Distributed Reinforcement Learning Algorithm for Dynamic Economic Dispatch with Unknown Generation Cost Functions

Dai, Pengcheng; Yu, Wenwu; Wen, Guanghui; Baldi, Simone

DOI
10.1109/TII.2019.2933443

Publication date
2020

Document Version
Accepted author manuscript

Published in
IEEE Transactions on Industrial Informatics

Distributed Reinforcement Learning Algorithm for Dynamic Economic Dispatch with Unknown Generation Cost Functions

Pengcheng Dai, Wenwu Yu, *Senior Member, IEEE*, Guanghui Wen, *Senior Member, IEEE* and Simone Baldi, *Member, IEEE*

***Abstract*—In this paper, the dynamic economic dispatch (DED) problem for smart grid is solved under the assumption that no knowledge of the mathematical formulation of the actual generation cost functions is available. The objective of the DED problem is to find the optimal power output of each unit at each time so as to minimize the total generation cost. To address the lack of a priori knowledge, a new distributed reinforcement learning optimization algorithm is proposed. The algorithm combines the state-action-value function approximation with a distributed optimization based on multipliers splitting. Theoretical analysis of the proposed algorithm is provided to prove the feasibility of the algorithm, and several case studies are presented to demonstrate its effectiveness.**

***Index Terms*—Distributed reinforcement learning, dynamic economic dispatch, state-action-value function approximation, multipliers splitting.**

I. Introduction

The power grid is undergoing significant changes due to the integration of distributed energy resources, the development of smart technologies, the high demand of transactions and energy management and so on [1], [2]. Within this context, smart grids have received increasing attention [3]. The smart grid technology makes full use of communication and sensing in an effort to attain safe, efficient, stable and sustainable power services [4]–[6]. In smart grid, the DED problem has attracted much attention. The aim of DED is to find the optimal power output of each generator at each time to minimize the total generation cost in a given time horizon. In most practical cases, the DED problem needs to be solved in a distributed way. It has been learned from existing literature that multi-agent systems theory [7]–[9] is an appealing framework to solve such a problem. The static economic dispatch (SED) problem is a special case of DED which has also been studied in the framework of multi-agent systems [10]–[20]. Specifically, a fully distributed λ-consensus algorithm was proposed in [10] for smart grid with a directed topology. The authors of [11] proposed a distributed discrete-time consensus algorithm under a jointly connected switching undirected topology. In [12], under a uniformly jointly strong connected directed graph with time-varying delays, some distributed gradient push-sum algorithms were discussed for SED. A distributed Laplacian-gradient algorithm was proposed in [13] with feasible initial point. Yi $et\ al.$ [14] solved the SED problem via an initialization-free distributed algorithm based on the multipliers splitting method. Guo $et\ al.$ [15] proposed an average consensus algorithm and the distributed projection gradient algorithm to solve SED with consideration of wind turbines and energy storage system. A distributed auction-based algorithm was proposed in [16] to solve a non-convex SED. In the presence of communication uncertainties, an adaptive incremental cost consensus-based algorithm was proposed in [18]. In contrast, few results on DED problem are reported in the literature due to the complexity of this problem [21]–[23]. A distributed primal-dual dynamic algorithm was proposed in [21]. Zhao $et\ al.$ [22] deal with a fully decentralized optimization for the multi-area DED through the cutting plane consensus algorithm. More recently, by integrating the average consensus protocol and alternating direction method of multipliers (ADMM), a distributed coordination algorithm has been proposed in [24] to solve the dynamic social welfare problem. In practice, the accurate mathematical expression of the cost functions in a DED problem may be unavailable as the cost functions are affected by various factors, such as operating conditions and aging of the generator. Note that most of the aforementioned algorithms no longer work when the accurate mathematical formation of the cost function is unavailable. Hence, it is of both theoretical and practical interest to design an algorithm to solve the DED problem with little information of the actual cost functions.

Reinforcement learning [25] is a method through which an agent can find the optimal policy by interacting with the environment. This has motivated the application of reinforcement learning algorithms in control and optimization problems, sometimes in the context of multi-agent systems [26]–[31]. The reinforcement learning-based approach is used to investi-

This work was supported by the National Natural Science Foundation of China under Grant Nos. 61673107 and 61673104, the National Ten Thousand Talent Program for Young Top-notch Talents under Grant No. W2070082, the General joint fund of the equipment advance research program of Ministry of Education under Grant No. 6141A020223, the Six Talent Peaks of Jiangsu Province through Grant No. 2019-DZXX-006, the Fundamental Research Funds for the Central Universities under Grant No. 4007019109 (RECON-STRUCT), the special guiding funds for double first-class under Grant No. 4007019201, and the Jiangsu Provincial Key Laboratory of Networked Collective Intelligence under Grant No. BM2017002. (*Corresponding author: Wenwu Yu.*)

P. Dai, W. Yu, and G. Wen, are with the School of Mathematics, Southeast University, Nanjing 211189, China (e-mail: Jldaipc@163.com; wwyu@seu.edu.cn; wenguanghui@gmail.com).

S. Baldi is with School of Mathematics, Southeast University, 211189 Nanjing, China, and guest with Delft Center for Systems and Control, TU Delft, 2628CD Delft, The Netherlands (e-mail: s.baldi@tudelft.nl).

gate the optimal tracking control problem in [26]. Data-driven optimal control based on reinforcement learning was proposed in [27] for discrete-time multi-agent systems with unknown dynamics. Wang $et\ al.$ [28] proposed a dual heuristic dynamic programming algorithm for a class of nonlinear discrete-time systems affected by time-varying delay. The method of policy iteration in reinforcement learning was used in [29] to find the optimal control for zero-sum games. Exciting applications of deep reinforcement learning are [30], [31], which show that an agent can learn to play Atari better than humans. In this paper, we draw inspiration from reinforcement learning techniques, especially from state-action-value function approximation and from nonlinear programming theories to solve the DED problem with little information of actual cost functions.

The contributions of this paper are as follows.

1) The techniques of state-action-value function approximation based on semi-gradient Q-learning and distributed optimization algorithm based on multipliers splitting are successfully combined in the proposed algorithm. This algorithm can deal with the situation of which the mathematical expression of the cost functions is not available.

2) The update of the operating policy depends not only on the optimal solution of the approximate state-action-value function but also on the last operating policy. This means that the cost can be proven to be monotonically non-increasing at each iteration.

3) Time-varying parameters in approximate state-action-value function are proposed. As compared to the use of a time-invariant parameters, they enable to reduce the error and preserve convexity of approximate state-action-value function. To the best of our knowledge, this is the first attempt to employ time-varying parameters in the approximation of the state-action-value function.

The rest of this paper is organized as follows. The DED problem is formulated in Section II. The distributed reinforcement learning optimization algorithm is proposed in Section III. Section IV confirms the feasibility of the distributed reinforcement learning optimization algorithm. Simulation results to demonstrate the effectiveness of the algorithm are provided in Section V. Finally, Section VI presents the conclusion and future work. The Appendix gives preliminaries about convex analysis, algebraic graph theory and reinforcement learning.

II. Problem Statement

A. Dynamic Economic Dispatch

We consider a smart grid setting where N units must make their electricity generation equal to the total power demand at each time slot t. The objective of the DED problem is to find the optimal electricity allocation such that the total generation cost of all units is minimized. The mathematical expression of this problem is:

$$\begin{aligned} \min \quad & \sum_{t=1}^{T}\sum_{i=1}^{N} F_i(p_{i,t}) \\ \text{s.t.} \quad & \sum_{i=1}^{N} p_{i,t} = D_t,\ t=1,2,\cdots,T, \\ & \underline{p}_i \le p_{i,t} \le \overline{p}^i,\ i=1,2,\cdots,N, \\ & |p_{i,t}-p_{i,t-1}| \le p_i^R,\ i=1,\cdots,N,\ t=1,\cdots,T, \end{aligned} \tag{1}$$

where $F_i(\cdot)$ is the generation cost function of unit i, $p_{i,t}$ is the power output of unit i at time t, D_t is the total power demand at time t, p_i^R is the ramp-rate limit of unit i. $\underline{p}_i$ and $\overline{p}^i$ are the minimum and maximum power output of unit i, respectively. For notational brevity, set $p_{i,0}+p_i^R=\overline{p}^i$, $p_{i,0}-p_i^R=\underline{p}_i$, and $D_t-\sum_{i=1}^{N}p_i^R \le D_{t+1} \le D_t+\sum_{i=1}^{N}p_i^R,\ i=1,2,\cdots,T-1$. We denote $\mathcal{P}_i=[\underline{p}_i,\overline{p}^i]$ as the set of admissible power output of unit i.

Various forms of the generation cost function have been proposed in literatures. The most common generation cost function is: $F_i(p_{i,t})=a_ip_{i,t}^2+b_ip_{i,t}+c_i$, where a_i, b_i, and c_i are some coefficients for unit i [19]. The cost function considered in this work is a more general sinusoidal cost function inspired by [33]:

$$F_i(p_{i,t})=a_ip_{i,t}^2+b_ip_{i,t}+c_i+|e_i\cdot\sin(f_i\cdot(\underline{p}_i-p_{i,t}))|,$$

where the additional coefficients e_i and f_i are related to the capacity of unit i. Note that the mathematical expression of this cost function is known for simulation purposes, but it is unknown for the purpose of controller design.

When considering the above cost function, the following challenges should be taken into account: (i) the non-convex objective function invalidates existing algorithms based on convex optimization problems; (ii) only the value of the generation cost is known while the mathematical formulation of cost function is unknown. Fortunately, reinforcement learning algorithm can be applied to tackle such challenges.

Remark 1: In the DED problem, the total demand D_t, the feasible power output combination (FPOC) of units and the generation cost at each time slot can be seen as the state, action and reward in the mind of reinforcement learning. Furthermore, the generation cost at each time slot is also important and should be fully considered with dealing with the DED problem. Hence, the discount factor γ introduced in the step of reinforcement learning (cf. Appendix C) is set as 1 in the DED problem.

Two standard assumptions are made to guarantee existence of an optimal distributed solution to (1):

Assumption 1: There exists at least one FPOC $(p_{1,1},\cdots,p_{N,1},\cdots,p_{1,T},\cdots,p_{N,T})^T$ at all time such that $\sum_{i=1}^{N}p_{i,t}=D_t, p_{i,t}\in\mathcal{P}_i,\ |p_{i,t}-p_{i,t-1}|\le p_i^R,\ t=1,\cdots,T,\ i=1,\cdots,N.$

Assumption 2: The graph topology about the units is undirected and connected. At each time slot t, each agent i can only access the local power demand $D_{i,t}$, adjust the local power output $p_{i,t}$ and obtain the local generation cost $F_i(p_{i,t})$.

III. Distributed reinforcement learning Optimization Algorithm

In order to solve the DED problem with unknown cost functions, we apply reinforcement learning ideas. Suppose each agent corresponding to each unit was assigned a unique identifier ID, e.g., its IP address. By using the graph discovery algorithm proposed in [15], each agent can get the total number of agents. A distributed reinforcement learning optimization algorithm is proposed based on seven steps.

1) *Discover the total demand at time slot t*: Define $\bar{D}_t[0] = (D_{1,t}, D_{2,t}, \cdots, D_{N,t})^T$. Apply the average-consensus protocol (18) for each agent i as follows:

$$\bar{D}_t[k+1] = \bar{D}_t[k] - \epsilon L \bar{D}_t[k], \tag{2}$$

where L is the Laplacian matrix of graph $\mathcal{G}$, $\epsilon \in (0, \frac{1}{\max_i l_{ii}})$. From Lemma 1 in Appendix B, we get $\lim_{k\to\infty} \bar{D}_t[k] = (\frac{1}{N} D_t)\mathbf{1}_N$ where $\mathbf{1}_N$ is a N-dimensional column vector with each entry being 1. Hence, the local estimation of the average power demand converges to the actual average power demand at time t. As result, the total demand at time t can be obtained as D_t.

2) *Find a FPOC at time slot t*: Choose $p_{i,t} \in (\max\{\underline{p}_i, p_{i,t-1} - p_i^R\}, \min\{\bar{p}_i, p_{i,t-1} + p_i^R\})$. Define the mismatch of demand-generations $m_t[0] = (D_{1,t} - p_{1,t}, \cdots, D_{N,t} - p_{N,t})^T$, and apply Lemma 1 in Appendix B again as follows:

$$m_t[k+1] = m_t[k] - \epsilon L m_t[k]. \tag{3}$$

It holds that $\lim_{k\to\infty} m_t[k] = \frac{1}{N}\sum_{i=1}^{N}(D_{i,t} - p_{i,t})\mathbf{1}_N = \alpha \mathbf{1}_N$. Adjust $p_{i,t}$ according to the following policy:

$$p_{i,t} \leftarrow \begin{cases} p_{i,t} + \mathrm{sign}(\alpha)\min\{\min\{\bar{p}_i, p_{i,t-1} + p_i^R\} \\ \qquad - p_{i,t}, \alpha\}, \qquad \alpha \geq 0, \\ p_{i,t} + \mathrm{sign}(\alpha)\min\{-\max\{\underline{p}_i, p_{i,t-1} - p_i^R\} \\ \qquad + p_{i,t}, |\alpha|\}, \quad \alpha < 0, \end{cases} \tag{4}$$

where $\mathrm{sign}(\cdot)$ is symbolic function. Repeat (3) and (4) till $\alpha = 0$.

Note that, when $\alpha = 0$, $P_t = (p_{1,t}, p_{2,t}, \cdots, p_{N,t})^T$ is a FPOC at time slot t.

3) *Measure the total generation cost at time slot t*: Define $\bar{c}_t[0] = (c_{1,t}, \cdots, c_{N,t})^T$, and $\bar{c}_t$ as the local estimation of the average generation cost at time slot t, where $c_{i,t} = F_i(p_{i,t})$ for each agent i. Apply the average-consensus protocol:

$$\bar{c}_t[k+1] = \bar{c}_t[k] - \epsilon L \bar{c}_t[k]. \tag{5}$$

As a result of Lemma 1 in Appendix B, we can obtain $\lim_{k\to\infty} \bar{c}_t[k] = \bar{c}_t \mathbf{1}_N$, i.e., the local estimation of the average generation cost converges to the actual average generation cost at time slot t, then the total generation cost is $N\bar{c}_t$.

4) *Update the parameters of approximate function at time slot t*: Define $J_t(D_t, P_t, \theta^t) = \phi(P_t)^T\theta^t$ to be the approximate state-action-value function, where $\phi(P_t)$ is a feature vector. The update of the parameters θ^t is

$$\begin{cases} \theta^t \leftarrow \theta^t + \beta[N\bar{c}_t + \min_{P_{t+1}} J_{t+1}(D_{t+1}, P_{t+1}, \theta^{t+1}) \\ \qquad - J_t(D_t, P_t, \theta^t)]\phi(P_t). \end{cases} \tag{6}$$

The feature vector may be constructed from P_t in many different ways. For easier analysis, it is smart to design $\phi(P_t)$ such that the approximate state-action-value function is a convex function. For example, let $\phi(P_t) = (p_{1,t}, \cdots, p_{N,t}, p_{1,t}^2, \cdots, p_{N,t}^2)^T$, $\theta^t = (\theta_1^t, \cdots, \theta_{2N}^t)^T$, and $f_i(p_{i,t}) = \theta_i^t p_{i,t} + \theta_{i+N}^t p_{i,t}^2$. Then $J_t(D_t, P_t, \theta^t) = \phi(P_t)^T\theta^t = \sum_{i=1}^{N} f_i(p_{i,t})$, (6) becomes

$$\begin{cases} \theta_i^t \leftarrow \theta_i^t + \beta[N\bar{c}_t + \min_{P_{t+1}} J_{t+1}(D_{t+1}, P_{t+1}, \theta^{t+1}) \\ \qquad - J_t(D_t, P_t, \theta^t)]p_{i,t}, \\ \theta_{i+N}^t \leftarrow \theta_{i+N}^t + \beta[N\bar{c}_t + \min_{P_{t+1}} J_{t+1}(D_{t+1}, P_{t+1}, \\ \qquad \theta^{t+1}) - J_t(D_t, P_t, \theta^t)]p_{i,t}^2. \end{cases} \tag{7}$$

Remark 2: $\min_{P_{t+1}} J_{t+1}(D_{t+1}, P_{t+1}, \theta^{t+1})$ in (7) can be obtained through step 5). Taking into account the particularity of the finite horizon in (1), we use time-varying parameters θ^t for each time slot t. This is done in order to guarantee that the approximate state-action-value function is a convex function (necessary for the analysis in Sect. IV). Note that (7) can be seen as a semi-gradient method applied to the state-action-value function [25].

5) *Obtain $\min_{P_t} J_t(D_t, P_t, \theta^t)$ in a distributed way*: Solve the following problem about approximate state-action-value function

$$\begin{aligned} \min \quad & \sum_{i=1}^{N} f_i(p_{i,t}) \\ \text{s.t.} \quad & \sum_{i=1}^{N} p_{i,t} = D_t, \\ & p_{i,t} \in \mathcal{P}_i, \ i = 1, 2, \cdots, N, \\ & |p_{i,t} - p_{i,t-1}^{a*}| \leq p_i^R, \ i = 1, 2, \cdots, N. \end{aligned} \tag{8}$$

where $p_{i,0}^{a*} = p_{i,0}$ for each i. Before moving on, let $\mathcal{P}_{i,t}^{new} = \mathcal{P}_i \cap [p_{i,t-1}^{a*} - p_i^R, p_{i,t-1}^{a*} + p_i^R]$. Problem (8) can be solved under the following standard assumption:

Assumption 3: There exists a finite optimal solution P_t^{a*} to problem (8). The Slater's constraint condition is satisfied for (8), that is there exist $\hat{p}_{i,t} \in \mathrm{int}(\mathcal{P}_{i,t}^{new})$, $\forall\ i$, such that $\sum_{i=1}^{N} \hat{p}_{i,t} = D_t$.

Here is the procedure to solve (8). The duality of (8) with $\lambda \in \mathbb{R}$ is

$$\max_{\lambda\in\mathbb{R}} \sum_{i=1}^{N} q_i(\lambda) = \sum_{i=1}^{N} \inf_{p_{i,t}\in\mathcal{P}_{i,t}^{new}} \{f_i(p_{i,t}) - \lambda p_{i,t} + \lambda \frac{1}{N} D_t\}.$$

We formulate a constrained optimization problem with Laplacian matrix L and $\Lambda = (\lambda_1, \lambda_2, \cdots, \lambda_N)^T \in \mathbb{R}^N$ as

$$\begin{aligned} \max_{\Lambda} \quad & \sum_{i=1}^{N} q_i(\lambda_i) \\ \text{s.t.} \quad & L\Lambda = 0_N. \end{aligned} \tag{9}$$

The augmented Lagrangian duality of (9) with multipliers $Z = (z_1, z_2, \cdots, z_N)^T \in \mathbb{R}^N$ is

$$\min_{Z} \max_{\Lambda} \sum_{i=1}^{N} q_i(\lambda_i) - Z^T L\Lambda - \frac{1}{2}\Lambda L \Lambda.$$

The distributed algorithm for agent i is given as follows:

$$\begin{cases} \dot{p}_{i,t} = \mathbb{P}_{\mathcal{P}_{i,t}^{new}}(p_{i,t} - \nabla f_i(p_{i,t}) + \lambda_i) - p_{i,t}, \\ \dot{\lambda}_i = (\frac{1}{N} D_t - p_{i,t}) - \sum\limits_{j\in\mathcal{I}_i} (z_i - z_j) - \sum\limits_{j\in\mathcal{I}_i} (\lambda_i - \lambda_j), \\ \dot{z}_i = \sum\limits_{j\in\mathcal{I}_i} (\lambda_i - \lambda_j). \end{cases} \tag{10}$$

From the KKT condition, the equilibrium point of (10) is the optimal solution to (8) (cf. analysis in Sect. IV). Denote one of such equilibrium points is $col(P_t^{a*}, \Lambda^{a*}, Z^{a*})$ as the column vector stacked with vectors P_t^{a*}, Λ^{a*}, and Z^{a*}. Then, the value of $\sum\limits_{i=1}^{N} (F_i(p_{i,t}^{a*}))$ can be obtained by Lemma 1 in Appendix B.

6) *Renew the local operating policy*: Renew the local operating policy according to the following algorithm.

Denote $W_{a*} = \sum\limits_{t=1}^{T} \sum\limits_{i=1}^{N} (F_i(p_{i,t}^{a*}))$, $W_p = \sum\limits_{t=1}^{T} \sum\limits_{i=1}^{N} (F_i(p_{i,t}))$, $W_\pi = \sum\limits_{t=1}^{T} \sum\limits_{i=1}^{N} (F_i(\pi_i(D_t)))$, then the local operating policy can be renewed by

$$\pi_i(D_t) \leftarrow \begin{cases} p_{i,t}^{a*}, & \text{if } \ W_{a*} = \min\{W_{a*},\ W_p,\ W_\pi\}, \\ p_{i,t}, & \text{if } \ W_p = \min\{W_{a*},\ W_p,\ W_\pi\}, \\ \pi_i(D_t), & \text{otherwise}, \end{cases} \tag{11}$$

where $P_t^{a*} = (p_{1,t}^{a*}, \cdots, p_{N,t}^{a*})^T = \arg\min\limits_{P_t} J_t(D_t, P_t, \theta^t)$. In particular, $\pi(D_t)$ is a determined policy in DED problem.

7) *Balance exploration and exploitation*: In order to balance exploration and exploitation, we use the ε-greedy policy, i.e., selecting the action $(\pi_i(D_t), \cdots, \pi_N(D_t))^T$ with probability $1 - \varepsilon$, and other FPOC with probability ε.

The distributed reinforcement learning optimization algorithm for the DED problem is summarized in Algorithm 1.

Remark 3: In the process of developing the distributed algorithm, the key difficulties are: (i) How to determine the total power demand at each time by agents in a distributed way in the absence of a centralized decision-making agent with global information? (ii) How to find a FPOC in a distributed way? (iii) How to renew the local operating policy in a distributed manner? For issue (i), the total power demand D_t can be obtained by the average-consensus protocol (2). The aim of (3) and (4) is to solve issue (ii) by finding a FPOC in a distributed way. Issue (iii) is addressed through (11).

Algorithm DED with distributed reinforcement learning optimization

Initialize $t = 0$, $k = 0$;
Initialize ε with ε-greedy policy;
Repeat
  $t \leftarrow t + 1$;
  Obtain the total power demand D_t at time t via (2);
  Initialize the parameters θ^t of the approximate state action-value function;
  Set J_t with $\theta^t = \mathbf{0}$;
Until $t = T$
Define $J_{T+1} = 0$.
Repeat
  $k \leftarrow k + 1$;
  $\tilde{r} = rand(1)$;
  Reset $t = 1$, $W_p = 0$, $W_{a*} = 0$;
  Repeat
    If $k \geq 2$ and $\tilde{r} \geq \varepsilon$ **then**
      Repeat
        Choose power output as $\pi(D_t)$;
        Obtain immediate generation cost of $\pi(D_t)$ via (5);
        Update the parameter θ^t through (7);
        $W_p \leftarrow W_p + N\bar{c}_t$;
        Find the P_t^{a*} of (8) by (10);
        Obtain immediate generation cost of P_t^{a*} via (5);
        $W_{a*} \leftarrow W_{a*} + N\bar{c}_t^{a*}$;
        $t \leftarrow t + 1$;
      Until $t = T + 1$
    Else
      Repeat
        Propose a power output $p_{i,t}$ of unit i;
        Repeat
          Predict the average demand-generation mismatch α based on (3);
          Adjust $p_{i,t}$ according to (4);
        Until $\alpha \to 0$
        If $k = 1$ **then**
          Denote the local operation policy $\pi(D_t)$ as P_t;
          $W_\pi \leftarrow W_\pi + N\bar{c}_t$;
        Else
          Choose power output as P_t;
          Obtain immediate generation cost via (5);
          Update the parameter θ^t through (7);
          $W_p \leftarrow W_p + N\bar{c}_t$;
          Find the P_t^{a*} of (8) by (10);
          Obtain immediate generation cost of P_t^{a*} via (5);
          $W_{a*} \leftarrow W_{a*} + N\bar{c}_t^{a*}$;
        End if
        $t \leftarrow t + 1$;
      Until $t = T + 1$
    End if
  Until $t = T + 1$
  Update the local operation policy by (11);
  $W_\pi = \min\{W_{a*},\ W_p,\ W_\pi\}$;
Until $k = K$
/* K is the maximum number of trials */

IV. Theoretical Analysis

In this section, the main theoretical results of the proposed distributed reinforcement learning optimization algorithm are provided and proven via convex analysis and projection.

First of all, the equilibrium point of (10) with P_t^{a*} is analyzed to be the optimal solution of (8), and the convergence of (10) to the exact optimal solution of (10) is also proved. Denote

$$\mathcal{P}_t^{new} = \mathcal{P}_{1,t}^{new} \times \mathcal{P}_{2,t}^{new} \times \cdots \times \mathcal{P}_{N,t}^{new},$$

$$P_t = (p_{1,t}, p_{2,t}, \cdots, p_{N,t})^T,$$

$$\Lambda = (\lambda_1, \lambda_2, \cdots, \lambda_N)^T,$$

$$Z = (z_1, z_2, \cdots, z_N)^T,$$

$$\nabla f(P_t) = (\nabla f_1(p_{1,t}), \nabla f_2(p_{2,t}), \cdots, \nabla f_N(p_{N,t}))^T.$$

Then, the compact form of (10) is

$$\begin{cases} \dot{P}_t = \mathbb{P}_{\mathcal{P}_t^{new}}(P_t - \nabla f(P_t) + \Lambda) - P_t, \\ \dot{\Lambda} = -L\Lambda - LZ + \frac{1}{N} D_t \mathbf{1}_N - P_t, \\ \dot{Z} = L\Lambda. \end{cases} \tag{12}$$

The following theorem is given for the equilibrium point of (12), which indicate that P_t^{a*} in the equilibrium point $(P_t^{a*}, \Lambda^{a*}, Z^{a*})$ of (12) is corresponding to the optimal solution of (8).

Theorem 1: Suppose that Assumptions 1-3 hold and the with equilibrium point of distributed algorithm (12) with $(P_t^{a*}, \Lambda^{a*}, Z^{a*})$, then P_t^{a*} is the optimal solution of (8).

Proof: By the property of the equilibrium point $(P_t^{a*}, \Lambda^{a*}, Z^{a*})$ of (12), we get the following equations:

(i) $L\Lambda^{a*} = 0$ i.e., $\Lambda^{a*} = \lambda^{a*}\mathbf{1}_N, \lambda^{a*} \in \mathbb{R}$, because the undirected graph $\mathcal{G}$ is connected.

(ii) $-L\Lambda^{a*} - LZ^{a*} + \frac{1}{N}D_t\mathbf{1}_N - P_t^{a*} = 0$, which implies that $D_t = \mathbf{1}_N^T P_t^{a*}$ i.e., $\sum_{i=1}^{N} p_{i,t}^{a*} = D_t$.

(iii) $\mathbb{P}_{\mathcal{P}_t^{new}}(P_t^{a*} - \nabla f(P_t^{a*}) + \Lambda^{a*}) - P_t^{a*} = 0$, which implies that $-\nabla f(P_t^{a*}) + \Lambda^{a*} \in N_{\mathcal{P}_t^{new}}(P_t^{a*})$.

Therefore, the equilibrium point $(P_t^{a*}, \Lambda^{a*}, Z^{a*})$ of (12) satisfies the KKT condition for (8)

$$\begin{cases} 0 \in \nabla f_i(p_{i,t}^{a*}) - \lambda^{a*} + N_{\mathcal{P}_t^{new}}(p_{i,t}^{a*}), \\ \sum_{i=1}^{N} p_{i,t}^{a*} = D_t. \end{cases} \tag{13}$$

Hence, P_t^{a*} in the equilibrium point $(P_t^{a*}, \Lambda^{a*}, Z^{a*})$ of (12) is the optimal solution of (8). ■

Based on the above result, our next task is to prove that the trajectories of (12) with P_t will convergence to the optimal solution P_t^{a*}.

Theorem 2: Under Assumptions 1-3, given the initial points $p_{i,t} \in \mathcal{P}_{i,t}^{new}, i \in 1, 2, \cdots, N$, the trajectories of the algorithm of (12) are bounded and the power output $p_{i,t}$ of agent i convergences to $p_{i,t}^{a*}$.

Proof: Denote $\overline{\mathcal{P}}_t^{new} = \mathcal{P}_t^{new} \times \mathbb{R}^N \times \mathbb{R}^N$. We define a new vector $M = col(P_t, \Lambda, Z)$ and the function $F(M) : \mathbb{R}^{3N} \to \mathbb{R}^{3N}$ as

$$F(M) = \begin{pmatrix} \nabla f(P_t) - \Lambda \\ L\Lambda + LZ - (\frac{1}{N}D_t\mathbf{1}_N - P_t) \\ -L\Lambda \end{pmatrix}. \tag{14}$$

Then, (12) can be written as $\dot{M} = \mathbb{P}_{\overline{\mathcal{P}}_t^{new}}(M - F(M)) - M$.

Define $H(M) = \mathbb{P}_{\overline{\mathcal{P}}_t^{new}}(M - F(M))$, and the dynamics become $\dot{M} = H(M) - M$. Consider the candidate Lyapunov function

$$V = -\langle F(M), H(M) - M\rangle - \frac{1}{2}\|H(M) - M\|^2 + \frac{1}{2}\|M - M^{a*}\|^2,$$

where $M^{a*} = col(P_t^{a*}, \Lambda^{a*}, Z^{a*})$ is the equilibrium point of (12). Via convex analysis and projection, we obtain

$$\begin{aligned} V =& -\langle F(M), H(M) - M\rangle - \frac{1}{2}\|H(M) - M\|^2 \\ &+ \frac{1}{2}\|M - M^{a*}\|^2 \\ =&\frac{1}{2}[\|M - F(M) - M\|^2 - \|H(M) - (M - F(M))\|^2] \\ &+ \frac{1}{2}\|M - M^{a*}\|^2 \\ \geq&\frac{1}{2}\|M - H(M)\|^2 + \frac{1}{2}\|M - M^{a*}\|^2. \end{aligned}$$

Hence, $V = 0$ if and only if $M = M^{a*}$. The derivative of V along (12) is

$$\begin{aligned} \dot{V} =&(F(M) - [J_F(M) - I](H(M) - M))^T(H(M) - M) \\ &+ (M - M^{a*})^T(H(M) - M), \end{aligned} \tag{15}$$

where $J_F(M)$ is the Jacobian matrix of $F(M)$

$$J_F(M) = \begin{pmatrix} \nabla^2 f(P_t) & -I & 0 \\ I & L & L \\ 0 & -L & 0 \end{pmatrix}, \tag{16}$$

which is positive semidefinite.

With the property of projection, it is obvious that $\langle M - F(M) - H(M), H(M) - M^{a*}\rangle \geq 0$, which implies $\langle M - H(M) - F(M), H(M) - M + M - M^{a*}\rangle \geq 0$. Hence, $\langle H(M) - M, M - M^{a*} + F(M)\rangle \leq -\|H(M) - M\|^2 - \langle F(M), M - M^{a*}\rangle$. We may further get that

$$\begin{aligned} \dot{V} =&\langle M - M^{a*} + F(M), H(M) - M\rangle + \|H(M) - M\|^2 \\ &- (H(M) - M)^T J_F(M)(H(M) - M) \\ \leq& - (H(M) - M)^T J_F(M)(H(M) - M)\rangle \\ &- \langle F(M), M - M^{a*}\rangle \\ \leq& - \langle F(M), M - M^{a*}\rangle \\ \leq& - \langle F(M) - F(M^{a*}), M - M^{a*}\rangle - \langle F(M^{a*}), M - M^{a*}\rangle \\ \leq&0. \end{aligned}$$

The last inequality holds because the Laplacian matrix is positive semidefinite, $f(P_t)$ is convex and because of the variational inequality of the optimal solution M^{a*}. Therefore, there exists a forward compact invariance set given as

$$IS = \{M | \frac{1}{2}\|M - M^{a*}\|^2 \leq V(M(0))\}.$$

From the KKT condition, there exist $p^{a*} \in N_{\mathcal{P}_t^{new}}(P_t^{a*})$ such that $p^{a*} = -\nabla f(P_t^{a*}) + \Lambda^{a*}$. Furthermore, we can obtain

$$\begin{aligned} \dot{V} \leq& - \langle F(M), M - M^{a*}\rangle \\ =& - \langle P_t - P_t^{a*}, \nabla f(P_t) - \Lambda - \nabla f(P_t^{a*})\rangle \\ &- \langle \Lambda - \Lambda^{a*}, L\Lambda + LZ - (\frac{1}{N}D_t\mathbf{1}_N - P_t^{a*})\rangle \\ &- \langle Z - Z^{a*}, -L\Lambda\rangle - \langle P_t - P_t^{a*}, \Lambda^{a*} - p^{a*}\rangle \\ \leq& - \langle P_t - P_t^{a*}, \nabla f(P_t) - \nabla f(P_t^{a*})\rangle \\ &+ \langle P_t - P_t^{a*}, p^{a*}\rangle - \langle \Lambda - \Lambda^{a*}, L(\Lambda - \Lambda)\rangle \\ \leq& - \langle P_t - P_t^{a*}, \nabla f(P_t) - \nabla f(P_t^{a*})\rangle \\ &- \langle \Lambda - \Lambda^{a*}, L(\Lambda - \Lambda)\rangle. \end{aligned}$$

Denote the set $\mathcal{M} = \{M | \dot{V} = 0\}$. Because of the positive definite Hessian matrix $\nabla^2 f(P_t)$ and the null space for Laplacian matrix L, we can obtain $\mathcal{M} = \{P_t = P_t^{a*}, \Lambda \in span\{\mathbf{1}_N\}\}$.

Next, we claim that the maximal invariance set within the set $\mathcal{M}$ is the equilibrium point of (8). Because of $\Lambda \in span\{\mathbf{1}_N\}\}$, then $Z = Z^{a*}$. According to (13), it is obvious that $\dot{\Lambda} = LZ^{a*} - (\frac{1}{N}D_t\mathbf{1}_N - P_t^{*a})$. We claim that $LZ^{a*} - (\frac{1}{N}D_t\mathbf{1}_N - P_t^{a*}) = 0$. Assume that $LZ^{a*} - (\frac{1}{N}D_t\mathbf{1}_N - P_t^{a*}) \neq 0$, then Λ will go to infinity, which contradicts that $\mathcal{M}$ is a compact set within IS. Hence, $\dot{\Lambda} = 0$ and $\Lambda = \Lambda^{a*}$. By the LaSalle invariance principle, the power output $p_{i,t}$ of agent i convergence to $p_{i,t}^{a*}$. ■

V. SIMULATION

In this section, the proposed distributed reinforcement learning optimization algorithm is tested through several examples.

TABLE I: Parameters of generation Units

Unit number	$\underline{p}_i$	$\overline{p}^i$	a_i	b_i	c_i	e_i	f_i	p_i^R
1	200	600	0.0020	10	500	300	0.03	50
2	100	400	0.0025	8	300	200	0.04	50
3	100	300	0.0050	6	100	150	0.05	50
4	50	200	0.0060	5	90	130	0.06	50

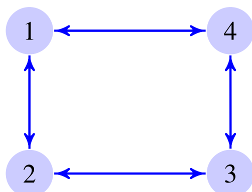

Fig. 1: Communication graph in Example 1.

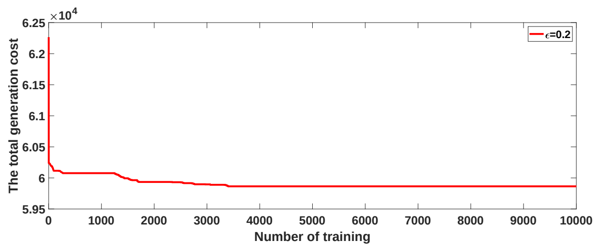

Fig. 2: The total generation cost of policy produced by the distributed reinforcement learning optimization algorithm in Example 1.

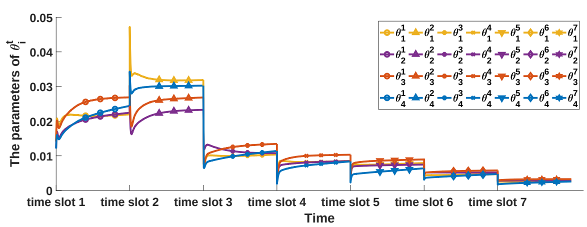

Fig. 3: The time-varying parameters θ_i^t in approximate state-action-value function.

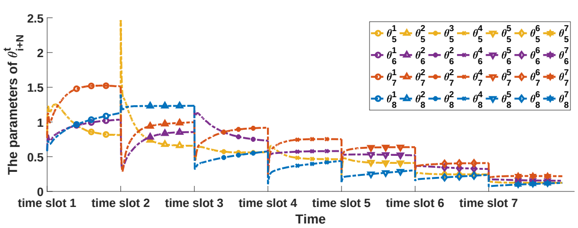

Fig. 4: The time-varying parameters θ_{i+N}^t in approximate state-action-value function.

Example 1. Consider four units connected via the undirected graph shown in Fig. 1. The cost function for each unit i is taken as $F_i(p_i) = a_i p_i^2 + b_i p_i + c_i + |e_i \cdot \sin(f_i \cdot (\underline{p}_i - p_i))|$, with coefficients shown in Table I (known only to the purpose of simulation). The admissible power outputs of each unit are set as follows: $\mathcal{P}_1 = [200, 600]$, $\mathcal{P}_2 = [100, 400]$, $\mathcal{P}_3 = [100, 300]$, and $\mathcal{P}_4 = [50, 200]$ (MW). The total power demand D_t is 800,

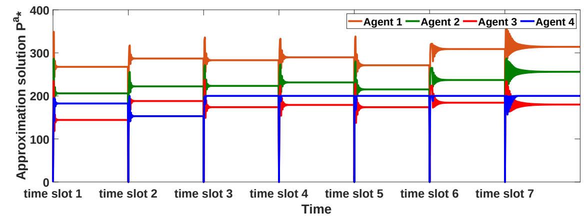

Fig. 5: The P^{a*} of approximate state-action-value function after training.

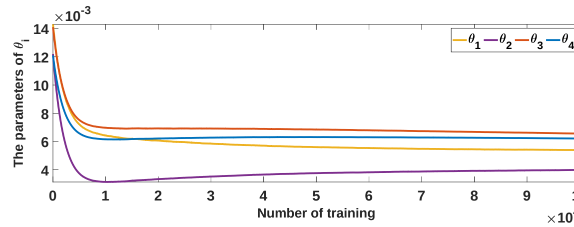

Fig. 6: The time-invariant parameters θ_i in approximate state-action-value function.

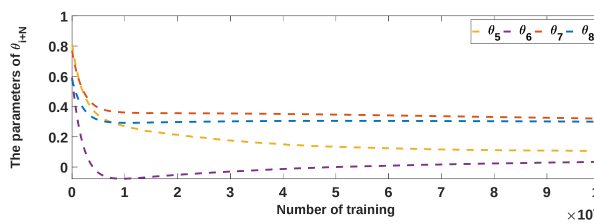

Fig. 7: The time-invariant parameters θ_{i+N} in approximate state-action-value function.

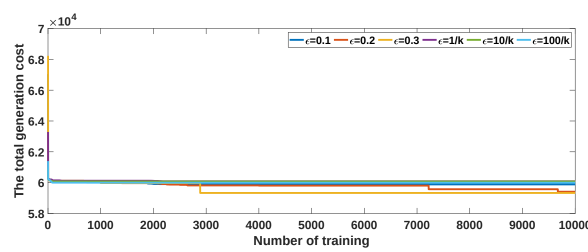

Fig. 8: The evolution of the total generation cost of updated policies in difference ε.

850,880, 900, 860, 930, and 950 (MW) for time periods $[0, 2)$, $[2, 6)$, $[6, 8)$, $[8, 18)$, $[18, 22)$, and $[22, 24)$, respectively.

We take for simplicity ε in the ε-greedy policy to be constant and equal to 0.2. As shown in Fig. 2, the total generation cost of updated policy is getting better and better during the training process. Figs. 3-4 show the time-varying parameters θ^t in approximate state-action-value functions for all time slots. In this example, the approximate state-action-value functions take the form $J_t(D_t, P_t, \theta^t) = \sum_{i=1}^{N} (\theta_i^t p_{i,t} + \frac{1}{4}\theta_{i+N}^t p_{i,t}^2)$. The optimal solutions P_t^{a*} of the approximate function for all time slots after training are shown in Fig. 5.

Remark 4: As the approximate state-action-value function $J_t(D_t, P_t, \theta^t)$ is the sum of total generation cost from time slot t to time slot T in the DED problem considered in this paper. It can seen from Figs. 3-4 that θ_t^i and θ_{i+N}^t are almost decreasing from time slot 1 to time slot T. Note that θ_i^t for time slot 2 is larger than θ_i^t over time slot 1 which does not satisfy the property of decreasing, however it has no effect according to the form of approximate state-action-value function.

In order to show the advantage of using time-varying parameters θ^t in the function approximation, a time-invariant parameters θ will be considered for all time slots. In other words, the approximate function takes the form $J(D_t, P_t, \theta) = \sum_{i=1}^{N}(\theta_i p_{i,t} + \frac{1}{4}\theta_{i+N} p_{i,t}^2)$. The parameters θ_i and θ_{i+N} are updated according to:

$$\begin{cases} \theta_i \leftarrow \theta_i + \beta[N\bar{c}_t + \min_{P_{t+1}} J(D_{t+1}, P_{t+1}, \theta) \\ \qquad - J(D_t, P_t, \theta)]p_{i,t}, \\ \theta_{i+N} \leftarrow \theta_{i+N} + \frac{\beta}{4}[N\bar{c}_t + \min_{P_{t+1}} J(D_{t+1}, P_{t+1}, \theta) \\ \qquad - J(D_t, P_t, \theta)]p_{i,t}^2. \end{cases} \tag{17}$$

Figs. 6-7 show updating process. As shown in Fig. 7, θ_6 goes below 0, which contradicts the assumption of convexity of approximate state-action-value function. In this case, the step 5) cannot be performed as the necessary assumptions are violated.

Remark 5: By the definition of the state-action-value function, one gets that using time-invariant parameters θ for each time slot will cause severe fluctuations for the update of θ. Note that the reinforcement learning optimization algorithm associated with time-varying parameters θ^t for each time slot t can reduce the concussion in the process of update of θ^t. It is also worth pointing out that using time-varying parameter is also an efficient way when there exist same FPOC in different time slots.

For the purpose of considering the effect of different ε in the ε-greedy policy, we take fixed $\varepsilon = 0.1$, $\varepsilon = 0.2$ and $\varepsilon = 0.3$ and also take $\varepsilon = \frac{1}{k}$, $\varepsilon = \frac{10}{k}$ and $\varepsilon = \frac{100}{k}$ which decreases gradually such that the operating policy is greedy limit with infinite exploration (GLIE) in ε-greedy. Fig. 8 shows the evolution of total generation cost of each updated policy through 10000 times training. As shown in Fig. 8, distributed reinforcement learning optimization algorithm yields a favorable policy when taking $\varepsilon = 0.3$.

Remark 6: It can be seen from the results given in Example 1 that the exploration in the distributed reinforcement learning optimization algorithm is very important as the number of FPOC is infinite in each time slot.

TABLE II: Parameters of units

Unit number	a_i	b_i	c_i	$\underline{p}_i$	$\overline{p}_i$	p_i^R
1	0.0072	5.56	30	60	339.69	50
2	0.0168	4.32	25	25	479.10	50
3	0.0216	6.60	25	28	290.4	50
4	0.0141	7.90	16	40	306.34	50
5	0.0273	7.54	6	35	593.80	50
6	0.0054	3.28	54	29	137.19	50
7	0.0159	7.31	23	45	595.40	50
8	0.0189	2.45	15	56	162.17	50
9	0.0084	7.63	20	12	165.1	50
10	0.0138	4.76	12	30	443.41	50

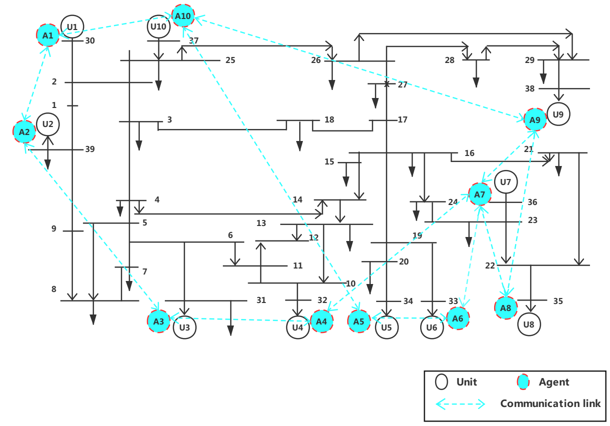

Fig. 9: IEEE 39-bus system.

Example 2. We consider the IEEE-39 bus system with 10 units. The communication network of these agents, which is described by the blue lines in Fig. 9, is undirected and connected. The cost function of each unit i is determined as $F_i(p_i) = a_i p_i^2 + b_i p_i + c_i$, where the coefficients are shown in Table II together with the minimum and maximum power generation of each unit. In this Example, we consider the DED problem in five time slots. The power demand D_t is assumed to be 1500, 1600, 1700, 1800, and 1900 (MV) for time slot 1, 2, 3, 4, and 5, respectively. At first, considering the object function is quadratic convex function and the feasible set is also convex set. We use the distributed optimization algorithm based on multipliers splitting method to find the exact optimal solution at time slot 1, 2, 3, 4, and 5 in Fig. 10. However, we do not know the form of the cost functions and the exact parameters in cost functions of units actually. Under this premise, we use the distributed reinforcement optimization algorithm to find the optimal policy. The Fig. 11 shows that the evolution of operating policy produced by distributed reinforcement learning optimization algorithm after 1087 times training in this Example. The exact optimal solution and the operating policy after 1087 times training are respectively shown in Table III and Table IV. The error between exact optimal cost and the operating policy cost is less than 4% of exact optimal cost. In contrast to the ED problem studied in [34], the DED problem under consideration is more difficult as the ramp-rate limit in each time slot.

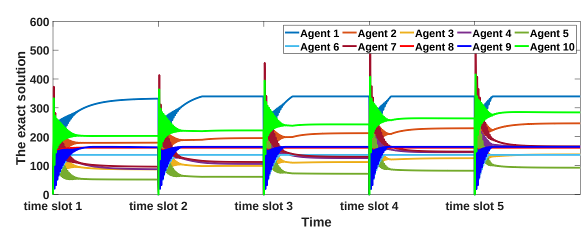

Fig. 10: The exact optimal solution.

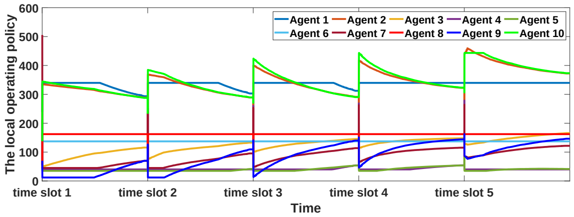

Fig. 11: The evolution of policy after 1087 times training.

TABLE III: Exact optimal solution for all time.

Agent	P_1^*	P_2^*	P_3^*	P_4^*	P_5^*
1	332.01	339.68	339.69	339.69	339.69
2	179.47	195.27	212.33	229.38	246.42
3	87.01	99.12	112.38	125.64	138.90
4	87.58	105.82	126.13	146.45	166.77
5	51.78	61.27	71.79	82.31	92.82
6	137.18	137.18	137.19	137.19	137.19
7	96.35	112.44	130.48	148.53	166.58
8	162.16	162.16	162.17	162.17	162.17
9	163.43	165.09	165.10	165.10	165.10
10	202.98	221.90	242.71	263.52	284.32

TABLE IV: The operating policy after 1087 times training.

Agent	$\pi(D_1)$	$\pi(D_2)$	$\pi(D_3)$	$\pi(D_4)$	$\pi(D_5)$
1	293.62	302.95	312.25	339.69	339.69
2	288.83	289.10	291.46	322.48	372.48
3	115.78	132.80	145.35	147.90	165.72
4	40	40	54.01	54.25	40
5	35	41.41	53.54	53.73	41.41
6	137.19	137.19	137.19	137.19	137.19
7	69.84	95.43	113.71	115.26	121.75
8	162.17	162.17	162.17	162.17	162.17
9	69.91	109.37	139.93	143.96	146.25
10	287.64	289.55	290.36	323.31	373.31

VI. Conclusion and Future Work

In this paper, we have formulated a DED problem with a little a prior information of the generation cost functions in smart grid. To solve the DED problem, we combined the state-action-value function approximation and the distributed optimization algorithm based on multipliers splitting to get a distributed reinforcement learning optimization algorithm. Each step in the proposed algorithm is fully distributed. Theoretical analysis as well as case studies have been presented to demonstrate the effectiveness of these proposed algorithms.

With respect to future works, the case that the total power demand D_{t+1} is decided by the feasible power output P_t at time slot t should be considered. Some constraints such as energy storage can be also considered in the future.

Appendix

A. Preliminaries on Convex Analysis

The following definitions and properties about convex set, convex function and projection can be found in [32].

A set $\Omega \subset \mathbb{R}^n$ is called convex set, if $\alpha x + (1-\alpha)y \in \Omega, \forall\, x, y \in \Omega, \forall\, \alpha \in [0,1]$. A function $f(\cdot):\ \Omega \to \mathbb{R}$ called to be convex function, if $f(\alpha x + (1-\alpha)y) \le \alpha f(x) + (1-\alpha)f(y),\ \forall\, x, y \in \Omega,\ \forall\, \alpha \in [0,1]$. If $f(\cdot):\ \Omega \to \mathbb{R}$ is differentiable at $x \in \Omega$, its gradient denoted by $\nabla f(x)$. $f(\cdot):\ \Omega \to \mathbb{R}$ is called differentiable on Ω, if $f(x)$ is differentiable at any point $x \in \Omega$. Denote $N_\Omega(x)$ as the normal cone of Ω at x, that is, $N_\Omega(x) = \{y : \langle y, x' - x\rangle \le 0, \quad \forall x' \in \Omega\}$.

For a closed set Ω, define the projection of x onto Ω is $\mathbb{P}_\Omega(x) = argmin_{y\in\Omega}\|x-y\|$. The common properties of projection as follows

$$\langle x - \mathbb{P}_\Omega(x), \mathbb{P}_\Omega(x) - x'\rangle \ge 0, \quad \forall x' \in \Omega,\ \forall x \in \mathbb{R}^n.$$

$$\|x-\mathbb{P}_\Omega(x)\|^2 + \|\mathbb{P}_\Omega(x)-x'\|^2 \le \|x-x'\|^2, \forall x' \in \Omega, \forall x \in \mathbb{R}^n.$$

Further, the normal cone $N_\Omega(x)$ can also be defined as $N_\Omega(x) = \{y : \mathbb{P}_\Omega(x+y) = x\}$.

B. Algebraic Graph Theory

The interaction topology of a system consisting of N units can be described by a graph. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph with the set of nodes (i.e., units) $\mathcal{V} = \{1, 2, \cdots, N\}$, the set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. A directed edge $e_{ij} \in \mathcal{E}$ represents that node i can get the information from node j, the graph $\mathcal{G}$ is said to be undirected when $e_{ij} \in \mathcal{E}$ if and only if $e_{ji} \in \mathcal{E}$. The in-degree neighbors $\mathcal{I}_i$ of node i is the set of nodes who can send their information to node i, i.e., $\mathcal{I}_i = \{j | e_{ij} \in \mathcal{E}\}$. A path is a sequence of distinct nodes in $\mathcal{V}$ such that any consecutive nodes in the sequence correspond to an edge of graph. The undirected graph is connected, if there exists at least one path between any two nodes. The adjacency matrix A has the entries $a_{ij} = 1$ if $e_{ij} \in \mathcal{E}$, and $a_{ij} = 0$, otherwise. The Laplacian matrix $L = [l_{ij}]_{N\times N}$ of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined as

$$l_{ij} = \begin{cases} -a_{ij}, & i \ne j, \\ \sum_{k=1,k\ne i}^{N} a_{ik}, & i = j. \end{cases}$$

Lemma 1: [7] Assume that the undirected graph $\mathcal{G}$ is connected, the first-order discrete-time protocol:

$$x[k+1] = x[k] - \epsilon L x[k], \tag{18}$$

where $\epsilon \in (0, \frac{1}{\max_i l_{ii}})$, achieves asymptotic average consensus, i.e., $\lim_{k\to\infty} x_i[k] = \frac{1}{N}\sum_{i=1}^{N} x_i[0],\ \forall\ i \in \{1, 2, \cdots, N\}$, where $x_i[k]$ is the i-th element of $x[k]$.

C. Reinforcement Learning

Reinforcement learning is a framework of the problem of learning from interaction to achieve a goal. The learner is called the agent, which interacts with the environment by getting some immediate reward as a consequence of taking an action. Reinforcement learning with discrete states and actions is usually formulated as a Markov Decision Process (MDP). The MDP is defined as a tuple $\{\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma\}$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions. $\mathcal{T}:\ \mathcal{S}\times\mathcal{A}\times\mathcal{S} \to [0,1]$ is the state transition function, $\mathcal{R}:\ \mathcal{S}\times\mathcal{A}\times\mathcal{S} \to \mathbb{R}$ represents the reward function, and $\gamma \in [0,1]$ is a discount factor. A policy $\pi:\ \mathcal{S}\times\mathcal{A} \to [0,1]$ is a probability distribution over actions for each state. The state-action-value function $q_\pi(s,a)$ under policy π is defined as the expected discount of the long term reward to the agent at the initial state s, taking action a and then following policy π. The aim of reinforcement learning is to find the optimal policy π^*. The policy π^* to maximize (minimize) cumulative reward is called to be the optimal policy, if $q_{\pi^*}(s,a) \geq q_\pi(s,a)$ $(or\ q_{\pi^*}(s,a) \leq q_\pi(s,a))$, $\forall s \in \mathcal{S},\ a \in \mathcal{A},\ \forall \pi$. In standard reinforcement learning problem, the environment is unknown, i.e., the transition function $\mathcal{T}$ and reward function $\mathcal{R}$ are unknown but static.

For large state and action spaces, function approximation in reinforcement learning is usually employed. Let $J(s,a,\theta)$ be an approximate function of the state-action-value function. We assume that $J(s,a,\theta)$ is differential function of parameter vector θ for all $s \in \mathcal{S},\ a \in \mathcal{A}$. The update of θ as follows

$$\theta \leftarrow \theta + \kappa\delta\nabla_\theta J(s,a,\theta),$$

where $\kappa \in (0,1)$ and δ is the one-step temporal difference (TD) error given by

$$\delta = r + \gamma J(s',a',\theta) - J(s,a,\theta),$$

where r is immediate reward after taking action a on state s, γ is the discount factor and (s',a') is state-action pair immediately after (s,a).

References

[1] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart grid-The new and improved power grid: A survey," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 944-980, Fourth Quarter. 2012.

[2] H. Farhangi, "The path of the smart grid," *IEEE Power Energy Mag.*, vol. 8, no. 1, pp. 18-28, Jan./Feb. 2010.

[3] P. Siano, "Demand response and smart grids-A survey," *Renew. Sustain. Energy Rev.*, vol. 30, pp. 461-478, 2014.

[4] V. C. Güngör *et al.*, "Smart grid technologies: Communication technologies and standards," in *IEEE Trans. Ind. Informat.*, vol. 7, no. 4, pp. 529-539, Nov. 2011.

[5] M. Pipattanasomporn, H. Feroze, and S. Rahman, "Multi-agent systems in a distributed smart grid: Design and implementation," in *Proc. PSCE,* Seattle, WA, USA, 2009, pp. 1-8.

[6] P. Gaj, J. Jasperneite, and M. Felser, "Computer communication within industrial distributed environmentłA survey," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 182-189, Feb. 2013.

[7] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520-1533, Sept. 2004.

[8] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48-61, Jan. 2009.

[9] A. Nedic, A. Ozdaglar and A.P. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Trans. Autom. Control*, vol. 55, no. 4, pp. 922-938, Apr. 2010.

[10] S. Yang, S. Tan, and J. Xu, "Consensus based approach for economic dispatch problem in a smart grid," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4416-4426, Nov. 2013.

[11] Z. Yang, J. Xiang, and Y. Li, "Distributed consensus based supply-demand balance algorithm for economic dispatch problem in a smart grid with switching graph," *IEEE Trans. Ind. Electron.*, vol. 64, no. 2, pp. 1600-1610, Feb. 2017,

[12] T. Yang *et al.*, "A distributed algorithm for economic dispatch over time-varying directed networks with delays," *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 5095-5106, Jun. 2017.

[13] A. Cherukuri, and J. Cortés, "Distributed generator coordination for initialization and anytime optimization in economic dispatch," *IEEE Trans. Control Netw. Syst.*, vol. 2, no. 3, pp. 226-237, Sept. 2015.

[14] P. Yi, Y. Hong, and F. Liu, "Initialization-free distributed algorithms for optimal resource allocation with feasibility constraints and application to economic dispatch of power systems," *Automatica*, vol. 74, no. 1, pp. 259-269, Dec. 2016.

[15] F. Guo, C. Wen, J. Mao, and Y.-D. Song, "Distributed economic dispatch for smart grids with random wind power," *IEEE Trans. Smart Grid*, vol. 7, no. 3, pp. 1572-1583, May 2016.

[16] G. Binetti, A. Davoudi, D. Naso, B. Turchiano, and F. L. Lewis, "A distributed auction-based algorithm for the nonconvex economic dispatch problem," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1124-1132, May 2014.

[17] P. Yi, Y. Hong, and F. Liu, "Distributed gradient algorithm for constrained optimization with application to load sharing in power system," *Syst. Control Lett.*, vol. 83, no. 9, pp. 45-52, 2015.

[18] G. Wen, X. Yu, Z. Liu, and W. Yu, "Adaptive consensus-based robust strategy for economic dispatch of smart grids subject to communication uncertainties," *IEEE Trans. Ind Informat.*, vol. 14, no. 6, pp. 2484-2496, Jun. 2018.

[19] W. Yu, C. Li, X. Yu, G. Wen, and J. Lü, "Economic power dispatch in smart grids: a framework for distributed optimization and consensus dynamics," *Sci. China-Inf. Sci.*, vol. 61, no. 1, pp. 1-16, 2018.

[20] C. Li, X. Yu, W. Yu, T. Huang, and Z.-W. Liu, "Distributed event-triggered scheme for economic dispatch in smart grids," *IEEE Trans. Ind. Informat.*, vol. 12, no. 5, pp. 1775-1785, Oct. 2016.

[21] X. He, J. Yu, T. Huang, and C. Li, "Distributed power management for dynamic economic dispatch in the multimicrogrids environment," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 4, pp. 1651-1658, Jul. 2019.

[22] W. Zhao, M. Liu, J. Zhu, and L. Li, "Fully decentralised multi-area dynamic economic dispatch for large-scale power systems via cutting plane consensus," *IET Generat., Transmiss. Distrib.*, vol. 10, no. 10, pp. 2486-2495, 2016.

[23] G. Chen, C. Li, and Z. Dong, "Parallel and distributed computation for dynamical economic dispatch," *IEEE Trans. Smart Grid*, vol. 8, no. 2, pp. 1026-1027, Mar 2017

[24] J. Qin, Y. Wan, X. Yu, F. Li, and C. Li, "Consensus-based distributed coordination between economic dispatch and demand response," *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 3709-3719, Jul. 2019

[25] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction.* Cambridge, MA: MIT Press, 1998.

[26] H. Zhang, Q. Wei, and Y. Luo, "A novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear systems via the greedy HDP iteration algorithm," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 937-942, Aug. 2008.

[27] H. Zhang, H. Jiang, Y. Luo, and G. Xiao, "Data-driven optimal consensus control for discrete-time multi-agent systems with unknown dynamics using reinforcement learning method," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 4091-4100, May 2017

[28] B. Wang, D. Zhao, C. Alippi, and D. Liu, "Dual heuristic dynamic programming for nonlinear discrete-time uncertain systems with state delay," *Neurocomputing*, vol. 134, pp. 222-229, 2014.

[29] K. G. Vamvoudakis and F. L. Lewis, "Online solution of nonlinear twoplayer zero-sum games using synchronous policy iteration," *Int. J. Robust Nonlin. Control*, vol. 22, no. 13, pp. 1460-1483, 2012.

[30] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," arXiv preprint arXiv:1312.5602, 2013.

[31] V. Mnih, K. Kavukcuoglu, D. Silver, et al, "Human-level control through deep reinforcement learning," *Nature.* 518(7540): 529, 2015.

[32] R. T. Rockafellar, *Convex Analysis*. Princeton, NJ, USA: Princeton Univ. Press, 1970.

[33] A. J. Wood and B. F. Wollenberg, *Power Generation, Operation, and Control*. New York: John Wiley & Sons, 2012.

[34] F. Li, J. Qin, and Y. Kang, "Multi-agent system based distributed pattern search algorithm for non-convex economic load dispatch in smart grid," *IEEE Trans. Power Syst.*, vol. 34, no. 3, pp. 2093-2102, May 2019.

Pengcheng Dai (S'19) received the B.S. degree in statistics from Yancheng Normal University, Yancheng, China, in 2016, and the M.S. degree in applied mathematics from Southeast University, Nanjing, China, in 2019, where he is currently pursuing the Ph.D. degree in applied mathematics.

His current research interests include distributed optimization and reinforcement learning.

Wenwu Yu (S'07-M'12-SM'15) received the B.Sc. degree in information and computing science and M.Sc. degree in applied mathematics from the Department of Mathematics, Southeast University, Nanjing, China, in 2004 and 2007, respectively, and the Ph.D. degree from the Department of Electronic Engineering, City University of Hong Kong, Hong Kong, China, in 2010. Currently, he is the Founding Director of Laboratory of Cooperative Control of Complex Systems and the Deputy Associate Director of Jiangsu Provincial Key Laboratory of Networked Collective Intelligence, an Associate Director in the Research Center for Complex Systems and Network Sciences, an Associate Dean in the School of Mathematics, and a Full Professor with the Young Endowed Chair Honor in Southeast University, China. Dr. Yu held several visiting positions in Australia, China, Germany, Italy, the Netherlands, and the USA. His research interests include multi-agent systems, complex networks and systems, disturbance control, distributed optimization, neural networks, game theory, cyberspace security, smart grids, intelligent transportation systems.

Dr. Yu severs as an Editorial Board Member of several flag journals, including *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *IEEE Transactions on Circuits and Systems II*, *Science China Information Sciences*, *Science China Technological Sciences*, etc.

He was listed by Clarivate Analytics/Thomson Reuters Highly Cited Researchers in Engineering in 2014 -2018. He publishes about 100 SCI journal papers with more than ten thousand citations. Moreover, he was awarded a National Natural Science Fund for Excellent Young Scholars in 2013, the National Ten Thousand Talent Program for Young Top-notch Talents in 2014, and the Cheung Kong Scholars Programme of China for Young Scholars in 2016. Dr. Yu is also the recipient of the Second Prize of State Natural Science Award of China in 2016.

Guanghui Wen (S'11–M'13–SM'17) received the Ph.D. degree in mechanical systems and control from Peking University, China, in 2012.

Currently, he is a Professor with the Department of Systems Science, School of Mathematics, Southeast University, Nanjing, China. His current research interests include cooperative control of multi-agent systems, analysis and synthesis of complex networks, cyber-physical systems, and resilient control.

Dr. Wen was the recipient of the Best Student Paper Award in the 6th Chinese Conference on Complex Networks in 2010. Dr. Wen was named a Highly Cited Researcher by Clarivate Analytics in 2018. As a co-advisor and a co-author, he has been a finalist for the ISCAS 2014 Best Student Paper Award. Moreover, he was awarded a National Natural Science Fund for Excellent Young Scholars in 2017. He is a reviewer for American Mathematical Review and is an active reviewer for many journals. He currently serves as an Editorial Board Member and an Associate Editor of the *IEEE Transactions on Systems, Man, and Cybernetics: Systems* and *Asian Journal of Control*.

Simone Baldi received the B.Sc. degree in electrical engineering, and the M.Sc. and Ph.D. degrees in automatic control systems engineering from the University of Florence, Italy, in 2005, 2007, and 2011, respectively. He is currently professor at the School of Mathematics, Southeast University, with a guest position at the Delft Center for Systems and Control, Delft University of Technology, where he was assistant professor. Previously, he held postdoctoral researcher positions at the University of Cyprus, and at the Information Technologies Institute, Centre for Research and Technology Hellas. He was awarded outstanding reviewer of Applied Energy (2016), Automatica (2017), and IET Control Theory and Applications (2018). Since March 2019, he is subject editor of International Journal of Adaptive Control and Signal Processing. His research interests include adaptive and learning systems with applications in networked control systems, smart energy and intelligent vehicle systems.