The distributed economic dispatch of smart grid based on deep reinforcement learning

Yang Fu1 | Xiaoyan Guo1 | Yang Mi1 | Minghan Yuan2 | Xiaolin Ge1 | Xiangjing Su1 | Zhenkun Li1

1 College of Electrical Engineering, Shanghai University of Electric Power, Shanghai, People’s Republic of China
2 College of Electrical Engineering, Shanghai University, Shanghai, People’s Republic of China

Abstract
In order to solve the problems of inefficient, inflexible and insecure for traditional centralized algorithm in the process of optimization dispatch, and with the application of artificial intelligence technology to smart grids, the novel distributed solution is proposed by using the deep reinforcement learning and the consensus theory to optimize the economic dispatch. Firstly, the optimal commitment sequence of massive units is realized through constructing deep reinforcement learning model. Secondly, the optimal unit output and efficient economic dispatch can be obtained by utilizing the improved consensus algorithm together with Adam’s algorithm. Finally, simulation results of IEEE-14 and IEEE-162 node systems may demonstrate the effectiveness of the proposed solution for the smart grids with complex network structures, which can not only solve the problem of massive data processing, but also it may reduce the dependence on the exact objective function when dealing with extremely complex load distribution scenes and distributed powers.

INTRODUCTION
With the development of large-scale renewable energy, the Smart Grid with high-density intermittent energy structure needs sufficient controllable resources to guarantee the efficiency, security and reliability for the power system operation. Flexible load may play an important role to maintain system balance compared with the traditional controllable generators. Therefore, how to take the connections among various parts of the “source-network-load-storage” and manage effectively diversified and scattered demand responses for the global optimization [1, 2] have attracted great attention in the field of science and technology.

The early solution to economic dispatch (ED) problem usually adopts centralized algorithms. All the distributed power (DG) and flexible loads send parameters to the central controller for centralized computing, processing and unified release. However, this centralized algorithm cannot be suitable for massive amounts of complex data due to large-scale intermittent renewable energy integration, the increased customers and the complex user types [3], nor can it meet the calculation requirements for access to DG with the plug-and-play features. Therefore, for distributed computing of ED [4–9], a distributed algorithm based on the consensus principle has been applied in many multi-agent systems [4]. It allows local agents to share information iteratively through a two-way communication link to find global optimal decisions. A consensus can be reached when all agents agree on the value of the information [5]. Then, for computing massive amounts of complex data, a distributed control strategy is proposed which is used to avoid single point failure and reduce the computing costs [6]. This strategy can reduce the core role of traditional central controllers in ED, and considers the expansibility of calculation and communication load in terms of each DG. In addition, for the convergence speed and global optimum of the consensus-based algorithm, a primal-dual principle is proposed in the literature [5] to be used

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.
© 2021 The Authors. IET Generation, Transmission & Distribution published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology

IET Gener. Transm. Distrib. 2021;1–14. wileyonlinelibrary.com/iet-gtd 1
to decompose the main problem into several sub-problems and then make parallel decision-making calculations to improve the convergence speed. In the literature [7], principles of sub-gradient steps are combined with the conventional consensus principle to update local optimization variables cyclically and reach the global optimum gradually.

The distributed algorithm based on the consensus principle has many specific advantages in solving the ED problem, but it may be rely on the predicted generation and load conditions [10]. In practice, the actual unit commitment is uncertain, non-static under a long-term scale, only considering real-time power allocation alone is of no practical significance for the ED [11]. In addition, it is difficult to represent the model with accurate mathematical formulas in some cases, for example, the cost function parameters of wind turbines can be very hard to be estimated since they are greatly affected by changing environmental factors [12]. Therefore, the appropriate ED solution need to be designed, which can not only eliminate the difficulty of establishing mathematical model, but also can make optimal decisions based on the previous or historical operating data [13], is of great significance in theory and practice.

However, when the data to be processed in a large-scale smart grid is too large, the “continuous loop” calculation mode cannot meet the system's requirements for the algorithm, especially when the environmental parameters change during the calculation process. In addition, when the objective function model is unknown or the order is too high, the above method cannot be used. Therefore, the deep reinforcement learning (DRL), which can save all the data having been calculated in the form of experience and learn, memory and utilize the data, can effectively improve computing efficiency. In the management of distributed energies [14], DRL is utilized to process high-dimensional data in smart grids, calculates energy optimization strategies online, and improves the efficiency of electricity through real-time feedback and control. In [15], quadratic function of DRL can be used to approximate the value function of each agent to achieve the optimal cooperative control with the purpose of solving the optimal consensus problem of multi-agent system which is capable of measuring input/output information. However, the DRL seems incompetent in the face of the “plug and play” [16] characteristics of distributed energies, dealing with continuous variables and other issues.

Inspired by the above research developments, the distributed ED solution is proposed by using the commitment of deep reinforcement learning and consensus theory, to study the ED of smart grids. The major contributions of this paper include:

1. The distributed method is utilized for source-load-storage economic power allocation in smart grids and to realize the coordination of the selected consensus variables among the agents. The process of the iterative calculation is completed in each local agent device without control centre for centralized calculation. Different from [3], the distributed method proposed in this paper is more simple and reliable, and is suitable for large-scale smart grid. Compared with [4–6], due to the existence of “adjustment item” in this method, various constraints can be satisfied in the calculation process automatically, and the “plug and play” requirements of DG can be realized, which the inherent shortcomings of DRL are eliminated [14, 15].

2. In response to the problems existing in the consensus algorithm, an improved algorithm is proposed by using the Adam algorithm [17]. Different from the [17], the theory of Adam and node agglomeration method is applied to the conventional consensus algorithm, strengthening the relationship between computing nodes and realizing the distributed form of Adam algorithm. The stability and computational efficiency are improved by the proposed algorithm. Compared with [8, 9], the proposed algorithm has better convergence efficiency and faster calculation speed.

3. The DRL is combined with the principle of consensus resulting in a solution to discrete problems of unit commitment and continuous problems of power allocation. Compared with [12, 13], the computational load of DRL can be reduced greatly by the proposed algorithm and the optimality and stability of the calculation results can be ensured. And the objective function of unit commitment is digitized, which no longer depends on the specific mathematical formula [13].

This paper is organized as follows. Models of ED are presented in Section 2. The ED solution theory is provided in Section 3. The algorithm convergence analysis is presented in Section 4. The Simulation results are formulated in Section 5. Finally, Section 6 is acknowledgments and Section 7 concludes this paper.

## 2 | SYSTEM MODEL AND PROBLEM FORMULATION

### 2.1 | Power distribution

The goal of load distribution is to find the optimal dispatch strategy. The total load is distributed among the $N'$ adjustable units in operation, so that the total cost is minimized, that is,

$$\min_{i=1}^{N'} \sum C_i(P_i)$$

(1)

where $C_i$ and $P_i$ denote the cost function and the power of adjustable unit $i$, respectively, that is, $P$ indicates the generating power for the generator, the dissipation power for the flexible load; $N'$ represents the number of adjustable unit in operation.

The cost function is formulated as

$$C_i(P_i) = a_i P_i^2 + b_i P_i + c_i$$

(2)

where $a_i$, $b_i$, and $c_i$ are the cost coefficients of the adjustable unit $i$. Load distribution is restricted by both the power balance constraint and the capacity constraint of each adjustable unit.
The power balance constraint is formulated as

$$\sum_{i=1}^{N} P_i + P_{loss} = D$$  \hspace{1cm} (3)

where $P_{loss}$ represents the transmission loss of line transmission power. Line transmission power constraints have a certain effect on the convergence process and optimization results of the algorithm. Generally, transmission loss accounts for about 3–7% of the total load [18]. $D$ is the total rigid load that cannot be adjusted. The capacity constraint of an adjustable unit is formulated as

$$P_i^{\text{min}} \leq P_i \leq P_i^{\text{max}}$$  \hspace{1cm} (4)

where $P_i^{\text{min}}$ and $P_i^{\text{max}}$ represent the minimum and maximum output power of the adjustable unit $i$, respectively. Other constraints are shown in [18].

### 2.2 Unit commitment

In the situation where a smart grid with a large number of generating units supplies total load $D$, the major goal of the unit commitment is finding a dispatch strategy to minimize the total cost within a period of time $T$, that is,

$$\min \sum_{i=1}^{N} \sum_{t=1}^{T} [C_i(P_{i,t})I_{i,t} + C_{i,\text{SU}}(t) + C_{i,\text{SD}}(t)]$$  \hspace{1cm} (5)

where $N$ is the total number of generating units. $I_{i,t}$ indicates that during the period $t$, the power generation unit $i$ is in operation or not. $C_{i,\text{SU}}$ represents possible downtime costs and $C_{i,\text{SD}}$ the start-up cost [18].

In addition to meeting the requirements of the load distribution in each time period, all power generation units $i = 1, 2, ..., N$ are subject to the minimum continuous up-down constraints [19],

$$[X_{i,\text{ON}}(t-1) - T_{i,\text{U}}] (I_{i,t-1} - I_{i,t}) \geq 0$$

$$[T_{i,\text{D}} - X_{i,\text{OFF}}(t-1)] (I_{i,t-1} - I_{i,t}) \geq 0$$  \hspace{1cm} (6)

and the generation’s ramp constraints [19],

$$-R_{i,\text{U}} \leq (P_{i,t} - P_{i,t-1}) I_{i,t-1} I_{i,t} \leq R_{i,\text{U}}$$  \hspace{1cm} (7)

where $R_{i,\text{U}}$ and $R_{i,\text{D}}$ are the ramp up and ramp down constraints, respectively. $T_{i,\text{U}}$ and $T_{i,\text{D}}$ are the minimum uptime and the minimum continuous downtime of a power generation unit $i$, respectively. $X_{i,\text{ON}}(t)$ is the period of continuous commitment of the generating unit $i$ at $t$, while $X_{i,\text{OFF}}(t)$ represents the time during which unit $i$ has been continuously stopped at $t$. Other detailed constraints are shown in [18, 19].

### 2.3 Solution strategy

The power distribution is characterized by continuity while the unit commitment has discrete features in the power system. For the continuous and discrete objects, the traditional strategy turns them into the same type of object to achieve the solution, which can hardly realize the optimal ED of smart grids and expense the efficiency and accuracy of calculation. This paper proposes an optimal strategy including a series of algorithms to cope with two different types of problems. Here an improved DRL algorithm is adopted to solve the discrete unit commitment problem firstly. Specifically, the optimal unit commitment sequence and the approximate unit power output are obtained by fixing some necessary variables. The improved distributed consensus algorithm is thereafter utilized to obtain the detailed unit power output based on the above results.

### 3 DISTRIBUTED DRL-BASED SOLUTION

To solve the above model, this section proposes a combined ED solution of DRL and the improved distributed algorithm based on the consensus principle.

#### 3.1 DRL algorithm

It can be known from Equation (3) that the state-action value function can be formulated in a recursive form, which can be used to calculate the optimal solution. In this paper, $s_{i,t}$ and $a_{i,t}$ are defined as the current stop/start state and the current start-stop action of each unit $i$, respectively, and $Q_i(s_{i,t}, a_{i,t})$ as the state-action value function. When the amount of data is large and the function $Q$ is difficult to formulate, it can be solved by the deep $Q$ network (DQN) algorithm.

#### 3.1.1 Updated state-action value functions

Each time, the $(s_{i,t}, a_{i,t})$ is accessed, state-action value function $Q(s_{i,t}, A_{i,t})$ is updated according to the formula as follows [20].

$$Q(S_{i,t}, A_{i,t}) \leftarrow (1 - \eta)Q(S_{i,t}, A_{i,t})$$

$$+ \eta (\text{rew}_{i,t} + \gamma \min_{A_{i,t+1}} Q(S_{i,t+1}, A_{i,t+1}))$$  \hspace{1cm} (8)

where $\eta$ represents the learning rate when the agent $i$ in the state space $S_{i,t}$ takes action space $A_{i,t}$ in the $t$-th time, and $\text{rew}_{i,t}$ denotes the reward and punishment feedback generated by the decision at the current moment. $\gamma$ represents the discount coefficient. The state space $S_{i,t}$ and the action space $A_{i,t}$ accord
to the following formula.

\[
S_{\ell, t} = \{P_{1, t}, P_{2, t}, \ldots, P_{N_t}; P_{t} = 90\% P_{N_t}\}
\]

\[A_{\ell, t} = \{I_{1, t}, I_{2, t}, \ldots, I_{N_t}; I_{t} \in 0 \text{ or } 1\}
\]

In Equation (8), the first term represents the processing of historical information, and the second term introduces new information brought by each access.

### 3.1.2 Exploration and exploitation

The balance of exploration and exploitation is achieved by selecting actions according to the idea of \(\varepsilon\)-greedy algorithm [21]. Specifically, the action with the highest value can be obtained in the selected dispatch strategy \(\pi(S_{\ell, t})\) with probability \(1-\varepsilon\), and start-stop \(a_{i,t}\) of the unit \(i\) is randomly selected with probability \(\varepsilon\).

### 3.1.3 Dispatch strategy adjustment

The dispatch strategy \(\pi(S_{\ell, t})\) is improved according to the current state-action value function \(Q(S_{\ell, t}, A_{\ell, t})\). \[
\pi(S_{\ell, t}) \leftarrow \arg \min_{A_{\ell, t+1}} Q(S_{\ell, t}, A_{\ell, t+1})
\]

If there are multiple \(A_{\ell, t+1}\) that minimize the target value, one of them is randomly selected.

### 3.1.4 Learning process of DQN model

When the objective function model Equation (1) is unknown or non-convex function, or the state space \(S_{\ell, t}\) is so complex that it cannot be represented by detailed mathematical formulas, the function \(Q\) cannot be obtained in the form of a state transition function or a table. In this case, the function \(Q\) needs to be represented by a function approximation method. The DQN model is also a suitable algorithm for approximating the tabular function \(Q\) by using the convolutional neural network (CNN) [22].

Take IEEE-14 node system with 4 thermal power units as an example, and please refer to [23] for detailed data. During the learning process, let \(P_{\text{load}}\) be the current interaction power with the main grid, and \(P_{\text{load}}^{+}\) be the power with the main grid at the next moment. Let \(P_{\text{load}}^{+}, S_{\ell, t}, S_{\ell, t+1}, A_{\ell, t}\), and \(A_{\ell, t+1}\) be the inputs to the CNN model. The data type of the input layer is \(18 \times 230\), representing the inputs' date during 230 h. The first and second layer is 36 convolution kernels of \(2 \times 1\) and 72 of \(2 \times 2\), respectively. Tanh function is used as the activation function. The max-pooling method is used in the pooling layer to transform the data into a one-dimensional vector with a length of 352, which is input to the fully connected layer. The fully connected layer includes 2 hidden layers, of which the number of neurons is 60 and 30, respectively. The number of neurons in the output layer is 1, which indicates the value \(Q\) obtained from the current start-stop status of the units. The ReLU function is used as activation function.

The CNN in the DQN algorithm is used to approximate the function \(Q\). In DQN [24], the updated formula of the function at this moment is shown in Equation (12) as follows:

\[
\omega_{t+1} = \omega_t + \xi [\text{rew}_{\ell, t} + \gamma \min_{A_{\ell, t+1}} Q(S_{\ell, t+1}, A_{\ell, t+1}, \omega) - Q(S_{\ell, t}, A_{\ell, t}, \omega)] V_\omega Q(S_{\ell, t}, A_{\ell, t}, \omega)
\]

where \(\xi\) is learning rate of DQN, \(\omega_t\) is a weight parameter for fitting \(Q\) in DQN. According to the constraints Equations (6) and (7) of the model in Equation (5), the reward and punishment feedback function \(\text{rew}_{\ell}\) is defined in Equations (13)–(17) as follows:

\[
\text{rew}_{\ell} = r_1 + r_2 + r_3 + r_4
\]

\[
r_1 = \begin{cases} 
1, & -R_{\ell, D} \leq (P_{i,t} - P_{i,t-1})I_i, I_{t-1} \leq R_{\ell, U} \\
-1, & \text{other}
\end{cases}
\]

\[
r_2 = \begin{cases} 
1, & [X_{\ell, ON}(t-1) - T_{\ell, U}](I_{t-1} - I_t) \geq 0 \\
-1, & \text{other}
\end{cases}
\]

\[
r_3 = \begin{cases} 
1, & [T_{\ell, D} - X_{\ell, OFF}(t-1)](I_{t-1} - I_t) \geq 0 \\
-1, & \text{other}
\end{cases}
\]

\[
r_4 = \begin{cases} 
0, & 0 \leq \sum_{i=1}^{n} I_{i,t} P_{i,t} - D < \Delta p_1 \\
-\zeta_1 \sum_{i=1}^{n} I_{i,t} P_{i,t} - D, & \Delta p_1 \leq \sum_{i=1}^{n} I_{i,t} P_{i,t} - D < \Delta p_2 \\
-\zeta_2 \sum_{i=1}^{n} I_{i,t} P_{i,t} - D, & \sum_{i=1}^{n} I_{i,t} P_{i,t} - D \leq \Delta p_2
\end{cases}
\]

where \(r_1\) is the reward item of climbing constraint, \(r_2\) and \(r_3\) are the reward items of minimum continuous upper and lower bound constraint, and \(r_4\) is the reward item of power balance offset. \(\xi\) is the deviation reward coefficient, and \(\Delta p\) is the deviation limit value.

When training DQN, the mean square error is used to define the error function of the network, as shown in the formula as follows:

\[
L(\omega) = E[(\text{rew}_{\ell} + \gamma \min_{A_{\ell, t+1}} Q(S_{\ell, t+1}, A_{\ell, t+1}, \omega) - Q(S_{\ell, t}, A_{\ell, t}, \omega))^2]
\]

An optimal strategy can be achieved by finding the gradient of the error function with respect to \(\omega\). The stochastic gradient descent is used to update the parameters to obtain the optimal value \(Q\).
3.2 | Consensus algorithm improvement

3.2.1 | Fully distributed implementation

As can be seen from consensus algorithm [25], the state transition matrix $W'$ is determined by the communication network topology and usually replaced by a Laplace matrix. Taking the IEEE-14 node system as an example, the $14 \times 14$ adjacency matrix $D$ of 14 controllable units is established based on graph theory. If the unit $i$ and the unit $j$ have a directly adjacent relationship, then the element $d_{ij}$ in the matrix $D$ is assigned as 1, otherwise as 0. The element $d_{ij}$ is assigned as 0.

However, taking the same adjacency matrix elements for adjacent nodes will limit the convergence speed of the consensus variable. In order to solve the problem above, the idea of different weights is integrated into the adjacency matrix since the final convergence accuracy is not affected by the change of elements in the Laplace matrix if the convergence coefficient remains unchanged [25]. Therefore, the concept of node agglomeration is introduced in this paper to perfect the state transition matrix $W'$ and improve the convergence speed.

Specifically, the node agglomeration method [26] is equivalent to centring on two points of which the degree of connection is to be judged, agglomerating them into a core node, and taking the degree of a node connected to the core node into account in the calculation of importance. The importance of node $\alpha$ and the element $w'_{ij}$ of improved state transition matrix $W'$ are defined as follows:

$$\alpha_{ij} = \frac{x_i y_j}{\sum_{i,j\in V} x_i y_j}$$

$$w'_{ij} = \frac{\alpha_{ij} l_j}{\sum_{j=1}^{n} \alpha_{ij} l_j}$$

where $n$ is the total number of nodes; $V$ is a set of all nodes in the network; $x_i$ and $y_j$ represent the degrees and edges of the core nodes after node $i$ and node $j$ are agglomerated, respectively. $l_j$ is an element of the Laplace matrix.

3.2.2 | Update of power adjustment items

In general, the equal incremental cost criterion introduced in [8] is a classic solution to the problem of power system dispatch optimization. When the slight cost increase rate $\mu$ is selected as the consensus variable of the first-order consensus algorithm and constitutes the consensus item, it can be known from [8] that in the process of continuous iteration, $\mu$ will gradually approach a specific value which is not necessarily $\mu'$. That is, various constraints cannot be satisfied and only the consensus term cannot solve the model correctly. Therefore, an adjustment item needs to be added to feedback, correct and make the result approach $\mu'$. The consensus variable is modified and the formula is updated as follows [27]:

$$\mu_i = 2\alpha_i P_i + b_i$$

$$\mu_i(k + 1) = \sum_{j \in X_i} w'_{ij} \mu_j(k) + \delta \varphi_i(k)$$

$$\varphi_i(k + 1) = \sum_{j \in X_i} v_{ij} \varphi_j(k) - [P_i(k + 1) - P_i(k)]$$

where $\mu_i$ is the slight increase rate of the controllable unit $i$, which is usually defined as the derivative value of the cost function relative to power in Equation (2). $\varphi_i$ is the adjustment term and $\delta$ is the adjustment coefficient which is a small positive number. $v_{ij}$ is the element of the matrix $W'$ after transposition and $X_i$ is the set of indices of the neighbouring agents of agent $i$. The network used in this paper is a standard IEEE network model, and topological structure is called a strong connection [19], and it is used by the distributed algorithm. Thus, the proposed model cannot cause congestions in the grid.

In the process of iterative calculation, the power adjustment item determines the convergence direction of the consensus variable by using Equation (23), so that the results of power decision continuously approach the optimal solution that satisfies various constraints in Equation (3). The process of verification is detailed as follows:

For simplicity, the Equation (23) is written in vector/matrix form as Equation (24).

$$\varphi(k + 1) = V^T \varphi(k) - [P(k + 1) - P(k)]$$

$$1^T \varphi(k + 1) = 1^T V \varphi(k) - 1^T [P(k + 1) - P(k)]$$

Since $\varphi$ is a non-negative column stochastic matrix, that is, the elements in each column vector are non-negative and the sum is 1, so $1^T V = 1^T$. Then it can be derived from Equation (25) as follows:

$$1^T [\varphi(k + 1) + P(k + 1)] = 1^T [\varphi(k) + P(k)]$$

Then reducing Equation (26) into variable expression form, that is

$$\sum_{i \in X_i} [P_i(k + 1) + \varphi_i(k + 1)] = \sum_{i \in X_i} [P_i(k) + \varphi_i(k)]$$

Therefore, using $\sum [P_i(0) + \varphi_i(0)] = 0$ when setting the initial value can make the $\varphi_i$ in the calculation process as the negative feedback to converge towards 0. When all $\varphi_i$ converges to 0, it means that the active power shortage of the system is 0, and the constraint in Equation (3) is satisfied. At the same time, the consensus variable $\mu_i$ in Equation (22) is iterated to the convergence value $\mu_i'$. 


3.2.3 Improvement based on Adam algorithm

The gradient descent can be used for optimization the objective function in Equation (1) which is continuously differentiable. This paper uses Adam algorithm, an effective stochastic gradient descent optimization method, to optimize the results calculated by the consensus algorithm with the purpose of accelerating the overall convergence. The proposed algorithm is completely distributed, so distributed improvements are conducted to Adam algorithm in this part.

The objective function is known as \( C(P) \) in this paper, the exponential moving average of the gradient \( m_k \) and the square gradient \( v_k \) is updated respectively in the calculation and the hyper-parameter \( \beta_1, \beta_2 \in [0, 1) \) is utilized to control the exponential decay rate of these moving averages [17].

The updated iterative formula for biased estimation \( m(k) \) and unbiased estimation \( v(k) \) is presented as follows [17]:

\[
\begin{align*}
\hat{m}(k) &= \hat{m}(k-1) + (1 - \beta_1)\nabla P C(P(k-1)) \\
\hat{v}(k) &= \hat{v}(k-1) + (1 - \beta_2)\nabla^2 P C(P(k-1))
\end{align*}
\]

where \( \beta_1^k \) and \( \beta_2^k \) are \( \beta_1 \) and \( \beta_2 \) to the power \( k \), respectively. The estimations after correction are as follows [17]:

\[
\begin{align*}
\hat{m}^k(k) &= \hat{m}(k)/ (1 - \beta_1^k) \\
\hat{v}^k(k) &= \hat{v}(k)/ (1 - \beta_2^k)
\end{align*}
\]

The centralized updated formula [17] of \( P \) is as follows:

\[
P(k) = P(k-1) - \frac{\hat{m}^k(k) / (\sqrt{\hat{v}^k(k) + \epsilon})}{\sqrt{\hat{v}^k(k) + \epsilon}}
\]

where \( \theta \) is the step size and is generally set to 0.001; \( \epsilon \) is the coefficient for fine-tuning the step size.

Therefore, the distributed form of Equation (30) is proposed in this paper as follows:

\[
P(k) = P(k-1) - \sum_{j \in X_i} w''_{ij} m'_{j}(k) / (\sqrt{v'_{j}(k) + \epsilon})
\]

where \( w'' \) is the element of the transform matrix \( W'' \), which \( 1^T W'' = 0^T \) and \( W'' 1 = 0 \) is satisfied in the case of vector. The \( w'' \) is formulated as follows:

\[
w''_{ij} = \begin{cases} 
-2/(x_i + x_j) & j = i' \in X_i' \\
-\sum w''_{ij} & j = 1 \\
0 & \text{otherwise}
\end{cases}
\]

where \( x_i \) is the degree of the agent \( i \).

By the improved Equations (31), (32), the distributed form of the Adam algorithm is implemented and is better integrated into the distributed consensus algorithm. Simultaneously, the convergence speed is further improved due to the gradient descent principle of Adam algorithm being integrated into the consensus algorithm. Consequently, the calculation efficiency of the entire algorithm proposed is improved and the total calculation time is reduced.

4 ALGORITHM CONVERGENCE ANALYSIS

In this section, the convergence analysis of the fully distributed optimization algorithm based on the consensus theory and Adam algorithm is mentioned to prove the effectiveness theoretically.

Aiming at finding the optimal value of the independent variable \( P \) in the function \( C(P) \), the test function \( E(K) \) is defined to represent the degree of algorithm convergence, which is the sum of all the previous difference between the online prediction \( C(P(k)) \) and the best fixed point parameter \( C(P^*) \) from a feasible set \( X \) for all the previous steps \( K \). Specifically, the test function is defined as Equation (33):

\[
E(K) = \sum_{k=1}^{K} [C(P(k)) - C(P^*)]
\]

where \( P^* = \arg \min_{P \in X} C(P) \), and \( C(P) \) represents Equation (2).

**Lemma 1.** If a function \( f: \mathbb{R}^d \rightarrow \mathbb{R} \) is convex, then for all \( x, y \in \mathbb{R}^d \),

\[
f(y) \geq f(x) + \nabla f(x)^T (y - x)
\]

The above lemma can be used to verify the presence of upper bound in the test function. And the proof for the main theorem is constructed by the updated rules proposed in this paper.

**Theorem 1.** Assume that the function \( C(P) \) has bounded gradients, \( \|\nabla P(C)\|_2 \leq G, \|\nabla P(C)\|_\infty \leq G_{\infty} \) for all \( P \in \mathbb{R}^d \) and distance between any \( P(k) \) generated by algorithm is bounded: \( \|P(m) - P(n)\|_2 \leq D_1 \|P(m) - P(n)\|_\infty \leq D_{\infty} \) for any \( m, n \in \{1, 2, ..., K\} \). The algorithm proposed in this paper achieves the following guarantee for all \( K \geq 1 \).

\[
E(K) \leq \frac{KD_1^2 G_{\infty}}{2} + \frac{\beta_1 D_1^2 G_{\infty}}{2(1 - \beta_1)} + \left( \frac{\beta_1}{2(1 - \beta_1)} + \frac{\sum w''}{2(1 - \beta_1)} \right) \frac{1}{1 - \beta_1^2 / \sqrt{\beta_2}} \frac{1}{\sqrt{1 - \beta_2}}
\]

**Proof.** Let \( \nabla C(P(k)) = g_k, P(k) = P_k, m_k = m_k(k) = n_k, v(k) = v_k, m'_k = m'_k(k) = v'_k(k) = v'_k(k) \). Using Lemma 1, then,

\[
C(P_k) - C(P^*) \leq g_k(P_k - P^*)
\]
From the updated rules Equation (31) presented in the fully distributed optimization algorithm, then,

\[
P_{k+1} = P_k - \sum u'' m'_k / \sqrt{v_k'}
\]

\[
= P_k - \sum_{1 \leq k \leq K} u'' (\beta_1 m_{k-1} + g_k (1 - \beta_1))
\]

\[
= P_k - \sum_{1 \leq k \leq K} u'' (1 - \beta_1) g_k (P_k - P^*) / \sqrt{v_k'}
\]

(37)

Square both sides of the update rule Equation (28), and then,

\[
(P_{k+1} - P^*)^2 = (P_k - P^*)^2 + \sum u''^2 (m'_k / \sqrt{v_k'})^2
\]

\[-2 \sum_{1 \leq k \leq K} u'' (1 - \beta_1) g_k (P_k - P^*) / \sqrt{v_k'}
\]

\[
- \frac{2}{1 - \beta_1} \beta_1 m_{k-1} + (1 - \beta_1) g_k (P_k - P^*) / \sqrt{v_k'}
\]

(38)

Because \( \alpha b / (\beta^2 / 2 + \beta^2 / 2) \), then the Equations (36) and (38) can be combined into the following Equation (39).

\[
g_k(P_k - P^*) = \frac{(1 - \beta_1) \sqrt{v_k'}}{2 \sum u'' (1 - \beta_1)} \left[ (P_k - P^*)^2 - (P_{k+1} - P^*)^2 \right]
\]

\[
+ \beta_1 m_{k-1} P_k + \sum_{1 \leq k \leq K} u''^2 (1 - \beta_1) g_k (P_k - P^*) / \sqrt{v_k'}
\]

\[
\leq \frac{\sqrt{v_k'}}{2 \sum u'' (1 - \beta_1)} (P_k - P^*)^2 + \beta_1 \sqrt{v_{k-1}} (P^* - P_k)^2
\]

\[
+ \frac{\beta_1 m_{k-1} P_k}{2(1 - \beta_1) \sqrt{v_k}} + \frac{\sum u'' m_k^2}{2(1 - \beta_1) \sqrt{v_k'}}
\]

(39)

According to Equations (39) and (36), and then,

\[
E(K) \leq \frac{1}{1 \sum u'' (1 - \beta_1)} \sum_{k=1}^{K} \left[ \sqrt{v_k} (P_k - P^*)^2 \right]
\]

\[
+ \frac{\beta_1}{2(1 - \beta_1) \sum_{k=1}^{K} \sqrt{v_k}} (P_k - P^*)^2
\]

\[
+ \frac{\beta_1}{2(1 - \beta_1)} \sum_{k=1}^{K} m_{k-1}^2 + \sum_{k=1}^{K} u'' m_{k-1}^2 / \sqrt{v_k'}
\]

\[
+ \frac{\beta_1}{2(1 - \beta_1)} \sum_{k=1}^{K} m_{k-1}^2 / \sqrt{v_k'}
\]

\[
\leq \frac{\sum_{k=1}^{K} m_{k-1}^2 / \sqrt{v_k'}}{2(1 - \beta_2) - \beta_1^2 \sqrt{v_k'}}
\]

(40)

(41)

Finally, it shows that the test function converges.

\[
\frac{E(K)}{K} = O \left( \frac{1}{\sqrt{K}} \right), \lim K \to \infty \frac{E(K)}{K} = 0
\]

5 | SIMULATION ANALYSIS

In this section, an IEEE-14 node system and an IEEE-162 node system are built to evaluate the performance of the proposed solution. The overall calculation process of the proposed algorithm is presented. The effectiveness of the algorithm under different working conditions is verified by the simulation examples in the IEEE-14 node system, and the calculation speed of the algorithm in large-scale smart grid systems is further verified by the simulation examples in the IEEE-162 node system.

5.1 | IEEE-14 node system

In this part, the IEEE-14 node system is constructed to simulate the intelligent distribution network and test the performance of the proposed algorithm. In the system, nodes 1, 2, 3, and 6 are thermal power generators, node 7 is an energy storage device, and the rest nodes have adjustable flexible loads. In addition, the node 1 is also connected to the main grid for power complementation. The simulated apparatus is a smart main grid. The system topology [8] is illustrated in Figure 1. And the basic parameters of IEEE-14 node system is illustrated in Table 1.
**Algorithm** Distributed ED optimization algorithm

1: Initialize \( t = 0, \eta = 0.005, \gamma = 0.001, \xi = 0.05, \delta = 0.01, \varepsilon = 0. \)

2: Collect the initial start-stop status and initial power output of each unit, and build the initial \( Q \) function value table in the form of Equations (9) and (10).

3: repeat

4: \( t = t + 1 \)

5: Use the \( \varepsilon \)-greedy algorithm to select \( A_{ij,t} \).

6: Seek value scores as current reward feedback \( r_{aw} \).

7: Calculate the current function value \( Q \) via Equation (8).

8: The scheduling strategy \( \pi(s_{i,t}) \) is calculated via Equation (11).

9: until meet constraints and optimal solution requirements.

10: Build a CNN network model and train historical data. Achieve offline learning and online calculation.

11: Get and analyse new network topology data.

12: Build the matrix \( W' \) and \( W'' \) via Equations (14), (15) and (23).

13: \( k = 0 \)

14: repeat

15: \( k = k + 1 \)

16: Update consensus variables via Equations (17) and (18).

17: Get current power \( P_i(k) \)

18: Update biased estimate \( m_k \) and \( v_k \) via (19).

19: Compute bias-corrected estimate via Equation (20).

20: Update parameters \( P_i(k) \) via Equation (22).

21: until meet the convergence conditions

22: return parameters \( P_i(k) \).

23: end

5.1.1 Examples of fitting \( Q \) functions using CNN

The interactive power data of the simulated intelligent main power grid and power distribution network are collected as a feature. The optimal start-stop sequence of the unit is calculated based on the changing trend of the feature, the current environment variables and the state variables. 280 data of interactive power are collected as shown in Figure 2.

The unit start-stop sequence and the power output sequence corresponding to each data are respectively calculated by the deep reinforcement learning algorithm without the specific objective function of unit commitment. The first 230 data is used as training samples and the last 50 data as testing samples. \( 18 \times 230 \) training set data and \( 1 \times 230 \) value-\( Q \) label data are respectively input to the input/output layer of the CNN network for training. The simulation results after training are illustrated in Figure 3.

Figure 3 reflects the quality of the CNN network model. The MSE value is obtained by comparing the output data of the CNN network model with the test data to reflect whether the CNN network model can fit the objective function well, so that optimal unit combination can be completed quickly even when the objective function is fuzzy and non-convex or...
there are too many data. It can be seen from the Figure 3 that the fitting effectiveness of the first 230 data is better, and the MSE value is lower. The testing result of the last 50 data is basically within the qualified range. This is mainly because the difference between the first 230 training data and the last 50 testing data is slightly larger, which results in a lower DQN effectiveness, but the overall effectiveness is better. The digitization of the objective function of unit commitment is basically realized, and it is no longer dependent on the specific mathematical formula.

5.1.2 Single dispatch instruction

It is assumed that the interactive power from the simulated intelligent main power grid makes it necessary to start all four generators, and the total power imbalance at the moment is 0.86 MW. That means that the simulated intelligent distribution network receives single dispatch instruction (SDI) and performs a single action. The simulation experimental results by using the proposed algorithm are shown in figures below:

It can be seen from Figures 4–6 that the output power of the 14 adjustable units tends to be stable, the active power of the system finally reaches a balance state, and the consensus variable tends to a value, which proves that the proposed algorithm has a significant convergence performance.

It is important to note that the proposed algorithm is able to converge to the near-optima much faster and smoother than other distributed methods in [8], [9] and [6]. For example, compared with the experiments shown in [8], the proposed algorithm has higher convergence efficiency under the same experimental conditions, and the optimization results are the same. The superiority and correctness of this algorithm are proved. And one of the published experiments in [9] showed that the similar ED problem was solved after 80 iterations, while the proposed algorithm can be able to finish this problem at the 52nd iteration.

The generation supply and load demand significantly change respectively in the process of the algorithm iteration, but the total net power deviation is small. Figure 5 illustrates that the maximum power deviation of the entire simulated intelligent distribution network system is about 5.61 MW, which is a very small part (5.61 / 210 = 0.027) compared with the total capacity, and cannot cause a large frequency deviation. It indicates that the algorithm can meet the demand for a basic stability of power systems [8]. Therefore, the algorithm proposed in this paper also has good research value and application prospect in the operation and control of smart grid.

5.1.3 SDI considering transmission loss

Line transmission power constraints have a certain effect on the convergence process and optimization results of the algorithm. Generally, transmission loss (TL) accounts for about 3% to 7% of the total load [28]. Taking 4% for example, the Equation (21)
is updated as:

\[
\begin{aligned}
\mu_{i,t} &= \frac{(2a_i P_{i,t} + b_i)}{(1 + 0.04)} \\
\mu_{i,t} &= 2a_i P_{i,t} + b_i
\end{aligned}
\]  

(44)

where \(\iota\) and \(\iota''\) represent the serial number of the power generation unit and the flexible load, respectively. Under the same experimental environment and conditions as in Section 5.1.2, the SDI simulation experimental results considering the transmission loss are shown in the figures below.

Figure 7 illustrates that when the transmission loss is taken into account in the model, the consensus variables of all adjustable units converge to a lower common value compared with that demonstrated in Figure 4. Therefore, as shown in Figure 8, the output power of the generation unit is reduced, and the power absorbed by the flexible load is also reduced, but it has little effect on the algorithm’s convergence performance.

5.1.4 Multiple dispatch instruction

In reality, the interactive power is not static, the dispatch instructions will change as the balance between the load of the entire smart grid and the power of the total generator set is slightly broken [29]. Therefore, in order to verify the timeliness of the proposed algorithm, the interactive power is set to 46.7, 73, 132.6, and 194.7 MW in turn, and the time cycle of multiple dispatch instruction (MDI) is set to 0.4 s, with the purpose of simulating the operation of the algorithm in real dispatch. The simulation figures are given in Figure 9–11.

It can be seen from Figures 9, 10, and 11, the proposed algorithm can meet the basic practical application requirements: within each dispatch cycle, the power output of each adjustable unit is stable, and the active power of the system can eventually reach balance. With the increase of interactive power, the consensus variables of the system also gradually increases. All variables achieve convergence and obtain the optimal solution,
and there is no case where the algorithm does not complete the convergence within the prescribed period. The simulation results demonstrate that the proposed algorithm can meet the basic dispatch requirements of the system in complex environments.

5.1.5 Power exceeds limit

In order to verify the effectiveness of the proposed algorithm when the power output of some adjustable units reaches the limit, the interactive power of the simulated smart grid is set to 441.8 MW, so that the power output of some adjustable units reaches the upper limit. The convergence results of the algorithm are shown in Figures 12–14.

Figure 14 illustrates that when iterating to 10 times, the power generation $G_1$ reaches a maximum of 50 MW, and the consensus variable also reaches a maximum. To maintain the balance of active power in smart distribution networks, other adjustable units will undertake more power balancing tasks. Figure 13 shows that the effectiveness of the proposed algorithm is verified since the unbalanced power of the system finally reaches 0.

5.1.6 Plug and play

The distributed power should meet the plug-and-play (PAP) requirements in the operation of smart grids [29]. To verify the effectiveness of the proposed algorithm in the case of plug-and-play, the following distributed power scenarios are constructed: the initial environmental conditions of the simulation experiment are the same as those in Section 4.1.1. When a dispatch period (0.4 s) elapses, the distributed power $G_{15}$ depicted by dotted lines in the Figure 1 is connected to the 13th node of the system.

It can be seen from the Figures 15 and 17 that when the distributed $G_{15}$ is connected to the system, it undertakes a part of the power distribution resulting in the reduction of the output
power which the remaining adjustable generators make. Consequently, the consensus variable is also reduced in the system.

Figure 16 illustrates that during the process of connecting the distributed G15 to the system, the total net power deviation of the entire simulated intelligent distribution network is so small that it cannot cause a large frequency deviation that means, the algorithm meets the demand for a basic stability of the power system. Therefore, the proposed algorithm can satisfy the plug-and-play function of distributed power in smart grids.

5.1.7 Contrast with traditional centralized algorithm

A contrastive simulation with two commonly used centralized algorithms is conducted to verify the performance of the proposed algorithm in this part. It is assumed that the total unbalance power is 100 MW, and the rest of the simulation initial conditions is the same as that in Section 5.1.1. The efficiency of the convergence of each algorithm is reflected clearly through observing the process of the unbalanced power converging to 0. The simulation results are shown in Figure 18. The results of convergence from the centralized consensus algorithm and the algorithm based on traditional gradient descent principle are represented by curve 1 and 2, respectively. Curve 3 shows the convergence result by using the complete distribution based on the improved Adam algorithm.

As shown in the figure, the first and second algorithms converge after 95 and 88 iterations, respectively. The third one converges after 58 iterations. Moreover, the computing efficiency of the proposed algorithm in the early stage of iteration is obviously better than that of the other two. It can be seen that the iteration efficiency of the proposed algorithm is significantly higher than that of the commonly used traditional algorithms.
IEEE-162 node system

An IEEE-162 node system is built with the purpose of verifying the effectiveness of the proposed algorithm in a huge energy network with massive adjustable units in this part. The testing case consists of 162 buses, 17 generators, 284 lines, 9 transformers and 91 loads. The proposed algorithm is used to optimize the dispatch of 126 adjustable units in the case of this paper. It is assumed that the designed topology is a strong connection graph, the total load demand is 18,422 MW, and the interactive power is increased by 4800 MW. The initial conditions of the simulation are the same as those given in [8]. Due to the excessive amount of the detailed basic data, please see reference [8].

The proposed algorithm is used to perform optimization calculations, the simulation results are presented in the Figure 19.

Based on the comparison of the results illustrated in Figure 19, the simulation results can prove the proposed algorithm has a significant effectiveness compared with the experiments shown in [8], [9] and [6] when processing massive data in large-scale smart grids. Particularly in one of the simulations [8], it showed that the same simulation was solved after 500 iterations, but the proposed algorithm can get the same results at the 397th iteration in less time.

6 | CONCLUSION

In this paper, the distributed mixture solution is proposed for the optimal dispatch of unit commitment and load distribution. The proposed solution based on DRL and consensus theory can achieve calculation cost reduction while having good dynamic performance in calculating massive data for large-scale power grids. The convergence efficiency is significantly improved by integration of the improved Adam algorithm and the consensus principle. In the context of data-driven artificial intelligence, the application of DRL to the proposed solution can effectively improve the decision by optimizing dispatch strategies and broaden the methodological ideas for ED. Generally, under the background that the global power grid dispatch has attracted unprecedented attention, this approach has broad prospects of application.

ACKNOWLEDGMENTS

The authors would like to thank the National Natural Science Foundation of China (No. 61873159), Shanghai Committee of Science and Technology Projects (18020500700), “Electrical Engineering” Shanghai class II Plateau Discipline and Shanghai Green Energy Grid Connected Technology Engineering Research Center (No. 13DZZ251900).

REFERENCES

1. Huang, W., et al.: From demand response to integrated demand response: Review and prospect of research and application. Prot. Control Mod. Power Syst. 4(12), (1–13) 2019.
2. Li, H., et al.: Optimal energy management for industrial microgrids with high-penetration renewables. Prot. Control Mod. Power Syst. 2(1), 12 (2017).
3. Madumbari, R., Dasgupta, S., Cho, B.B.: Distributed control for optimal economic dispatch of a network of heterogeneous power generators. IEEE Trans. Power Syst. 27(4), 1750–1760 (2012). https://doi.org/10.1109/TPWRS.2012.2188048
4. Chen, G., Yang, Q.: An ADMM-based distributed algorithm for economic dispatch in islanded microgrids. IEEE Trans. Ind. Inf. 14(9), 3892–3903 (2018). https://doi.org/10.1109/TII.2017.2785366
5. Hao, R., et al.: Distributed piecewise approximation economic dispatch for regional power systems under non-ideal communication. IEEE Access 7, 45533–45543 (2019). https://doi.org/10.1109/ACCESS.2019.2908680
6. Rahbari-Astr, N., et al.: Incremental welfare consensus algorithm for cooperative distributed generation/demand response in smart grid. IEEE Trans. Smart Grid 5(6), 2836–2845 (2014). https://doi.org/10.1109/TSG.2014.2346511
7. Mai, V.S., Abed, E.H.: Local prediction for enhanced convergence of distributed optimization algorithms. IEEE Trans. Control Network Syst. 5(4), 1962–1975 (2018). https://doi.org/10.1109/TCNS.2017.2776084
8. Xu, Y., Li, Z.: Distributed optimal resource management based on the consensus algorithm in a microgrid. IEEE Trans. Ind. Electron. 62(4), 2584–2592 (2015)
9. Hug, G., Kar, S., Wu, C.: Consensus + Innovations Approach for Distributed Multiagent Coordination in a Microgrid. IEEE Trans. Smart Grid 6(4), 1893–1903 (2015). https://doi.org/10.1109/TSG.2015.240053
10. Liang, Y., Liu, F., Mei, S.: Distributed real-time economic dispatch in smart grids: A state-based potential game approach. IEEE Trans. Smart Grid 9(5), 4194–4208 (2018). https://doi.org/10.1109/TSG.2017.2652919
11. He, X., Zhao, Y., Huang, T.: Optimizing the dynamic economic dispatch problem by the distributed consensus-based ADMM approach. IEEE Trans. Ind. Inf. 16(5), 3210–3221 (2020). https://doi.org/10.1109/TII.2019.2908450
12. Chen, G., Li, C., Dong, Z.: Parallel and distributed computation for dynamical economic dispatch. IEEE Trans. Smart Grid 8(2), 1026–1027 (2017). https://doi.org/10.1109/TSG.2016.2623980
13. Xing, H., et al.: Distributed bisection method for economic power dispatch in smart grid. IEEE Trans. Power Syst. 30(6), 3024–3035 (2015). https://doi.org/10.1109/TPWRS.2014.2376935
14. Mocanu, E., et al.: On-line building energy optimization using deep reinforcement learning. IEEE Trans. Smart Grid 10(4), 3698–3708 (2019). https://doi.org/10.1109/TSG.2018.2834219
15. Zhang, J., et al.: Data-based optimal control of multiagent systems: A reinforcement learning design approach. IEEE Trans. Cybern. 49(12), 4441–4449 (2019). https://doi.org/10.1109/TCYB.2018.2868715
16. Guo, F., et al.: Distributed economic dispatch for smart grids with random wind power. IEEE Trans. Smart Grid 7(3), 1572–1583 (2016). https://doi.org/10.1109/TSG.2015.2434831
17. Kingma, D. P., Ba, J.: Adam: A method for stochastic optimization. Comp. Sci. (1412.6980, 1–15 2014)
18. Saravanan, B., Asudevan, E. V., Korhari, D.: Unit commitment problem solution using invasive weed optimization algorithm. Int. J. Electr. Power Energy Syst. 55, 21–28 (2014)
19. Li, F., Qin, J., Zheng, W. X.: Distributed Q-learning-based online optimization algorithm for unit commitment and dispatch in smart grid. IEEE Trans. Cybern. 50(9), 4146–4156 (2020). https://doi.org/10.1109/TCYB.2019.2921475
20. Liu, W., et al.: Distributed economic dispatch in microgrids based on cooperative reinforcement learning. IEEE Trans. Neural Networks Learn. Syst. 29(6), 2192–2203 (2018). https://doi.org/10.1109/TNNLS.2018.2801880
21. Lin, S., et al.: Learning capability of relaxed greedy algorithms. IEEE Trans. Neural Networks Learn. Syst. 24(10), 1598–1608 (2013). https://doi.org/10.1109/TNNLS.2013.2265397
22. Xi, L., et al.: A novel multi-agent DDQN-AD method-based distributed strategy for automatic generation control of integrated energy systems. IEEE Trans. Sustainable Energy 11(4), 2417–2426 (2020). https://doi.org/10.1109/TSTE.2019.2988361
23. Abu-Hashim, R., et al.: Test systems for harmonics modeling and simulation. IEEE Trans. Power Delivery 14(2), 579–587 (1999). https://doi.org/10.1109/61.754106
24. Dai, P., et al.: Distributed reinforcement learning algorithm for dynamic economic dispatch with unknown generation cost functions. IEEE Trans. Ind. Inf. 16(4), 2258–2267 (2020). https://doi.org/10.1109/TII.2019.2933443
25. Ren, W., Beard, R.W.: Consensus seeking in multiagent systems under dynamically changing interaction topologies. IEEE Trans. Autom. Control 50(5), 655–661 (2005). https://doi.org/10.1109/TAC.2005.846556
26. Sumant, P. S., Cangellaris, A. C., Aluru, N. R.: A node-based agglomeration AMG solver for linear elasticity in thin bodies. Commun. Numer. Methods Eng. 25(3), 219–236 (2009)
27. Yang, S., Tan, S., Xu, J.: Consensus based approach for economic dispatch problem in a smart grid. IEEE Trans. Power Syst. 28(4), 4416–4426 (2013). https://doi.org/10.1109/TPWRS.2013.2271640
28. Zhang, W., et al.: Online optimal generation control based on constrained distributed gradient algorithm. IEEE Trans. Power Syst. 30(1), 35–45 (2015)
29. Murty, V.V.S.N., Kumar, A.: Multi-objective energy management in microgrids with hybrid energy sources and battery energy storage systems. Prot. Control Mod. Power Syst. 5(1), 1–20 (2020)