A Methodology for Building Generation Trajectories to Balance Continuous-time Load Profiles

Menghan Zhang, Yue Zhou, *Member, IEEE*, Mingxu Xiang, *Member, IEEE*, Juan Yu, *Senior Member, IEEE*, Wenyuan Li, *Life Fellow, IEEE*, and Zhifang Yang, *Senior Member, IEEE*

***Abstract* — In power systems, maintaining a balance between generation and load is crucial. Traditional discrete-time dispatch methods often fall short, as they do not account for continuous-time changes in the load profiles throughout the time span. This oversight can lead to inaccuracies in tracing load profiles and even cause ramping resource shortages. In this paper, we propose the idea of continuous-time generation trajectories as dispatch results, to align with continuous-time load profiles. To ensure the solvability of the continuous-time dispatch, we propose an iterative dispatch methodology, which includes two stages: trajectory construction and constraint verification. In the trajectory construction stage, we use a parametric programming model to divide the continuous-time load profiles into multiple segments. Subsequently, we build the generation trajectories for each segment using parametric solutions. In the constraint verification stage, we specifically check the continuous-time ramping constraints. This stage identifies the infeasible trajectories, which will be updated during the next iteration. We repeat this iterative process until each unit has a feasible continuous-time generation trajectory throughout the time span. The effectiveness of our methodology is demonstrated in an illustrative 5-bus system and an actual 661-bus system.**

***Index Terms*—Continuous-time dispatch, Continuous-time generation trajectories, Discrete-time dispatch, Parametric programming.**

I. Introduction

A. Research Motivation

Power systems achieve a balance between generation and load through commitment, dispatch, and control [1]. Ideally, continuous-time dispatch is required to match continuous-time load profiles, representing a perfect balance between generation and load. Obtaining continuous-time results (continuous-time generation trajectories) in the dispatch problem enables direct sampling and derivation of discrete-time results. Thus, continuous-time results serve as a more general and precise criterion for dispatch decisions, predicting outcomes under broader operational conditions.

However, traditional dispatch methods operate on a discrete-time horizon to ensure solvability. While discrete-time dispatch models are mathematically manageable, they bring numerous challenges. Discrete-time methods invariably result in intra-interval energy deviations in both centralized systems (in most US markets) and self-scheduling systems (in most European markets). These deviations indicate less efficient dispatch and cause deterministic frequency deviations in actual operation, significantly impacting system security and reliability. For example, an energy deviation of 1300 MW can cause a frequency deviation of 50 mHz in load-frequency control areas, whereas frequency deviations exceed 100 mHz around the turn of the hour [2][3]. Frequency deviations offer relevant insights into intra-interval energy deviations from an operational perspective. Indeed, temporary episodes of energy shortfalls and surpluses can also impact energy quality, load curtailment, and renewable consumption [4][5].

Notably, there is no direct method to measure the benefits of continuous-time results, since they have yet to be implemented in practice. Instead, we assess the deficiencies of discrete-time results through the lens of the tracing limitation and the ramping resource shortage. Discrete-time dispatches convert continuous-time load profiles into hourly demands (or other time intervals, such as 30, 15 or 5 minutes), using a linear spline approximation to estimate the continuous-time balance. This discrete-time conversion naturally leads to intra-interval energy deviations and further requires the imbalance costs [6]. The tracing limitation refers to the inability of generation outputs to match continuous-time load profiles throughout the time span. Moreover, the continuous-time ramping behaviors of units are not captured by the discrete-time model, which simplifies continuous-time ramping constraints into discrete-time ones. The ramping resource shortage describes a unit's failure to meet its continuous-time ramping constraints.

In market-based power systems, operators (system or market operators) are responsible for ensuring the procurement of energy and ancillary services to support real-time balance [7]. Discrete-time dispatch not only omits critical intra-interval load profiles (tracing limitation) but also leads to an overestimation of ramping capabilities (ramping resource shortage). There is increasing evidence that such issues significantly impact the balance [8]. These intra-interval energy deviations indirectly highlight the advantages of continuous-time dispatch.

These intra-interval energy deviations stem from the discrete-time model, and we aim to convert the discrete-time results into continuous-time ones. In this paper, we propose a methodology to build feasible continuous-time generation trajectories for all units and ensure adherence to the continuous-time ramping constraints of each unit. Continuous-time generation trajectories can more accurately respond to intra-interval changes in load profiles, thereby alleviating issues such as the tracing limitation and the ramping resource shortage. Although this methodology faces the challenge of predicting continuous-

This work was supported in part by the National Natural Science Foundation of China (52307082) and in part by the Young Elite Scientists Sponsorship Program by CAST (2023QNRC001). *Corresponding author: Zhifang Yang (email: zfyang@cqu.edu.cn).*

M. Zhang, M. Xiang, J. Yu, W. Li, and Z. Yang are with State Key Laboratory of Power Transmission Equipment Technology, College of Electrical Engineering, Chongqing University, Chongqing 400044, China.

Y. Zhou is with the School of Engineering, Cardiff University, Cardiff CF24 3AA, UK.

time load profiles, it represents a practical step towards achieving continuous-time balance.

B. Literature Review

In market-based power systems, operators aim to ensure the procurement of energy and ancillary services. However, as load variations and renewables increase, intra-interval imbalances, including temporary episodes of energy shortfalls and surpluses, occur frequently and require further management. These shortfalls and surpluses illustrate the failure of discrete-time dispatch to accurately trace continuous-time load profiles.

In practice, this tracing limitation affects the transition from day-ahead dispatch (such as hourly intervals) to real-time operations (such as 5-minute intervals). In real-time operations, the load profiles are divided into two components: the forecasted day-ahead load profiles and the deviations from these profiles to the actual real-time loads. Even perfect day-ahead forecasts can lead to energy deviations due to model errors, which arise from the inconsistent resolution of load profiles. This is because intra-interval load variations are always excluded in day-ahead dispatch. Additionally, day-ahead load profiles are always imperfect, with forecast errors due to uncertainties. As a result, current uncertainties stem from both model errors and forecast errors.

Recent research has developed two main streams within the discrete-time framework to address these issues.

1) Modeling Emerging Resources: These approaches focus on modeling emerging resources within the existing discrete-time framework. These resources introduce additional cost components and constraints, which require special adjustments to the dispatch model. For instance, probabilistic formulations have been instrumental in determining additional energy resources for the forecasted scenarios [9][10]. Additionally, specialized flexible response products [11] and ramping mileages [12] have been introduced as new types of ancillary services to address ramping issues. The integration of ramping polytope and cut generation techniques has improved ramping abilities by adding ramping constraints [13].

These emerging resources in the US market (as well as balancing services in the European market) are on standby in real-time to compensate for intra-interval energy deviations. They are poised to enhance the ability to trace load variations under corresponding conditions. However, these resources and corresponding constraints have not yet been verified against continuous-time load profiles. Moreover, they also complicate market structures. Such complexity raises concerns regarding the fair allocation of additional costs.

2) Enhancing Time Resolution: Another line of research focuses on high-time resolution formulations to more accurately trace continuous-time load profiles [14]. These formulations tighten dispatch intervals to better approximate continuous-time load profiles [15]. This enhancement is relatively easy to achieve because these methods do not fundamentally alter the mathematical formulation. However, the increased time resolution presents computational challenges. Therefore, some researchers have proposed adaptive formulations to balance computational efficiency and accuracy. Clustering methods have been used to develop time-adaptive formulations [16]. Furthermore, clustering results can achieve feasible dispatch outcomes at the original time resolution [17]. However, tracing load within denser intervals remains a concern, as the results are still inherently in discrete-time, not continuous-time.

Recent advancements have proposed a continuous-time model using spline functions for continuous-time dispatch [18] and pricing approaches [19]. Additionally, these advancements extend to stochastic models [20]. Due to necessary approximations for solvability, such as quadratic or cubic interpolation, some energy deviations still exist between the continuous-time generation trajectories and continuous-time load profiles. Nevertheless, these methods provide forward-looking and practical insights with acceptable accuracy loss.

To sum up, from a discrete-time perspective, uncertainties stem from both model errors and forecast errors. Currently, ancillary services or balancing services after the day-ahead market can handle these uncertainties. However, this approach can obscure the true sources of the uncertainties and dampen incentives to improve intra-interval forecasting capabilities.

The continuous-time model is a promising approach to correct model errors. From a continuous-time perspective, intra-interval changes can be directly balanced, eliminating the need for rebalancing caused by model errors. However, the current model still struggles to provide feasible continuous-time generation trajectories that accurately match the continuous-time load profiles, incurring some energy deviations.

C. Contributions

We propose a methodology to build continuous-time dispatch results (continuous-time generation trajectories) once the continuous-time load profiles are known. More specifically, we propose an iterative dispatch methodology to build feasible continuous-time generation trajectories for all units over a specified time span. These trajectories adhere to each unit's continuous-time ramping constraints and ensure alignment with continuous-time load profiles. Notably, this iterative dispatch methodology does not approximate or sample the known load profiles but instead directly builds the continuous-time generation trajectories to balance them.

The transition from discrete-time models to continuous-time formulations introduces inherent complexities. With the emergence of variational optimization, variables and constraints become infinite-dimensional vectors or matrices. Furthermore, continuous-time ramping constraints are transformed into differential constraints.

To ensure the solvability of continuous-time dispatch, we propose an iterative dispatch methodology that comprises two stages: trajectory construction and constraint verification. During the trajectory construction stage, a parametric programming model is employed to transform the continuous-time dispatch into a parametric programming problem. This model uses time as a parameter, converting decision variables and constraints from infinite-dimensional to finite-dimensional, and subsequently divides continuous-time load profiles into multiple segments. For these segments, generation trajectories are calculated through parametric solutions. In the constraint verification

stage, the differential constraints and the continuity of trajectories are externally checked. Generation trajectories that violate differential constraints are flagged and set for updates during the next trajectory construction stage. By using feasible trajectories, continuous-time generation trajectories are finally represented by piecewise affine functions of time.

The continuous-time generation trajectories effectively bridge the gap between day-ahead dispatch and real-time operations, facilitating smoother operational transitions. We present operational processes specifically designed for this application, with case studies that confirm the effectiveness.

II. Mathematical Formulation

To thoroughly explain the transition from the discrete-time model to the continuous-time formulation, this section begins by outlining the linear optimization in the discrete-time dispatch. Then, the discussion transitions to variational optimization related to the continuous-time formulation. Finally, the assumptions behind the model and the reasons for the simplifications are discussed.

A. Discrete-time Dispatch Model

In this paper, three foundational terms are introduced for precise articulations and analyses: 'period', 'interval', and 'segment'. The 'period' is the time span for which dispatch results are required. The 'interval' denotes the subdivisions of time within the overarching 'period'. The 'segment' refers to the continuous-time load profiles correlated with each 'interval'.

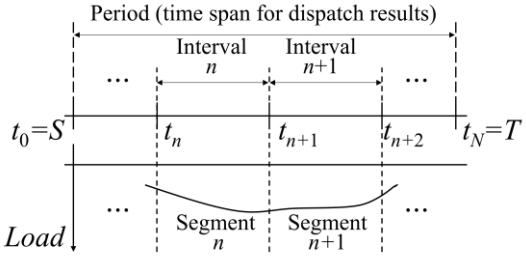


Fig. 1. Three foundational terms in this paper.

As shown in Fig. 1, the traditional discrete-time framework divides the period $\mathcal{T}=[S,T]$ into N intervals. $t_0=S$ represents the start of the period and $t_N=T$ represents the end. The intervals are represented as $\mathcal{T}_n=[t_n, t_{n+1}]$, $\mathcal{T}=\cup_{n=0}^{N-1}\mathcal{T}_n$, where n is the interval index. The intervals are uniform in length, denoted as $\Delta t=t_{n+1}-t_n$. The discrete-time load profiles, denoted as $D(t_n)$, are sampled from the continuous-time load profiles $D(t)$. The discrete-time load profiles $D(t_n)$ are the segment endpoints and the t_n values are the corresponding interval endpoints. The decision variables $\mathbf{G}(t_n)$ are optimized to satisfy the discrete-time load profiles $D(t_n)$. The discrete-time dispatch is presented below.

$$\min_{\mathbf{G}(t_n)} \sum_{n=0}^{N-1}\left(C\left(\mathbf{G}\left(t_n\right)\right)\right)\Delta t \tag{1}$$

$$s.t. \quad \sum\nolimits_K G_k\left(t_n\right) = D\left(t_n\right), \forall n \tag{2}$$

$$G_k^{\min} \le G_k\left(t_n\right) \le G_k^{\max}, \forall k, n \tag{3}$$

$$R_k^D \Delta t \le \left(G_k\left(t_{n+1}\right) - G_k\left(t_n\right)\right) \le R_k^U \Delta t, \forall k, n \tag{4}$$

where k is the unit index and K is the total number of units. The expression $C(\mathbf{G}(t_n))=\Sigma_K C_k(G_k(t_n))$ denotes the sum of the production costs for all units within interval n. The superscripts min and max represent the minimum and maximum production of unit k, respectively. R_k^D / R_k^U represent the downward/upward ramping capacities per unit of time for unit k, respectively.

The objective function (1) aims to minimize production costs for $\mathbf{G}(t_n)=(G_1(t_n),\ldots,G_K(t_n))^{\mathrm{Tr}}$, where the superscript Tr represents the transpose. Constraints (2) model the power balance at segment endpoints. Constraints (3) set bounds on the production of the units. Constraints (4) describe the ramping limitations of the units.

The disparity in time resolution is represented through the different interval lengths of Δt. To unify the expression associated with Δt, R_k^D and R_k^U within the constraints (4) are expressed per unit of time. Then, the constraints (4) incorporate the interval length of Δt as an additional component.

Typically, the discrete-time dispatch for all units is provided as the generation levels at interval endpoints. The continuous-time action of a single unit k can be understood as a generation trajectory, characterized by piecewise constant behavior. This concept is exhibited in technical literature, such as in source [21]. A rigorous interpretation of piecewise constant generation trajectory exists. It suggests that units should instantaneously transition to the subsequent generation levels at the start of each interval, as shown in Fig. 2(a). However, this generation trajectory is unrealistic for a physical unit.

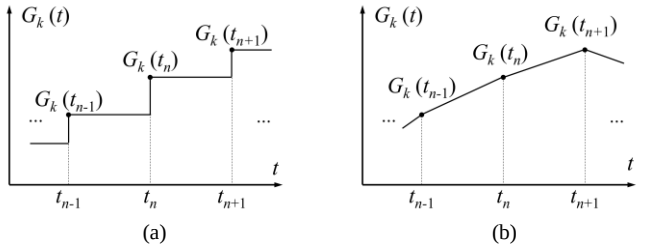


Fig. 2. Generation trajectory from discrete-time dispatch for a single unit k (a) constant generation trajectory, (b) linear generation trajectory.

In practice, operations are conducted to ensure a smooth transition along the linear generation trajectory. Fig. 2(b) illustrates the linear generation trajectory of unit k, defined by two interval endpoints. For example, $G_k(t_n)$ and $G_k(t_{n+1})$ determine the linear generation trajectory within $\mathcal{T}_n=[t_n, t_{n+1}]$. The unit k adheres to the linear generation trajectory, transitioning from one discrete-time generation level to the next one. For $\forall t \in \mathcal{T}_n$, the linear generation trajectory of the unit k is presented below.

$$G_k\left(t\right) = \frac{\left(G_k\left(t_{n+1}\right) - G_k\left(t_n\right)\right)}{\Delta t}\left(t - t_n\right) + G_k\left(t_n\right), \forall t \in \mathcal{T}_n \tag{5}$$

The linear generation trajectory of the unit k is a common treatment in the discrete-time framework. However, it serves as an approximation because the method in (5) utilizes the finite difference between two generation levels. By focusing on interval endpoints, it entirely neglects variations in continuous-time load profiles. Therefore, it causes a failure to accurately achieve the continuous-time balance.

B. Continuous-time Dispatch Model

Recognizing the limitations of discrete-time dispatch, we propose the idea of continuous-time generation trajectories as

dispatch results to offer a more accurate solution. The transition from the discrete-time dispatch to the continuous-time formulation causes essential changes. To start, increasing the time resolution of the discrete-time model (1)-(4) allows for broader coverage of continuous-time load profiles $D(t)$. As $\Delta t \to 0$, the derivative function of generation trajectory (of a single unit k) is finally defined in (6).

$$G_k^{'}(t) \triangleq \lim_{\Delta t \to 0} \frac{\left(G_k(t_{n+1}) - G_k(t_n)\right)}{\Delta t}, \forall t \in \mathcal{T}_n \tag{6}$$

When time resolution becomes infinitely fine (Δt=0, $N \to \infty$), the ramping constraints (4) are totally reformulated below.

$$R_k^D \le G_k^{'}(t) \le R_k^U \tag{7}$$

The continuous-time dispatch is formulated below.

$$\min_{\mathbf{G}(t)} \int_{\mathcal{T}} \left(C\left(\mathbf{G}(t)\right)\right) dt \tag{8}$$

$$s.t. \quad \sum\nolimits_K G_k(t) = D(t), \forall t \tag{9}$$

$$G_k^{\min} \le G_k(t) \le G_k^{\max}, \forall k, t \tag{10}$$

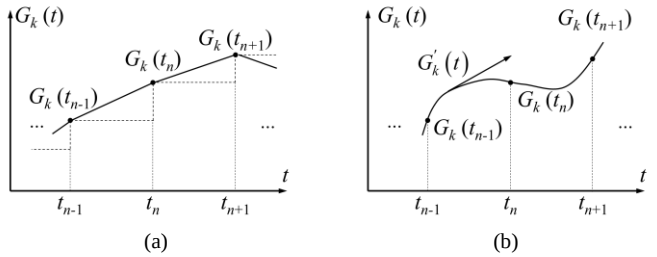
$$R_k^D \le G_k^{'}(t) \le R_k^U, \forall k, t \tag{11}$$

where $C(\mathbf{G}(t))=\Sigma_K C_k(G_k(t))$ represents the sum of production costs of all units at time t.

The objective function (8) aims to minimize the total continuous-time production costs for $\mathbf{G}(t)= (G_1(t),\ldots,G_K(t))^{\mathrm{Tr}}$ throughout the period $\mathcal{T}=[S,T]$. Since time t can take any value within the period $\mathcal{T}$, the decision variables and constraints are infinite-dimensional. Moreover, constraints (11) introduce the derivative functions of $\mathbf{G}(t)$. The model (8)-(11) provides feasible continuous-time generation trajectories, represented by $\mathbf{G}(t)$, to match the continuous-time load profiles $D(t)$.

The model (8)-(11) includes three types of infinite-dimensional constraints, and constraints (11) become differential constraints. These typical formulations mark the model's transformation into a variational optimization problem. The incorporation of $D(t)$ in the continuous-time model allows for the exact modeling of continuous-time ramping constraints, which cannot be achieved in the discrete-time dispatch model.

Fig. 3 compares the generation trajectory of a single unit k across the intervals. The generation levels at interval endpoints, such as $t=t_n$ and $t=t_{n+1}$, are set to align with the discrete-time results, ensuring a relevant comparison.



(a) (b)

Fig. 3. Different generation trajectory for a single unit k (a) discrete-time generation trajectory, (b) continuous-time generation trajectory.

Fig. 3(a) illustrates the linear generation trajectory of unit k from discrete-time dispatch, which cannot achieve a continuous-time balance. Conversely, the continuous-time generation trajectory $G_k(t)$, shown in Fig. 3(b), is explicitly expressed as a function of t. If the continuous-time load profiles $D(t)$ are nonlinear, the generation trajectory $G_k(t)$ should reflect these features. The derivative function $G_k^{'}(t)$, which satisfies differential constraints, is also verified directly. If all units follow the continuous-time generation trajectories $\mathbf{G}(t)$, the continuous-time load profiles $D(t)$ can be accurately traced while satisfying the continuous-time ramping constraints.

C. Assumptions and Simplifications

The continuous-time unit commitment is not within the scope of this paper. A foundational assumption is that the commitment results are fixed throughout the period. However, if the committed units are insufficient, this assumption may lead to infeasible dispatch results. To address this, we propose two output forms for continuous-time dispatch: continuous-time generation trajectories and recommitment instructions. If the initial commitment results are infeasible in the continuous-time dispatch, the recommitment instructions indicate this infeasibility and signal the need for additional units.

Another foundational assumption is that the continuous-time load profiles $D(t)$ are known in advance. Considering that there are differential constraints in the model (8)-(11), the derivative function of $D(t)$ must exist. For the methodology proposed later in the paper, the specific functional form of the continuous-time load profiles is not crucial. The requirements for the predetermined continuous-time load profiles are that they must be single-valued functions of time and have derivative functions. In this paper, we focus on model errors arising from discrete-time dispatch and aim to achieve a continuous-time balance. Therefore, the accurate acquisition of load profiles $D(t)$ is important, but it is not within the scope of this paper. Some discussions on continuous-time load forecasting can be found in Refs. [22] and [23].

In this paper, we apply simplifications to continuous-time dispatch to enhance clarity and focus. The startup, shutdown, and no-load costs, which are typically associated with commitment results, are omitted since the commitment results are predetermined in continuous-time dispatch. Additionally, specific constraints, such as transmission constraints, are also omitted. Our methodology simplifies the analysis, allowing us to focus on continuous-time ramping constraints and avoid interactions with more complex constraints.

III. The Outline of Iterative Dispatch Methodology

Currently, continuous-time generation trajectories can bridge the gap caused by the inconsistent resolution of load profiles between day-ahead dispatch and real-time operations. While day-ahead dispatch produces hourly results, real-time operations require more granular 5-minute intervals within a rolling 60-minute period. This difference may render real-time operations infeasible, even when real-time loads perfectly align with day-ahead forecasts. The reason is that intra-hourly load profiles are excluded in day-ahead dispatch, which we refer to as model errors in Section I.B. Moreover, any deviation between the actual real-time loads and day-ahead forecasts leads to forecast errors, further complicating real-time operations. Both types of errors can result in discrepancies between day-ahead dispatch and real-time operations, making it difficult to pinpoint

the exact sources of these errors.

Continuous-time generation trajectories can address this issue. They are adaptable to any time resolution and theoretically eliminate the approximation errors in the discrete-time model. More importantly, they can incorporate sub-hourly loads into day-ahead dispatch, which addresses the root cause of discrepancies. As a result, the transition from day-ahead dispatch to real-time operations is affected solely by the forecast errors associated with continuous-time load profiles.

Considering this application, we aim to build continuous-time generation trajectories for an hourly (60-minute) period $\mathcal{T}=[S,T]$. This period starts at S=0 (minute) and ends at T=60 (minute). The operational process is visually summarized in Fig. 4. Notably, continuous-time generation trajectories are scalable for a 24-hour cycle, provided that this process is replicated throughout all 24 hours.

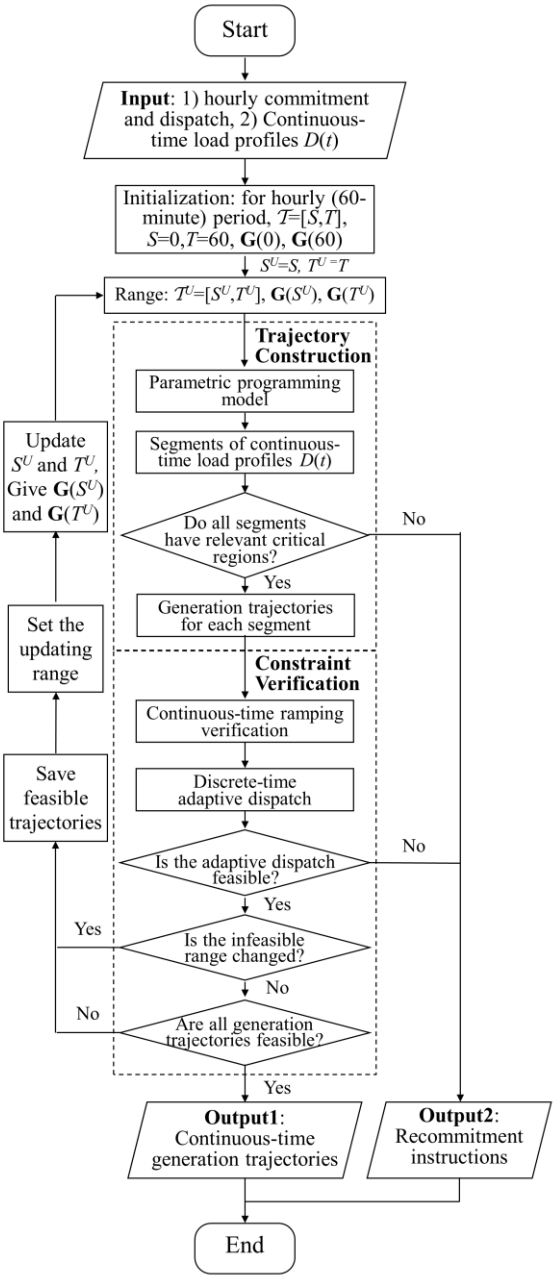


Fig. 4. Operational process of iterative dispatch methodology.

Before applying the iterative dispatch methodology, initial inputs are derived from the day-ahead market, which includes the hourly commitment and dispatch. According to Section II.C, the assumption is that the commitment results remain constant within hourly periods. These inputs act as adaptations for the actual process, ensuring that the output at the start and end intervals aligns with and protects the existing hourly results. The initialization sets $\mathbf{G}(0)$ and $\mathbf{G}(60)$ from hourly day-ahead commitment and dispatch. Additionally, we have prior knowledge of the continuous-time load profiles $D(t)$ within the hourly period $\mathcal{T}$ from the load forecasting. These load profiles are also inputs, and we aim to accurately balance them. Utilizing these inputs, the iterative dispatch methodology systematically outputs either continuous-time generation trajectories, or recommitment instructions.

After the initialization, the continuous-time dispatch model (8)-(11) can be formulated for the hourly period $\mathcal{T}$. However, this model is not readily solvable due to the infinite-dimensional decision space and the differential constraints. Instead, we propose the iterative dispatch methodology to build continuous-time generation trajectories for the hourly period $\mathcal{T}$, following the process in Fig. 4. Our methodology comprises a trajectory construction stage and a constraint verification stage.

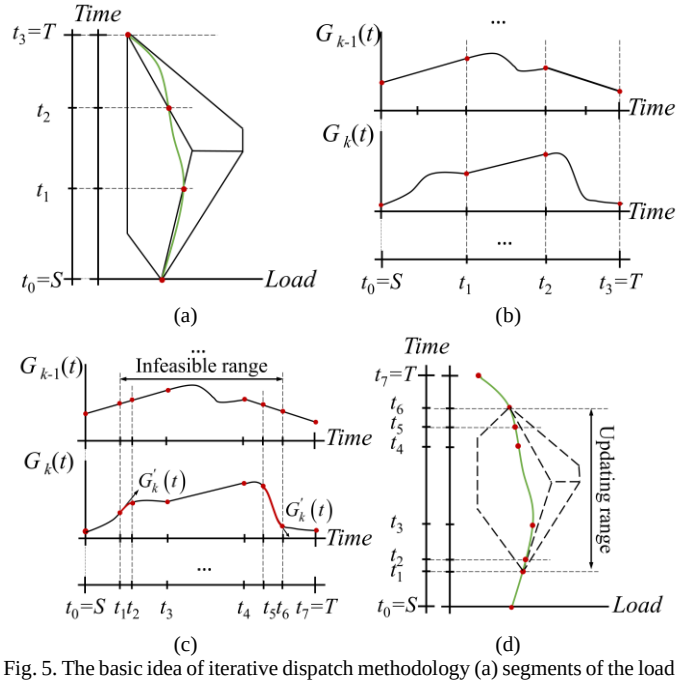


Fig. 5. The basic idea of iterative dispatch methodology (a) segments of the load profiles, (b) generation trajectories for units, (c) identification of the infeasible range, (d) the updating range and the next trajectory construction stage.

During the trajectory construction stage, we manage the infinite-dimensional problem by employing a parametric programming model. As shown in Fig. 5(a), we segment the continuous-time load profiles $D(t)$ by identifying critical regions (critical regions outlined with black lines), which are determined by changes in active constraints. These segments, informed by the model's inherent constraints rather than subjective decisions, result in variable-length intervals, as depicted by the red endpoints in Fig. 5(a). The piecewise affine functions of decision variables corresponding to these segments serve as generation trajectories, as shown in Fig. 5(b).

The subsequent constraint verification stage is tasked with verifying these generation trajectories against the continuous-time ramping constraints and ensuring their continuity. This stage is crucial because differential constraints are not modeled in the trajectory construction stage. Generation trajectories that do not adhere to continuous-time ramping constraints—like $G_k(t)$ during intervals $[t_1, t_2]$ and $[t_5, t_6]$ in Fig. 5(c)—are identified as ramping violations. By detecting these violations, we introduce new interval endpoints. We then utilize a discrete-time adaptive dispatch to assist in judging continuity and define the infeasible range, such as the range $[t_1, t_6]$ in Fig. 5(c).

For the next trajectory construction stage, the updating range, like the range $[t_1, t_6]$ in Fig. 5(d), is set to further rebuild generation trajectories. During the next iteration, the generation trajectories, like those within ranges $[t_0, t_1]$ and $[t_6, t_7]$, are verified as feasible trajectories and they remain unchanged. The dotted lines in Fig. 5(d) represent the identified critical regions in the next trajectory construction stage. Using these critical regions, we further rebuild the generation trajectories within the updating range.

The iterative dispatch methodology, illustrated in Fig. 5(a)-(d), progressively refines the continuous-time load profiles $D(t)$, rebuilds generation trajectories, and checks the differential constraints. The iterative process repeats until either feasible continuous-time generation trajectories are attained throughout the period or recommitment instructions are issued. Notably, each iterative process relies on the feasible trajectories determined during the previous constraint verification stage. Our methodology does not guarantee the optimality of the original continuous-time dispatch model (8)-(11). Furthermore, as shown in Fig. 4, some conditional operations determine the path of the process, which are detailed in the following sections.

IV. Trajectory Construction Stage

This stage recasts the infinite-dimensional problem into a finite-dimensional one. It offers a segmentation of continuous-time load profiles $D(t)$ based on changes in active constraints. Subsequently, this stage builds generation trajectories for each segment to balance the corresponding continuous-time load profiles $D(t)$.

A. Parametric Programming Model

In this paper, we introduce a parametric programming model that incorporates time t as a distinctive parameter, reflecting that decisions change over time. This model defines the constraints for all t in the range $\mathcal{T}^U=[S^U,T^U]$, where S^U and T^U denote the start and end of the $\mathcal{T}^U$, respectively. The superscript U represents the range during the trajectory construction stage. Load profiles D^P are also incorporated as an individual parameter, capturing the variations in continuous-time load profiles. The superscript P represents the parametric profiles in the parametric programming model. The parametric programming model is formulated below.

$$\min_{\mathbf{G}^P\left(t,D^P\right)} C\left(\mathbf{G}^P\left(t,D^P\right)\right) \tag{12}$$

$$s.t. \qquad \sum\nolimits_K G_k^P\left(t,D^P\right)=D^P \tag{13}$$

$$G_k^{\min} \le G_k^P\left(t,D^P\right) \le G_k^{\max}, \forall k \tag{14}$$

$$R_k^D(T^U-t) \le \left(G_k\left(T^U\right)-G_k^P\left(t,D^P\right)\right) \le R_k^U(T^U-t), \forall k \tag{15}$$

$$R_k^D(t-S^U) \le \left(G_k^P\left(t,D^P\right)-G_k\left(S^U\right)\right) \le R_k^U(t-S^U), \forall k \tag{16}$$

where decision variables $\mathbf{G}^P(t,D^P)$ depend on parameters t and D^P. The $G_k(S^U)$ and $G_k(T^U)$ are the given dispatch results of unit k at the start and end of $\mathcal{T}^U$, respectively.

The objective function (12) aims to minimize the production costs for $\mathbf{G}^P(t,D^P)=(\,G_1^P(t,D^P)\,,\ldots,\,G_K^P(t,D^P)\,)^{\mathrm{Tr}}$. Constraints (13) model the power balance for $\forall t\in\mathcal{T}^U$. Constraints (14) set bounds on the production of units. Constraints (15)-(16) describe the ramping limitations with two given dispatch results, forcing the $G_k^P(t,D^P)$ to align with given dispatch results $G_k(S^U)$ and $G_k(T^U)$ at $t=S^U$ and $t=T^U$.

The $\mathcal{T}^U$ is initialized as period $\mathcal{T}=[S,T]=[0,60]$ for the first iteration, as shown in Fig. 4. The given dispatch results, $\mathbf{G}(0)$ and $\mathbf{G}(60)$, are provided by the day-ahead hourly results. In subsequent iterations, $\mathcal{T}^U=[S^U,T^U]$ is adjusted to match the updating range. $\mathbf{G}(S^U)$ and $\mathbf{G}(T^U)$ are provided by the discrete-time adaptive model in the constraint verification stage.

By treating time t as a parameter, the infinite-dimensional problem is represented as a finite-dimensional one with parametric profiles. Given the inherent temporal nature of the continuous-time load profiles $D(t)$, it is more appropriate to represent $D(t)$ within the time t directly in the parametric programming model. However, the introduction of non-linear load profiles results in non-linear constraints. Therefore, we strategically treat the load profiles as an independent parameter. The objective is to maintain the linearity within the parametric problem's formulation.

Significantly, the model (12)-(16) does not include the continuous-time ramping constraints (11). This exclusion arises because the derivative of $G_k(t)$ challenges the model's linearity. This limitation requires us to check the continuous-time ramping constraints separately later.

The output of the model (12)-(16) includes some critical regions and a compact representation of the parametric solutions. These parametric solutions link the parameters t and D^P to decision variables $\mathbf{G}^P(t,D^P)$ through piecewise affine functions within the corresponding critical regions. These parametric solutions serve as the basis for subsequent analysis.

B. Segments of Continuous-time Load Profiles

We divide the continuous-time load profiles $D(t)$ into multiple segments based on the parametric solutions. By incorporating piecewise affine functions of $\mathbf{G}^P(t,D^P)$ into constraints (13)-(16), these constraints are divided into active and inactive constraints, which become parametric constraints.

$$\begin{cases} \mathbf{h}_J\left(t,D^P\right)=\mathbf{0} \\ \mathbf{h}_I\left(t,D^P\right)\le\mathbf{0} \end{cases} \tag{17}$$

where subscript J and I represent the active and inactive constraints, respectively. $\mathbf{h}_J$ and $\mathbf{h}_I$ represent the active and inactive constraints expressed by parameters, respectively.

This categorization facilitates the identification of constraints that influence the decision space. The decision variables

$\mathbf{G}^P(t,D^P)$ that are bounded by the same inactive constraints can be gathered into a single region, denoted as Ω_i. In order to distinguish each region and its affine functions, we introduce the subscript i as the region index. The region Ω_i forms a polyhedron defined by the parametric constraints.

$$\Omega_i \triangleq \left\{\left(t, D^P\right) \in \mathbb{R}^2 : \mathbf{h}_{I,i}\left(t, D^P\right) \le \mathbf{0}\right\} \tag{18}$$

Feasibility is ensured by incorporating $\mathbf{G}^P(t,D^P)$ into the inactive inequalities of Ω_i. Additionally, the Karush-Kuhn-Tucker conditions are applied through $\boldsymbol{\lambda}_i(t,D^P)$. Moreover, taking into account the parameter bounds, the critical region H_i is defined in (19).

$$H_i \triangleq \left\{\left(t, D^P\right) \in \mathbb{R}^2 : \mathbf{h}_{I,i}\left(t, D^P\right) \le \mathbf{0}, \boldsymbol{\lambda}_i\left(t, D^P\right) \ge \mathbf{0}, S^U \le t \le T^U\right\} \tag{19}$$

Once we determine all the critical regions throughout $\mathcal{T}^U$, we then introduce the continuous-time load profiles $D(t)$ for $\forall t \in \mathcal{T}^U$. Intersections of $D(t)$ with critical regions pinpoint segments of $D(t)$ within these critical regions. Each segment is marked by two interval endpoints: the incidence t_n and the exit t_{n+1}. For instance, Fig. 5(a) illustrates interval endpoints in red, dividing $D(t)$ into three segments between t_0 and t_3.

As we iterate, the updating range $\mathcal{T}^U=[S^U,T^U]$ is set from the results based on the previous constraint verification stage. The $\mathcal{T}^U$ is subdivided and its corresponding critical regions are also changed. Other intervals that remain unchanged retain their last parametric solutions and critical regions. For instance, the delineated critical regions (dotted line) in Fig. 5(d) intersect with $D(t)$ during the updating range $[t_1, t_6]$, while the results of intervals $[t_0, t_1]$ and $[t_6, t_7]$ remain unchanged.

After several iterations, if continuous-time load profiles $D(t)$ intersect N critical regions throughout the period $\mathcal{T}$, we segment $\mathcal{T}$ into N intervals. These intervals are represented as $\mathcal{T}_n=[t_n, t_{n+1}]$ and the entire period $\mathcal{T}=\bigcup_{n=0}^{N-1}\mathcal{T}_n$ is divided accordingly. The transition from one critical region to the next occurs at respective interval endpoints.

However, if continuous-time load profiles $D(t)$ exit a critical region without entering a new one, it indicates an absence of relevant critical regions for certain segments. In such instances, the continuous-time dispatch problem becomes infeasible due to the violation of some constraints, rendering generation trajectories unachievable. The trajectory construction stage then issues recommitment instructions, as depicted in Fig. 4.

C. Generation Trajectories for Each Segment

Each interval $\mathcal{T}_n=[t_n, t_{n+1}]$ corresponds to a distinct segment of the $D(t)$. Within these intervals, decision variables $\mathbf{G}^P(t,D^P)$ are described by the same piecewise affine functions. For clear representation, the piecewise affine functions are marked as $\mathbf{G}_n^P(t, D^P)$ for a distinct interval $\mathcal{T}_n$. These functions link with critical regions H_i and are expressed below.

$$\mathbf{G}_n^P\left(t, D^P\right) = \mathbf{A}_i \begin{bmatrix} t \\ D^P \end{bmatrix} + \mathbf{B}_i, \forall t \in \mathcal{T}_n \tag{20}$$

where $\mathbf{A}_i$ is a $K\times2$ constant matrix. $\mathbf{B}_i$ is a column vector of dimension $K\times1$.

The matrices $\mathbf{A}_i$ and $\mathbf{B}_i$ correspond to the affine functions associated with the critical region H_i. The region index i is associated with the interval index n.

By introducing $D(t)$ to replace D^P, the decision variables $\mathbf{G}_n^P(t, D^P)$ are rewritten as $\mathbf{G}_n^C(t)$ below.

$$\mathbf{G}_n^C\left(t\right) = \mathbf{A}_i \begin{bmatrix} t \\ D(t) \end{bmatrix} + \mathbf{B}_i, \forall t \in \mathcal{T}_n \tag{21}$$

The affine functions $\mathbf{G}_n^C(t)$ represent generation trajectories for all units within the interval $\mathcal{T}_n$, which are functions of a single dependent variable (time t). The superscript C denotes the generation trajectories from the trajectory construction stage. If the continuous-time load profiles $D(t)$ are not linear, the generation trajectories $\mathbf{G}_n^C(t)$ will also be non-linear.

If the continuous-time load profiles $D(t)$, for $\forall t \in \mathcal{T}$, intersect the N critical regions, generation trajectories $\mathbf{G}^C(t)=[\mathbf{G}_0^C(t),\ldots, \mathbf{G}_{N-1}^C(t)]$ can be represented by the piecewise affine functions. However, if some segments lack relevant critical regions, it indicates that the affine functions cannot be derived from (21), resulting in infeasible dispatch for these segments. It is necessary to recommit more units, as previously discussed in Section IV.B.

V. Constraint Verification Stage

To achieve a linear formulation, the trajectory construction stage does not model differential constraints. We propose the constraint verification stage to verify the generation trajectories against the continuous-time ramping constraints and ensure the continuity of the generation trajectories.

A. Continuous-time Ramping Verification

The trajectory construction stage lacks continuous-time ramping constraints, leading to potential infeasibilities in generation trajectories $\mathbf{G}_n^C(t)$ during transitions from t_n to t_{n+1} within the interval $\mathcal{T}_n$. Based on the assumption described in Section II.C, the derivative functions of continuous-time load profiles $D(t)$ are calculated and denoted as $D'(t)$. Then, we calculate the derivative functions $\mathbf{G}_n'(t)$ according to (21).

$$\mathbf{G}_n'\left(t\right) = \mathbf{A}_i \begin{bmatrix} 1 \\ D'(t) \end{bmatrix}, \forall t \in \mathcal{T}_n \tag{22}$$

These derivative functions are crucial as they enable us to directly check continuous-time ramping constraints. We compare the derivative functions $\mathbf{G}_n'(t)$ against the downward /upward ramping capacities (R_k^D / R_k^U) for each unit k, as outlined in (23).

$$R_k^D \le G_{k,n}'\left(t\right) \le R_k^U, \forall t \in \mathcal{T}_n \tag{23}$$

Ramping violations of (23) trigger a subdivision of the generation trajectories, categorized into those that comply with (23) and those that violate (23). This subdivision identifies interval endpoints where (23) is not satisfied.

Consequently, each iteration undergoes two identifications of interval endpoints: initially during the trajectory construction stage, where interval endpoints are identified through critical regions; and subsequently during the constraint verification stage, where ramping violations are considered for the trajectories. As shown in Fig. 5(c), the generation trajectory $G_k(t)$ of unit k exhibits ramping violations from t_1 to t_2 and t_5 to t_6, which

determine additional four interval endpoints. As a result, three segments of continuous-time load profiles $D(t)$ in Fig. 5(a) are further refined to seven segments in Fig. 5(d).

Verification at interval endpoints presents a challenge due to the differing affine functions, such as $\mathbf{G}_{n-1}^{C}(t_n)$ and $\mathbf{G}_{n}^{C}(t_n)$. They are effective in adjacent intervals $[t_{n-1}, t_n]$ and $[t_n, t_{n+1}]$ respectively. However, the derivative functions at the endpoints may not always exist as shown in (24). Therefore, the verification of continuity at the endpoints cannot use (23).

$$\lim_{\varepsilon\to 0^+}\mathbf{G}_{n-1}^{'}(t_n-\varepsilon)\neq\lim_{\varepsilon\to 0^+}\mathbf{G}_{n}^{'}(t_n+\varepsilon) \tag{24}$$

To ensure the continuity of the interval endpoints, the differing affine functions must satisfy (25).

$$\lim_{\varepsilon\to 0^+}\mathbf{G}_{n-1}^{C}(t_n-\varepsilon)=\lim_{\varepsilon\to 0^+}\mathbf{G}_{n}^{C}(t_n+\varepsilon) \tag{25}$$

B. Discrete-time Adaptive Dispatch

The trajectory construction stage only models ramping constraints based on two specific dispatch results, omitting continuous-time ramping constraints. Section V.A elaborates on using (23) to identify ramping violations and determine corresponding endpoints. However, addressing these ramping violations may necessitate simultaneous adjustments at several endpoints, due to the continuous nature of ramping issues. The generation trajectories with the affected endpoints may be inaccurate and these trajectories also require further adjustments. Notably, simply satisfying (25) at certain endpoints does not ensure that they are unaffected by the changes at other endpoints. Therefore, identifying the affected endpoints becomes a crucial task. We propose a discrete-time adaptive dispatch model to address this issue. This model provides additional information and support in identifying the affected endpoints when verifying the continuity of endpoints.

After the continuous-time ramping verification, we assume that there is a total of N intervals. The discrete-time adaptive dispatch, specifically designed for the interval endpoints, differs from the earlier models in Section II.A. Here, the length of intervals is not uniform but adaptive. The length of interval $\mathcal{T}_n$ is denoted as $\Delta t_n = t_{n+1} - t_n$, with the subscript n added. The decision variables $\mathbf{G}^D(t_n)$ are marked with the superscript D to indicate results from discrete-time adaptive dispatch. The discrete-time adaptive dispatch is formulated below.

$$\min_{\mathbf{G}^D(t_n)}\sum\nolimits_{n=0}^{N-1}\left(C\left(\mathbf{G}^D(t_n)\right)\right)\Delta t_n \tag{26}$$

$$s.t.\quad \sum\nolimits_K G_k^D(t_n)=D(t_n),\forall n \tag{27}$$

$$G_k^{\min}\le G_k^D(t_n)\le G_k^{\max},\forall k,n \tag{28}$$

$$R_k^D\Delta t_n\le\left(G_k^D(t_{n+1})-G_k^D(t_n)\right)\le R_k^U\Delta t_n,\forall k,n \tag{29}$$

$$\mathbf{G}^D(t_0)=\mathbf{G}(0),\mathbf{G}^D(t_N)=\mathbf{G}(60) \tag{30}$$

Constraints (29) represent ramping constraints at endpoints. Constraints (30) ensure that dispatch results align with inputs from the day-ahead market dispatch. Infeasibility in the model (26)-(30) implies that the units cannot match the discrete-time load profiles $D(t_n)$. In this case, the continuous-time dispatch is also infeasible, because it imposes stricter ramping constraints from a continuous-time perspective. Such infeasibility necessitates the recommitment of units, as depicted in Fig. 4.

Conversely, feasibility in the model (26)-(30) verifies the ramping constraints at endpoints. If the discrete-time adaptive results and generation trajectories achieve consistency at some endpoints, these endpoints are isolated from adjacent points. In such scenarios, ramping constraints (29) are inactive at these isolated endpoints, ensuring that the piecewise affine functions of generation trajectories remain unaffected.

Then, (31) serves as the criterion for verifying continuity instead of (25). If both $\mathbf{G}_{n-1}^{C}(t_n)$ and $\mathbf{G}_{n}^{C}(t_n)$ match $\mathbf{G}^D(t_n)$, the continuity of endpoint t_n is verified. Otherwise, the endpoint t_n is affected by the adjacent endpoints and it is discontinuous. The discontinuity of endpoint t_n implies that constraints (29) become active at endpoint t_n, suggesting the need for a more precise division of critical regions and for new affine functions of generation trajectories.

$$\lim_{\varepsilon\to 0^+}\mathbf{G}_{n-1}^{C}(t_n-\varepsilon)=\mathbf{G}^D(t_n)=\lim_{\varepsilon\to 0^+}\mathbf{G}_{n}^{C}(t_n+\varepsilon) \tag{31}$$

The mismatch errors in (32) serve as convergence conditions. The generation trajectories are considered continuous at interval endpoint t_n only when the mismatch errors across all units are sufficiently small. Conversely, if the interval endpoints are discontinuous, the generation trajectories will be rebuilt in the next iteration.

$$\begin{cases}\boldsymbol{\Delta}_n(t_n)=\mathbf{G}^D(t_n)-\mathbf{G}_n^C(t_n)\\ \boldsymbol{\Delta}_{n-1}(t_n)=\mathbf{G}^D(t_n)-\mathbf{G}_{n-1}^C(t_n)\end{cases} \tag{32}$$

C. Feasible Trajectories and Updating Ranges

When generation trajectories within the interval $\mathcal{T}_{n-1}=[t_{n-1}, t_n]$ comply with (23) and show continuity at endpoints, these trajectories are considered feasible. If all generation trajectories from t_0 to t_n satisfy these criteria, then $[t_0, t_n]$ is preserved as a feasible trajectory range. Similarly, if all generation trajectories from t_m to t_N are feasible, then $[t_m, t_N]$ is preserved. Therefore, starting from the $t_0=S$ and $t_N=T$ of the period, two feasible trajectory ranges emerge in each iteration. These ranges are instrumental in identifying the infeasible range.

The infeasible range indicates the occurrence of inaccurate generation trajectories. For example, as depicted in Fig. 6, a discontinuous endpoint t_{n+1} indicates an inability to reach $G_k^D(t_{n+1})$ when transitioning from t_n to t_{n+1}. Thus, the generation trajectories within $[t_n, t_{n+1}]$ are inaccurate. In this case, if $[t_0, t_n]$ is a feasible trajectory range, t_n marks the start of the infeasible range. Similarly, if $[t_m, t_N]$ is a feasible trajectory range and t_{m-1} is discontinuous, then the t_m marks the end of the infeasible range. Although some generation trajectories within the infeasible range $[t_n, t_m]$ may be feasible, they are rebuilt to consider the effect of discontinuities at t_{n+1} and t_{m-1}. Thus, each iteration results in a single infeasible range and the infeasible range is flanked by two feasible trajectory ranges.

Based on the infeasible range, the updating ranges are set for the next iteration. Fig. 4 illustrates conditional operations guiding each iteration. These conditional operations result in two patterns of updating ranges. In the first pattern, as shown in Fig. 6(a), the updating range $\mathcal{T}^U=[S^U,T^U]$ aligns with the infeasible range $[t_n, t_m]$, considering dispatch results $\mathbf{G}^D(t_n)$ and $\mathbf{G}^D(t_m)$ in the next trajectory construction stage. Then, the next iteration subdivides the updating range $[t_n, t_m]$ to build new generation

trajectories and generate a new infeasible range.

If the new infeasible range remains unchanged, the iteration must proceed to the second pattern. This means that the endpoints t_{n+1} and t_{m-1} remain discontinuous and their corresponding ramping constraints cannot be ignored. Then, we refine the updating ranges to $[t_n, t_{n+1}]$ or $[t_{m-1}, t_m]$, as depicted in Fig. 6(b). The second pattern provides two updating ranges, and either of them can be chosen for the next iteration. Take updating range $[t_n, t_{n+1}]$ as an example, the given results $\mathbf{G}^D(t_n)$ and $\mathbf{G}^D(t_{n+1})$ are introduced in the next iteration. Since the new generation trajectories must pass through $\mathbf{G}^D(t_n)$ and $\mathbf{G}^D(t_{n+1})$, this iteration must change the infeasible range. Notably, a similar method can be applied to the updating range $[t_{m-1}, t_m]$.

Therefore, the updating ranges are dynamically set in two patterns, adapting to changes in the infeasible range. The ultimate goal is to achieve that all continuous-time generation trajectories are feasible, marked by the absence of the infeasible range. The process, governed by two conditional operations, is shown in Fig. 4.

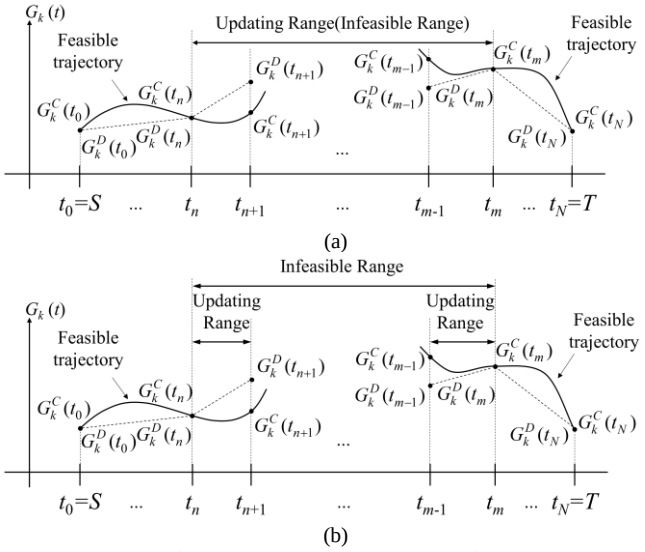


Fig. 6. Illustration of discontinuities and two patterns of updating ranges determined by the unit k (a) the first pattern, (b) the second pattern.

To sum up, the external constraint verification stage iteratively identifies and updates any infeasible generation trajectories until they adhere to the complete continuous-time ramping constraints (differential constraints). In each iteration, inequality (23) is used to externally verify the continuity of generation trajectories, while equality (31) is used to ensure the continuity of endpoints. By setting the mismatch errors to be sufficiently small, the endpoints achieve practical continuity. Mathematically, this continuity is only an approximation.

VI. Case Studies

In practice, ISOs, regardless of their market structure, implement additional scheduling stages that bridge the gap between the day-ahead market and real-time operations. The operational process discussed in Section III aims to build hourly continuous-time generation trajectories. These trajectories are crucial in smoothing the transition from day-ahead dispatch to real-time operations, and they support additional scheduling stages. We first validate our methodology in an illustrative 5-bus system, demonstrating its ability to identify ramping resource shortages and build continuous-time generation trajectories. We then extend this validation to an actual 661-bus system, confirming the scalability and effectiveness of the proposed methodology.

Since there are no actual continuous-time load profiles and the available load data are all discrete-time, we need to predetermine continuous-time load profiles. In Section II.C, we state that the predetermined continuous-time load profiles must be single-valued functions of time and have derivative functions. In this paper, we use 5-minute forecast load data (original data) within the 60-minute dispatch period ($\mathcal{T}$=[0,60]), and then we choose the most common polynomial to fit continuous-time load profiles as inputs.

These load profiles are fitted in advance with a seventh-degree polynomial function and then used as inputs. The continuous-time load profiles $D(t)$ are formulated as seventh-degree polynomial functions (33).

$$D(t) = \alpha_1 t^7 + \alpha_2 t^6 + \ldots + \alpha_7 t + \alpha_8 \tag{33}$$

where α_1~α_8 are the polynomial fitting coefficients.

Optimizations are solved by CPLEX v12.7.1 in the MATLAB environment on a ThinkPad X1 2021 with an Intel(R) Core (TM) i5-1135G7 CPU. The parametric solutions are obtained by the publicly available MPT3 tool. The convergence conditions are set at 0.001 MW. The results of continuous-time generation trajectories, represented by piecewise functions of time, are detailed in [24].

A. Ramping Resource Shortages in 5-bus System

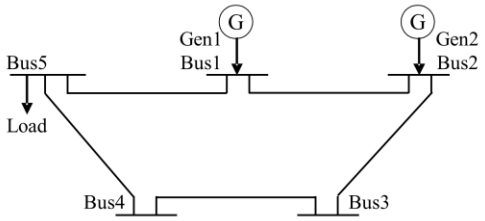


Fig. 7. The Illustrative 5-bus System.

Fig. 7 showcases the illustrative 5-bus system, comprising two thermal units, Gen1 and Gen2.

Table I details the physical characteristics of the two units.

TABLE I
Characteristics of Units in the Illustrative 5-bus System

Resource	Type	Bid ($/MWh)	Max. bid capacities (MW)	Min. bid capacities (MW)	R_k^D / R_k^U (MW/h)
Gen1	thermal	25	700	200	120
Gen2	thermal	30	500	200	80

Initially, we demonstrate the ability of our methodology to identify ramping resource shortages, highlighting the limitations of hourly day-ahead results. At t=0, Gen1 (205.32MW) is online while Gen2 is offline. At t=60, Gen1 (200.34MW) is online while Gen2 remains offline. Here, only Gen1 follows the continuous-time load profiles $D(t)$.

In Fig. 8(a), the red segment indicates an absence of the relevant critical region. The red segment intersects a black line at t=31.44 and t=41.39. The slope (black line) of 2.00 MW/min

(120MW/h) represents Gen1's ramping limitation. Then, connecting points on the red segment to $D(60)$ results in a slope steeper than Gen1's ramping limitation, indicating that the system experiences ramping resource shortages from t=31.44 to t=41.39.

This result identifies the red segment as infeasible due to ramping resource shortages and thus, the continuous-time dispatch is infeasible. It necessitates issuing recommitment instructions for additional units to enhance the ramping capacities. Consequently, the proposed methodology enables proactive actions for ramping resource shortages between the day-ahead dispatch and real-time operations.

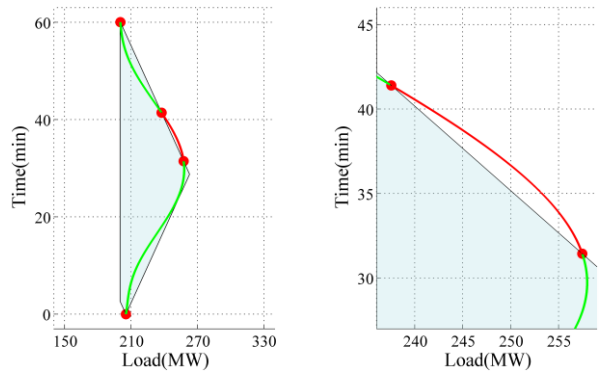


Fig. 8. Identification of ramping resource shortages (a) segments of continuous-time load profiles, (b) an enlargement of the infeasible segment.

B. Continuous-time Generation Trajectories in 5-bus System

An additional scenario is presented to build continuous-time generation trajectories. The physical characteristics of the units are consistent with Table I. Specifically, the coefficient α_8 is increased by 200 MW to align with the units' minimum output, while keeping other coefficients from Section VI.A unchanged.

At t=0, Gen1 and Gen2 are online, with outputs of 205.32MW and 200MW, respectively. At t=60, Gen1 and Gen2 remain online and have outputs of 200.34MW and 200MW, respectively. These results are used to execute initialization.

Post-initialization, the iterative dispatch methodology encompasses various sub-results derived from both the trajectory construction and constraint verification stages. To demonstrate the full process, we present different results. M1 represents the piecewise affine functions forming the continuous-time generation trajectories from the trajectory construction stage. M2 shows the outcomes of the discrete-time adaptive dispatch from the constraint verification stage. SE displays the dispatch results for the interval endpoints as determined by M2.

The trajectory construction stage for hourly $\mathcal{T}$=[0,60] identifies three critical regions as shown in Fig. 9(a), where continuous-time load profiles $D(t)$ intersect two critical regions and are divided into three segments. Fig. 9(b) delineates the continuous-time generation trajectories (M1) for each segment.

During the constraint verification stage, some trajectories violate the ramping constraints outlined in (23). This is depicted in Fig. 9(c), where the continuous-time load profiles with ramping violations are marked in red. These violations lead to four new segment endpoints, making a total of eight. Fig. 9(d) presents discrete-time adaptive dispatch results (M2), showing feasible trajectory ranges [0, 9.62] and [51.69,60]. Mismatches at t_3=23.47 and t_6=41.39 show their discontinuities and identify the infeasible range [9.62, 51.69]. Subsequently, the updating range [9.62, 51.69] is established based on the first pattern.

The trajectory construction stage for the updating range [9.62, 51.69], as illustrated in Fig. 9(e), determines two additional segment points, t_3=27.63 and t_4=30.96. The number of segment endpoints increases to ten. Fig. 9(f) compares the generation trajectories (M1) and the discrete-time adaptive dispatch results (M2) for each segment. The infeasible range still exists after the constraint verification stage, prompting a new updating range [23.48, 31.44] based on the first pattern.

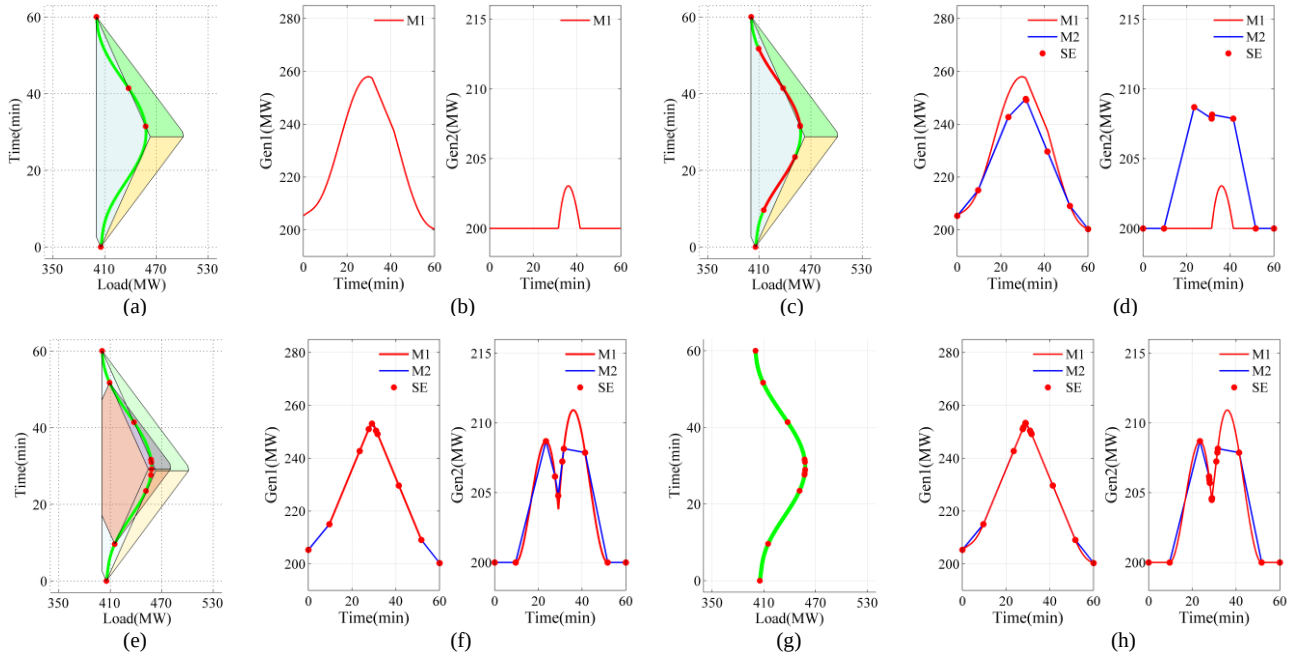


Fig. 9. Results for the illustrative 5-bus system during iterative dispatch methodology.

After seven iterations, our methodology divides the continuous-time load profiles into twenty segments and the final results are shown in Fig. 9(g). The generation trajectories (M1) and the discrete-time adaptive dispatch (M2) achieve the convergence conditions at SE and they are deemed all continuous (Fig. 9(h)). Finally, the continuous-time generation trajectories, represented by explicit piecewise affine functions [24], are successfully obtained.

C. *Results in Actual 661-bus System*

In this case, we formulate continuous-time load profiles $D(t)$ as seventh-degree polynomial functions as inputs, by employing the forecasted real-time load data at 5-minute intervals over a 60-minute period. In the final results, nine units have active ramping constraints, while the rest maintain consistent operation at their maximum or minimum generation levels. For simplicity, Fig. 10 represents the output of nine units, focusing on their alignment with net load profiles (minus the power outputs of other units with fixed generation levels).

As depicted in Fig. 10(a), after 11 iterations, the continuous-time net load profiles are segmented into 43 segments, and continuous-time generation trajectories are built for all units. Fig. 10(b) shows that the net load profiles (black line) are precisely traced by the power outputs of nine units (shown in different colors), indicating that the sum of piecewise affine functions can align with the functions of the net load profiles. The final affine functions for these nine units are elaborated in [24]. The accurate alignment indicates that the model errors inherent in the current discrete-time model are effectively addressed and our methodology achieves continuous-time balance. This result demonstrates that our methodology can enhance the precision of load tracing in power systems.

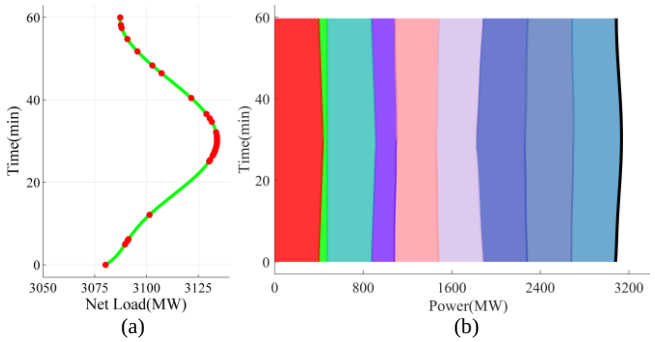


Fig. 10. Results (a) final segments of continuous-time net load profiles, (b) continuous-time balance between generation trajectories and load profiles.

We continue to analyze the computational performance of the proposed methodology. The iterative dispatch methodology requires two calculations in each iteration: one for the parametric programming model (M1) in the trajectory construction stage, and one for the discrete-time adaptive dispatch (M2) in the constraint verification stage.

To demonstrate computational performance, Table II provides the time taken for each calculation across 11 iterations. The parametric programming model (M1) is a two-dimensional problem, and its computational performance depends on the specific conditions of the updating range in each iteration, particularly the length of the updating range and the number of inactive constraints. These varying conditions in each iteration cause the computational performance of M1 to be somewhat unstable.

For the discrete-time adaptive dispatch (M2), the number of segment endpoints increases with each iteration. Initially, the calculation time of M2 increases rapidly as more segment endpoints are determined. However, as the updating range shortens, the number of new endpoints and the calculation time increase more slowly. Due to the variability in segment endpoints and constraints, the computational performance of M2 is also somewhat unstable.

TABLE II
THE COMPUTATIONAL TIME IN THE ACTUAL 661-BUS SYSTEM

Iterations	Total time (s)	M1 (s)	M2 (s)
1	8.7785	8.7750	0.0035
2	11.4531	11.4484	0.0047
3	12.1563	12.1506	0.0057
4	12.6332	12.6257	0.0075
5	9.8907	9.8833	0.0074
6	13.6094	13.6012	0.0082
7	7.6250	7.6161	0.0089
8	8.7032	8.6945	0.0087
9	8.0000	7.9918	0.0082
10	7.9857	7.9777	0.0080
11	11.4688	11.4608	0.0080

We also analyze the economic performance of the proposed methodology by comparing the production costs of four different discrete-time methods over an hourly period. The discrete-time methods include 60, 15, 5, and 1-minute intervals. Fig. 11 provides the comparison results for different discrete-time intervals. The discrete-time cost values use the production costs at the start of each interval (red points) to represent the entire intervals (red lines). Notably, the production costs from the continuous-time dispatch results are also piecewise affine functions of time (black curves).

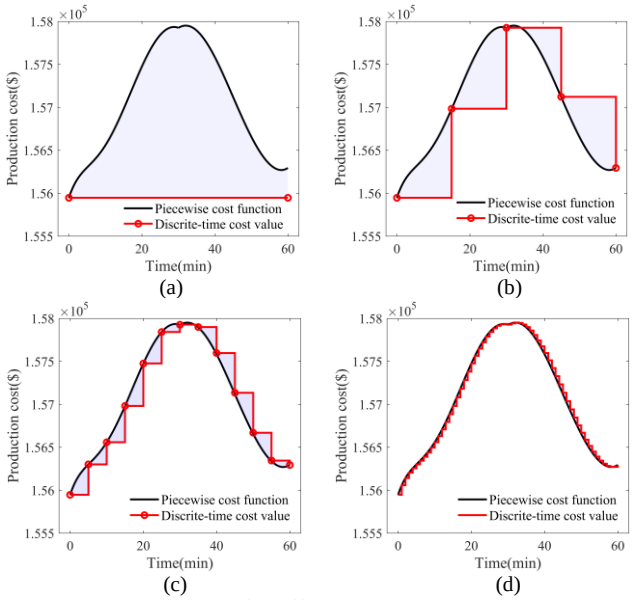


Fig. 11. Comparison results for different discrete-time intervals (a) 60-minute interval, (b) 15-minute interval, (c) 5-minute interval, (d) 1-minute interval.

In Fig. 11, it is clear that the production costs of the discrete-time methods exhibit model errors (blue shaded areas) when

compared with the continuous-time production costs. These errors reveal that the discrete-time production costs are approximations of the continuous-time costs. As the time step decreases, the approximation becomes more accurate.

Based on Fig. 11, we use specific values to further analyze the economic performance. The continuous-time production costs are integrated from the piecewise affine functions throughout the hourly period to directly obtain the total production costs. The discrete-time production costs are calculated for each interval first and then summed to obtain the total production costs. In Table III, it can be seen that as the time step becomes smaller, the total production costs gradually approach the costs of continuous-time dispatch.

Additionally, we estimate the imbalance costs by calculating the absolute value of the approximation errors (blue shaded areas). The logic behind this estimation is as follows: In the continuous-time dispatch, the units' output balances the continuous-time load profiles. However, in discrete-time dispatch, the units only balance the load profiles at the starting point of the interval and maintain a fixed output within the dispatch interval. The difference in output between the discrete-time and continuous-time results represents an intra-interval imbalance (the marginal costs of units are assumed to be fixed in this estimation). As the time step becomes smaller, the imbalance costs also decrease, as shown in Table III and illustrated by the shrinking blue shaded areas in Fig. 11.

By adding the total production costs and the imbalance costs, the final costs are obtained. As shown in Table III, the final costs of discrete-time results are greater than or equal to those of continuous-time results. The final costs of the 60-minute interval are equal to the continuous-time result because the 60-minute output only has one blue shaded area in Fig. 11(a). For the other three discrete-time results, when considering the absolute value of approximation errors (imbalance costs of each interval), the final costs are all greater than the continuous-time results.

Practically, intra-interval imbalances require much more expensive resources, such as specific ancillary services or even emergency resources, making the imbalance costs and final costs much higher than estimated here. If we adopt continuous-time dispatch, the additional imbalance costs caused by discrete-time intervals can be eliminated. Essentially, this is the economic advantage of eliminating model errors.

TABLE III
THE ECONOMIC PERFORMANCE OF DIFFERENT METHODS

Method	Total production costs ($/Hour)	Imbalance costs ($/Hour)	Final costs ($/Hour)
60-minute interval	1.5595e+05	1.1262e+03	1.5707e+05
15-minute interval	1.5699e+05	4.8665e+02	1.5748e+05
5-minute interval	1.5705e+05	1.5674e+02	1.5721e+05
1-minute interval	1.5706e+05	3.0336e+01	1.5710e+05
continuous-time	1.5707e+05	—	1.5707e+05

VII. CONCLUSION AND FUTURE WORK

This paper aims to balance continuous-time load profiles by providing continuous-time dispatch results. This continuous-time balance eliminates the approximation errors inherent in the discrete-time model. The iterative dispatch methodology consists of two stages: trajectory construction and constraint verification. This proposed methodology can build feasible continuous-time generation trajectories, represented as piecewise affine functions of time.

Case studies confirm the effectiveness of our methodology in aligning continuous-time generation trajectories with continuous-time load profiles. This alignment signifies progress towards a continuous-time balance. In practice, the applications of our methodology include the identification of ramping resource shortages and the facilitation of smooth transitions from day-ahead dispatch to real-time operations.

Further research is necessary to develop robust forecasting techniques for building continuous-time load profiles. These advancements are crucial for bringing our methodology to real implementation in actual control centers of power utilities.

REFERENCES

[1] Y. Chen et al., "Security-constrained unit commitment for electricity market: Modeling, solution methods, and future challenges," *IEEE Trans. Power Syst.*, vol. 38, no. 5, pp. 4668-4681, 2023.

[2] UCTE, "Frequency quality investigation, excerpt of the final report," 2008. [Online]. Available: https://www.entsoe.eu/fileadmin/user_upload/_library/publications/ce/otherreports/090330_UCTE_FrequencyInvestigationReport_Abstract.pdf

[3] ENTSO-E, "50-Hertz-Netzfrequenz," 2022. [Online]. Available: https://www.netzfrequenzmessung.de/

[4] J. E. Caicedo, D. Agudelo-Martínez, E. Rivas-Trujillo and J. Meyer, "A systematic review of real-time detection and classification of power quality disturbances," *Prot. Control of Mod. Power Syst.*, vol. 8, no. 1, pp. 30-66, 2023.

[5] R. Philipsen, G. Morales-España, M. Weerdt, and L. Vries, "Trading power instead of energy in day-ahead electricity markets," *Appl. Energy*, vol. 233-234, pp. 802-815, 2019.

[6] G. Morales-España, L. Ramírez-Elizondo, and B. F. Hobbs, "Hidden power system inflexibilities imposed by traditional unit commitment formulations," *Appl. Energy*, vol. 191, pp. 223–238, 2017.

[7] Y. Yao, C. Gao, S. Li, Y. Zhou, D. Wang, and M. Song, "Comparative study on distributed generation trading mechanisms in the UK and China," *Energy Convers. Econ.*, vol. 3, no. 3, pp. 122-141, 2022.

[8] California ISO, "Flexible ramping products: Incorporating FMM and EIM," 2014. [Online]. Available: https://www.caiso.com/Documents/SDGEComments_FlexibleRampingProductDraftFinalProposal.pdf

[9] Y. Cho, T. Ishizaki, N. Ramdani, and J. I. Imura, "Box-based temporal decomposition of multi-period economic dispatch for two-stage robust unit commitment," *IEEE Trans. Power Syst.*, vol. 34, no. 4, pp. 3109-3118, 2019.

[10] L. Zhang, P. Gribik, and T. Peng, "A probabilistic optimization reliability assessment commitment framework," 2011. [Online]. Available: https://cms.ferc.gov/sites/default/files/2020-05/20110628072957-Jun28-SesB1-Zhang-MISO_0.pdf

[11] C. Wu, G. Hug, and S. Kar, "Risk-limiting economic dispatch for electricity markets with flexible ramping products," *IEEE Trans. Power Syst.*, vol. 31, no. 3, pp. 1990-2003, 2016.

[12] California ISO, "SIBR interface specification for bidding services," 2017. [Online]. Available: https://www.caiso.com/Documents/SIBR InterfaceSpecificationWebServicesFall2017Release.pdf

[13] B. Knueven, J. Ostrowski, and J. Wang, "The ramping polytope and cut generation for the unit commitment problem," *INFORMS J. Comput.*, vol. 30, no. 4, pp. 739-749, 2018.

[14] Federal Energy Regulatory Commission, "Integration of variable energy resources," 2012. [Online]. Available: https://www.ferc.gov/sites/default/files/2020-06/OrderNo.764.pdf

[15] M. Kazemi, P. Siano, D. Sarno, and A. Goudarzi, "Evaluating the impact of sub-hourly unit commitment method on spinning reserve in presence of intermittent generators," *Energy*, vol. 113, pp. 338–354, 2016.

[16] S. Pineda, R. Fernández-Blanco, and J. M. Morales, "Time-adaptive unit commitment," *IEEE Trans. Power Syst.*, vol. 34, no. 5, pp. 3869–3878, 2019.

[17] M. Zhang, Z. Yang, W. Lin, J. Yu, W. Dai, and E. Du, "Enhancing economics of power systems through fast unit commitment with high time resolution," *Appl. Energy*, vol. 281, 2021, Art. no. 116051.
[18] M. Parvania and A. Scaglione, "Unit commitment with continuous-time generation and ramping trajectory models," *IEEE Trans. Power Syst.*, vol. 31, no. 4, pp. 3169–3178, 2016.
[19] M. Parvania and R. Khatami, "Continuous-time marginal pricing of electricity," *IEEE Trans. Power Syst.*, vol. 32, no. 3, pp. 1960–1969, 2017.
[20] K. Hreinsson, A. Scaglione, and B. Analui, "Continuous time multi-stage stochastic unit commitment with storage," *IEEE Trans. Power Syst.*, vol. 34, no. 6, pp. 4476-4489, 2019.
[21] California ISO, "Business practice manual for market operations," 2020. [Online]. Available: https://bpmcm.caiso.com/BPM%20Document%20Library/Market%20Operations/BPM_for_Market%20Operations_V65_redline.pdf
[22] R. Nikjoo, A. Estebsari, and M. Nazari, "Non-parametric regression model for continuous-time day-ahead load forecasting with Bernstein polynomial," 2019 IEEE International Conference on Environment and Electrical Engineering and 2019 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe), Genova, Italy, 2019, pp. 1-5.
[23] R. Khatami, M. Parvania, P. Khargonekar, and A. Narayan, "Continuous-time stochastic modeling and estimation of electricity load," 2018 IEEE Conference on Decision and Control (CDC), Miami, FL, USA, 2018, pp. 3988-3993.
[24] M. Zhang, "Results of continuous-time generation trajectories," 2023. [Online]. Available: https://figshare.com/articles/dataset/Results_of_Continious-time_generation_trajectories/24547873

Menghan Zhang received the B.S. and Ph.D. degrees in Electrical Engineering from Chongqing University, Chongqing, China, in 2019 and 2024, respectively. His major research interests include power system optimization and electricity markets.

Yue Zhou (Member, IEEE) received his B.S., M.S. and Ph.D. degrees in electrical engineering from Tianjin University, Tianjin, China, in 2011, 2016 and 2016, respectively. He currently works as a Lecturer in Cyber Physical Systems in the CIREGS centre of School of Engineering of Cardiff University. His research interests include power system demand side response, peer-to-peer energy trading, and distributed ledger technology (blockchain).

Mingxu Xiang (Member, IEEE) received the Ph.D. degree in electrical engineering from Chongqing University, Chongqing, China in 2022. He is currently an Assistant Research Fellow at Chongqing University. His research interests include power system operation and electricity market.

Juan Yu (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from Chongqing University, China, in 2007. Currently, she is a Full Professor at Chongqing University. Her research interests include big data analytics and power system analysis.

Wenyuan Li (Life Fellow, IEEE) graduated from Tsinghua University, Beijing, China, in 1968, and received the M.S. and Ph.D. degrees from Chongqing University, Chongqing, China, in 1982 and 1987, respectively, all in electrical engineering. He is currently a Professor with Chongqing University, Chongqing, China. Dr. Li is a Foreign Member of the Chinese Academy of Engineering and a Fellow of the Canadian Academy of Engineering. He received several awards, including the IEEE Power Engineering Society Roy Billinton Power System Reliability Award in 2011, the International PMAPS Merit Award in 2012, and the IEEE Canada Power Medal in 2014.

Zhifang Yang (Senior Member, IEEE) received his Ph.D. degree in electrical engineering from Tsinghua University in 2018. He currently works as a Full Professor at Chongqing University. His research interests include power system optimization and electricity market.