IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received April 14, 2018, accepted May 12, 2018, date of publication May 17, 2018, date of current version June 26, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2837663

An ADMM Based Distributed Finite-Time Algorithm for Economic Dispatch Problems

PENG LI AND JIANGPING HU, (Senior Member, IEEE)
School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

Corresponding author: Jiangping Hu (hjp_lzu@163.com)

This work was supported in part by the National Nature Science Foundation of China under Grants 61473061, 71503206, and 61104104, and in part by the Program for New Century Excellent Talents in University under Grant NCET-13-0091.

ABSTRACT In this paper, we present a distributed algorithm based on an alternating direction method of multipliers (ADMM), which is applied to solve economic dispatch problems (EDPs). First, with the help of two indicator functions, an EDP is transformed to an equivalent optimization problem with only equality constraint and thus can be dealt with ADMM. Second, a centralized algorithm is proposed to solve the transformed EDP, and furthermore, a distributed algorithm is designed with the help of finite-time average-consensus control strategy. Compared with the existing algorithms for EDP, the distributed algorithm can solve the economic dispatch problem on directed graphs. Moreover, the proposed algorithms can ensure that the generator constraints are satisfied during the whole computation process. Finally, some simulation results are also provided to demonstrate the effectiveness of the proposed algorithms.

INDEX TERMS Alternating direction method of multipliers, economic dispatch problem, finite-time consensus algorithm, smart grid.

I. INTRODUCTION

The economic dispatch problem (EDP) is very significant in electric power industry and has drawn much attention by communities of systems control and power systems. The EDP is essentially to minimize the total generation cost under power balance constraint and power generation constraint [1]. Till now, some algorithms have been proposed to solve this problem, such as the gradient search method, the classical lambda iteration method, dynamic, evolutionary algorithms, and heuristic techniques [1]–[4]. However, as pointed out in [5], these algorithms are centralized, which means that they generally require an information fusion center to process all the data from the whole network. Thus, these traditional algorithms can not work effectively for a very large-scale power systems because the information exchange may lead the information fusion center to be saturated quickly and the centralized infrastructures may be fragile for intentional attack.

To overcome the defects of centralized algorithms, many distributed optimization algorithms have been developed for EDPs. Generally, distributed algorithms divide an EDP of a large-scale power system into subproblems for individual sub-regions, which can adapt to any topological network and provide many advantages such as plug-and-play, enhanced robustness, reduction in communication cost, and better privacy and security. In recent years, many important distributed algorithms have been proposed to solve EDPs in a distributed fashion, varying from the Auxiliary Problem Principle (APP) strategies [6], the incremental cost consensus (ICC) based methods [7], [8] to Lagrangian relaxation (LR) based approaches [9]–[11]. In [7], an incremental cost consensus (ICC) strategy was proposed to solve an EDP with power balance constraint. The difference between the demand and the supply was used as a global feedback signal in the discrete-time update law of the incremental costs (ICs) for generators. A Laplacian-nonsmooth-gradient consensus algorithm was proposed in [12] to solve an EDP with power balance constraint. Furthermore, a dynamic average consensus algorithm was proposed in [13] to solve an EDP by using the estimate of the difference of the demand and the supply. Two classes of continuous-time projection-based gradient algorithms were proposed in [14] to solve EDP in an initialization-free and scalable manner. In [15], a θ-logarithmic barrier function was firstly used to reformulate the cost function and then a fast gradient based distributed optimization method was given to guarantee incremental cost consensus for generators. In [16], distributed incremental cost consensus algorithms were proposed for the switching communication network among generators. Guo *et al.* [17] presented a distributed economic dispatch based on projected gradient and finite time average consensus algorithms for smart grid systems. It is noted that all these algorithms are only applicable for undirected communication networks. It is essential to develop a distributed algorithm for directed communication networks because directed communication network has lower cost than undirected ones. In [18], a distributed bisection algorithm was proposed for a strongly connected directed network. Yang *et al.* [19] presented a distributed dynamic consensus algorithm for incremental costs to solve an EDP with minimum time steps.

Till now, EDPs have been solved mainly by using distributed gradient method or distributed sub-gradient method. Recently, a distributed algorithm based on alternating direction method of multipliers (ADMM) has been extended to solve distributed estimation problems as well [9], [10], [20]. The ADMM is an algorithm that decomposes an original problem into smaller subproblems which are easier to be handled, and then sequentially solves these subproblems at each iteration. Some ADMM based solution strategies have been proposed for distributed power systems and multi-agent systems. For example, an ADMM approach was developed to solve a semidefinite relaxation-based state estimation problems for AC power systems in [21]. Some distributd ADMM algorithms were respectively proposed in [22] and [23] to solve the optimal power flow problem for electrical grids. A distributed ADMM algorithm was developed in [24] for a general unconstrained optimization problem of a multi-agent network and its convergence rate was proved to be $O(1/k)$. In [25], by judiciously integrating the proximal minimization method with ADMM, the author proposed a distributed optimization algorithm where the constraints include the polyhedra constraints, which makes the subproblems efficiently solvable. In [26], a low-complexity algorithm was presented for problems with complicated structures or large dimensions and used an inexact step for each ADMM update to perform cheap computation. To our knowledge, there are very few results using ADMM to solve EDP. For example, an ADMM based consensus algorithm was proposed to solve an EDP with time-varying power demand for generators with undirected communication topology in [28]. A distributed algorithm based on alternating direction method of multipliers is proposed to address economic dispatch problem with general general cost functions [29].

Furthermore, it is noted that most distributed optimization algorithms require a long enough time for the convergence. Fortunately, some finite-time consensus control strategies were proposed in [27] and [30] to ensure that consensus can be reached in a finite time independently with initial states. Based on some existing results on ADMM and finite-time consensus algorithms, this paper presents two ADMM based energy dispatch algorithms in a centralized and decentralised fashion, respectively, for the economic dispatch problem. The main contributions of this paper are as follows. First, an ADMM solution strategy is introduced to solve an EDP by defining two indicator functions to transform the EDP such that ADMM can handle. Second, the communication topology among generators is a strongly connected directed network, in which the convergence analysis of energy dispatch algorithms is more challenging than that in undirected networks. Third, a distributed ADMM based consensus algorithm is proposed to compute the the optimal Lagrange multiplier in finite-time. The convergence of the proposed algorithm is analyzed theoretically. Moreover, the proposed algorithm can ensure that the generator constraints hold during the whole computation procedure. To the best of our knowledge, generator constraints can only be guaranteed at the final stage for most EDP solution strategies.

This paper is organized as follows. Section II presents some notions on graph theory and the formulation of the economic dispatch problem. In section III, a centralized ADMM based solution strategy is firstly proposed and then a distributed algorithm is designed by using a finite-time consensus strategy. Section IV gives some simulation results to demonstrate the effectiveness of the proposed algorithms. Section V presents some conclusion remarks and future research directions.

II. PRELIMINARIES AND PROBLEM FORMULATION

In this section, some notations in algebraic graph theory [31] are introduced for distributed systems. Then an economic dispatch problem (EDP) is firstly formulated for a power system.

A. GRAPH THEORY

The power system consisting of many generators can be described by a directed graph. A directed graph can be defined as $G = (V, E, P)$, where $V = \{1, 2, \cdots, N\}$ is the vertex set, $E \subseteq V \times V$ is the set of directed edge and P is a weighted and nonnegative adjacency matrix. If node i can receive information from node j, the edge e_{ji} belongs to the set E and $p_{ij} > 0$, $j \neq i$. A path between node v_j and node v_i is a sequence of edges $(v_j, v_{j1}), (v_{j1}, v_{j2}) \cdots (v_{jm}, v_i)$. A digraph is said to be strongly connected if there is a path from v_j to v_i for any pair nodes (v_j, v_i).

The nodes which can receive information from node v_i are called out-neighbors of node v_i and belong to the set $N_i^- = \{v_j \in V | e_{ij} \in E\}$. The cardinality of N_i^- is called the out-degree of a node v_i and is denoted by $d_i^- = |N_i^-|$. The nodes that regard node i as an out-neighbor are called the in-neighbors of node i and are denoted as $N_i^+ = \{v_j \in V | e_{ji} \in E\}$. The in-degree of node i is $d_i^+ = |N_i^+|$. If $d_i^+ = d_i^-$, for $i = 1, \cdots, N$, a digraph is called a balanced graph. Matrix P is called a column(row) stochastic matrix if it is a square and nonnegative matrix whose columns (rows) sum is 1. Matrix P is called a double stochastic matrix if it is not only column stochastic but also row stochastic.

In this paper, we only consider the digraph satisfying the following assumption.

Assumption 1: The graph $G = (V, E, P)$ is strongly connected. Each node knows its out-degree d_i^- and sets the weights on its self-link and outgoing links to be $1/(1 + d_i^-)$. Hence, the weight matrix P is nonnegative, column stochastic, and has entries $p_{ji} = 0$ only if the associated edge $e_{ij} \notin E$.

B. ECONOMIC DISPATCH PROBLEM

We assume that there are N power generators in the power system. The cost function for each generator i is denoted by $f_i(x_i)$, where $x_i \in R$ is the output power of generator i. The EDP can be expressed as,

$$\min f(\boldsymbol{x}) = \sum_{i=1}^{N} f_i(x_i), \tag{1}$$

subject to the following two constraints,

(1) Generator constrains

$$x_i^{min} \le x_i \le x_i^{max}, \tag{2}$$

where the x_i^{min} and x_i^{max} are the lower and upper bounds of the i generator capacity, respectively.

(2) Supply-demand balance constraints

$$\sum_{i=1}^{N} x_i = P_d \tag{3}$$

where the constant P_d satisfies $\sum_{i=1}^{N} x_i^{min} \le P_d \le \sum_{i=1}^{N} x_i^{max}$. In this paper, the cost functions $f_i(\cdot)$ satisfy the assumption 2.

Assumption 2: For each $i \in \{1, \ldots, N\}$, the cost function $f_i(\cdot) : R_+ \to R_+$ is strictly convex and continuously differentiable, where R_+ denotes the set of nonnegtive real numbers.

III. ADMM BASED ALGORITHMS FOR EDPS

In this section, a centralized solution strategy is firstly presented for the economic dispatch problem with the help of the Alternating Direction Method of Multipliers (ADMM). Then a distributed algorithm is proposed based on a finite-time consensus strategy.

In order to utilize the ADMM to solve the EDP, we reformulate the problem (1)-(3). We firstly define two convex sets with $\boldsymbol{x} = (x_1, x_2, \cdots, x_N)^T, \boldsymbol{y} = (y_1, y_2, \cdots, y_N)^T$

$$\Omega_1 = \{\boldsymbol{x} \in R^N | x_i^{min} \le x_i \le x_i^{max}, \quad \text{for } i = 1, \cdots, N\},$$
$$\Omega_2 = \{\boldsymbol{y} \in R^N | \sum_{i=1}^{N} y_i = P_d\}, \tag{4}$$

and two indicator function $g_1(\boldsymbol{x}),\ g_2(\boldsymbol{y})$ for the sets Ω_1 and Ω_2 as follows

$$g_1(\boldsymbol{x}) = \begin{cases} 0, & \text{if } \boldsymbol{x} \in \Omega_1, \\ +\infty, & \text{otherwise}, \end{cases}$$
$$g_2(\boldsymbol{y}) = \begin{cases} 0, & \text{if } \boldsymbol{y} \in \Omega_2, \\ +\infty, & \text{otherwise}. \end{cases} \tag{5}$$

Then the EDP (1)-(3) can be transformed to the follow optimization problem,

$$\min f(\boldsymbol{x}) + g_1(\boldsymbol{x}) + g_2(\boldsymbol{y}),$$
$$\text{s.t } \boldsymbol{x} - \boldsymbol{y} = 0, \tag{6}$$

where $\boldsymbol{x}, \boldsymbol{y} \in R^N$.

A. CENTRALIZED ADMM ALGORITHM

The alternating direction method of multipliers (ADMM) is a variant of the augmented Lagrangian scheme. Compared with the augmented Lagrangian scheme, it uses partial updates for the dual variables. An ADMM based centralized algorithm is given to update the primal variables x, y and the Lagrange multiplier $\boldsymbol{\lambda} \in R^N$ of the problem (6) as follows

$$\boldsymbol{x}(k+1) = \arg\min_{\boldsymbol{x}} L_\rho(\boldsymbol{x}, \boldsymbol{y}(k), \boldsymbol{\lambda}(k)), \tag{7}$$
$$\boldsymbol{y}(k+1) = \arg\min_{\boldsymbol{y}} L_\rho(\boldsymbol{x}(k+1), \boldsymbol{y}, \boldsymbol{\lambda}(k)), \tag{8}$$
$$\boldsymbol{\lambda}(k+1) = \boldsymbol{\lambda}(k) + \rho(\boldsymbol{x}(k+1) - \boldsymbol{y}(k+1)), \tag{9}$$

where the $L_\rho(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda})$ is the augmented Lagrangian function of the problem (6) and is given as follows

$$L_\rho(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + g_1(\boldsymbol{x}) + g_2(\boldsymbol{y}) + \boldsymbol{\lambda}^T(\boldsymbol{x} - \boldsymbol{y}) + \frac{\rho}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2. \tag{10}$$

To make the iteration schemes (7)-(9) converge to the optimal solutions, the following assumption and definition are made for the problem (6).

Assumption 3: [9] The unaugmented Lagrangian function $L_0 = f(\boldsymbol{x}) + g_1(\boldsymbol{x}) + g_2(\boldsymbol{y}) + \boldsymbol{\lambda}^T(\boldsymbol{x} - \boldsymbol{y})$ has a saddle point, i.e., there exists an optimal solution $(\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{\lambda}^*)$ such that

$$L_0(\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{\lambda}) \le L_0(\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{\lambda}^*) \le L_0(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda}^*) \tag{11}$$

holds for all $\boldsymbol{x} \in \Omega_1, \boldsymbol{y} \in \Omega_2, \boldsymbol{\lambda} \in R^N$.

Definition 1: The primal residual and dual residual are defined respectively as

$$\boldsymbol{r}(k) = \boldsymbol{x}(k) - \boldsymbol{y}(k), \quad \boldsymbol{s}(k) = -\rho(\boldsymbol{y}(k) - \boldsymbol{y}(k-1)). \tag{12}$$

Lemma 1: [9] If the Assumption 3 holds and $\rho > 0$, then the iteration (7)-(9) can converge to the optimal solution $\boldsymbol{x}^*$, $\boldsymbol{y}^*$, and the optimal Lagrange multiplier $\boldsymbol{\lambda}^*$ of the problem (6), with

$$\lim_{k\to\infty} \|\boldsymbol{r}(k)\| = 0 \text{ and } \lim_{k\to\infty} \|\boldsymbol{s}(k)\| = 0.$$

Now a specific computation procedure is given for the optimal solution with the centralized algorithm (7), (8) and (9). First, combining (7) and (10) yields

$$\begin{aligned} \boldsymbol{x}(k+1) &= \arg\min_{\boldsymbol{x}} L_\rho(\boldsymbol{x}, \boldsymbol{y}(k), \boldsymbol{\lambda}(k)) \\ &= \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}) + g_1(\boldsymbol{x}) + g_2(\boldsymbol{y}) \\ &\quad + \boldsymbol{\lambda}^T(k)(\boldsymbol{x} - \boldsymbol{y}(k)) + \frac{\rho}{2}\|\boldsymbol{x} - \boldsymbol{y}(k)\|_2^2 \\ &= \arg\min_{\boldsymbol{x}\in\Omega_1} f(\boldsymbol{x}) + \boldsymbol{\lambda}^T(k)(\boldsymbol{x} - \boldsymbol{y}(k)) \\ &\quad + \frac{\rho}{2}\|\boldsymbol{x} - \boldsymbol{y}(k)\|_2^2. \end{aligned}$$

To simplify the equation above, we choose $\boldsymbol{\sigma}(k) = \frac{\boldsymbol{\lambda}(k)}{\rho} = [\sigma_1, \cdots, \sigma_N] \in R^N$ and get the following equivalent equation,

$$\begin{aligned}\boldsymbol{x}(k+1) &= \arg\min_{\boldsymbol{x}\in\Omega_1} f(\boldsymbol{x}) + \frac{\rho}{2}\|\boldsymbol{x} - \boldsymbol{y}(k) + \boldsymbol{\sigma}(k)\|_2^2 \\ &= \arg\min_{\boldsymbol{x}\in\Omega_1} \sum_{i=1}^{N} f_i(x_i) + \frac{\rho}{2}\sum_{i=1}^{N}\big(x_i - y_i(k) + \sigma_i(k)\big)^2.\end{aligned}$$

Let $C_i(x_i) = f_i(x_i) + \frac{\rho}{2}\big(x_i - y_i(k) + \sigma_i(k)\big)^2$, then one has

$$\begin{aligned}&\min \sum_{i=1}^{N} C_i(x_i), \\ &s.t.\ x_i^{min} \le x_i \le x_i^{max}, \quad \text{for } i = 1, 2, \cdots, N. \quad (13)\end{aligned}$$

From the problem (13), the cost function of agent i is only relevant to its own constraint, so the new state $x_i(k+1)$ is given as follows

$$x_i(k+1) = \min\{\max\{\nabla C_i^{-1}(0), x_i^{min}\}, x_i^{max}\}, \quad (14)$$

for $i = 1, 2, \cdots, N$. ∇C_i^{-1} denotes the inverse function of ∇C_i which is the derivative of C_i. Based on the assumption 2, ∇f_i is continuous and strictly increasing and thus ∇C_i^{-1} is also continuous and strictly increasing. We can adopt the bisection method to compute the numerical solution of $x_i(k+1)$ within a finite steps. Moreover, we can see that the output power don't violate the generator constraint from the equation (14).

Similarly, from (8), (10) and $\boldsymbol{\sigma}(k) = \frac{\boldsymbol{\lambda}(k)}{\rho}$, one has

$$\begin{aligned}\boldsymbol{y}(k+1) &= \arg\min_{\boldsymbol{y}} L_\rho(\boldsymbol{x}(k+1), \boldsymbol{y}, \boldsymbol{\lambda}(k)) \\ &= \arg\min_{\boldsymbol{y}} g_2(\boldsymbol{y}) + \frac{\rho}{2}\|\boldsymbol{x}(k+1) - \boldsymbol{y} + \boldsymbol{\sigma}(k)\|_2^2 \\ &= \arg\min_{\boldsymbol{y}\in\Omega_2} \frac{\rho}{2}\|\boldsymbol{x}(k+1) - \boldsymbol{y} + \boldsymbol{\sigma}(k)\|_2^2. \quad (15)\end{aligned}$$

The problem (15) can be rewritten by the following form,

$$\begin{aligned}&\min \sum_{i=1}^{N} \frac{\rho}{2}\big(y_i - \varsigma_i(k)\big)^2, \\ &st.\ \sum_{i=1}^{N} y_i = P_d, \quad (16)\end{aligned}$$

where $\varsigma_i(k) = x_i(k+1) - \sigma_i(k)$. By using Lagrange multiplier method, we can transform the problem (16) to an unconstrained optimization problem, that is,

$$L = \sum_{i=1}^{N} \frac{\rho}{2}\big(y_i - \varsigma_i(k)\big)^2 + \eta(P_d - \sum_{i=1}^{N} y_i)$$

where η is the Lagrange multiplier associated with the equality constraint. On the basis of the KKT condition, one has

$$\begin{aligned}&\frac{\partial L}{\partial y_i} = \rho\big(y_i - \varsigma_i(k)\big) - \eta = 0 \quad \text{for } i = 1, \cdots, N, \\ &P_d - \sum_{i=1}^{N} y_i = 0. \quad (17)\end{aligned}$$

By solving the equations (17), we get the optimal Lagrange multiplier,

$$\eta^* = \rho\Big(\frac{P_d}{N} - \frac{\sum_{i=1}^{N}\varsigma_i(k)}{N}\Big). \quad (18)$$

Furthermore, according to the relationship $\rho\big(y_i - \varsigma_i(k)\big) - \eta = 0$, for $i = 1, \cdots, N$, the optimal solution $y_i(k+1)$ is given by

$$y_i(k+1) = \frac{\eta^*}{\rho} + \varsigma_i(k), \quad \text{for } i = 1, \cdots, N. \quad (19)$$

Substituting $x_i(k+1)$ and $y_i(k+1)$ into (9) yields

$$\begin{aligned}\lambda_i(k+1) = \lambda_i(k) + \rho(x_i(k+1) - y_i(k+1)), \\ \text{for } i = 1, \cdots, N. \quad (20)\end{aligned}$$

From Lemma 1, the optimal solution is thus obtained for the EDP (1)-(3).

Algorithm 1 Centralized Algorithm to Solve EDP

Input: $\rho, x_i^{min}, x_i^{max}, x_i(0), y_i(0), \lambda_i(0)$, for $i = 1, \ldots, N$
Output: Optimal power $\boldsymbol{x}^*$

```
for k = 0,1,··· do
    σ_i(k) = λ_i(k)/ρ,  for i = 1,··· ,N
    for i=1,2,··· ,N do
        x_i(k+1) = min{max{∇C_i^{-1}(0), x_i^min}, x_i^max}
    end for
    ς_i(k) = x_i(k+1) − σ_i(k),  for i = 1,··· ,N
    for i=1,2,··· ,N do
        η* = ρ(P_d/N − (Σ_{i=1}^N ς_i(k))/N)
        y_i(k+1) = η*/ρ + ς_i(k)
        λ_i(k+1) = λ_i(k) + ρ(x_i(k+1) − y_i(k+1))
    end for
end for
return x*
```

Remark 1: When the cost functions are quadratic, i.e., $f_i(x_i) = \alpha_i x_i^2 + \beta_i x_i + \gamma_i$, where the coefficient $\alpha_i > 0$, $\beta_i > 0$ and $\gamma_i > 0$, the update scheme (14) has a closed-form expression as

$$x_i(k+1) = \min\{\max\{\frac{\rho y_i(k) - \rho\sigma_i(k) - \beta_i}{2\alpha_i + \rho}, x_i^{min}\}, x_i^{max}\}.$$

Remark 2: From (18) and (19), the primal variable $y_i(k+1)$ is computed in a centralized way since computing the optimal value η^* requires the global information. In the next subsection, we will propose a distributed algorithm to compute the optimal Lagrange multiplier η^*.

B. DISTRIBUTED ADMM ALGORITHM

Based on Subsection III-A, we firstly divide the optimal Lagrange multiplier η^* into two parts so that each part can be calculated in a distributed manner. Secondly, distributed algorithms are designed for the two parts of η^* with the help of a finite-time consensus strategy proposed by [30].

Firstly, we define two notations as follows,

$$u^* = \frac{P_d}{N}, \quad v^* = \frac{\sum_{i=1}^{N} \varsigma_i(k)}{N}.$$

Then η^* can be expressed by

$$\eta^* = \rho(u^* - v^*).$$

We now propose two distributed algorithms to compute u^*, v^*, respectively. Firstly, a distributed algorithm is given to compute u^* as follows,

$$\begin{aligned} u_{i,1}(k+1) &= p_{ii}u_{i,1}(k) + \sum_{v_j \in N_i^+} p_{ij}u_{j,1}(k), \\ u_{i,2}(k+1) &= p_{ii}u_{i,2}(k) + \sum_{v_j \in N_i^+} p_{ij}u_{j,2}(k), \end{aligned} \tag{21}$$

where $p_{li} = 1/(1 + d_i^-)$ for $v_l \in N_i^- \cup \{v_i\}$ and, $p_{li} = 0$, otherwise. The initial values are given by $u_{i,1}(0) = y_i(0)$, $u_{i,2}(0) = 1$, for $i = 1, \cdots, N$, where $\sum_{i=1}^{N} y_i(0) = P_d$. Next, a distributed algorithm is proposed to compute v^* as follows,

$$\begin{aligned} v_{i,1}(k+1) &= p_{ii}v_{i,1}(k) + \sum_{v_j \in N_i^+} p_{ij}v_{j,1}(k), \\ v_{i,2}(k+1) &= p_{ii}v_{i,2}(k) + \sum_{v_j \in N_i^+} p_{ij}v_{j,2}(k), \end{aligned} \tag{22}$$

where the initial values are given by $v_{i,1}(0) = \varsigma_i(k)$, $v_{i,2}(0) = 1$ for $i = 1, \cdots, N$. The coefficients of linear iterations (22) are the same with the linear iterations (21).

In order to guarantee the convergence of the ADMM iterations (7)-(9), the time of computing $y_i(k+1)$ should be finite at each iteration. Next, we analyze the finite-time convergence of the distributed algorithms (21) and (22). Since the two algorithms have the same structure, we just analyze the convergence of the linear iteration (21) and the convergence analysis of the other algorithm is similar.

The convergence of the linear iteration (21) depends on the normalized kernel of Hankel matrices. Two square Hankel matrices with dimension $j \times j$ are constructed as follows,

$$H_i^j\{\overline{u}_{i,1}\} = \begin{bmatrix} \overline{u}_{i,1}(1) & \overline{u}_{i,1}(2) & \cdots & \overline{u}_{i,1}(j) \\ \overline{u}_{i,1}(2) & \overline{u}_{i,1}(3) & \cdots & \overline{u}_{i,1}(j+1) \\ \vdots & \vdots & \ddots & \vdots \\ \overline{u}_{i,1}(j) & \overline{u}_{i,1}(j+1) & \cdots & \overline{u}_{i,1}(2j-1), \end{bmatrix}$$

$$H_i^j\{\overline{u}_{i,2}\} = \begin{bmatrix} \overline{u}_{i,2}(1) & \overline{u}_{i,2}(2) & \cdots & \overline{u}_{i,2}(j) \\ \overline{u}_{i,2}(2) & \overline{u}_{i,2}(3) & \cdots & \overline{u}_{i,2}(j+1) \\ \vdots & \vdots & \ddots & \vdots \\ \overline{u}_{i,2}(j) & \overline{u}_{i,2}(j+1) & \cdots & \overline{u}_{i,2}(2j-1), \end{bmatrix}$$

where j denotes the iteration time, $\overline{u}_{i,1}(k) = u_{i,1}(k) - u_{i,1}(k-1)$, $\overline{u}_{i,2}(k) = u_{i,2}(k) - u_{i,2}(k-1)$, for $k = 1, 2, \cdots, 2j-1$.

At each iteration time j, we need to check the rank of the two Hankel matrices. Once the Hankel matrices $H_i^j\{\overline{u}_{i,1}\}$ and $H_i^j\{\overline{u}_{i,2}\}$ becomes singular at some time for arbitrary initial conditions $u_{i,1}(0)$ and $u_{i,2}(0)$ $(i = 1, 2, \cdots, N)$ except a set of initial conditions with Lebesgue measure zero, one can compute the normalized kernel $\xi_i = (\xi_i(0), \cdots, \xi_i(M_i - 1), 1)$ of $H_i^j\{\overline{u}_{i,1}\}$, where $M_i + 1$ is minimum degree of the monic polynomial of the Hankel matrix. From [30, Th. 1], one has

$$u^* = \lim_{k \to \infty} \frac{u_{i,1}(k)}{u_{i,2}(k)} = \frac{u_{M_i,1}^T \xi_i}{u_{M_i,2}^T \xi_i},$$

where $u_{M_i,1}^T = (u_{i,1}(0), u_{i,1}(1), \cdots, u_{i,1}(M_i))$, $u_{M_i,2}^T = (u_{i,2}(0), u_{i,2}(1), \cdots, u_{i,2}(M_i))$. Thus, the node i can compute u^* within $2(M_i + 1)$ steps. For all the nodes, u^* can be computed with at least $2(M_u + 1)$ steps. Similarly, the minimum times to compute v^* is given by $2(M_v + 1)$.

A main result is given to show that the optimal solution of the EDP can be computed in a distributed manner with the three iterations (7), (8) and (9).

Theorem 1: If $\rho > 0$ in (20) and Assumption 3 holds, then the update schemes given by the algorithms (7), (8) and (9) together with the distributed algorithms (21) and (22) can converge to the optimal solution $\boldsymbol{x}^*, \boldsymbol{y}^*$ and $\boldsymbol{\lambda}^*$.

Proof: We firstly show that the three updates $\boldsymbol{x}(k+1)$, $\boldsymbol{y}(k+1)$ and $\boldsymbol{\lambda}(k+1)$ are calculated in a distributed manner. Since η^* can be computed via the distributed algorithms (21) and (22), thus the update of $y_i(k+1)$ needs no more global information, which results in the distributed computation of $\boldsymbol{x}(k+1)$ and $\boldsymbol{\lambda}(k+1)$.

Next, we prove that the iterations (14), (19), (20) converge to the optimal solution of the EDP (1)-(3). From Lemma 1, under Assumption 1, the ADMM iteration (14), (19) and (20) converge to the optimal solution $\boldsymbol{x}^*, \boldsymbol{y}^*$ and $\boldsymbol{\lambda}^*$ with the primal residual $\boldsymbol{r}(k)$ and the dual residual $\boldsymbol{s}(k)$ satisfying $\lim_{k\to\infty} \|\boldsymbol{r}(k)\| = 0$, $\lim_{k\to\infty} \|\boldsymbol{s}(k)\| = 0$. The proof is completed. ■

Remark 3: From the update (14), $x_i(k)$ satisfies the local constraint $x_i^{min} \leq x_i \leq x_i^{max}$ during the whole iteration procedure, which is very meaningful in real energy dispatch for power systems.

Remark 4: In this paper, we firstly proposed a centralized algorithm to solve EDPs and then distributed algorithms are proposed to overcome the constraint that global information is needed to compute $y_i(k)$. Moreover, the distributed algorithms can guarantee a finite-time convergence. It is noticed that, even though the distributed algorithms are more robust, the convergence rate of the centralized algorithm is faster than that of the distributed one.

IV. SIMULATION RESULTS

In this section, we give some simulation results to validate the proposed algorithms. Suppose that the power system consists of 6 generators and the parameters of the generators are presented in Tabel 1. The demand power is given by $P_d = 150$. The network topology associated with the power system is illustrated in Figure 1.

Algorithm 2 Distributed Algorithm to Solve EDP

Input: weight matrix P and ρ, x_i^{min}, x_i^{max}, $x_i(0)$, $y_i(0)$, $\lambda_i(0)$, for $i = 1, \ldots, N$
Output: Optimal power $\boldsymbol{x}^*$
for k = 0,1,$\cdots$ **do**
  $\sigma_i(k) = \frac{\lambda_i(k)}{\rho}$, for $i = 1, \cdots, N$
  for i=1,2,$\cdots$,N **do**
    $x_i(k+1) = \min\{\max\{\nabla C_i^{-1}(0), x_i^{min}\}, x_i^{max}\}$
  end for
  for $k_1 = 0,1,\cdots,$ **do**
    For each node v_i, run the iteration (21) to get u^* with the initial value $u_{i,1}(0) = y_i(0)$, $u_{i,2}(0) = 1$, and store the successive values of $u_{i,1}(k)$ and $u_{i,2}(k)$, respectively.
    if k is odd **then**
      Compute the $u_{i,1}(k)$ and $u_{i,2}(k)$, and construct the Hankel matrix $H_i^j\{\overline{u}_{i,1}\}$ and $H_i^j\{\overline{u}_{i,2}\}$
      if $H_i^j\{\overline{u}_{i,1}\}$ and $H_i^j\{\overline{u}_{i,2}\}$ are singular **then**
        Store their first defective matrix, compute its normalized kernel ξ_i and compute the average consensus value u^*,

$$u^* = \frac{u_{M_i,1}^T \xi_i}{u_{M_i,2}^T \xi_i}$$

      end if
    end if
    Similar to computing the u^*, compute the average consensus value v^*.
  end for
  for i=1,2,$\cdots$,N **do**
    $\eta^* = \rho(u^* - v^*)$, $y_i(k+1) = \frac{\eta^*}{\rho} + \varsigma_i(k)$,
    $\lambda_i(k+1) = \lambda_i(k) + \rho(x_i(k+1) - y_i(k+1))$
  end for
end for
return x^*

TABLE 1. Generator parameters.

$Node$	α_i	β_i	γ_i	x_i^{min}	x_i^{max}
1	0.08	2	5	20	60
2	0.07	3.5	7	10	55
3	0.058	2.5	6.4	10	40
4	0.065	4	9.2	5	45
5	0.06	3	8.3	5	20
6	0.038	3.6	5.8	10	35

From the network topology, the weight matrix P is given for the distributed algorithms (21) and (22) as follows,

$$P = \begin{pmatrix} 1/2 & 0 & 0 & 0 & 0 & 1/3 \\ 1/2 & 1/2 & 0 & 0 & 0 & 1/3 \\ 0 & 1/2 & 1/3 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 1/2 & 0 & 0 \\ 0 & 0 & 1/3 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/3 \end{pmatrix}.$$

For the updates (7)-(9), the initial values are given by $\boldsymbol{x}(0) = [35, 25, 30, 28, 12, 20]^T$, $\boldsymbol{\lambda}(0) = \boldsymbol{0}$ and $\boldsymbol{y}(0) = \boldsymbol{x}(0)$,

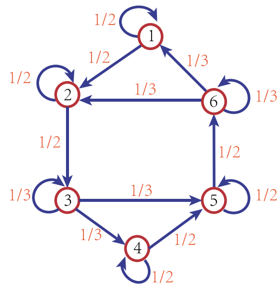


FIGURE 1. The network topology associated with the power system.

respectively, and the parameter ρ is selected as $\rho = 1$. For the distributed algorithms (21) and (22), the initial values are given by $u_{i,1}(0) = y_i^0$ and $v_{i,1}(0) = \varsigma_i(0)$, respectively.

In Figure 2, the output power of each generator converges to their optimal values, which are $x_1^* = 26.50$, $x_2^* = 19.58$, $x_3^* = 32.24$, $x_4^* = 17.26$, $x_5^* = 20$, $x_6^* = 34.42$ at the 45th step and the output powers of all the generators satisfy the generator constrains all the time. From Figure 3 and Figure 4, both the primal residual $\|\boldsymbol{r}(k)\|$ and dual residual $\|\boldsymbol{s}(k)\|$ converge to zero as the iteration proceeds, which indicates that the output powers can converge to the optimal powers. Figure 5 shows that the supply-demand balance is not satisfied at the initial stage but can be guaranteed after 10 iteration steps.

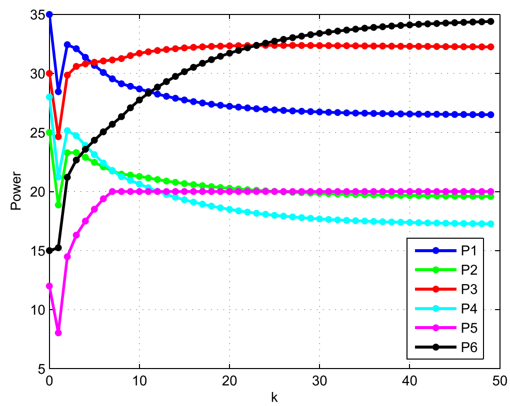


FIGURE 2. The evolution of the output powers.

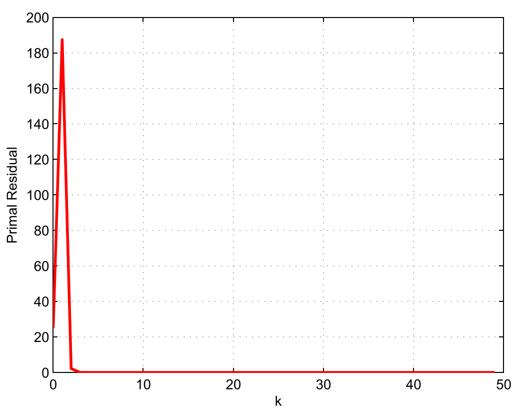


FIGURE 3. The evolution of the primal residuals.

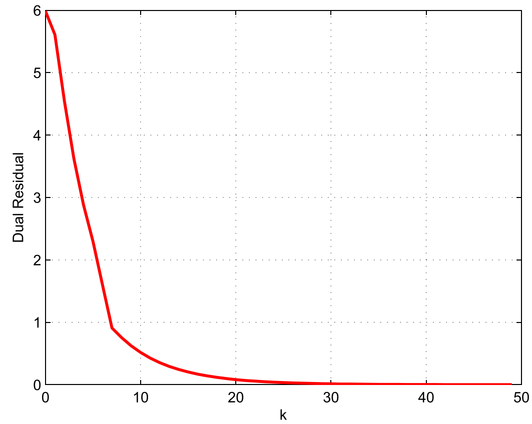


FIGURE 4. The evolution of the dual residuals.

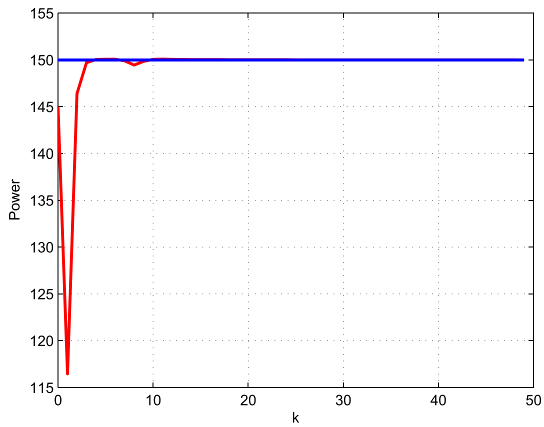


FIGURE 5. The evolution of the supply-demand balance.

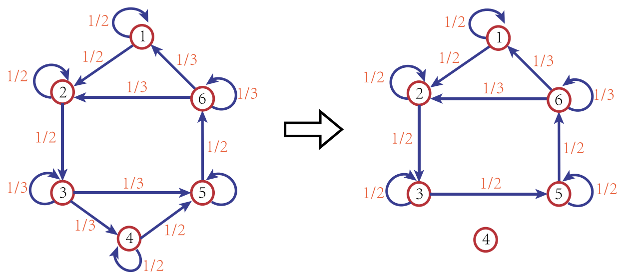


FIGURE 6. The communications topology of the generator plug-and-play case.

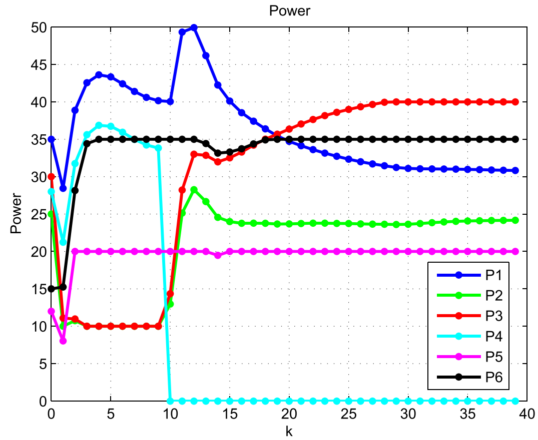


FIGURE 7. The output power under plug-and-play operation.

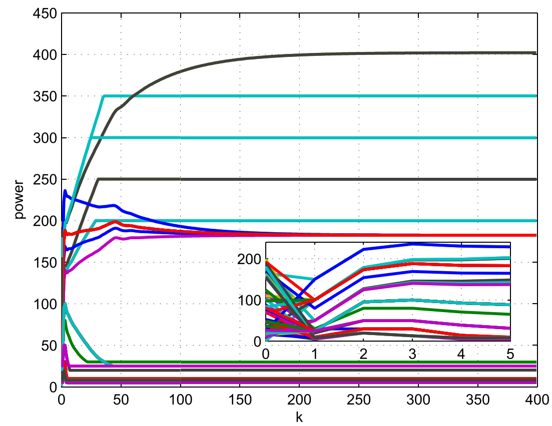


FIGURE 8. The evolution of the output powers of the IEEE 118-bus system.

In the following scenario, we study the proposed algorithm for the case of the plug-and-play. At the initial moment, all the generators support a total load of 150. The generator 4 is plugged out at the iteration $k = 10$, and thus the communication topology will be reconfigured, as illustrated in Figure 6. From Figure 7, the remaining generators still support the total load and the corresponding output powers of the generators are also satisfying the constraints. At the same time, the powers converge to the optimal solutions as time goes. Thus this experiment result demonstrates that the proposed algorithm is still effective for the case of plug-and-play.

We further apply the proposed algorithm for the IEEE 118-bus system, where there are 54 generators and the cost functions are quadratic, i.e., $f_i(x_i) = \alpha_i x_i^2 + \beta_i x_i + \gamma_i$, for $i = 1, 2, \ldots, 54$. The coefficients of the cost functions are given in the ranges $\alpha_i \in [0.0024, 0.0697]$, $\beta_i \in [8.3391, 37.6968]$ and $\gamma_i \in [6.78, 74.33]$. The communication topology is a directed cycle with nodes $1, \ldots, 54$. The power demand is 4600. Figure 8 illustrates the transient behaviors of the power allocation.

V. CONCLUSION

This paper proposed an ADMM based solution strategy for economic dispatch problems. The original EDP was firstly transformed to a form which ADMM can handle. Then a centralized algorithm was proposed to compute the optimal solution of the EDP, and furthermore, distributed algorithms were designed to compute the optimal Lagrange multiplier. The finite-time convergence of the distributed algorithms has been analyzed by checking the ranks of two Hankel matrices. Moreover, the proposed algorithms can guarantee that the generator constraints are satisfied during the whole iteration process. The future direction includes designing distributed algorithms to solve economic dispatch problems with additional physical constraints, such as transmission line loss and power flow and transmission line flow constraints.

REFERENCES

[1] A. J. Wood and B. F. Wollenberg, *Power Generation, Operation, and Control*. New York, NY, USA: Wiley, 1996.

[2] J. P. Zhan, Q. H. Wu, C. X. Guo, and X. X. Zhou, "Fast λ-iteration method for economic dispatch with prohibited operating zones," *IEEE Trans. Power Syst.*, vol. 29, no. 2, pp. 990–991, Mar. 2014.

[3] G. Abbas, J. Gu, U. Farooq, A. Raza, M. U. Asad, and M. E. El-Hawary, "Solution of an economic dispatch problem through particle swarm optimization: A detailed survey—Part II," *IEEE Access*, vol. 5, pp. 24426–24445, Nov. 2017.

[4] M. F. Zaman, S. M. Elsayed, T. Ray, and R. A. Sarker, "Evolutionary algorithms for dynamic economic dispatch problems," *IEEE Trans. Power Syst.*, vol. 31, no. 2, pp. 1486–1495, Mar. 2016.

[5] Y. Wang, S. Wang, and L. Wu, "Distributed optimization approaches for emerging power systems operation: A review," *Electr. Power Syst. Res.*, vol. 144, pp. 127–135, Mar. 2017.

[6] Y. Ren, Y. Tian, and H. Wei, "Parameter evaluation of auxiliary problem principle for large-scale multi-area economic dispatch," *Int. Trans. Elect. Energy Syst.*, vol. 24, no. 12, 1782–1790, Oct. 2013.

[7] Z. Zhang and M.-Y. Chow, "Convergence analysis of the incremental cost consensus algorithm under different communication network topologies in a smart grid," *IEEE Trans. Power Syst.*, vol. 27, no. 4, pp. 1761–1768, Nov. 2012.

[8] W. T. Elsayed and E. F. El-Saadany, "A fully decentralized approach for solving the economic dispatch problem," *IEEE Trans. Power Syst.*, vol. 30, no. 4, pp. 2179–2189, Jul. 2015.

[9] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[10] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, "Optimal parameter selection for the alternating direction method of multipliers (ADMM): Quadratic problems," *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 644–658, Mar. 2015.

[11] A. Kargarian, Y. Fu, and Z. Li, "Distributed security-constrained unit commitment for large-scale power systems," *IEEE Trans. Power Syst.*, vol. 30, no. 4, pp. 1925–1936, Jul. 2015.

[12] 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, Sep. 2015.

[13] A. Cherukuri and J. Cortés, "Initialization-free distributed coordination for economic dispatch under varying loads and generator commitment," *Automatica*, vol. 74, pp. 183–193, Dec. 2016.

[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, pp. 259–269, Dec. 2016.

[15] 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, May 2016.

[16] Z. Q. Yang, J. Xiang, and Y. J. 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.

[17] 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.

[18] H. Xing, Y. Mou, M. Fu, and Z. Lin, "Distributed bisection method for economic power dispatch in smart grid," *IEEE Trans. Power Syst.*, vol. 30, no. 6, pp. 3024–3035, Nov. 2015.

[19] T. Yang, D. Wu, Y. Sun, and J. Lian, "Minimum-time consensus based approach for power system applications," *IEEE Trans. Ind. Electron.*, vol. 63, no. 2, pp. 1318–1328, Feb. 2016.

[20] J. F. C. Mota, J. Xavier, P. Aguiar, and M. Püschel. (2011). "A proof of convergence for the alternating direction method of multipliers applied to polyhedral-constrained functions." [Online]. Available: https://arxiv.org/abs/1112.2295

[21] H. Zhu and G. B. Giannakis, "Power system nonlinear state estimation using distributed semidefinite programming," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 6, pp. 1039–1050, Dec. 2014.

[22] T. Erseghe, "Distributed optimal power flow using ADMM," *IEEE Trans. Power Syst.*, vol. 29, no. 5, pp. 2370–2380, Sep. 2014.

[23] Y. Wang, L. Wu, and S. Wang, "A fully-decentralized consensus-based ADMM approach for DC-OPF with demand response," *IEEE Trans. Smart Grid*, vol. 8, no. 6, pp. 2637–2647, Nov. 2017.

[24] E. Wei and A. E. Ozdaglar, "Distributed alternating direction method of multipliers," in *Proc. CDC*, Maui, HI, USA, Dec. 2012, pp. 5445–5450.

[25] T.-H. Chang, "A proximal dual consensus ADMM method for multi-agent constrained optimization," *IEEE Trans. Signal Process.*, vol. 64, no. 14, pp. 3719–3734, Jul. 2016.

[26] T. H. Chang, M. Hong, and X. Wang, "Multi-agent distributed optimization via inexact consensus ADMM," *IEEE Trans. Signal Process.*, vol. 63, no. 2, pp. 482–497, Jan. 2015.

[27] Y. Yuan, G.-B. Stan, L. Shi, M. Barahona, and J. Goncalves, "Decentralised minimum-time consensus," *Automatica*, vol. 49, no. 5, pp. 1227–1235, 2013.

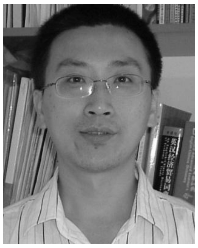
[28] H. Xing, Z. Lin, and M. Fu, "An ADMM + consensus based distributed algorithm for dynamic economic power dispatch in smart grid," in *Proc. 34th Chin. Control Conf.*, Hangzhou, China, Jul. 2015, pp. 9048–9053.

[29] G. Chen and Q. Yang, "An ADMM-based distributed algorithm for economic dispatch in islanded microgrids," *IEEE Trans. Ind. Informat.*, Dec. 2017, doi: 10.1109/TII.2017.2785366.

[30] T. Charalambous, Y. Yuan, T. Yang, W. Pan, C. N. Hadjicostis, and M. Johansson, "Distributed finite-time average consensus in digraphs in the presence of time delays," *IEEE Trans. Control Netw. Syst.*, vol. 2, no. 4, pp. 370–381, Dec. 2015.

[31] C. Godsil and F. R. Gordon, *Algebraic Graph Theory*. Berlin, Germany: Springer, 2013.

PENG LI received the B.S. degree in automation from the Henan University of Technology, China, in 2013, and the M.S. degree in control science and engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2016, where he is currently pursuing the Ph.D. degree with the School of Automation Engineering. His research interests include distributed optimization, multi-agent systems, and smart grid.



JIANGPING HU (SM'15) received the B.S. degree in applied mathematics and the M.S. degree in computational mathematics from Lanzhou University, China, in 2000 and 2004, respectively, and the Ph.D. degree in complex system modeling and control from the Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China, in 2007. He has held various faculty/research/visiting positions at the Royal Institute of Technology (KTH), Sweden, the City University of Hong Kong, Hong Kong, Sophia University, Japan, and the University of Western Sydney, Australia. He is currently a Professor with the School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu, China. His current research interests include distributed control and optimization, machine learning, and sensor networks.

Dr. Hu has been serving as an Associate Editor for the journal *Kybernetika* since 2016.