A Fast Solution Method for Large-Scale Unit Commitment Based on Lagrangian Relaxation and Dynamic Programming

Jiangwei Hou, Student Member, IEEE, Qiaozhu Zhai, Member, IEEE, Yuzhou Zhou, Member, IEEE, and Xiaohong Guan, Fellow, IEEE

Abstract—The unit commitment problem (UC) is crucial for the operation and market mechanism of power systems. With the development of modern electricity, the scale of power systems is expanding, and solving the UC problem is also becoming more and more difficult. To this end, this article proposes a new fast solution method based on Lagrangian relaxation and dynamic programming. Firstly, the UC solution is estimated to be an initial trial UC solution by a fast method based on Lagrangian relaxation. This initial trial UC solution fully considers the system-wide constraints. Secondly, a dynamic programming module is introduced to adjust the trial UC solution to make it satisfy the unit-wise constraints. Thirdly, a method for constructing a feasible UC solution is proposed based on the adjusted trial UC solution. Specifically, a feasibility-testing model and an updating strategy for the trial UC solution are established in this part. Numerical tests are implemented on IEEE 24-bus, IEEE 118-bus, Polish 2383-bus, and French 6468-bus systems, which verify the effectiveness and efficiency of the proposed method.

Index Terms—Dynamic programming, fast solution, Lagrangian relaxation, large-scale unit commitment.

NOMENCLATURE

A. Indices and Sets

\[ N, L, M, T \]

Total number of the units, the lines, the load buses, and the scheduling periods.

\[ c, e, i, k, l, m \]

Index of the paths in NSTD, \( c = 1, 2, \ldots \)

Index of the edges in the path \( c \).

Index of units, \( i = 1, 2, \ldots, N \)

Index of iterations of the method, \( k = 1, 2, \ldots \)

Index of transmission lines, \( l = 1, 2, \ldots, L \)

Index of load buses, \( m = 1, 2, \ldots, M \)

B. Variables

\[ z, p, v_{i,t}^+, v_{i,t}^-, v_0^+, v_0^- \]

Non-negative and dispatch variables

\[ \hat{z}, \hat{p} \]

An approximate dual solution.

\[ \lambda_{0,t}, \lambda_{i,t}^+, \lambda_{i,t}^- \]

Dual variables associated with the power transfer distribution factors (PTDF) for the mth net nodal load at period t (MW).

\[ \hat{\lambda}_{k}^+, \hat{\lambda}_{k}^- \]

A constant threshold for determining \( \lambda_{k}^+ \).

\[ \hat{\lambda}_{i}^+, \hat{\lambda}_{i}^- \]

A constant threshold for determining \( \lambda_{i}^+ \).

\[ s_d, s_m, s_F, s_R \]

Minimum up/down time of the unit \( i \) (hour).

\[ \Delta_{i}, U_i, R \]

Ramp up/down limits of unit \( i \) (MW).

\[ \varepsilon, \tau_i, T_i \]

Convergence tolerance of the method.

Index of scheduling periods, \( t = 1, 2, \ldots, T \)

Commitment and dispatch variables

Non-negative and dispatch variables

Initial step-size, and step-sizes for updating \( \Delta_{i} \).

The mth net nodal load at period t (MW).

Thermal limit of transmission line \( l \) (MW).

The graph with NSTD of unit \( i \).

Minimum and maximum generation capacities (MW).

Scaling factors for the loads, minimum up/down time, thermal limits, and ramp limits.

Commitment decisions associated with the edge \( e \) and the path \( c \) in NSTD.

A constant threshold for determining \( \Delta_{i} \).

Power transfer distribution factors (PTDF) related with the ith unit and the mth load, respectively.

Ramp up/down limits of unit \( i \) (MW).

Coefficient matrices of the minimum up/down time constraints.

Digital Object Identifier 10.1109/TPWRS.2023.3287192

0885-8950 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
\( \delta \)

Thresholds for optimizing the initial step size \( c_0 \).

\( u \)

Upper bound of the initial step size \( c_0 \).

**D. Functions**

\( q(\cdot), \hat{q}(\cdot) \)

Lagrangian dual function of the UC problem, and an approximate Lagrangian dual function.

\( g_{r_k}^{\ast} \)

The approximate sub-derivative of \( q(\lambda) \) at \( \lambda_{rk}^0 \).

\( S(\cdot), C(\cdot) \)

Functions of start-up costs and fuel costs (\$).

\( \beta(\cdot) \)

Function for determining \( \hat{z}^\ast \) with \( \beta^0 \).

\( \Omega(\cdot) \)

Function defined in (18) and (19).

\( \Lambda(\cdot), N(\cdot) \)

Generation cost, and normalized generation cost.

**E. Abbreviations**

trial UC

A trial UC solution.

DPLR

The proposed method based on dynamic programming and Lagrangian relaxation.

NSTD

A new state transition diagram for the unit sub-problems of LR methods.

**I. INTRODUCTION**

UNIT commitment (UC) is one of the most critical problems focused on by the operators of power systems. The main objective of UC problems is to find a day-ahead scheduling plan for the on/off status of generating units. This scheduling plan achieves the minimum generation cost, and it guarantees the satisfaction of various system-wide and unit-wise constraints about generation, transmission, and system stability [1], [2], [3].

With the rapid expansion of power systems, the solving of UC problems becomes more and more difficult, and obtaining fast solutions for large-scale UC problems has become a pressing issue [4]. The main difficulty originates from that the mixed variables (binary and continuous) in the problem are heavily coupled by various constraints. In particular, the system-wide constraints couple the variables of all the generating units at each scheduling period, and the unit-wise constraints couple the variables of a unit in adjacent periods [4], [5]. Indeed, this problem is proven to be NP-hard [4], and for decades it has been drawing much attention from academies and the electric power industry.

There have been various approaches for solving the UC problem, mainly including the mix-integer linear programming (MILP) [6], [7], [8], [9], [10], [11], [12], [13], Lagrangian relaxation (LR) methods [14], [15], [16], [17], [18], dynamic programming [19], data-driven methods [20], [21], and heuristic methods [22]. These methods keep a proper balance between the efficiency and quality of the solutions, for UC problems of different sizes and characteristics. The heuristic methods, such as the genetic algorithm [22], are generally efficient in searching for locally optimum solutions, but their computation time is unpredictable; the data-driven method improve the accuracy and efficiency based on prior knowledge of historical data, which do not account for newly emerging system data [21]. To this end, this article aims at providing a fast solution of high quality for large-scale UC problems without resorting to prior knowledge.

The existing methods of this kind are mainly relevant to MILP methods [6], [7], [8], [9], [10], [11], [12], [13] and LR methods [14], [15], [16], [17], [18].

The MILP methods enumeratively search for the optimal UC solution in a tree structure of the binary variables with the branch-and-bound method, and they have nowadays been the researchers’ preferred solution method mainly due to the improved performance of MILP solvers. The improvements along with their variants make it easier for the researchers to formulate and solve MILP models with more modeling details. For example, the tight MILP formulations in [6], [7], [8] are said to be more computationally efficient since their strengthened linear programs provide improved lower bounds of the objective; some other MILP formulations finer reflect the operational constraints of the systems that are usually coarsely approximated [9]. In addition, the Benders decomposition method and its accelerated versions of MILP also considerably improve the efficiency [10], [11], [12], [13]. However, their computational complexity grows exponentially with the problem size, and it is reported that at times existing solvers cannot effectively solve the MILP problems within the time window for clearing the electricity market [14].

LR methods usually relax the coupling constraints and penalize violations of these constraints in the objective with Lagrange multipliers. This makes the problem separate into sub-problems that can be efficiently solved. For example, the system-wide constraints are relaxed in [23], which yields unit sub-problems for every unit. These sub-problems can be solved in polynomial time by some efficient dynamic programming methods [15], [16]. When the unit-wise constraints are further relaxed, the unit sub-problem can be further separated into unit-period sub-problems of different periods whose UC solution can be calculated with an analytical function in [14]. These UC solutions are found to be close to the optimal UC solution [14], [15], however, they are generally infeasible due to the relaxations. Thus, the solution and the associated multipliers are usually iteratively adjusted with sub-gradient methods to generate trial solutions converging to the optimal solution [15]. This approach usually has low computational complexity in practice, but it incurs the problems of zigzagging trial solutions [24]. And improvement of the convergence has perennially been an active research area. For example, the surrogate sub-gradient developed and utilized in [23], [25] improves the solution speed by not fully optimizing the relaxed problem; and in [26] the Lagrangian function is augmented with penalties of squared constraint violations to make the trial solutions converge faster. However, the convergence of large-scale problems is still slow and heuristics for recovering a feasible solution is wildly used [27].

It can be seen that MILP methods and LR-based methods focus on the following working directions.

- **Constraints reduction**

One of the main reasons for the difficulties in solving large scale security-constrained UC/ED is that there are too many constraints, for example, the transmission capacity/security related constraints. Considerable reduction of these constraints can significantly reduce the computational complexity of the problem. In [33], an analytical condition is proposed to fast...
identify redundant transmission constraints in the problem, which can eliminate over 80% of the transmission constraints. The method in [34] observes that there are many similar constraints in the SCUC problem, and it merges these constraints to further reduce the scale of the constraints. Constraints reduction is also widely used in the presolve process of most commercial solvers.

- Variables reduction

The existence of large-scale integer variables in the SCUC/TCUC is also one of the main reasons for the difficulties in solving the problem. This could easily render the problem intractable since the computational difficulty grows exponentially with the number of the variables, and thus reducing or fixing the number of the integer variables are used by many researchers to significantly reduce the computational complexity. In [14], an analytical function with respect to LR multipliers is proposed to fast calculate an approximate UC solution; and this analytical approximate UC is used to fix most of the UC variables in the original UC problem, resulting in a much smaller UC problem and significant improvements on solution speed. Similar idea is also found in [13], which fixes and unlocks the binary variables to obtain a feasible UC solution based on Benders’ cuts.

- Problem decomposition

Decomposing a large UC problem into smaller subproblems is wildly and perennially found in literature. One of the most well-known decomposition schemes is with Lagrangian relaxation, which generally relaxes the system-wide coupling constraints and obtains unit subproblems with reduced computational complexity [35]. A deeper relaxation of the hourly coupling constraints is found in [14], which gives rise to unit-period subproblems whose analytical UC solution can be obtained with an analytical function. As another decomposition scheme, the Benders’ decomposition usually separates the binary variables and the continuous variables into its master problem and its subproblem of smaller scale [10], which also draws much attention recently due to improvements on the Benders’ cuts [11].

- Exploiting the problem structures

Many researches also focus on exploiting the special structures of the UC problem. For examples, three set of binary variables in [36] are used to formulate the unit sub-problem in such a way that the formulation provides a convex hull for the minimum up/down time constraints, which achieves dramatic improvements on efficiency. The structures of the minimum up/down time constraints are exactly captured by a specially designed state transition diagram (the NSTD) [16], and the optimal UC can be obtained with $O(T^3)$ ($T$ is the number of scheduling periods) computational complexity.

And it is seen from the sub-bullets that both the NSTD in [16] and the analytical UC in [14] have the advantage of producing fast trial UC solutions; however, the disadvantage is that these trial UC solutions are generally infeasible to the relaxed constraints, and thus it is necessary to recover a feasible solution with MILP methods, which can be time-consuming. Nevertheless, we found that these two methods complement each other in their way of incorporating the constraints. Namely, the NSTD trial UC strictly satisfies the unit-level constraints and the analytical UC fully considers the system-wide constraints. This hints us the possibility that these two fast methods can be merged into a fast solution method that can directly produce fast feasible solutions, without the need for recovering a feasible solution. This is important for large-scale UC problems and is the main cause of the article.

In this article, a systematic method for producing a fast near-optimal UC solution is devised by combining the advantages of the two methods in [14] and [16], based on the framework of Lagrangian Relaxation. Its computational complexity grows polynominally in the problem size, and its relative duality gap is generally within 1%. It differs from the methods in [14], [16] in that it directly produces fast feasible near-optimal solutions without the need for recovering a feasible solution. In numerical tests, the method finds the solution after a few trial solutions have been explored, and the problems of a 6468-bus system can be solved within minutes. The main contributions of this article are summarized as follows.

1) A systematic framework that properly combines the advantages of the analytical function and the NSTD is proposed. In particular, the UC solution given by the analytical function is adjusted into a trial UC with the NSTD in such a way that this trial UC is likely to satisfy all the constraints.

2) A corrective method for making the trial UC solution strictly feasible is proposed. Specifically, a feasibility-testing problem is constructed in such a way that, it strictly checks the feasibility of the trial UC and yields corrective information for improving the feasibility. In addition, the initial multipliers and updating step size are also optimized to make the solution near-optimal.

The rest of this article is organized as follows. Section II describes the UC model and the fast analytical UC solution in [14]. Section III firstly shows the main idea of the method and then elaborates on it. This method consists of fast settings of its trial UC solution, an adjustment method for the trial UC solution, and a method for constructing a feasible near-optimal UC solution. Section IV presents numerical results. Section V concludes this article.

II. BASIC UC PROBLEM AND A FAST TRIAL UC SOLUTION

In this section, the basic formulation of the UC problem and the fast analytical UC solution based on LR are introduced for later analysis.

The transmission-constrained UC model is adopted in this article and is presented as follows.

$$\min_{z,t} \sum_i S_i(z_i) + \sum_{i,t} C_i(z_{i,t}, P_{i,t})$$  \hfill (1) \\
subject to: \quad \sum_t P_{i,t} - \sum_{m} d_{m,t} = 0, \quad \forall t; \hfill (2) \\
\sum_i \Gamma_{i,l}^{D} P_{i,t} - \sum_{m} \Gamma_{l,m}^{D} d_{m,t} \leq F_{l}, \quad \forall l, t; \hfill (3) \\
- \sum_i \Gamma_{i,l}^{U} P_{i,t} + \sum_{m} \Gamma_{l,m}^{U} d_{m,t} \leq F_{l}, \quad \forall l, t; \hfill (4) \\
\sum_{i} P_{i,t} \leq P_{i,t}^{\bar{P}}, \quad \forall i; \hfill (5)$$
\[ \Delta_t^i (z_{i,t}, z_{i,t-1}) \leq p_{i,t} - p_{i,t-1} \leq \Delta_t^i (z_{i,t}, z_{i,t-1}), \quad \forall i, t; \]  
\[ Uz \leq R, \quad z \in \{0, 1\}^{I \times T} \]  

The objective (1) is to minimize the total generation cost, including the start-up costs and the fuel costs. (2) is the power demand balance constraint. Inequalities (3)–(4) are the transmission capacity/security constraints based on the direct current (DC) power flow model. Constraints (5) and (6) represent the generation capacities and ramp up/down limits. (7) is the minimum up/down time constraints. According to [28], spinning reserve constraints can also be incorporated in (7). The system-wide constraints (2)–(4) and the unit-wise constraints (5)–(7) incorporate representative coupling relationships of the variables. Detailed information about this UC model could be referred to in [3].

For (1)–(7), the LR methods usually relax the system-wide constraints, and the corresponding relaxed problem is as (8), where the multipliers \( \lambda_{0,t}, \lambda_{0,t}^+, \lambda_{0,t}^-, \lambda_{l,t}^+, \lambda_{l,t}^- \) are associated with the constraints (2), (3), and (4), respectively. Then, the LR methods generate a series of trial solutions that converge to the optimal solution by alternately solving (8) and maximizing \( q(\lambda) \) over \( \lambda \).

However, the LR methods yield many computationally demanding unit sub-problems for large-scale UC problems [15]. This computational difficulty can be well addressed by further relaxing the unit-wise constraints (6)–(7) [14]. In particular, after the system-wide constraints are relaxed and the unit-wise constraints are ignored, the Lagrange dual function for (1)–(7) is as (9)–(10).

\[
\begin{align*}
q(\lambda) &= \min_{z,p} \sum_i S_i(z_i) + \sum_{i,t} C_t(z_{i,t}, p_{i,t}) + \\
&\quad \sum_t \lambda_{0,t} (\sum_i p_{i,t} - \sum_m d_{m,t}) + \\
&\quad \sum_t \lambda_{0,t}^+ (\sum_i p_{i,t} - \sum_m d_{m,t}) + \\
&\quad \sum_t \lambda_{0,t}^- (\sum_i p_{i,t} - \sum_m d_{m,t}) - F_t) + \\
&\quad \sum_t \lambda_{l,t}^+ (\sum_i p_{i,t} - \sum_m d_{m,t} - F_t) + \\
&\quad \sum_t \lambda_{l,t}^- (\sum_i p_{i,t} - \sum_m d_{m,t} - F_t) \\
&\text{s.t. } \Delta_t^i (z_{i,t}, z_{i,t-1}) \leq p_{i,t} - p_{i,t-1} \leq \Delta_t^i (z_{i,t}, z_{i,t-1}), \quad \forall i, t; \\
&\quad Uz \leq R, \quad z \in \{0, 1\}^{I \times T} \\
\end{align*}
\]

where the hat sign “\( \hat{\lambda} \)” is for denoting \( q(\lambda) \) as a relaxation of \( q(\lambda) \). Then, with the optimal solution to (9)–(10) (which will be given later), the multiplier vector \( \lambda \) (i.e., the dual variables) is to be optimized with the following dual problem.

\[
\begin{align*}
\max_{\lambda} q(\lambda) \\
\text{s.t. } \lambda_0, \lambda_l \geq 0, \forall l \\
\end{align*}
\]

At this time, the optimal solution pair \((\hat{z}, \hat{p}, \lambda^*)\) to (9)–(12) is to be easily obtained. Specifically, the problem (9)–(10) is separable both in periods and in units, and this problem can be separated into \( I \times T \) unit-period sub-problems. The unit-period sub-problem of unit \( i \) at period \( t \) is as follows.

\[
\begin{align*}
\hat{q}_{i,t}(\lambda_t) &= \min_{z_{i,t}, p_{i,t}} C_t(z_{i,t}, p_{i,t}) + \beta_{i,t}(\lambda_t) p_{i,t} \\
\text{s.t. } z_{i,t} P_{i,t} &\leq p_{i,t} \leq z_{i,t} \hat{P}_{i,t} z_{i,t} \in \{0, 1\} \\
\end{align*}
\]

where \( \beta_{i,t}(\lambda_t) = -\lambda_{0,t} + \sum_m (\lambda_{0,t}^m - \lambda_{l,t}^- m - F_i) \) (of which the constant part is omitted). With any given multiplier vector \( \lambda_t \), the associated optimal binary solution \( \hat{z}_{i,t}^*(\lambda_t) \) to (13)–(14) can be expressed as the following analytical function, and the dispatch solution \( \hat{p}_{i,t}^*(\lambda_t) \) of this problem can also be easily obtained [14].

\[
\begin{align*}
\hat{z}_{i,t}^*(\lambda_t) &= \begin{cases} 
1, & \text{if } \beta_{i,t}(\lambda_t) < \beta_t^0 \\
0, & \text{if } \beta_{i,t}(\lambda_t) \geq \beta_t^0 
\end{cases} \\
\end{align*}
\]

where the threshold \( \beta_t^0 \) for determining the binary solution \( \hat{z}_{i,t}^*(\lambda_t) \) is (16). It needs to be clarified that, (13) does not include the commitment cost since it becomes constant in the deduction of (15) [14].

\[
\beta_t^0 = \max \{-a_i - b_i/P_t, -a_i - b_i/P_t\} \\
\]

With the solution pair \((\hat{z}, \hat{p}, \lambda^*)\), the dual problem (11)–(12) becomes the following single-level optimization problem.

\[
\begin{align*}
\max_{\lambda} q(\lambda) : \lambda_0, \lambda_l \geq 0, \forall l, t \\
\end{align*}
\]

which is separable in periods and whose optimal solution \( \hat{\lambda}^* \) can be easily obtained with T linear programs [14]. Then, with this optimal multiplier vector \( \hat{\lambda}^* \), the analytical function (15) directly gives the optimal UC solution \( \hat{z}(\hat{\lambda}^*) \) to (9)–(10). This analytical UC solution fully considers the system-wide constraints; however, due to the relaxation, it is generally infeasible and needs adjustments.

### III. THE PROPOSED FAST SOLUTION METHOD FOR LARGE-SCALE UC PROBLEM

A concise explanation about the aim of the proposed method is that it focuses on producing a fast feasible near-optimal UC solution to the original UC problem. This is realized by efficiently adjusting the fast analytical UC solution \( \hat{z}(\hat{\lambda}^*) \) into a feasible near-optimal UC solution, based on the framework of Lagrangian relaxation. The fast analytical UC \( \hat{z}(\hat{\lambda}^*) \) is firstly adjusted into a trial UC with the NSTD in such a way that this trial UC is likely to satisfy all the constraints. Then, a feasibility-testing problem is used to efficiently check the feasibility of this trial UC and to provide corrective information for updating the multipliers, which will be used to re-calculate
a fast analytical UC with (15). By iterating the above process, the method produces a fast feasible near-optimal solution.

A. Detailed Steps of the Method

The method is further explained in the following steps (see Fig. 1).

Step 0. Initialize: Set $k = 0$, set the initial LR multiplier vector $\hat{\lambda}_0$ as the optimal solution $\hat{\lambda}^*$ to (17); go to step 1.

Step 1. Calculate analytical UC: With $\hat{\lambda}_k$, calculate the analytical UC solution (denoted as $\hat{z}(\hat{\lambda}_k)$) with (15). According to (9)–(10), this analytical UC fully considers the system-wide constraints, but it may be infeasible to the original UC problem and will be adjusted. Go to step 2.

Step 2. Adjust the analytical UC $\hat{z}(\hat{\lambda}_k)$: With NSTD, the analytical UC $\hat{z}(\hat{\lambda}_k)$ is adjusted to a trial UC solution (denoted as $\hat{z}(\hat{\lambda}_k)$) in such a way that this trial UC $\hat{z}(\hat{\lambda}_k)$ is feasible with the unit-wise constraints and has the minimum difference with the analytical UC. At this time, this trial UC $\hat{z}(\hat{\lambda}_k)$ is likely to be feasible to all the constraints. Go to step 3.

Step 3. Test and improve the feasibility of the trial UC: Firstly, set the binary variables in the original UC problem (1)–(7) as the trial UC $\hat{z}(\hat{\lambda}_k)$. This yields a linear program that may not be feasible. Secondly, add non-negative slack variables to this linear program to replace the possible violations of its system-wide constraints, and set its minimizing objective as the sum of the slack variables. To this end, this linear program must be feasible and serves as a feasibility-testing problem for the trial UC. Thirdly, if the slack variables (violations) are all minimized to zeros or their sum is less than a convergence tolerance, the trial UC is regarded as feasible and the iteration stops; otherwise, these minimized violations turn out to constitute an approximate sub-derivative $g_{\hat{\lambda}_k}$ of the Lagrangian dual function $q(\lambda)$ at the multiplier vector $\hat{\lambda}_k$. Go to step 4.

Step 4. Update the multipliers: Update $\hat{\lambda}_k$ to $\hat{\lambda}_{k+1}$ with the approximate sub-derivative $g_{\hat{\lambda}_k}$ and with a proper step size (to be optimized in (34)); set $k = k + 1$, and go to step 1.

Fig. 1. The basic idea of the proposed method.

B. Adjustment Method for the Fast Trial UC Solution

In Fig. 1, the analytical UC $\hat{z}(\hat{\lambda}_k)$ is already provided by (15); this subsection explains how to adjust the analytical UC to the trial UC. Namely, the trial UC is obtained with the NSTD in such a way that it strictly satisfies the unit-wise constraints and has the minimum difference from the analytical UC.

The NSTD serves as the basis for the adjustment. As seen in Fig. 2, this state transition diagram for each unit consists of hundreds of transition states scattered in multiple stages. Each of the states represents a scheduling period and is labeled with the corresponding number; each stage corresponds to a decision of turning the unit on (or off) for a time span (which is longer than the minimum up/down time), and each of the paths connecting the two states labeled with 0 and $T+1$ yields a UC decision chain for the unit during the horizon. For example, the path of the red dotted line segments in Fig. 2 corresponds to a UC decision chain that starts from turning the unit on at period 1; then, this “on” decision lasts for T-1 hours ($\geq \tau_i = 4$) until the period T-1 at the 3rd stage; then, the next “off” decision lasts until the period $T+1$ at the 4th stage (which is in the next scheduling horizon). It is pointed out that the on/off status of the unit determined by each of the paths satisfies the unit-wise constraints [16]. And since the analytical UC may violate the unit-wise constraints, there may be a difference between the on/off status of each path and the analytical UC. This difference is essential in adjusting the analytical UC to the trial UC.

The analytical UC is to be adjusted to the on/off status determined by a particular one of the paths; also, this on/off status has the minimum difference from the analytical UC, and is actually the trial UC. This is realized by applying dynamic programming on NSTD with the following procedures.

Firstly, to apply dynamic programming on NSTD, the transition cost associated with each of the edges is defined. In particular, the on/off status determined by any one of the edges may be different from the analytical UC of the unit at the corresponding period. This difference is defined as the transition cost of this edge, as follows.

$$\Omega_{t,c}(\hat{z}(\lambda), z^c) = \sum_{t_1}^{t_2-1} |\hat{z}_{t_1}(\lambda) - z_{t_1}^c|$$ (18)
where \( t_1, t_2 \) are endpoint periods associated with the edge \( e \), \( \tilde{z}^* (\lambda) \) is the analytical UC, and \( \{ z_{t_1}^e, t_1 \leq t \leq t_2 - 1, t \text{ integer} \} \) is the on/off status determined by this edge.

Secondly, the transition cost associated with any one of the paths \( e \) in \( G_t \) is defined as (19), which is the sum of the transition costs (18) of the edges therein.

\[
\Omega_{t,e}(\tilde{z}^*(\lambda), z^*) = \sum_{e} \Omega_{t,e}(\tilde{z}^*(\lambda), z^*), e = 1, 2, \ldots
\]  

(19)

where \( e \) is the index of the edges in the path \( e \). Then, the optimal path \( c^* \) with the minimum total cost \( \Omega_{t,e} \) is obtained by solving the following problem (20) with dynamic programming. It needs to be explained that the computational complexity of (20) is around \( O(T^3) \). In the case of the minimum up/down time being one for a unit, there will be \( T + 2 \) stages in Fig. 1, and the second stage has \( T + 1 \) states, which is greater than that of the other stages. Then, at any stage \( t \) in Fig. 1, (20) with backward recursion/induction needs to determine which state in stage \( t \) is the optimal succeeding state for every state in stage \( t \); this requires at most \( T + T \) additions and \( T + T \) comparisons, which amounts to an \( O(T^2) \) computational complexity for stage \( t \); thus, considering all the \( T + 1 \) stages, the total computational complexity is around \( O(T^3) \).

\[
c^* = \arg \min_{c \in C_t} \Omega_{t,c}(\tilde{z}^*(\lambda), z^*)
\]  

(20)

It is seen that, the on/off status determined by the optimal path \( c^* \) has the minimum difference with the analytical UC \( \tilde{z}^*(\lambda^*) \), and it is defined as the aforementioned trial UC \( \tilde{z}^*(\lambda^*) \). In this way, this trial UC fully considers the system-wide constraints besides strictly satisfying the unit-wise constraints, and it is also of high quality in terms of the fuel cost. In addition, the commitment cost omitted in (13)–(14) can be easily calculated for this trial UC at this time.

Although the trial UC is likely to satisfy all the constraints, it does not guarantee to satisfy the system-wide constraints. Therefore, further adjustments to the trial UC are needed.

C. Construction Method for Feasible UC Solution

If the trial UC is infeasible to the original UC problem, it can only have violations of the system-wide constraints. In this case, the trial UC is to be adjusted in this subsection to accommodate these violations. This is realized by adjusting the multipliers that determine the trial UC.

The feasibility of the trial UC can be efficiently examined with the feasibility-testing problem (21)–(27). This problem is a linear program derived by fixing the binary variables in the original UC problem as the trial UC. If the trial UC is feasible, in the feasible region of this linear program, there exists a dispatch solution point that has zero violations of all the constraints, including the system-wide ones; otherwise, there is no such point, and violations of the system-wide constraints are inevitable. In the latter case, the minimum total violation of the system-wide constraints is to be obtained with the feasibility-testing problem, as follows.

\[
\min_{p,v} \sum_{t} \sum_{l,t} (v_{0,t}^+ + v_{l,t}^+) + \sum_{t} (v_{0,t}^- + v_{l,t}^-)
\]  

(21)

subject to:

\[
\sum_{l,t} \Gamma^P_{l,t} p_{1,t} - \sum_{m} \Gamma^D_{m,t} d_{m,t} - v_{l,t}^+ \leq F_t, \forall l, t;
\]

\[
- \sum_{l,t} \Gamma^P_{l,t} p_{1,t} + \sum_{m} \Gamma^D_{m,t} d_{m,t} - v_{l,t}^- \leq F_t, \forall l, t;
\]

\[
\sum_{i} p_{t,i} - \sum_{m} d_{m,t} + v_{0,t}^+ - v_{0,t}^- = 0, \forall t;
\]

\[
\tilde{z}_{t,j}^e(\lambda^*_k) P_j \leq p_{t,j} \leq \bar{z}_{t,j}^e(\lambda^*_k) P_j, \forall i, t;
\]

\[
\Delta_t (\tilde{z}^*_t, \bar{z}^*_t, t - 1) \leq p_{t,j} - p_{t,j - 1} \leq \Delta_t (\tilde{z}^*_t, \bar{z}^*_t, t - 1), \forall i, t;
\]

\[
v_{0,t}^+ + v_{0,t}^- + v_{l,t}^- \geq 0, \forall l, t;
\]

where the nonnegative auxiliary vector variable \( v \) is used to represent the non-negative violations (residuals) of the system-wide constraints. The model (21)–(27) must be feasible if the original UC problem is feasible. Then, as in the objective (21), all the auxiliary variables are minimized to see if there exists a dispatch solution \( \tilde{p} (\lambda^*_k) \) under \( \tilde{z}^*(\lambda^*_k) \) that has zero total violation of the system-wide constraints. Once the optimal objective is zero, the solution pair \( (\tilde{z}^*(\lambda^*_k), \tilde{p}(\lambda^*_k)) \) of the above feasibility-testing problem must be a feasible near-optimal solution to the original UC problem.

If the optimal objective is greater than zero, the optimized violations (i.e., the optimal values of the slack variables) are to be used to adjust the multipliers and thus the analytical/trial UC, iteratively. According to LR multiplier theory, the non-zero optimal values of the auxiliary variables \( v^* \) turn out to constitute an approximate sub-derivative of \( q(\lambda) \) at the multiplier vector \( \lambda^*_k \), as follows.

\[
g^*_k = (v^*_{0,t} - v^{-}_{0,t}, v^+_{l,t}, v^-_{l,t})
\]  

(28)

where \( k \) is the index of the iterations for the adjustments. This approximate sub-derivative is essential in updating the multiplier vector \( \lambda^*_k \) to a new multiplier vector \( \lambda^*_k + 1 \), with the sub-gradient method as (29),

\[
\lambda^*_k = \frac{1}{c_k} \lambda^*_k + c_k g^*_k
\]  

(29)

where \( c_k \) is a step size specified as \( c_0/k > 0 \) (\( c_0 \) is configured with the model (34) in the appendix). Compared with \( \lambda^*_k \), \( \lambda^*_k + 1 \) derives a UC solution achieving a better objective of the dual problem (8). Namely, this UC solution has a decreased violation of the system-wide constraints, with respect to the generation cost. This also holds for the analytical UC \( \tilde{z}(\lambda^*_k) \) and the trial UC \( \tilde{z}^*(\lambda^*_k + 1) \) calculated with the multipliers \( \lambda^*_k + 1 \), (15), and (20). In other words, besides attaining high quality in terms of the generation cost, the trial UC \( \tilde{z}(\lambda^*_k + 1) \) has improved feasibility compared with that of the previous trial UC \( \tilde{z}^*(\lambda^*_k) \).

A feasible trial UC is obtained with iterations of the above adjustments. Under the framework of sub-gradient methods, multiple iterations of the adjustments may be required to derive a feasible trial UC. Specifically, in the possible case that the trial UC \( \tilde{z}(\lambda^*_k + 1) \) is infeasible, the feasibility-testing problem will again derive an approximate sub-derivative \( g^*_k \). This approximate sub-derivative \( g^*_k \) is then used to update \( \lambda^*_k + 1 \) to
\( \lambda^*_{k+2} \) as the way of updating \( \tilde{\lambda}^*_{k} \) to \( \tilde{\lambda}^*_{k+1} \). With (15), (18)–(20), this newest multiplier vector \( \tilde{\lambda}^*_{k+2} \) will again yield a trial UC \( \tilde{z}(\tilde{\lambda}^*_{k+2}) \) with further improved feasibility. The above process illustrated in Fig. 1 repeats until a feasible trial UC appears, or the following total violation of the system-wide constraints is less than a specified convergence tolerance \( \varepsilon \).

\[
V = \sum_{l,t} (v^+_{l,t} + v^-_{l,t}) + \sum_{i} \left( v^+_{i,0,t} + v^-_{i,1,t} \right) \leq \varepsilon \quad (30)
\]

It is verified in numerical tests that after a few iterations, the solution \( (\tilde{z}^*, \tilde{p}^*) \) from (21)–(27) satisfies all the constraints in the original UC problem, or (30) strictly satisfies. From now on, the proposed fast solution method based on DP and LR is denoted as DPLR for convenience.

### IV. NUMERICAL TESTS

In this section, an overview of the performance of the method is described first. Then, the solution time and the solution quality are tested under normal constraints. Finally, DPLR is tested under varied objective costs and strengthened constraints. These numerical tests are implemented with Matlab 2018b and the solver Gurobi 9.5.0 running on a desktop with an Intel 3.3 GHz CPU and 32 GB RAM.

#### A. Basic Information and Performance Summary

The method is tested on cases of the IEEE 24-bus system, 118-bus system, a modified Polish 2383-bus system, and a modified French 6468-bus system, with variations of the system parameters (i.e., objective costs, load levels, ramp limits, minimum up/down time, and transmission limits). In addition, redundant constraints of the cases are eliminated to reduce the computational complexity, without changing the optimal solution [33]. Details of the tested cases are available in [29], [30], [31], [32]. Basic information about these cases and the parameters of DPLR are listed in Tables I and II, respectively. In Table II, the column “\( N \)” shows the ratio of the ramp limit to the corresponding maximum generation capacity of the unit; the column “\( P \)” shows the maximum of the minimum up/down times of the unit; \( \varepsilon \) is the specified convergence tolerance of DPLR; \( \delta \) and \( \mu \) are given quantities for optimizing the initial step size in the appendix. The performance is also compared with three methods: the Benders’ Decomposition method (Benders),

### TABLE I

| Sys.        | N   | L   | \( T \) | \( \sum P_i(MW) \) | \( \sum d_i(MW) \) |
|-------------|-----|-----|---------|---------------------|---------------------|
| 24-bus      | 32  | 38  | 24      | 3405                | 2850                |
| 118-bus     | 54  | 186 | 24      | 13373               | 6000                |
| 2383-bus    | 327 | 2986| 24      | 29834               | 24558               |
| 6468-bus    | 400 | 9000| 24      | 115596              | 52555               |

### TABLE II

| Parameters of the Cases | \( d_i(MW) \) | \( \delta / P_i \) | \( F(MW) \) | \( \varepsilon \) | \( \delta \) | \( \mu \) |
|-------------------------|------------|-----------------|------------|---------------|---------|------|
| \( 100\% \text{Peak Load} \) | \( \leq 0.6 \) | \( 100\% \) | \( \leq 6 \) | \( 10^{-4} \) | \( 0.5 \) | \( 1 \) |

the Augmented Lagrangian Relaxation method (with surrogate sub-gradient) (ALRS), and Gurobi. The Benders method is implemented based on reference [10], with an accelerating technique [11]. The ALRS is implemented according to reference [18], which also includes heuristics to recover a feasible solution based on its dual solution.

With the convergence tolerance \( \varepsilon \) in (30) set as \( 10^{-4} \), the overall solution time of DPLR is on average less than one-tenth of the time consumed by the Benders and the solver, and it is also far less than that of the ALRS. In particular, 1) it is found that the computation time of DPLR for solving problems of the Polish 2383-bus case is within several minutes; 2) DPLR only requires a few iterations to provide a feasible near-optimal solution even if the test cases are with very tight constraints; 3) the solution of DPLR achieves competitive solution quality and relative gaps compared with that of the solver.

#### B. Analysis of the Computational Efficiency

In this subsection, the computational efficiency is tested by examining the iteration number and the solution time of DPLR and by comparing these results with that of the three methods.

As in Table III, the time consumed by the methods is in the column “Time”, in which the “–” sign means that the corresponding method runs out the RAM. The solution costs of the methods are in the column “Solution cost”. The sub-columns “Var” and “Constr” in the columns “OUC” and “FT” list the number of the variables and the constraints in the original UC problem and the feasibility-testing problem, respectively.

It is seen in Table III that: 1) DPLR is much faster than the three methods, and its solution cost is comparable with that of the solver. 2) the feasibility-testing problem has much fewer variables compared to the original UC problem, which practically explains why DPLR is fast. This indicates that DPLR can be used to solve the problem of even larger systems. These advantages of DPLR can be attributed to the fact that all its computations (related with the analytical UC, the NSTD, and the feasibility-testing problem) are simple operations with low computational complexity.

Then, the number of iterations that DPLR requires to solve the problem is obtained under different settings of the initial step size. This is because the initial step size of sub-gradient methods has a significant influence on the number of iterations, and it usually needs to be tuned. In view of this, we have properly optimized the initial step size as \( c_0^+ \) with the problem (34) (see details in the appendix), which is adaptive to changes of case data. To test both the number of the required iterations and the effectiveness of the optimized initial step size \( c_0^+ \), 30 exponentially increasing initial step sizes are generated within the range of \( [0, c_0^+] \cup (c_0^+, 1], \) and the number of the total iterations of DPLR at each generated step size is obtained. As in Fig. 3, the horizontal axis labels the logarithm of the step size, and the vertical axis shows the number of iterations; arrows are added to indicate the optimized step size \( c_0^+ \) of each system. The DPLR is terminated if the number of the iterations exceeds 20; and the markers in the legend of Fig. 3 label the step sizes at which a feasible DPLR solution is obtained within 20 iterations.
TABLE III
PERFORMANCE OF THE METHODS WITH THE SYSTEMS

| Sys.    | OUC* | FT* | Time (sec.) | Performance | Solution cost (10^4$) |
|---------|------|-----|-------------|-------------|-----------------------|
|         | Var  | Constr | Var  | Constr | DPLR | ALRS | Gurobi | DPLR | ALRS | Gurobi | DPLR | ALRS | Gurobi |
| 24      | 1536 | 6456 | 577  | 1660  | 1.3  | 232.7 | 0.8  | 772.0 | 4.242967 | 4.272000 | 4.22076 | 4.23965 |
| 118     | 2592 | 16728 | 910  | 1892  | 2.6  | 684.5 | 1.0  | 2941.8 | 27.90351 | 28.17194 | 27.82404 | 27.95862 |
| 2383    | 15696 | 186120 | 2062 | 14619 | 23.9 | 1192.1 | 2463.5 | 7750.6 | 492.4875 | 495.2316 | 489.2914 | 491.4816 |
| 6488    | 19200 | 486120 | 2552 | 21082 | 67.1 | 1619.3 | --   | --   | --   | --   | --   | --   |

a OUC stands for the “original UC problem”, and the number of the variables and the constraints in OUC are listed in the subcolumns “Var” and “Constr”.

b FT stands for the “feasibility-testing problem”, and the number of the variables and the constraints in FT are listed in the subcolumns “Var” and “Constr”.

C. Analysis of the Solution Quality

In this subsection, solution quality in terms of the generation cost and relative gap is tested under variations of the objective cost data and system load, in comparison with the solution quality of the three methods.

The generation cost of a UC solution is accordingly defined as follows.

\[ \Lambda_u(z, p) = \sum_i S_i(z_i) + \sum_{i,t} C_i(z_{i,t}, p_{i,t}) \] (31)

And for convenience, we also define the following normalized generation cost of a DPLR solution.

\[ N_c = \Lambda_c(z^*, p^*) / \Lambda_c(z^*, p^*) \] (32)

where \((z^*, p^*)\) is a DPLR solution and \((z^*, p^*)\) is a solution of the solver.

For the 2383-bus case with the parameters in Table II and the logarithmic initial step size of DPLR being \(-13.3\), Fig. 4 shows the normalized generation cost of DPLR solution under 30 load levels (actual load / peak load) evenly sampled in the range of 0.6~1.0. It shows that DPLR solution cost is basically comparable with that of the solver. To further examine the solution quality, relative MIP/duality gaps of DPLR and the three methods are compared with the same 2383-bus case under the sampled load levels (see Fig. 5). It is seen from the comparison in Fig. 5 that the relative gaps of DPLR is totally acceptable. This indicates that the DPLR solution is indeed of high quality and near-optimal in terms of generation costs. Tests of other systems show similar results.

D. Stability of the Proposed Method

This subsection shows that DPLR has a stable performance as above even with varied objective cost data and varied tightness
of the constraints, such as heavier load level, lower transmission limits, smaller ramp rates, and longer minimum up/down time.

We first test DPLR under varied objective costs. The original and varied objective cost data for the 24-bus system and the 2383-bus system are shown in Figs. 6 and 7(a), where the vertical axis shows the coefficients of the affine dispatch costs for all the units, and the horizontal axis is the index of the units which are sorted in an ascending order of the coefficients. The objective cost data is varied in such a way that, the units are first divided into many groups and units of each group are then set to have the same dispatch and commitment cost (see Fig. 6). Under this setting, tests show that DPLR still produces fast feasible solutions for all the systems under the varied objective cost data. This can also be verified in Fig. 7(b), which shows that, with the varied objective cost data, DPLR still produces fast feasible near-optimal solutions for the 2383-bus case under the 30 sampled load levels, and the relative gaps of these solutions are very small.

Then, DPLR is tested under varied tightness of the constraints. The tightness of the load levels, transmission limits, ramp limits, and minimum up/down time are measured by the following four scaling factors.

\[
s_d = \frac{d^\uparrow}{\bar{d}}, s_M = \frac{\bar{\tau}}{\bar{d}}, s_F = \frac{\bar{F}^\downarrow}{\bar{F}}, s_R = \Delta^{\downarrow}/\Delta^{\pm} \quad (33)
\]

where \(d^\uparrow\) and \(\bar{\tau}\) represent the uniformly increased loads and minimum up/down time, and \(\bar{F}^\downarrow\), \(\Delta^{\downarrow}\) represent the uniformly increased transmission limits and ramp up/down limits.

With each kind of constraint becoming tighter, there will be an extreme value of the corresponding scaling factor at which the DPLR (or the solver) will fail to generate a feasible solution since the problem is becoming infeasible; such extreme values of the scaling factors are denoted as \(s_d\), \(s_M\), \(s_F\), \(s_R\), respectively. As in Table IV, the values before the slash sign are the extreme scaling factors of the solver, the remaining values are the extreme scaling factors of DPLR. Notice that there is no value for the solver with the 6468-bus cases since under which the RAM runs out, and the value in the column \(s_M\) is truncated if the corresponding scaled minimum up/down time exceeds the length of the scheduling horizon.

It is seen from Table IV that, 1) the extreme values of the solver are slightly “tighter” than that of DPLR. This means DPLR cannot handle cases where the feasible region is extremely small; 2) however, tests show that such extreme cases also make the solver consume excessive computing time and resources, and these cases are rare in practice.

Then, after scaling the transmission limits (other constraints remain normal) of each system with the corresponding extreme value \(s_F\) in Table IV, there are four UC problems for the systems which have tightened transmission constraints. These problems are then solved by DPLR, and the associated total number of iterations and the normalized generation cost \(N_c\) of DPLR are listed in the columns “k∗” and “Nc” in Table V. Table VI shows the results of DPLR after scaling the ramp limits with \(s_R\), and Table VII shows the corresponding results of DPLR after scaling the minimum up/down time with \(s_M\). In Tables V, VI, and VII, the second column lists the corresponding extreme scaling factors \(s_M\), \(s_F\), and \(s_R\); the “Y” in the “solved?” columns means that feasible near-optimal solutions are obtained within
where the entries of the constant vector $\delta$ are given scalar thresholds, the constant $g_{0,1}^c$ is given by (28), and the constant $u$ is a given upper bound of the variable $c_0$ representing the initial step size, and the variables $\hat{\lambda}_i^*$ are the updated multipliers at the first iteration expressed as a function of the variable $c_0$ in the first constraint. The second condition is for keeping the binary solutions $z_{i,t}^*$ with bigger $|\beta_{i,t}^c(\hat{\lambda}_i^*) - \beta_{i,t}^0|$ unchanged in the first update of trial UC, and $\delta$ can control the ratio of the unchanged binary solutions to all the binary solutions; if the $|\beta_{i,t}^c(\hat{\lambda}_i^*) - \beta_{i,t}^0|$ of binary solutions $z_{i,t}^*$ are smaller than $\delta_i$, $z_{i,t}^*$ are allowed to be adjusted. The objective maximizes the initial step size to find the bound $[0, c_0]$ of the variable $c_0$ that satisfies the constraints in (34). This problem is feasible since $(0, \hat{\lambda}_i^*)$ is feasible. It is found in numerical tests that, with the multipliers calculated with (29) and (34), the method generally requires a few iterations to give the feasible near-optimal solution $(\hat{z}^*, \hat{p}^*)$, which is of high quality in terms of the generation cost.

V. CONCLUSION

Unit commitment problems for large-scale power systems are difficult to solve mainly due to their non-convex and non-differentiable nature. This difficulty in LR-based methods lies in the feasibility issues and the computational problems of their trial solutions. The computational difficulty can be greatly alleviated by the analytical function that gives a fast analytical UC solution under given multipliers. With this analytical function, a fast solution method for large-scale UC problems is established in this article based on Lagrange relaxation and dynamic programming. The method searches for a near-optimal UC solution with the NSTD rather than the tree structure used in MILP methods; and by constructing proper directions for updating the trial solutions, the method usually requires only a few iterations to provide the solution.

The numerical results show that DPLR is on average ten times faster than the commercial solver in solving large-scale UC problems, and DPLR solution is of high quality in terms of the generation cost. The generalization of DPLR for solving UC problems with uncertainty is in progress. DPLR is inherently generating a fast solution to large-scale UC problems under a single scenario of loads. In the future, the feasibility of DPLR solution under many of the scenarios or even all the scenarios of the uncertain loads will be assured.

APPENDIX

For sub-gradient methods, the initial step size usually needs to be tuned for a tested system, and this tuned step size may not be applicable to other systems. This results in re-tuning of the initial step size once the system is different. To find a proper initial step size without experimentally tuning it, an ideal approach is to start DPLR from a good-enough trial solution; in addition, the updated trial solutions should not deviate far from the initial trial solution. For DPLR, the good-enough trial solution is exactly the initial trial UC (i.e., the analytical UC solution to (9)–(10)), all we need to do at this time is to make the updated trial solutions close to it. This can be realized by optimizing the initial step size $c_0$ with the following model (which can be easily transformed into a linear program).

\[
\begin{align*}
\max &\quad c_0 \\
\text{s.t.} &\quad \hat{\lambda}_i^* = \lambda^* + c_0 g_{i,1}^c, \\
&\quad |\beta_{i,t}^c(\hat{\lambda}_i^*) - \beta_{i,t}^0| \geq \delta_i, \quad \text{if} \quad |\beta_{i,t}^c(\lambda^*) - \beta_{i,t}^0| \geq \delta_i, \forall i, t; \\
&\quad 0 \leq c_0 \leq u, \quad \hat{\lambda}_i^*, \hat{\lambda}_{i,t}, \hat{\lambda}_{i,t,1,1} \geq 0, \forall i, t.
\end{align*}
\]

(34)

\[
\begin{array}{c|ccc|c}
\text{System} & \tau_w & \text{Solved?} & k_s & \text{Ne} & \text{CPU time (sec)} \\
\hline
24\text{bus} & 1.3 & \text{N} & 20 & 1.01 & -- \\
118\text{bus} & 4.3 & \text{Y} & 5 & 1.02 & 6.9 \\
2383\text{bus} & 5.0 & \text{Y} & 8 & 1.008 & 66.1 \\
6468\text{bus} & 3.0 & \text{Y} & 2 & 1.005 & 105.6 \\
\end{array}
\]
[16] Q. Zhai, X. Guan, and F. Gao, “Optimization based production planning with hybrid dynamics and constraints,” IEEE Trans. Autom. Control, vol. 55, no. 12, pp. 2778–2792, Dec. 2010.

[17] P. Ramanan, M. Yildirim, E. Chow, and N. Gebraeel, “An asynchronous, decentralized solution framework for the large scale unit commitment problem,” IEEE Trans. Power Syst., vol. 34, no. 5, pp. 3677–3686, Sep. 2019.

[18] X. Sun, P. B. Luh, M. A. Bragin, Y. Chen, J. Wan, and F. Wang, “A novel decomposition and coordination approach for large day-ahead unit commitment with combined cycle units,” IEEE Trans. Power Syst., vol. 33, no. 5, pp. 5297–5308, Sep. 2018.

[19] H. Xiong, Y. Shi, Z. Chen, C. Guo, and Y. Ding, “Multi-stage robust dynamic unit commitment based on pre-extended-fast robust dual dynamic programming,” IEEE Trans. Power Syst., vol. 38, no. 3, pp. 2411–2422, May 2023.

[20] C. Ning and F. You, “Data-driven adaptive robust unit commitment under wind power uncertainty: A Bayesian nonparametric approach,” IEEE Trans. Power Syst., vol. 34, no. 3, pp. 2409–2418, May 2019.

[21] N. Yang et al., “Intelligent data-driven decision-making method for dynamic multisequence: An E-Seq2Seq-based SCUC expert system,” IEEE Trans. Ind. Informat., vol. 18, no. 5, pp. 3126–3137, May 2022.

[22] R. Ponciroli, N. E. Staff., J. Ramsey, F. Ganda, and R. B. Vilim, “An improved genetic algorithm approach to the unit commitment/economic dispatch problem,” IEEE Trans. Power Syst., vol. 35, no. 5, pp. 4005–4013, Sep. 2020.

[23] F. Feng, P. Zhang, M. A. Bragin, and Y. Zhou, “Novel resolution of unit commitment problems through quantum surrogate Lagrangian relaxation,” IEEE Trans. Power Syst., vol. 38, no. 3, pp. 2460–2471, May 2023.

[24] N. Z. Shor, “The subgradient method,” in Minimization Methods for Non-differentiable Functions, 1st ed. New York, NY, USA: Springer, 1985, pp. 22–47.

[25] M. A. Bragin, P. B. Luh, J. H. Yan, N. Yu, and G. A. Stern, “Convergence of the surrogate Lagrangian relaxation method,” J. Optim. Theory Appl., vol. 164, pp. 173–201, Apr. 2015.

[26] C. Beltran and F. J. Heredia, “Unit commitment by augmented Lagrangian relaxation: Testing two decomposition approaches,” J. Optim. Theory Appl., vol. 112, pp. 295–314, Feb. 2002.

[27] T. Seki, N. Yamashita, and K. Kawamoto, “New local search methods for improving the Lagrangian-relaxation-based unit commitment solution,” IEEE Trans. Power Syst., vol. 25, no. 1, pp. 272–283, Feb. 2010.

[28] Q. Zhai, J. Tian, and Y. Mao, “Spinning reserve width of generator unit: Concept, properties, and applications, part II: Applications in UC with up and down spinning reserve requirements,” Proc. CSEE, vol. 36, no. 16, pp. 6599–6609, 2016.

[29] C. Grigg, “The IEEE reliability test system-1996: A report prepared by the reliability test system task force of the application of probability methods subcommittee,” IEEE Trans. Power Syst., vol. 14, no. 3, pp. 1010–1020, Aug. 1999.

[30] I. Pena, C. B. Martinez-Anido, and B. M. Hodge, “An extended IEEE 118-bus test system with high renewable penetration,” IEEE Trans. Power Syst., vol. 33, no. 1, pp. 281–289, Jan. 2018.

[31] Z. R. Daniel, C. E. Murillo-Sánchez, and R. J. Thomas, “MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education,” IEEE Trans. Power Syst., vol. 26, no. 1, pp. 12–19, Feb. 2011.

[32] C. Josz, S. Fliscounakis, J. Maeght, and P. Panciatici, “AC power flow data of the IEEE reliability test system-1993: a report prepared by the 118-bus test system working group,” IEEE Trans. Power Syst., vol. 25, no. 1, pp. 9–17, 1992.

[33] Y. Yang, X. Guan, and Q. Zhai, “Fast grid security assessment with N-K contingencies,” in Proc. IEEE Power Energy Soc. Gen. Meeting, Chicago, IL, USA, Jul. 2017, Art. no. 17543715.

[34] X. Guan, P. B. Luh, H. Yan, and J. A. Amalfi, “An optimization-based method for unit commitment,” Int. J. Elect. Power Energy Syst., vol. 14, no. 1, pp. 1946–1954, Nov. 2010.

[35] C. Gentile, G. Morales-España, and A. Ramos, “A tight MIP formulation of the unit commitment problem with start-up and shut-down constraints,” EURO J. Comput. Optim., vol. 5, no. 1/2, pp. 177–201, 2017.

Qiaozhu Zhai (Member, IEEE) received the B.S. and M.S. degrees in applied mathematics and the Ph.D. degree in systems engineering from Xi’an Jiaotong University, Xi’an, China, in 1993, 1996, and 2005, respectively. He is currently a Professor with Systems Engineering Institute, Xi’an Jiaotong University. His research interests include optimization of large-scale systems and integrated resource bidding and scheduling in the electric power market.

Yuzhou Zhou, (Member, IEEE) received the B.S. degree in automation and the Ph.D. degree in control science and engineering from Xi’an Jiaotong University, Xi’an, China, in 2015 and 2021, respectively. From 2019 to 2020, he was a Visiting Scholar with the Stevens Institute of Technology, Hoboken, NJ, USA. He is currently an Assistant Professor with the Systems Engineering Institute of Xi’an Jiaotong University. His research interests include the planing and scheduling of energy systems with renewable energy resources and energy storage systems.

Xiaohong Gao (Fellow, IEEE) received the B.S. and M.S. degrees in automatic control from Tsinghua University, Beijing, China, in 1982 and 1985, respectively, and the Ph.D. degree in electrical engineering from the University of Connecticut, Storrs, CT, USA, in 1993. From 1993 to 1995, he was a Senior Consulting Engineer with PG&E, San Francisco, CA, USA. From January 1999 to February 2000, he visited the Division of Engineering and Applied Science, Harvard University, Cambridge, MA, USA. Since 1995, he has been with the Systems Engineering Institute, Xi’an Jiaotong University, Xian, China, and was appointed as the Cheung Kong Professor of systems engineering in 1999, and the Dean of the School of Electronic and Information Engineering in 2008. Since 2001, he has been the Director of the Center for Intelligent and Networked Systems, Tsinghua University. From 2003 to 2008, he was the Head of the Department of Automation. Since 2018, he has been a Member of the Chinese Academy of Sciences. His research interests include optimization of power and energy systems, electric power markets, and cyber-physical systems, such as smart grid and sensor networks.