h_{us_-}^{on} = n_- - H_u^{on} + 1 \quad \forall u \quad (26)$$

$$h_{u(s_-,s)}^{on} \ge 0 \quad \forall u \quad (27)$$

$$h_{u(s_-,s)}^{on} \ge \left(T_u^{on} - h_{us_-}^{on}\right) \quad \forall u \quad (28)$$

As explained in Section IV, $h_{u(s_-,s)}^{on}$ appears with positive terms in the objective function of SP_{s_-} with penalizing consistency constraints (17) and (18). As explained in [28], this penalization strategy along with (25)-(28) results in $h_{u(s_-,s)}^{on}$ taking the maximum of $\left(0, T_u^{on} - h_{us_-}^{on}\right)$. The minimization problem tries to reduce $h_{u(s_-,s)}^{on}$, and (25)-(28) enforce $h_{u(s_-,s)}^{on}$ to be larger than zero and $T_u^{on} - h_{us_-}^{on}$. In addition, $H_u^{on} \ge \Phi_{ut}^{on}$ means that H_u^{on} is larger than or equal to the largest element of vector Φ_u^{on}. Since the objective function tends to minimize and the penalty term $h_{u(s_-,s)}^{on}$ pertaining to H_u^{on} appears with positive terms in the objective function, H_u^{on} sticks to the smallest possible value that is the maximum of vector Φ_u^{on}. Other max functions can be modeled in a similar manner.

IV. Coordination Algorithm

We propose an accelerated analytical target casting (A-ATC) algorithm to coordinate SCUC subproblems. An acceleration technique and an initialization strategy are proposed to enhance the convergence performance of ATC.

A. Normal Parallel ATC

Consider SP_{s_-} and SP_s formulated in Section II. For brevity

of equations, we use a vector format to represent shared variables, e.g., $r_{s\_,s} = \{p_{ut_o s\_}, I_{ut_o s\_}, h^{on}_{u(s\_,s)}, h^{off}_{u(s\_,s)}\}$ for $SP_{s\_}$. We use parallel ATC, which is based on augmented Lagrangian relaxation, to coordinate subproblems [22]. SCUC subproblems are in level two, and a coordinator is in level one. The actual shared variables r (called responses in the context of ATC) are duplicated to create a set of auxiliary variables χ (called targets) for the coordinator and separate subproblems to make them solvable in parallel [22]. $r - \chi$ are consistency constraints between the coordinator and subproblems. Violations of consistency constraints are penalized in objective functions (3a) and (4a). The relaxed $SP_{s\_}$ at iteration k is written as:

$$\left(x^k_{s\_}, r^k_{s\_,s}\right) = \text{argmin} \sum_{t=1}^{n\_+1} \sum_{u} f(p_{ut}, I_{ut}) \quad (29)$$
$$-\lambda^{k\dagger}_{s\_,s}\left(\chi^{k-1}_{s\_,s} - r_{s\_,s}\right) + \rho\left\|\chi^{k-1}_{s\_,s} - r_{s\_,s}\right\|^2$$
$$s.t. \qquad (3b), (7)-(16)$$

where $x^k_{s\_}$ and $r^k_{s\_,s}$ are decision variables while $\chi^{k-1}_{s\_,s}$ are known target variables received from the coordinator. λ and ρ are Lagrange and penalty multipliers. Superscript † is the transpose operator. Analogously, SP_s is formulated as:

$$\left(x^k_s, r^k_{s,s\_}, r^k_{s,s_+}\right) = \text{argmin} \sum_{t=1}^{n+1} \sum_{u} f(p_{ut}, I_{ut}) \quad (30)$$
$$-\lambda^{k\dagger}_{s\_,s}\left(\chi^{k-1}_{s,s\_} - r_{s,s\_}\right) + \rho\left\|\chi^{k-1}_{s,s\_} - r_{s,s\_}\right\|^2$$
$$+\lambda^{k\dagger}_{s,s_+}\left(\chi^{k-1}_{s,s_+} - r_{s,s_+}\right) + \rho\left\|\chi^{k-1}_{s,s_+} - r_{s,s_+}\right\|^2$$
$$s.t. \qquad (4b), \qquad \underbrace{(7)-(16)}_{\text{for transition from } s\_ \text{ to } s} \qquad \& \qquad \underbrace{(7)-(16)}_{\text{for transition from } s \text{ to } s_+}$$

The coordinator receives r^k and solves the following problem to determine target variables χ^k.

$$\chi^k = \text{argmin}\, \lambda^\dagger(\chi - r^k) + \rho\|\chi - r^k\|^2 \quad (31)$$

For penalty terms with a positive sign for λ, Lagrange multipliers are updated as follows:

$$\lambda^k = \lambda^{k-1} + 2\rho^2(\chi^k - r^k) \quad (32)$$

and for penalty terms with a negative sign for λ, we have:

$$\lambda^k = \lambda^{k-1} + 2\rho^2(r^k - \chi^k) \quad (33)$$

This formulation can be generalized for multiple subproblems. λ and ρ need to be initialized in acceptable ranges. If ρ are selected too large, the convergence speed may increase; however, it increases the chance of losing optimality [24]. Although selecting small ρ enhances solution accuracy, the convergence speed degrades [24]. Knowledge about suitable ranges for λ and ρ will be gained by solving the problem multiple times and using historical information obtained from past implementations of the distributed SCUC.

B. Proposed Accelerated ATC

The convergence performance of ATC might degrade when more iterations are carried out and the solution becomes close to the optimal point, or when the optimal point is in a shallow area (or a ravine). Near the optimal point, the term $\chi^k - r^k$ might become small that leads to updated multipliers in iteration $k+1$ that are almost the same as them in iteration k, i.e., $\lambda^{k+1} \approx \lambda^k$. This slows the convergence speed.

We present an accelerated ATC based on the technique that first proposed by Nesterov to accelerate gradient descent methods [29, 30]. The proposed accelerated ATC utilizes a prediction type acceleration step. The concept of momentum is used to prevent the algorithm from deceleration while more iterations are carried out. After each iteration k, the cumulated gradient of previous iterations (i.e., momentum) is calculated as the predicted direction in the next iteration, and a big jump is made in that direction. Then, the gradient is measured, and a correction is made to avoid moving forward too fast.

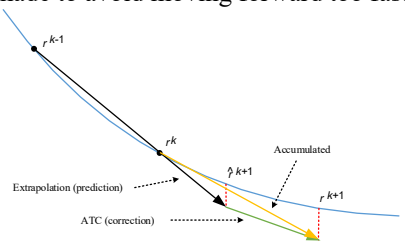


Fig. 4. Prediction and correction procedure.

Accelerated ATC (A-ATC): At iteration k, we predict that to what point ATC directs coupling variables and Lagrange multipliers at iteration $k+1$. To do so, we add the effects of momentum in ATC. Consider coupling variable r at two consecutive iterations, i.e., r^{k-1} and r^k, as shown in Fig. 4. To predict r^{k+1}, we connect a line between points $(k-1, r^{k-1})$ and (k, r^k). The line equation is:

$$y = (r^k - r^{k-1})x + kr^{k-1} - (k-1)r^k \quad (34)$$

By replacing $x = k+1$, we predict the next point as:

$$y = (r^k - r^{k-1})(k+1) + kr^{k-1} - (k-1)r^k = 2r^k - r^{k-1} \quad (35)$$

We reorganize (35) and name $y = \hat{r}^{k+1}$ that is the predicted r at iteration $k+1$.

$$\hat{r}^{k+1} = r^k + (r^k - r^{k-1}) \quad (36)$$

The first term is the current r and the second term is the line slope. We add a coefficient $\eta \in [0,1]$ to the slope to have a better prediction. In the first few iterations of ATC, large oscillations may be observed, and the jump should not be large, otherwise, it may cause a large error on $\hat{r}^{k+1}$. Thus, in the first few iterations $\eta \ll 1$ to prevent large error and it becomes gradually larger to enhance the speed. We use the following expressions to increase η gradually until $\eta \cong 1$ [29, 30].

$$\alpha_0 = 1 \quad \& \quad \alpha_{k+1} = (1 + \sqrt{1 + 4\alpha_k^2})/2 \quad (37)$$
$$\eta = \frac{\alpha_k - 1}{\alpha_{k+1}} \quad (38)$$

Since α_k has an increasing trend, η becomes closer to one over the course of iterations. This leads to a larger ATC step size, especially when r approaches the optimal point. Near the optimal point, the slope becomes smaller and does not vary considerably. By this prediction-correction procedure, the convergence performance of ATC is enhanced.

A-ATC for SCUC: The penalty term in the objective function (29) of $SP_{s\_}$ at iteration k is modified as:

$$-\hat{\lambda}^{k\dagger}_{s\_,s}\left(\hat{\chi}^{k-1}_{s\_,s} - r_{s\_,s}\right) + \rho\left\|\hat{\chi}^{k-1}_{s\_,s} - r_{s\_,s}\right\|^2 \quad (39)$$

where $\chi_{s\_,s}$ and $\lambda_{s\_,s}$ are replaced by $\hat{\chi}_{s\_,s}$ and $\hat{\lambda}_{s\_,s}$. The two penalty terms in the objective function (30) of SP_s at iteration k

are modified by replacing $\chi_{s,s\_}$, χ_{s,s_+}, $\lambda_{s\_,s}$, and $\lambda_{s,s\_}$ with $\hat{\chi}_{s,s\_}$, $\hat{\chi}_{s,s_+}$, $\hat{\lambda}_{s\_,s}$, and $\hat{\lambda}_{s,s_+}$. The new parameters shown by $(\hat{\cdot})$ are predictions of coordinator variables and Lagrange multipliers. The predicted values used at iteration $k+1$ are calculated based on the actual shared variables and penalty multipliers obtained at iterations k and $k-1$ as follows:

$$\alpha_{k+1} = \left(1+\sqrt{1+4\alpha_k^2}\right)/2 \tag{40}$$

$$\hat{\chi}^{k+1} = \chi^k + \frac{\alpha_k - 1}{\alpha_{k+1}}\left(\chi^k - \chi^{(k-1)}\right) \tag{41}$$

$$\hat{\lambda}^{k+1} = \lambda^k + \frac{\alpha_k - 1}{\alpha_{k+1}}(\lambda^k - \lambda^{k-1}) \tag{42}$$

Convergence analysis of A-ATC is provided in the Appendix. Implementing A-ATC is similar to the classical ATC with a few more steps. The proposed A-ATC can be used instead of the classical ATC to solve various distributed optimization problems, such as distributed SCUC and optimal power flow

Initialization Strategy: ATC, similar to most distributed algorithms, is sensitive to initialization. To choose a good-enough starting point, we take advantage of power system characteristics and propose a method to initialize coupling variables. We ignore the coupling intervals and intertemporal connectivity between subhorizons. That means the coupling variables between subproblems are disregarded and subproblems are considered to be completely independent. The SCUC subproblems are solved once. The achieved results might not be feasible as the connectivity between subproblems is ignored, but they are close to the optimal solution as all other SCUC constraints at most intervals are respected. Hence, this solution can be used as a good starting point for initializing the distributed algorithm. Our simulation results illustrate that this initialization strategy works well for all studied cases.

The pseudocode for the A-ATC-based distributed SCUC is given in the following Algorithm.

Algorithm Pseudocode for the proposed A-ATC-based distributed SCUC

1: Decompose the considered horizon into multiple subhorizons
2: Ignore the coupling intervals and shared variables
3: Solve SCUC subproblems in parallel
4: Use the obtained results to initialize $\hat{\chi}^0$
5: Initialize multipliers λ^0, ρ in their acceptable ranges and set $k=0$
6: **while** $|\chi^k - r^k| > \varepsilon$, $k = k+1$ **do**
7: Solve SCUC subproblems in parallel and determine r^k (outputs of SCUC subproblems and inputs for the coordinator)
8: Solve the coordinator problem to determine χ^k (outputs of the coordinator and inputs for SCUC subproblems)
9: Exchange r^k and χ^k between the coordinator and subproblems
10: Update λ^k by (32) or (33)
11: Calculate α_{k+1} by (40)
12: Update $\hat{\chi}^{k+1}$ and $\hat{\lambda}^{k+1}$ by (41) and (42) (inputs for the coordinator and SCUC subproblems)
13: **end while**

V. Case Study

The proposed algorithm is implemented on a 3-bus system, the IEEE 24-bus system, the IEEE 118-bus system, and a 472-bus system. All systems information is given in [31]. The considered scheduling problem includes 72 intervals, each representing an hour. For most cases, we have observed that decomposing the consider scheduling horizon into three subhorizons, each with 24 intervals, will lead to good time-saving. Having three subhorizons also shows the situation of modeling intertemporal constraints for a middle subproblem. Hence, without loss of generality, we have divided the scheduling horizon into three subhorizons. In addition, we study the impact of number of subhorizons on solution performance using the IEEE 24-bus system. To initialize Lagrange multipliers and penalty parameters multipliers, we have run multiple cases with λ^0 and ρ in the range of 0.01 to 100. For all cases, setting $\lambda^0 = \rho = 1$ for continuous shared variables and $\rho = 3$ for integer shared variables provides good results. All simulations are carried out using YALMIP toolbox [32] and Gurobi solver on a PC with 16GB of RAM. The reported times for the distributed SCUC include the initialization time and time that A-ATC takes to converge. The following convergence index is used to measure the relative distance between operation costs determined by the distributed SCUC, f^d, and benchmark results obtained by the centralized SCUC, f^*:

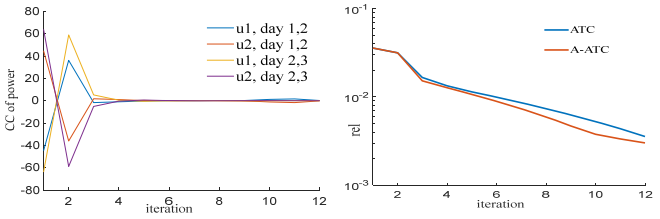
$$rel = \frac{|f^* - f^d|}{f^*} \tag{43}$$

A. 3-Bus System

This small case serves as a tutorial for the proposed algorithm and to show that the UC results obtained by the proposed algorithm and the centralized SCUC are the same, not only at the boundary intervals (shown in Tables I-V) but also at all other intervals. Two load patterns are considered to analyze all three possible situations explained in Section II. D.

Case1: The load pattern one is considered. Intervals 1-25, 25-49, and 49-72 belong, respectively, to subhorizons one, two, and three. A-ATC, without applying the suggested initialization strategy, converges after 12 iterations. As shown in Fig. 5a, consistency constraints (5) and (6) are roughly zero upon convergence that means power generated by units in coupling intervals are the same from the perspective of consecutive subproblems. Table I shows units' on/off status at the boundary intervals of SP_1 and SP_2 after the first iteration. The same results are obtained with and without considering on/off time consistencies since minimum on/off times (i.e., 5 hours) are already satisfied. Consider the on/off status of unit two. SP_1 sends $h^{on}_{2(1,2)} = 3$ to ask SP_2 to keep unit two on for three more hours. From the perspective of SP_2, this is satisfied as according to (12) $h^{on}_{2(2,1)} = 3$. This is situation 1 in Section II.D in which shared variables corresponding to on/off status of the units are equal from the first iteration and consistency constraints (17) are satisfied. Table II shows the units' on/off status at the boundary intervals between SP_2 and SP_3 after the first iteration. SP_2 sends $h^{on}_{2(2,3)} = 0$ to SP_3 for unit two. On the other hand, according to (12), $h^{on}_{2(3,2)} = \min\{0,2\} = 0$, which means unit two in SP_3 stays on only at the first two intervals. This is situation 3 in Section II.D in which consistency constraints (18) are satisfied at the first iteration. However, the generation in coupling intervals 25 and 49 must be equal to satisfy consistency constraints (5) and (6) for generation ramp constraints. All consistency constraints are satisfied upon convergence of the algorithm after 12 iterations.

Figure 5b shows the rel index, and Table III compares cost obtained by the distributed SCUC and the benchmark cost determined by the centralized SCUC. Upon convergence, the



(a) (b)

Fig. 5. a) The difference between shared variables in coupling intervals and b) the rel index for case 1of 3-bus system.

TABLE I
UNITS ON/OFF STATUS IN BOUNDARY INTERVALS BETWEEN DAYS ONE AND TWO AFTER THE FIRST ITERATION (CASE 1)

	Subhorizon one						Subhorizon two					
Hour	20	21	22	23	24	25	25	26	27	28	29	30
Unit 1	1	1	1	1	1	1	1	1	1	1	1	1
Unit 2	0	0	0	1	1	1	1	1	1	1	1	1

TABLE II
UNITS ON/OFF STATUS IN BOUNDARY INTERVALS BETWEEN DAYS TWO AND THREE AFTER THE FIRST ITERATION (CASE 1)

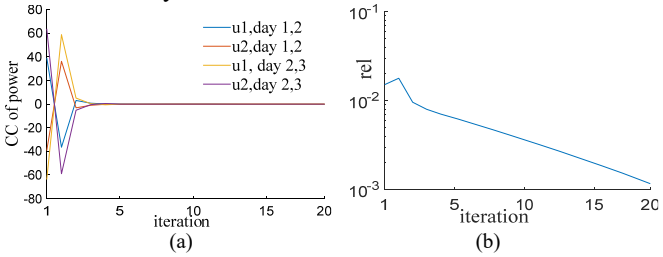
	Subhorizon two						Subhorizon three					
Hour	44	45	46	47	48	49	49	50	51	52	53	54
Unit 1	1	1	1	1	1	1	1	1	1	1	1	1
Unit 2	1	1	1	1	1	1	1	1	1	0	0	0

cost of distributed SCUC is \$668,713, and the rel index is 0.003. For the sake of comparison, we have simulated ATC and A-ATC. As shown in Fig. 5b, rel obtained by A-ATC is always less than that obtained by ATC. This means that after a specific number of iterations, A-ATC provides a more accurate solution than the normal ATC.

To enhance the distributed SCUC performance, we apply the suggested initialization strategy and re-run A-ATC. The algorithm converges after three iterations. The units' on/off status and power dispatches are the same as those obtained by the centralized SCUC. Operation costs obtained by A-ATC is \$666,709 that is equal to the cost determined by the classical centralized SCUC (i.e., $rel \approx 0$).

Case 2: Load pattern two is considered. Table IV shows the units' on/off status after the first iteration. According to (14), for unit two, SP_1 sends $h^{off}_{2(1,2)} = 3$ to SP_2. But, according to (16), from the view of SP_2, it is optimal to have $h^{off}_{2(1,2)} = 2$ (situation 2 in Section II.D). Therefore, if subproblems are solved separately, $h^{off}_{2(1,2)} \neq h^{off}_{2(2,1)}$ and SCUC results are infeasible.

With the stopping criterion of 0.01 MW, A-ATC, without applying the suggested initialization strategy, converges after 20 iterations. Figure 6 shows the difference between shared variables and rel over iterations. Table V depicts the units' on/off status at the boundary intervals upon convergence. The unit commitment results and power generations are similar to those obtained by the classical centralized SCUC.



(a) (b)

Fig. 6. a) The difference between shared variables in coupling intervals and b) the rel index for case 2 of the 3-bus system.

TABLE III
RESULTS FOR 3-BUS SYSTEM (CASES 1) W/WO INITIALIZATION STRATEGY

Algorithm	Cost (\$)	Iteration	rel
Centralized	666,709	-	-
Distributed A-ATC	668,713	12	0.003
Distributed A-ATC+ initialization	666,709	2+1	≈0

TABLE IV
UNITS ON/OFF STATUS IN BOUNDARY INTERVALS BETWEEN DAYS ONE AND TWO AFTER THE FIRST ITERATION (CASE 2)

	Subhorizon one						Subhorizon two					
Hour	20	21	22	23	24	25	25	26	27	28	29	30
Unit 1	1	1	1	1	1	1	1	1	1	1	1	1
Unit 2	1	1	1	0	0	0	0	0	1	1	1	1

TABLE V
UNITS ON/OFF STATUS IN BOUNDARY INTERVALS BETWEEN DAYS ONE AND TWO UPON CONVERGENCE (CASE 2)

	Subhorizon one						Subhorizon two					
Hour	20	21	22	23	24	25	25	26	27	28	29	30
Unit 1	1	1	1	1	1	1	1	1	1	1	1	1
Unit 2	1	1	1	1	1	1	1	1	1	1	1	1

TABLE VI
RESULTS FOR THE 3-BUS SYSTEM (CASE 2) W/WO INITIALIZATION STRATEGY

Algorithm	Cost (\$)	Iteration	rel
Centralized	817,755	-	-
Distributed A-ATC	818,710	20	0.001
Distributed A-ATC+ initialization	817,755	10+1	≈0

TABLE VII
RESULTS OF DIFFERENT ALGORITHMS FOR 24-BUS SYSTEM

Algorithm	Iteration	rel	Time (s)
Centralized NCUC	-	-	17
NCUC with A-ATC + initialization	2+1	≈0	2
Centralized SCUC	-	-	135
SCUC with A-ATC + initialization	3+1	≈ 3e-8	12

TABLE VIII
POWER OF UNITS IN COUPLING INTERVALS (24-BUS SYSTEM)

Unit No.	Hour 25, subhorizon 1	Hour 25, subhorizon 2	Hour 49, subhorizon 2	Hour 49, subhorizon 3
1	184.5	184.5	177.7	177.7
3	153.7	153.7	148.1	148.1
5	184.5	184.5	177.7	177.7
7	162.1	162.1	156.4	156.4
9	204.5	204.5	197.7	197.7

TABLE IX
RESULTS FOR THE IEEE 118-BUS SYSTEM

Algorithm	Iteration	rel	Time (s)
Centralized	-	-	3818
ATC + initialization	4+1	≈e-5	216
A-ATC + initialization	4+1	≈e-6	216

A-ATC is applied with the suggested initialization strategy. As shown in Table VI, operation cost is \$817,755 using distributed SCUC and the classical centralized SCUC. The suggested initialization strategy reduces the number of iterations from 20 to 11 and makes $rel \approx 0$.

B. IEEE 24-Bus System

Two cases are studied, network constrained UC (NCUC) with no contingency and SCUC with ten plausible contingencies. A-ATC with the initialization strategy is implemented for both cases. The distributed NCUC converges after three iterations within two seconds. As depicted in Table VII, costs determined by distributed and centralized NCUC algorithms are \$3,774,232. The distributed algorithm decreases the solution time by 88%. Table VIII shows power generated by several sample units in coupling intervals. The rel becomes zero upon convergence.

For SCUC, ten scenarios for transmission line outage are considered at each interval. This increases the size of the problem. The distributed SCUC converges to the same cost as of that for the centralized SCUC (i.e., $3,833,162) while being 91% faster. This implies that when the size of the problem increases, the effectiveness of the distributed SCUC is more prominent.

Number of Subhorizons: We have studied the impact of number of subhorizons on the solution time. We have decomposed the scheduling horizon into different numbers of subhorizons and run the distributed SCUC. Figure 7 shows the solution time versus the number of subhorizons for the 24-bus system. It is observed that increasing the number of subhorizons reduces the solution time; however, decomposition the scheduling horizons beyond a certain number, which is nine for this case, increases the solution time. Such a trend is widely observed in parallel computing approaches. Breaking the horizon into two and three subhorizons leads to the best time-saving. Increasing the number of subhorizons beyond three does not result in significant time-saving.

We have also studied the impact of load on the optimal number of subhorizons. Several low-load and high-load scenarios are analyzed. Although we have observed that the load changes the curve pattern shown in Fig. 7, one pattern, similar to Fig. 7, is obtained for all load conditions between 70%~100% of the baseload (Fig. 7 is plotted for the baseload) and another pattern for load conditions larger than the baseload. Hence, we conclude that one can categorize the load into low, medium, and high (e.g., seasonal load), decompose the problem into several subhorizons for each loading condition, and select the best number of subhorizons for each loading condition. This is an efficient approach as a user can use the same number of subhorizon for many similar load scenarios.

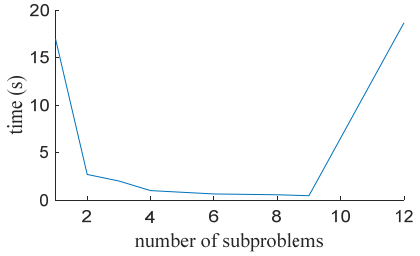


Fig. 7. Time versus number of subproblems for the IEEE 24-bus system.

TABLE X

RESULTS FOR THE 472-BUS SYSTEM

Algorithm	Iteration	Cost ($)	Time (hour)
Centralized	-	17,866,016	111
A-ATC + initialization	6+1	17,768,289	14

C. IEEE 118-Bus System

As shown in Table IX, the centralized approach provides a cost of $4,928,242 after 3818 seconds. The distributed algorithm converges to a cost of $4,928,268 after five iterations within 216 seconds that is 94% faster than the centralized approach. We also compare ATC and A-ATC. A-ATC provides a smaller rel index upon convergence.

To have a better comparison, we stop the centralized approach when a cost of $4,928,268 (the same as A-ATC) is obtained. While A-ATC converges after 216 seconds, the centralized approach takes 2406 seconds. This means that the centralized SCUC takes 91% more computation time to converge to the same solution as the distributed algorithm.

D. A 472-Bus System

The superiority of the proposed algorithm as compared to the centralized SCUC is more considerable for larger cases. We combine four IEEE 118-bus systems to build a 472-bus system. Table X shows the results. We stop the centralized approach after 111 hours, and the achieved cost is $17,866,016. However, the proposed distributed algorithm converges after seven iterations to a cost of $17,768,289 within 14 hours.

VI. CONCLUSION

A strategy is proposed to decompose SCUC over the scheduling time horizon, create several subproblems each for a subhorizon, and model temporal interdependencies between subproblems. The proposed strategy is called a temporal decomposition. The concept of coupling intervals is introduced to model ramping limitation of generating units between subproblems. In addition, several counting auxiliary variables are determined to coordinate minimum on/off times for transition between consecutive subproblems. An accelerated ATC algorithm with an initialization strategy is proposed to coordinate subproblems. The simulation results show that the proposed algorithm obtains the same SCUC results (i.e., binary variables for units' on/off status and power dispatch) as the centralized SCUC while reducing the computation time considerably. For instance, for the IEEE 118-bus system, the solution time decreases by 94%. We have observed that as the size of the problem increases, the distributed algorithm shows better performance than the centralized SCUC. The results show the privilege of A-ATC to ATC and effectiveness of the suggested initialization strategy to enhance the convergence performance. For the future work, we will focus on developing methods to find the best time intervals for decomposing the considered scheduling horizon and the optimal number of subhorizons.

APPENDIX

In this Appendix, we discuss the convergence of A-ATC. Consider the following problem:

$$\min F_1(r_1) + F_2(r_2) \qquad \text{(a1)}$$
$$s.t. \quad r_1 = r_2$$

Assume that F_1 and F_2 are strongly convex. In the normal parallel ATC, a coordinator exists to coordinate the two subproblems. Subproblems one and two send updated r_1 and r_2 to the coordinator and receive χ. Consider the coordinator's optimization problem as $G(\chi)$ and $H(r) = F_1(r_1) + F_2(r_2)$. We restate the problem (a1) as:

$$\min H(r) + G(\chi) \qquad \text{(a2)}$$
$$s.t. \quad \chi = r$$

Then:

$$r^+ = \operatorname{argmin} H(r) + \lambda(\chi - r) + \rho^2(\chi - r)^2 \qquad \text{(a3)}$$
$$\chi^+ = \operatorname{argmin} G(\chi) + \lambda(\chi^+ - r) + \rho^2(\chi^+ - r)^2 \qquad \text{(a4)}$$
$$\lambda^+ = \lambda + 2\rho^2\,(\chi^+ - r^+) \qquad \text{(a5)}$$

where $\chi^+ = \chi^{k+1}$ and $r^+ = r^{k+1}$. We write the dual conjugate function of (a2) as:

$$\max D(\lambda) = -H^*(-\lambda) - G^*(\lambda) \qquad \text{(a6)}$$

Introducing $\lambda^{1/2} = \lambda + 2\rho^2(\chi^+ - r)$ and $\lambda^+ = \lambda + 2\rho^2(\chi^+ - r^+)$, and using optimality conditions, we calculate r^+ and χ^+ as follows:

$$\partial H(r^+) = -\lambda^{1/2}, r^+ = \nabla H^*\left(-\lambda^{\frac{1}{2}}\right) \quad \text{(a7)}$$

$$\partial G(\chi^+) = +\lambda^+, \chi^+ = \nabla G^*(\lambda^+) \quad \text{(a8)}$$

To calculate the convergence rate of the parallel ATC, the relation between $(\lambda^*) - D(\lambda^+)$ and λ should be determined. According to (a6):

$$D(\lambda^*) - D(\lambda^+) = -H^*(-\lambda^*) - G^*(\lambda^*) + H^*(-\lambda^+) + G^*(\lambda^+) \quad \text{(a9)}$$

We rewrite the term $H^*(\lambda^*) - H^*(-\lambda^+)$ in (a9) as follows:

$$H^*(\lambda^*) - H^*(-\lambda^+) = H^*(\lambda^*) - H^*\left(-\lambda^{\frac{1}{2}}\right) + H^*\left(-\lambda^{\frac{1}{2}}\right) - H^*(-\lambda^+) \quad \text{(a10)}$$

After some calculations and using convex properties:

$$H^*(\lambda^*) - H^*(-\lambda^+) \geq (\lambda^* - \lambda^+)\left(\nabla H^*\left(-\lambda^{\frac{1}{2}}\right)\right) - \frac{1}{2\rho^2}(\|\lambda^+ - \lambda\|) \quad \text{(a11)}$$

Similarly, for the term $G^*(\lambda^*) - G^*(-\lambda^+)$ in (a9):

$$G^*(\lambda^*) - G^*(-\lambda^+) \geq (\lambda^* - \lambda^+)\left(\nabla G^*(\lambda^+)\right) \quad \text{(a12)}$$

By incorporating (a10) and (a11) into (a9), we then have:

$$D(\lambda^+) - D(\lambda^*) \geq \frac{1}{\rho}(\lambda^* - \lambda)(\lambda - \lambda^+) + \frac{1}{2\rho}\|\lambda^+ - \lambda\|^2 \quad \text{(a13)}$$

To convert (a3)- (a5) to the accelerated ATC, we replace $\lambda = \hat{\lambda}^+$ and $\chi = \hat{\chi}^+$.

$$D(\lambda^+) - D(\lambda^*) \geq \frac{1}{\rho}\left(\hat{\lambda} - \lambda^*\right)\left(\lambda^+ - \hat{\lambda}^+\right) + \frac{1}{2\rho}\left\|\lambda^+ - \hat{\lambda}^+\right\|^2 \quad \text{(a14)}$$

Using the telescopic summation and after some simplifications:

$$D(\lambda^*) - D(\lambda^+) \leq \frac{2\left\|\hat{\lambda}_1 - \lambda^*\right\|}{\rho^2(k+2)^2} \quad \text{(a15)}$$

This proves the convergence of A-ATC whose convergence rate is $O(1/k^2)$.

REFERENCES

[1] M. F. Anjos and A. J. Conejo, "Unit commitment in electric energy systems," *Foundations and Trends® in Electric Energy Systems,* vol. 1, no. 4, pp. 220-310, 2017.

[2] J. Zhu, "Security-Constrained Economic Dispatch," *Optimization of Power System Operation,* pp. 141-210, 2009.

[3] M. H. Amini *et al.*, "Decomposition Methods for Distributed Optimal Power Flow: Panorama and Case Studies of the DC Model," in *Classical and Recent Aspects of Power System Optimization*: Elsevier, 2018, pp. 137-155.

[4] A. Kargarian *et al.*, "Toward distributed/decentralized DC optimal power flow implementation in future electric power systems," *IEEE Transactions on Smart Grid,* vol. 9, no. 4, pp. 2574-2594, 2018.

[5] J. Mohammadi, G. Hug, and S. Kar, "Agent-based distributed security constrained optimal power flow," *IEEE Transactions on Smart Grid,* vol. 9, no. 2, pp. 1118-1130, 2018.

[6] Y. Wang, S. Wang, and L. Wu, "Distributed optimization approaches for emerging power systems operation: A review," *Electric Power Systems Research,* vol. 144, pp. 127-135, 2017.

[7] D. K. Molzahn *et al.*, "A survey of distributed optimization and control algorithms for electric power systems," *IEEE Transactions on Smart Grid,* vol. 8, no. 6, pp. 2941-2962, 2017.

[8] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning,* vol. 3, no. 1, pp. 1-122, 2011.

[9] A. J. Conejo, F. J. Nogales, and F. J. Prieto, "A decomposition procedure based on approximate Newton directions," *Mathematical programming,* vol. 93, no. 3, pp. 495-515, 2002.

[10] G. Cohen, "Auxiliary problem principle and decomposition of optimization problems," *Journal of optimization Theory and Applications,* vol. 32, no. 3, pp. 277-305, 1980.

[11] A. R. Malekpour and A. Pahwa, "Stochastic networked microgrid energy management with correlated wind generators," *IEEE Transactions on Power Systems,* vol. 32, no. 5, pp. 3681-3693, 2017.

[12] H. M. Kim, N. F. Michelena, P. Y. Papalambros, and T. Jiang, "Target cascading in optimal system design," *Journal of mechanical design,* vol. 125, no. 3, pp. 474-480, 2003.

[13] R. Baldick, B. H. Kim, C. Chase, and Y. Luo, "A fast distributed implementation of optimal power flow," *IEEE Transactions on Power Systems,* vol. 14, no. 3, pp. 858-864, 1999.

[14] S. Bahrami, M. H. Amini, M. Shafie-Khah, and J. P. Catalao, "A decentralized renewable generation management and demand response in power distribution networks," *IEEE Transactions on Sustainable Energy,* vol. 9, no. 4, pp. 1783-1797, 2018.

[15] G. Binetti, A. Davoudi, F. L. Lewis, D. Naso, and B. Turchiano, "Distributed consensus-based economic dispatch with transmission losses," *IEEE Transactions on Power Systems,* vol. 29, no. 4, pp. 1711-1720, 2014.

[16] J.-P. Watson and D. L. Woodruff, "Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems," *Computational Management Science,* vol. 8, no. 4, pp. 355-370, 2011.

[17] G. Cong, C. Meyers, D. Rajan, and T. Parriani, "Parallel strategies for solving large unit commitment problems in the California ISO planning model," in *2015 IEEE International Parallel and Distributed Processing Symposium*, 2015, pp. 710-719: IEEE.

[18] I. Aravena and A. Papavasiliou, "A distributed asynchronous algorithm for the two-stage stochastic unit commitment problem," in *2015 IEEE Power & Energy Society General Meeting*, 2015, pp. 1-5: IEEE.

[19] Y. Fu, M. Shahidehpour, and Z. Li, "AC contingency dispatch based on security-constrained unit commitment," *IEEE Transactions on Power Systems,* vol. 21, no. 2, pp. 897-908, 2006.

[20] A. Ahmadi-Khatir, A. J. Conejo, and R. Cherkaoui, "Multi-area energy and reserve dispatch under wind uncertainty and equipment failures," *IEEE Transactions on Power Systems,* vol. 28, no. 4, pp. 4373-4383, 2013.

[21] M. J. Feizollahi, M. Costley, S. Ahmed, and S. Grijalva, "Large-scale decentralized unit commitment," *International Journal of Electrical Power & Energy Systems,* vol. 73, pp. 97-106, 2015.

[22] A. Kargarian, Y. Fu, and Z. Li, "Distributed security-constrained unit commitment for large-scale power systems," *IEEE Transactions on Power Systems,* vol. 30, no. 4, pp. 1925-1936, 2015.

[23] F. Safdarian, O. Ciftci, and A. Kargarian, "A time decomposition and coordination strategy for power system multi-interval operation," in *2018 IEEE Power & Energy Society General Meeting (PESGM)*, 2018, pp. 1-5: IEEE.

[24] D. P. Bertsekas, "Nonlinear programming," *Journal of the Operational Research Society,* vol. 48, no. 3, pp. 334-334, 1997.

[25] S. DorMohammadi and M. Rais-Rohani, "Exponential penalty function formulation for multilevel optimization using the analytical target cascading framework," *Structural and Multidisciplinary Optimization,* vol. 47, no. 4, pp. 599-612, 2013.

[26] S. Tosserams, L. Etman, P. Papalambros, and J. Rooda, "An augmented Lagrangian relaxation for analytical target cascading using the alternating direction method of multipliers," *Structural and multidisciplinary optimization,* vol. 31, no. 3, pp. 176-189, 2006.

[27] Y. Sun, Z. Li, W. Tian, and M. Shahidehpour, "A Lagrangian Decomposition Approach to Energy Storage Transportation Scheduling in Power Systems," *IEEE Transactions on Power Systems,* vol. 31, no. 6, pp. 4348-4356, 2016.

[28] H. P. Williams, *Model building in mathematical programming*. John Wiley & Sons, 2013.

[29] Y. Nesterov, "A method of solving a convex programming problem with convergence rate O (1/k2)," in *Soviet Mathematics Doklady*, 1983, vol. 27, no. 2, pp. 372-376.

[30] T. Goldstein, B. O'Donoghue, S. Setzer, and R. Baraniuk, "Fast alternating direction optimization methods," *SIAM Journal on Imaging Sciences,* vol. 7, no. 3, pp. 1588-1623, 2014.

[31] Available: https://sites.google.com/site/aminkargarian/test-system-data/temporal-decomposition-for-security-constrained-unit-commitment

[32] J. Lofberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, 2004, pp. 284-289: IEEE.