Distributed Economic Dispatch for Smart Grids With Random Wind Power

Fanghong Guo, Changyun Wen, *Fellow, IEEE*, Jianfeng Mao, *Member, IEEE*, and Yong-Duan Song, *Senior Member, IEEE*

***Abstract*—In this paper, we present a distributed economic dispatch (ED) strategy based on projected gradient and finite-time average consensus algorithms for smart grid systems. Both conventional thermal generators and wind turbines are taken into account in the ED model. By decomposing the centralized optimization into optimizations at local agents, a scheme is proposed for each agent to iteratively estimate a solution of the optimization problem in a distributed manner with limited communication among neighbors. It is theoretically shown that the estimated solutions of all the agents reach consensus of the optimal solution asymptomatically. This scheme also brings some advantages, such as plug-and-play property. Different from most existing distributed methods, the private confidential information, such as gradient or incremental cost of each generator, is not required for the information exchange, which makes more sense in real applications. Besides, the proposed method not only handles quadratic, but also nonquadratic convex cost functions with arbitrary initial values. Several case studies implemented on six-bus power system, as well as the IEEE 30-bus power system, are discussed and tested to validate the proposed method.**

***Index Terms*—Distributed optimization, economic dispatch (ED), finite-time consensus, plug-and-play, projected gradient.**

NOMENCLATURE

λ_i	Eigenvalue of the Laplacian matrix $\mathcal{L}$.
∇c_k	Gradient of the cost function c_k.
ζ_l	Stepsize at iteration l.
c_k	Cost function of the kth agent.
C_{pwj}	Cost coefficient of the underestimation of the availability of the jth wind turbine (WT).
C_{rwj}	Cost coefficient of the overestimation of the availability of the jth WT.
d_j	Cost coefficient of the jth WT.
$G_i(k)$	Neighbor table obtained by agent i at time step k.
P_d	Total load demand.
P_i	Scheduled power output of the ith thermal generator (TG).
$P_{X_k}[\cdot]$	Projection operator of the kth agent.
S_G	Set of TGs.
S_L	Set of loads.
S_W	Set of WTs.
W_j	Scheduled power output of the jth WT.
$w_{ii}(m)$	Update gains of the ith agent for its own states at iteration m.
$w_{ij}(m)$	Update gains of the ith agent for neighboring states at iteration m.
$W_{j,\mathrm{av}}$	Available power output of the jth WT.
$W_{r,j}$	Rated power output of the jth WT.
x	Global vector of scheduled generated power.
$x^\star$	Optimal value of x.
$x^k(l)$	Estimate of the kth agent at time step l.
X_k	Constraint set of the kth agent.
y_i^m	Exchanged information of the ith agent at iteration m.

Manuscript received October 30, 2014; revised January 22, 2015 and April 8, 2015; accepted May 12, 2015. This work was supported in part by the Major State Basic Research Development Program 973 under Grant 2012CB215202 and Grant 2014CB249200, in part by the National Natural Science Foundation of China under Grant 61134001, in part by Singapore's National Research Foundation under Grant NRF-CRP8-2011-03, and in part by the Energy Research Institute@NTU. Paper no. TSG-01082-2014. *(Corresponding author: Yong-Duan Song.)*

F. Guo is with the Energy Research Institute@NTU, Interdisciplinary Graduate School, Nanyang Technological University, Singapore 639798 (e-mail: fguo003@e.ntu.edu.sg).

C. Wen is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: ecywen@ntu.edu.sg).

J. Mao is with the School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore 639798 (e-mail: jfmao@ntu.edu.sg).

Y.-D. Song is with the Key Laboratory of Dependable Service Computing in Cyber Physical Society of Ministry of Education, Chongqing University, Chongqing 400044, China, and also with the College of Automation, Chongqing University, Chongqing 400044, China (e-mail: ydsong@cqu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSG.2015.2434831

I. INTRODUCTION

ECONOMIC dispatch (ED) is considered as one of well-studied and key problems in the power system research. It deals with the power allocation among the generators in an economic efficient way while meeting the constraints of total load demand as well as the generator constraints [1]. Some algorithms have been proposed to solve the ED problem, such as quadratic programming [2], λ-iteration [3], Lagrangian relaxation technique [4], and so on. However, all these methods are realized in a centralized way, i.e., collect the global information of all the generators and conduct the optimization in a central node. As pointed out in [5] and [6], such a centralized optimization is usually costly both in computation and communication when the power system becomes larger and larger. Moreover, they are unable to meet the plug-and-play requirement of recent smart grid system. When some

generators are newly installed or uninstalled, such centralized optimization may need to be redesigned [7], [8].

Recently in order to overcome the drawbacks mentioned above, distributed algorithms have been proposed [9]–[15]. Their main idea is to decompose the central optimization into several local optimizations. By letting each local optimization agent communicate with their neighbors, the global objective cost function can be minimized. Compared to centralized algorithms, a distributed one has the following major advantages: 1) less computational and communication cost; 2) plug-and-play property required by smart grid systems, which makes algorithm design more flexible; 3) robust to single-point-failures; and 4) easy and simple to design and implement as it only handles local information. In [10], ED problem is formulated as the incremental cost λ-consensus problem. The incremental cost of the ith generator λ_i is updated by combining λ_j from its neighbors with the global power mismatch. However, the calculation of power mismatch term requires the global information of each generator output and the total load demand. A similar λ-consensus algorithm is proposed in [11]. Both incremental cost and power mismatch are obtained in a distributed way through two consensus algorithms. In [12], a distributed ED algorithm with transmission losses is proposed, which is based on two parallel consensus algorithms. An auction-based consensus protocol is proposed in [13].

In addition to λ-consensus, a distributed gradient method has also been applied in the ED problem. In [14], an improved distributed gradient method is proposed to handle both equality and inequality constraints. However, the stepsize should be carefully chosen when the variables reach their constraint bounds. In [15], a fast distributed gradient method consisting of θ-logarithmic barrier function is proposed to solve ED problem. Note that the distributed gradient method requires that the initial values should be carefully allocated to meet the equality constraint.

Recently, renewable energy generators have been integrated to power systems to deal with the energy and environmental challenges. Among various kinds of renewable energy generators, WT is widely developed for the advantages such as free availability and environmental friendliness of wind energy as well as maturity of turbine techniques [16], [17]. Hence, the ED problem needs to be reformulated not only considering the conventional TGs but also the renewable energy generators such as WTs. There are mainly two problem formulations and approaches to handle the ED with random wind power. One is based on the stochastic programming strategies, where only TG cost function is minimized and the wind power is considered as the stochastic constraint appearing in the equality constraint [18], [19]. The other is based on a deterministic model, where the overestimation and underestimation cost of the wind power is proposed [20]–[22].

In this paper, we consider and follow the latter one, where ED for a smart grid system (shown in Fig. 1) consisting of conventional TGs, WTs as well as loads is considered. To ensure high utilization of the intermittent wind power, energy storage systems (ESSs) are always cooperatively integrated with WTs [23]. Note that the quadratic cost function is assumed in most existing ED problem formulations.

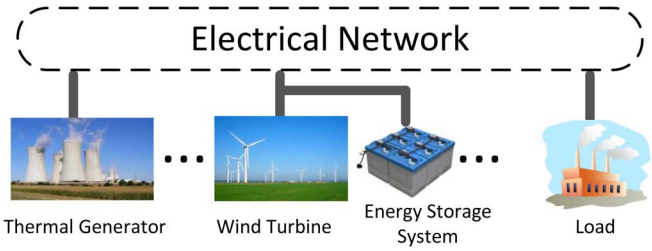


Fig. 1. Smart grid system.

However, when a WT and ESSs are included, their cost functions are not quadratic any more [20]. Hence some methods mentioned above may fail to work. In this paper, a distributed ED strategy based on projected gradient and finite-time average consensus algorithm (FACA) is proposed to solve this new ED problem. Our idea is to let each local agent iteratively estimate a solution of the optimization problem in a distributed manner by using its own and also available information from its neighbors. It is theoretically ensured that the estimated solutions of all the agents converge to the optimal solution of the problem. Several case studies implemented on a six-bus power system as well as an IEEE 30-bus power system are discussed and tested to validate the proposed method.

Besides the main advantages mentioned earlier in comparing with centralized approaches, our proposed method has some additional advantages over existing distributed schemes, as summarized below.

1) With the proposed method, private confidential information such as gradient or incremental cost is only known by each individual agent and is not used as communication information, which makes more sense in real applications.
2) Compared to λ-consensus algorithm, the cost function is not restricted to be quadratic. Our method can handle ED problem with nonquadratic convex cost function, such as that of WT. Compared to distributed gradient method, the initial values of our proposed method can be arbitrary, thus are not required to meet the stringent equality constraint.

The remainder of this paper is organized as follows. In Section II, we formulate the ED problem considering the penetration of WT and ESSs. Some preliminary knowledge including graph theory and finite-time average consensus (FAC) is introduced in Section III. In addition, to meet the requirement of distributed implementation architecture, a new algorithm is proposed to determine the graph topology and total load demand. Section IV presents the main results of proposed distributed ED strategy. Several cases studies are illustrated to show the effectiveness of proposed method in Section V. This paper is concluded in Section VI.

II. Problem Formulation

Mathematically speaking, the objective of traditional ED problem is to minimize the total generation cost subject to the demand supply constraint as well as the generator constraints [24]. In this paper, we consider a ED model which involves random wind power. The main goal of ED is to

minimize the system cost consisting of both TGs and WTs, which is given by [20]

$$C(P_i, W_j) = \sum_{i \in S_G} f_i(P_i) + \sum_{j \in S_W} g_j(W_j) \tag{1}$$

where S_G and S_W are the sets of TGs and WTs, respectively, P_i, W_j are the scheduled power output of the ith TG, $i \in S_G$ and the jth WT, $j \in S_W$.

The cost of conventional TG is usually approximated by a quadratic function [24]

$$f_i(P_i) = \alpha_i P_i^2 + \beta_i P_i + \gamma_i \tag{2}$$

where α_i, β_i, and γ_i are the cost coefficients of the ith TG.

In TG, the scheduled and generated power outputs are always the same. However, due to the random nature of wind speed, the available generated power $W_{j,\mathrm{av}}$ at the jth WT is a random variable, which may be different from the scheduled power W_j. Thanks to the integration of ESSs into the WTs, the total output of WT unit can be guaranteed to be equal to the scheduled one. For example, if the scheduled power output W_j is greater than $W_{j,\mathrm{av}}$, then the ESS can compensate the mismatch; if the scheduled power output W_j is less than $W_{j,\mathrm{av}}$, then the WT should clamp its output to W_j and the ESSs can be charged by the surplus wind power. In order to characterize the cost of the WT, the overestimation and underestimation cost has been proposed [20], [21]. Referring to [21], the overall cost for jth WT can be expressed as

$$g_j(W_j) = d_j W_j + C_{pwj} E(Y_{ue,j}) + C_{rwj} E(Y_{oe,j}) \tag{3}$$

where $d_j W_j$ is a linear cost function for wind power generation with d_j being the cost coefficient or the "price" of the jth WT, the terms $C_{pwj}E(Y_{ue,j})$ and $C_{rwj}E(Y_{oe,j})$ are the underestimation and overestimation costs with C_{pwj}, C_{rwj} being the cost coefficients, respectively, which are explained in detail in Appendix A.

Then considering both generator constraints and demand supply constraint, the ED problem with random wind power can be formulated as

$$\begin{cases} \min\limits_{P_i, W_j} & C(P_i, W_j) \\ \text{s.t.} & \sum_{i \in S_G} P_i + \sum_{j \in S_W} W_j = P_d \\ & P_i^{\min} \le P_i \le P_i^{\max}, i \in S_G \\ & 0 \le W_j \le W_{r,j}, j \in S_W \end{cases} \tag{4}$$

where $P_i^{\min}$ and $P_i^{\max}$ are the lower and upper bounds of the ith TG, $W_{r,j}$ is the rated wind power of the jth WT, and P_d is the total load demand satisfying $\sum_{i \in S_G} P_i^{\min} \le P_d \le \sum_{i \in S_G} P_i^{\max} + \sum_{j \in S_W} W_{r,j}$.

Note that the cost coefficient α_i of the TG is usually positive, which implies that (2) is a convex function. Meanwhile, it is also proved in [21] that (14) and (16) in Appendix A are also convex with respect to W_j, which yields the convexity of the WT cost function (1). Also the constraints are convex, thus the ED problem described in (4) can be considered as a convex optimization problem.

Remark 1: Compared to the ED problem formulated in [10]–[15], the cost function in (4) is not quadratic any more, which implies that the designed distributed methods in [10]–[13] may fail to work. In this paper, a new distributed optimization method will be introduced to solve the ED problem formulated in (4).

Suppose there are $N = n_G + n_W$ generators, consisting of $n_G = |S_G|$ TGs and $n_W = |S_W|$ WTs, and M loads in a smart grid system, shown in Fig. 1. We first treat every generator and load as an "agent," and each agent is assigned a unique ID. Without the loss of generality, we assign the first N agents as the TGs and WTs and denote their estimated generated power in a global vector as $x = \begin{bmatrix} x_1 & \cdots & x_{n_G} & \cdots & x_N \end{bmatrix}^T$. The cost function c_k and constraint set X_k of agent $k, k \in S_G \cup S_W$ are denoted as follows, respectively:

$$c_k(x) = \begin{cases} f_k(x_k), k \in S_G, & k = 1, \ldots, n_G \\ g_k(x_k), k \in S_W, & k = n_G + 1, \ldots, N \end{cases} \tag{5}$$

$$X_k = \begin{cases} \begin{matrix} P_k^{\min} \le x_k \le P_k^{\max} \\ \sum_{i=1}^N x_i = P_d \end{matrix}, & k \in S_G, k = 1, \ldots, n_G \\ \begin{matrix} 0 \le x_k \le W_{r,k} \\ \sum_{i=1}^N x_i = P_d \end{matrix}, & k \in S_W, k = n_G + 1, \ldots, N. \end{cases} \tag{6}$$

Then, (4) can be reformulated as

$$\begin{cases} \min_x & \sum_{k=1}^N c_k(x) \\ \text{s.t.} & x \in \cap_{k=1}^N X_k. \end{cases} \tag{7}$$

Lemma 1: The ED problem formulated in (7) has an optimal solution $x^\star$.

Proof: As the constraint set X_k in (6) is compact, thus the intersection $X = \cap_{k=1}^N X_k$ is also compact. Besides, the function $c_k(x)$ is a continuous convex function in $\mathbb{R}^{N \times 1}$, which implies that $\sum_{k=1}^N c_k(x)$ is also a continuous function. It follows from the well-known extreme value theorem that the ED problem formulated in (7) has an optimal solution $x^\star, x^\star \in \cap_{k=1}^N X_k$. ■

Note that the optimal solution $x^\star$ is unknown to each agent and both cost function $c_k(x)$ and constraint X_k of generator $k, k \in S_G \cup S_W$ are only known by agent k itself. Our idea is that each agent estimates the optimal solution by using the available information of its neighboring agents and itself iteratively. Denoting the estimate of kth agent at time step l as $x^k(l)$, then our aim is to propose a scheme to achieve that $\lim_{l \to \infty} x^k(l) = x^\star$, $k = 1, \ldots, N$. In other words, the estimates of all agents reach consensus of the optimal solution asymptotically.

III. Preliminaries

In this section, we first give a summary of graph theory [25] and present a FACA [26]. Then a new algorithm called graph discovery is proposed to make FACA be realized in a totally distributed way. Finally, a distributed load demand discovery method is introduced to determine P_d in the constraint (6) by applying the modified FACA algorithm.

A. Graph Theory

A graph is defined as $\mathcal{G} = (\mathcal{V}, \xi)$, where $\mathcal{V} = \{1, \ldots, n\}$ denotes the set of vertices, $\xi \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges between two distinct vertices. If for all $(i, j) \subseteq \xi$, then $(j, i) \subseteq \xi$, we call $\mathcal{G}$ is undirected; otherwise it is called

directed graph. The set of neighbors of the ith vertex is denoted as $\mathcal{N}_i \triangleq \{j \subseteq \mathcal{V} : (i,j) \subseteq \xi\}$. The graph $\mathcal{G}$ is connected means that there exists at least one path between any two distinct vertices. The elements of the adjacency matrix $\mathbf{A}$ are defined as $a_{ij} = a_{ji} = 1$ if $j \subseteq \mathcal{N}_i$, otherwise $a_{ij} = a_{ji} = 0$. Clearly, $\mathbf{A}$ is a symmetric matrix with the diagonal elements $a_{ii} = 0$ for undirected graph. The Laplacian matrix of $\mathcal{G}$ is defined as $\mathcal{L} = \Delta - \mathbf{A}$, where Δ is called in-degree matrix and is defined as $\Delta = \mathrm{diag}(\Delta_i) \subseteq \mathbb{R}^{n\times n}$ with $\Delta_i = \sum_{j\in\mathcal{N}_i} a_{ij}$. It is well known that the Laplancian matrix $\mathcal{L}$ of a undirected graph has one distinct zero eigenvalue and all the others are positive, i.e., $\lambda_1 = 0, 0 < \lambda_2 \leq \cdots \leq \lambda_n$, where λ_2 is called algebraic connectivity, which is positive if and only if the undirected graph is connected [25].

B. Finite-Time Average Consensus Algorithm

In order to have finite-step consensus convergence, an FACA has been proposed [26]. Compared to the conventional average consensus algorithms such as in [27], this algorithm has the following advantages.

1) The consensus can be reached in finite steps.
2) It ensures all the agents reach consensus at the same time.

The general average consensus can be represented as

$$y_i^{m+1} = w_{ii}(m)y_i^m + \sum_{j\in\mathcal{N}_i} w_{ij}(m)y_j^m, \quad i = 1,\ldots,n \tag{8}$$

where y_i^m denotes the information of the ith agent at iteration m, $w_{ii}(m)$, and $w_{ij}(m)$ are the update gains of its own states and neighboring states at iteration m, respectively, $\mathcal{N}_i$ is the set of neighboring agents of the ith agent.

Lemma 2 [26]: Let $\lambda_2 \neq \lambda_3 \neq \cdots \neq \lambda_{K+1} \neq 0$ be the K distinct nonzero eigenvalues of the graph Laplacian matrix $\mathcal{L}$, $y_i, i = 1,\ldots,n$ in (8) can reach consensus in finite K steps, if the updating gains for agent i are chosen as

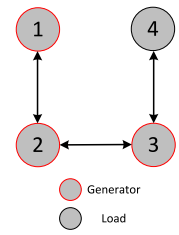
$$w_{ij}(m) = \begin{cases} 1 - \dfrac{n_i}{\lambda_{m+1}}, & j = i \\ \dfrac{1}{\lambda_{m+1}}, & j \in \mathcal{N}_i, \\ 0, & \text{otherwise} \end{cases} \qquad m = 1,\ldots,K \tag{9}$$

where $n_i = |\mathcal{N}_i|$, which is the number of the neighboring agents of agent i.

C. Distributed FACA

The FACA can reach consensus in finite steps, which is necessary for our proposed method developed in the next section. However, from (9), we know that the main limitation of the FACA is the assumption that each agent needs to know the nonzero eigenvalues of Laplacian matrix $\mathcal{L}$ of the whole communication graph, i.e., the whole graph topology, as a prior knowledge. This is very restrictive, as in practice each agent does not have the global information of whole graph topology such as the total generator number N, the total load number M, and the Laplacian matrix $\mathcal{L}$ from the beginning. In addition, these global information may change due to the addition and removal of certain agents. Clearly, this requirement results in the implementation of FACA nondistributed.

TABLE I
ILLUSTRATION EXAMPLE OF ALGORITHM 1

Step	Content	Communication Graph
$k=0$	$G_1(0) = \{ID(1)[ID(2)]\}$ $G_2(0) = \{ID(2)[ID(1), ID(3)]\}$ $G_3(0) = \{ID(3)[ID(2), ID(4)]\}$ $G_4(0) = \{ID(4)[ID(3)]\}$	
$k=1$	$G_1(1) = \{G_1(0), G_2(0)\}$ $G_2(1) = \{G_1(0), G_2(0), G_3(0)\}$ $G_3(1) = \{G_2(0), G_3(0), G_4(0)\}$ $G_4(1) = \{G_3(0), G_4(0)\}$	
$k=2$	$G_1(2) = \{G_1(0), G_2(0), G_3(0)\}$ $G_2(2) = \{G_1(0), G_2(0), G_3(0), G_4(0)\}$ $G_3(2) = \{G_1(0), G_2(0), G_3(0), G_4(0)\}$ $G_4(2) = \{G_2(0), G_3(0), G_4(0)\}$	
$k=3$	$G_1(3) = \{G_1(0), G_2(0), G_3(0), G_4(0)\}$ $G_2(3) = \{G_1(0), G_2(0), G_3(0), G_4(0)\}$ $G_3(3) = \{G_1(0), G_2(0), G_3(0), G_4(0)\}$ $G_4(3) = \{G_1(0), G_2(0), G_3(0), G_4(0)\}$	

To relax this requirement, we propose a new algorithm for each agent, named graph discovery to determine N, M, and $\mathcal{L}$ by themselves automatically. Similar to [28], we only impose the assumption that each agent i has been assigned a unique identifier $\mathrm{ID}(i)$, e.g., its IP address. Different type of agents has different type of IDs, for example

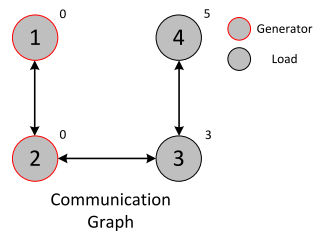
$$\mathrm{ID}(i) = \begin{cases} \mathrm{ID}_G(i), & i = 1,\ldots,N \\ \mathrm{ID}_L(i), & i = N+1,\ldots,N+M. \end{cases}$$

Algorithm 1 (Graph Discovery): Let $G_i(k)$ denote the neighbor table obtained by agent i, $i \in \mathcal{V}$ at time step k, which will be determined by the following steps.

1) At $k = 0$, each agent $i \in \mathcal{V}$ initializes the table as
$$G_i(0) = \big\{\mathrm{ID}(i)\big[\mathrm{ID}(j), j \in \mathcal{N}_i\big]\big\}$$
and sends this data to all its neighbors in $\mathcal{N}_i$.
2) At each step $k \geq 1$, agent i updates its table $G_i(k)$ as
$$G_i(k+1) = \bigcup_{j\in\mathcal{N}_i\cup\{i\}} G_j(k).$$
3) If $G_i(k) = G_i(k-1)$, then agent i stops exchanging information with its neighbors. Otherwise, go to step 2).
4) Let k_f be the first instant at which $G_i(k) = G_i(k-1)$, that is
$$k_f = \min\{k | G_i(k) = G_i(k-1)\}$$
then the total number of the agents $n = N + M = |G_i(k_f)|$.
5) Finally, the total generator number N can be extracted by counting the number of agents with ID $\mathrm{ID}_G(i)$, and the $n \times n$ Laplacian matrix $\mathcal{L}$ can be extracted from $G_i(k_f)$ according to the definition introduced in Section III-A, e.g., $\mathcal{L}$ can be extracted with the ith row being determined by $G_i(0)$ in $G_i(k_f)$.

An example with four agents to illustrate this algorithm is shown in Table I. For vertices 2 and 3, it takes $k_f = 2$ steps while for vertices 1 and 4, it takes $k_f = 3$ to discover the whole graph.

TABLE II
ILLUSTRATION EXAMPLE OF TOTAL LOAD DEMAND DISCOVERY

Step	Content	Communication Graph
$m=0$	$y^1(0)=0$ $y^2(0)=0$ $y^3(0)=3$ $y^4(0)=5$	
$m=1$	$y^1(1)=0$ $y^2(1)=5.1212$ $y^3(1)=1.2929$ $y^4(1)=1.5859$	
$m=2$	$y^1(2)=2.5606$ $y^2(2)=0.6465$ $y^3(2)=3.3535$ $y^4(2)=1.4394$	
$m=3$	$y^1(3)=2$ $y^2(3)=2$ $y^3(3)=2$ $y^4(3)=2$	$\mathcal{L}=\begin{bmatrix}1 & -1 & 0 & 0\\ -1 & 2 & -1 & 0\\ 0 & -1 & 2 & -1\\ 0 & 0 & -1 & 1\end{bmatrix}$

Note that this algorithm not only discovers the number of the agents but also the whole graph topology. By applying Algorithm 1 in the initial of the FACA, it can be realized in a totally distributed way.

D. Total Load Demand Discovery

Note that the total load demand P_d appears in the constraint set (6) of each agent. How to determine P_d in a distributed way is another problem to be handled in this section. Here, we propose to apply distributed FACA to find P_d for each agent. Let y^i be the communication state for the ith agent, and its initial value is defined as

$$y^i(0)=\begin{cases}0, i\in S_G\cup S_W, & i=1,\dots,N\\ P_d^i, i\in S_L, & i=N+1,\dots,N+M.\end{cases} \tag{10}$$

Lemma 3: The total load demand P_d can be determined by each agent i, $i\in S_G\cup S_W$ in finite K steps when using the FACA updating law (8) and (9), where K is defined in Lemma 2.

Proof: According to Lemma 2, $y^i(m), i=1,\dots,N+M$ will reach an average consensus in K steps, namely, $y^i(K)=y^j(K)=(\sum_{i=1}^{N+M}y^i(0)/N+M), \forall i,j=1,\dots,N+M$. Then the total load demand P_d can be obtained by each agent i, $i\in S_G\cup S_W$, that is

$$P_d=(N+M)y^i(K),\quad i=1,\dots,N. \tag{11}$$

■

An illustration example is shown in Table II. In this example, agents 1 and 2 are the generators while agents 3 and 4 are loads with the demand of 3 and 5, respectively. At $k=0$, each agent sets their initial value according to (10) as $y^1(0)=0, y^2(0)=0, y^3(0)=3$, and $y^4(0)=5$, respectively. By applying distributed FACA, they can reach a consensus $y^1(3)=y^2(3)=y^3(3)=y^4(3)=2$ in three steps and the total load demand P_d is obtained as $P_d=4*2=8$.

IV. DISTRIBUTED ECONOMIC DISPATCH

A. Distributed Projected Gradient Method

Motivated by the projection idea in [29] and constrained optimization in [30], we now propose a distributed projected gradient method (DPGM) to solve the ED problem formulated in (7). Different from existing distributed methods in [9]–[15], the communication information required here is the estimates of the scheduled generated power x^k, i.e., the estimated optimal solutions, of the local agent and its neighbors rather than the more private and confidential gradient or incremental gain. Recall that $x^k(l)$, $k=1,\dots,N+M$, denotes the estimate of the agent k at iteration l, which is an $N\times 1$ vector. As only generator agents estimate the power output, we define $x^k(l)=\begin{bmatrix}0 & \cdots & 0\end{bmatrix}^T$, $k=N+1,\dots,N+M$ for all load agents for any l. Unlike the distributed gradient method in [14] and [15], the initial value $x^k(0)$, $k=1,\dots,N$ is allowed to be arbitrary.

The kth agent updates its estimate by using the average information produced by distributed FACA, then taking a gradient step to minimize its own cost function c_k, and at last projecting the result on its constraint set X_k. This updating rule can be summarized as

$$\begin{cases} z_1^k(l)=w_{kk}(1)x^k(l)+\sum_{j\in\mathcal{N}_k}w_{kj}(1)x^j(l)\\ z_2^k(l)=w_{kk}(2)z_1^k(l)+\sum_{j\in\mathcal{N}_k}w_{kj}(2)z_1^j(l)\\ \qquad\vdots\\ z_K^k(l)=w_{kk}(K)z_{K-1}^k(l)+\sum_{j\in\mathcal{N}_k}w_{kj}(K)z_{K-1}^j(l)\\ z^k(l)=\dfrac{N+M}{N}z_K^k(l)\end{cases} \tag{12}$$

$$x^k(l+1)=P_{X_k}\Big[z^k(l)-\zeta_l\nabla c_k\Big(x^k(l)\Big)\Big] \tag{13}$$

where $w_{kk}(m)$, $w_{kj}(m)$, $m=1,\dots,K$ are the FACA updating gains determined in (9), $P_{X_k}[.]$ is the projection operator described in Appendix B, ζ_l is a stepsize at iteration l, and ∇c_k denotes the gradient of the cost function c_k.

Theorem 1: Let $\{x^k(l)\}$, $k=1,\dots,N$ be the estimates generated by the algorithm (12) and (13) and $X=\cap_{k=1}^N X_k$ be the intersection set. Then the sequence $\{x^k(l)\}$, $k=1,\dots,N$ converges to the optimal solution $x^\star$ with $x^\star\in X$, that is

$$\lim_{l\to\infty}x^k(l)=x^\star,\ \ k=1,\dots,N$$

if the stepsize ζ_l satisfies that $\sum_l\zeta_l=\infty$ and $\sum_l\zeta_l{}^2<\infty$.

Proof: According to Lemma 2, the average consensus process (12) can be reached in finite K steps if the update gains $w_{ii}(m)$ and $w_{ij}(m)$, $m=1,\dots,K$ are chosen as (9). This process is equal to $z^k(l)=1/N\sum_{j=1}^N x^j(l)$. Then result follows from the proof of [30, Proposition 5]. Due to the page limit, we omit the details here. ■

Remark 2: Compared to the projected subgradient algorithm in [30], our proposed DPGM is a fully distributed one. The average step (14) in [30] requires that each agent communicates with all the others in the entire system, which almost has the same communication cost as a centralized one. By applying the distributed FACA algorithm, the proposed DPGM can realize the same function with limited communication.

Remark 3: Compared to the existing methods for ED problem, the proposed DPGM has the

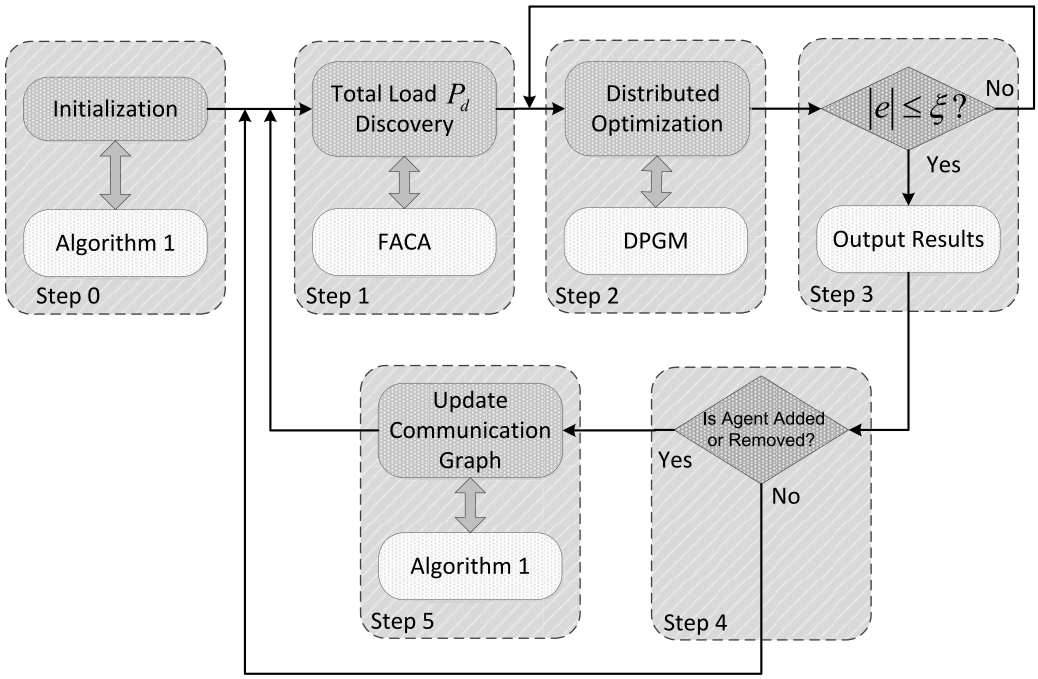


Fig. 2. Flowchart of distributed ED.

following advantages.

1) No private information, such as gradient or the incremental cost, is required to exchange with other generators.
2) It can solve any convex objective cost function rather than only quadratic function.
3) The initial value of the estimate can be chosen arbitrary by each agent individually.

B. Implementation of Distributed ED

Based on proposed Algorithm 1, distributed FACA and DPGM, we are ready to implement our distributed ED design. The flowchart of our proposed distributed ED procedure is shown in Fig. 2, with each corresponding step described as follows.

Step 0 *(Initialization and Graph Discovery):* As a starting point, the communication graph of all agents is predesigned as connected.[1] Using Algorithm 1, each agent can get the information of the whole communication graph such as the number of generator agents N, the number of load agents M, and the Laplancian matrix $\mathcal{L}$ in less than $N-1$ steps. Then using some available numerical methods to calculate the nonzero eigenvalues of $\mathcal{L}$.

Step 1 *(Total Load Demand Discovery):* In this step, each agent determines the total load demand P_d from (11) in K steps according to Lemma 3.

Step 2 *(Distributed Optimization):* The DPGM algorithm (12), (13) introduced in Section IV-A is applied here to execute the distributed ED.

Step 3 *(Stop Criterion Check):* In theory, the sequences of estimates generated by DPGM converge to the optimal solution asymptomatically. In practice, some iteration stop criterions are set. Here we set $|e^k| \le \xi, k = 1, \ldots, N$ as a stop criterion, where $e^k = x^k(l+1) - x^k(l)$, ξ is a user defined small positive number. If this condition is satisfied, then stop the iteration, output the results and go to step 4. If not, go to step 2.

Step 4 *(Plug-In and Plug-Out Reconfiguration):* At this step, each agent needs to check whether there is any agent added in or removed from the grid. If yes, execute the graph reconfiguration rule described below, and then go to step 5 to update the communication graph; otherwise go to step 1.

Graph Reconfiguration: If an agent (agent n) is newly added in, it tries to find its nearest neighbors, gets permission from them and then adds them in its neighbor list. If an agent (say agent i) is removed, its neighboring agent j, $j \in \mathcal{N}_i$ will delete agent i from its neighborhood list $\mathcal{N}_j$ and also tries to setup communication with other agent k, $k \in \mathcal{N}_i \backslash j$, which is also the neighbor of agent i. If $k \in \emptyset$, i.e., no other neighbor of agent i exists, then nothing is needed to be done but just deleting agent i.

Step 5 *(Graph Updating):* At this step, all agents need to update the communication graph using Algorithm 1 and then go to step 1.

A simple graph reconfiguration example is illustrated in Fig. 3. Suppose agent 3 is removed, its neighboring agent $\mathcal{N}_3 = \{1, 2, 4\}$ monitors this situation, respectively. For agent 1, it needs to set up a new communication channel with agent 4 (no need with agent 2 as they have already been connected); for agent 2, it also needs to setup a new communication channel with agent 4; for agent 4, it needs to set up communication with both agent 1 and 2.

[1]This is easily implementable, as each agent can choose the same communication graph as their physical connection graph at the initial step.

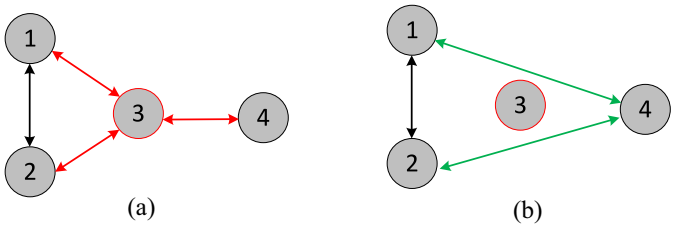


Fig. 3. Graph reconfiguration illustration example. (a) Before disconnection. (b) After disconnection.

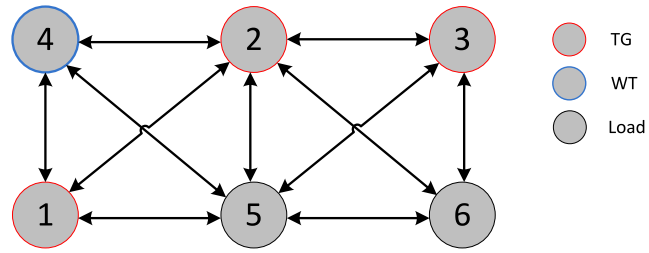


Fig. 4. Communication graph of test power system.

C. Complexity Analysis

Here, we analyze the computational performance of proposed distributed strategy. Note that our proposed DPGM (12), (13) mainly contains two parts, namely, FAC and projected gradient operation (PGO). According to Lemma 2, for a system with N agents, the FAC process can be fulfilled in less than $N-1$ steps, which is much more efficient than conventional average consensus algorithm possessing asymptotical convergence. For PGO, it is actually a combination of the gradient descent method and a projection operation. The projection operation in our specific problem, as discussed in Appendix B, is nothing but a simple algebraic operation. So, it has little contribution to the computational cost.

V. Case Studies

In order to test the effectiveness of the proposed distributed ED method, several case studies are presented and discussed in this section. First, a six-bus power system implementation without and with generator constraints are demonstrated. The second case study illustrates the plug-and-play property of proposed method including both generator and load node. Then the IEEE 30-bus system is used as a large network case to demonstrate the effectiveness of proposed method. Lastly, comparisons with genetic algorithm (GA) are carried out.

A. Case Study 1: Implementation on Six-Bus Power System

In this test case, a six-bus power system topology is adopted from [12]. It consists of three TGs, one WT, and two load nodes. We replace one TG with a WT in [12]. Its communication graph is shown in Fig. 4. The corresponding Laplacian matrix can be obtained by each agent using Algorithm 1, which is

$$\mathcal{L} = \begin{bmatrix} 3 & -1 & 0 & -1 & -1 & 0 \\ -1 & 5 & -1 & -1 & -1 & -1 \\ 0 & -1 & 3 & 0 & -1 & -1 \\ -1 & -1 & 0 & 3 & -1 & 0 \\ -1 & -1 & -1 & -1 & 5 & -1 \\ 0 & -1 & -1 & 0 & -1 & 3 \end{bmatrix}.$$

Its three distinct nonzero eigenvalues are $\lambda_1 = 2, \lambda_2 = 4$, and $\lambda_3 = 6$, which means that it needs only $K = 3$ steps for each agent to reach consensus when applying distributed FACA. The parameters of three different types of TGs are adopted from [12], while the WT parameters are from [21]. These parameters are listed in Tables III and IV, respectively.

First, the generator constraints are not imposed. In the initial, the load demand is ${P_d}^5 = 200$ MW, ${P_d}^6 = 200$ MW, and each generator is operating in the optimal condition with the generated power $P_1 = 174.0683$ MW, $P_2 = 100.00$ MW, $P_3 = 50.00$ MW, and $W_1 = 75.9317$ MW. Then load 5 are doubled, i.e., ${P_d}^5 = 400$ MW. The changed total load demand can be discovered by TG and WT generators in three steps as shown in Table V. Then each generator agent (agents 1–4) conducts the distributed ED using the proposed DPGM method. The initial value is chosen as $x^1(0) = \begin{bmatrix} 174.0683 & 0 & 0 & 0 \end{bmatrix}^T$, $x^2(0) = \begin{bmatrix} 0 & 100 & 0 & 0 \end{bmatrix}^T$, $x^3(0) = \begin{bmatrix} 0 & 0 & 50 & 0 \end{bmatrix}^T$, and $x^4(0) = \begin{bmatrix} 0 & 0 & 0 & 75.9318 \end{bmatrix}^T$. The iteration process is shown in Fig. 5(a). Clearly, all the estimated power outputs converge to the optimized solution ${P_1}^\star = 367.7996$ MW, ${P_2}^\star = 102.2463$ MW, ${P_3}^\star = 29.1174$ MW, and ${W_1}^\star = 100.8367$ MW with a total cost $C = 5611.8\$$. Also it is found that the incremental cost (or the gradient of the cost ∇c_k) of each generator reaches a consensus value 8.25.

TABLE III
Parameters of TGs

Generator	α_i $[\$/MW^2h]$	β_i $[\$/MWh]$	γ_i $[\$/h]$	${P_i}^{\min}$ $[MW]$	${P_i}^{\max}$ $[MW]$
1	0.00142	7.2	510	150	600
2	0.00194	7.85	310	100	400
3	0.00482	7.97	78	50	200

TABLE IV
Parameters of Wind and WT

Wind	v_{in}	v_{out}	v_r	(c,k)
	5	45	15	$(8,2)$
Wind Turbine	d_j	C_{pwj}	C_{rwj}	W_r
	6	3.1	3.1	160

TABLE V
Total Load Demand Discovery

Step	Content	Step	Content
$m=0$	$y^1(0)=0$ $y^2(0)=0$ $y^3(0)=0$ $y^4(0)=0$ $y^5(0)=400$ $y^6(0)=200$	$m=2$	$y^1(2)=50$ $y^2(2)=0$ $y^3(2)=50$ $y^4(2)=50$ $y^5(2)=400$ $y^6(2)=50$
$m=1$	$y^1(1)=200$ $y^2(1)=300$ $y^3(1)=300$ $y^4(1)=200$ $y^5(1)=-500$ $y^6(1)=100$	$m=3$	$y^1(3)=100$ $y^2(3)=100$ $y^3(3)=100$ $y^4(3)=100$ $y^5(3)=100$ $y^6(3)=100$

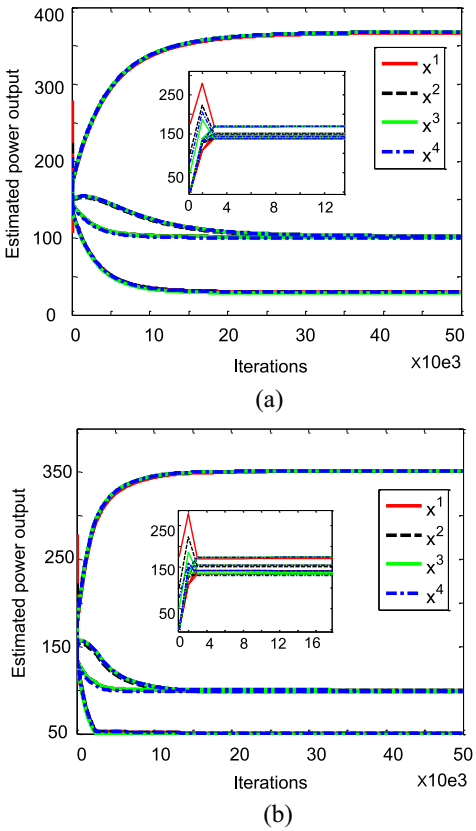


Fig. 5. Simulation results of six-bus power system. (a) Without and (b) with generator constraints.

Note that the third generator is conflict with its lower output bound $P_3^{\min} = 50$ MW.

Then, we consider the generator constraints. The results are shown in Fig. 5(b). The optimized power output are $P_1^\star = 351.7814$ MW, $P_2^\star = 100.0248$ MW, $P_3^\star = 50.0248$ MW, and $W_1^\star = 98.1691$ MW with a total cost $C = 5614.4\$$. In this case, note that all the generators' power output are within their constraints, respectively.

This case study shows that our proposed method can handle the ED problem both without and with generator constraints.

B. Case Study 2: Plug-and-Play Capability

This case study is to test the flexibility of the proposed method. The plug-and-play performance of both generator and load are considered.

1) Generator Plug-and-Play: In this subcase, the plug-and-play of TG is considered. The results of TG are shown in Fig. 6. From Fig. 6(a), it is clear that the estimated power outputs of all agents $x^k(l)$, $k = 1, \ldots, 4$ almost reach consensus in $l = 10 \times 10^3$ iterations. To clearly demonstrate the estimated power output, the estimates of agent 1 is shown in Fig. 6(b). In the initial, the load demands are $P_d^5 = 400$ MW and $P_d^6 = 200$ MW. After a few iterations the proposed method ensures convergence to the optimized power output $P_1^\star = 351.7814$ MW, $P_2^\star = 100.0248$ MW, $P_3^\star = 50.0248$ MW, and $W_1^\star = 98.1691$ MW with the total cost $C = 5614.4\$$. Suppose generator 1 (agent 1) is disconnected from the power system at the time step $l = 30 \times 10^3$. Then using the graph reconfiguration rule introduced in Section IV-B, each agent can reconfigure the communication graph as shown in Fig. 7. The remaining generators can converge to new optimized power output $P_1^\star = 0$ MW, $P_2^\star = 322.2944$ MW, $P_3^\star = 117.6226$ MW, and $W_1^\star = 160.0829$ MW. It is shown

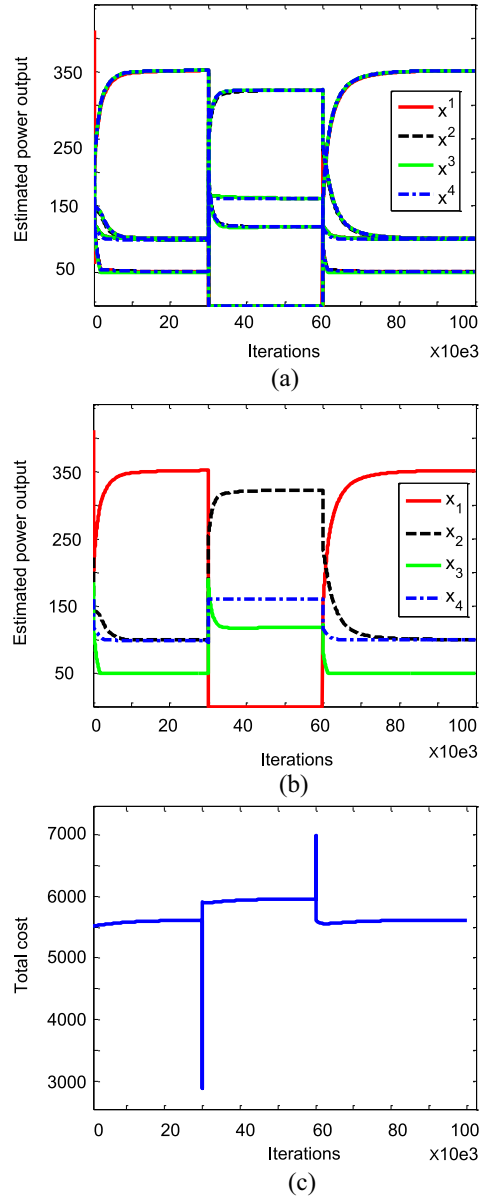


Fig. 6. Simulation results with generator plug-and-play. Estimates of (a) all agents and (b) agent 1. (c) Total cost.

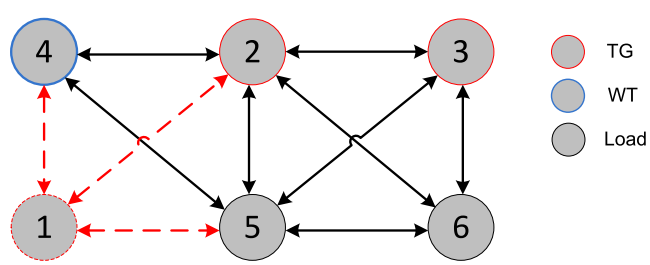


Fig. 7. Graph reconfiguration of generator 1 plug-and-play.

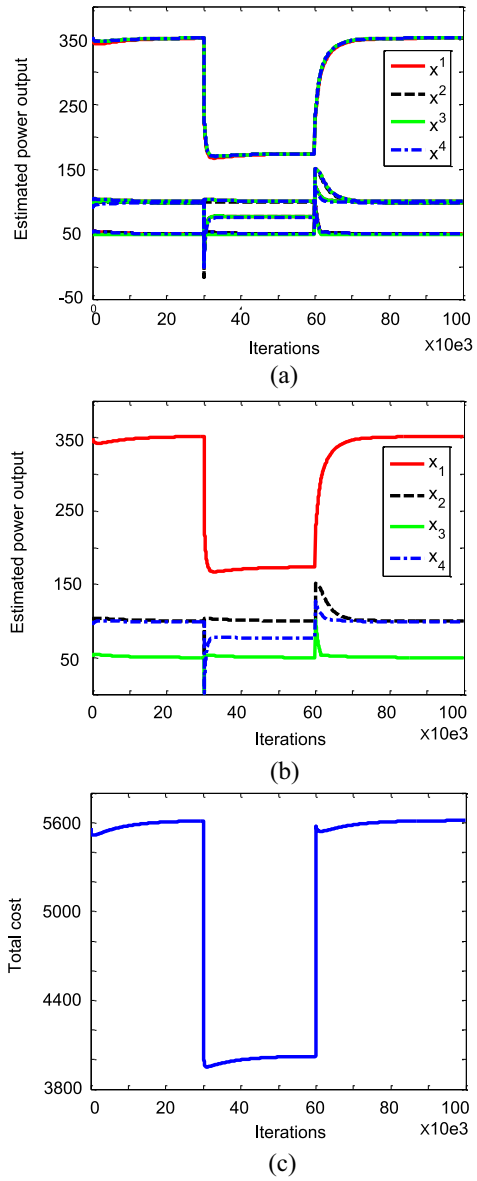


Fig. 8. Simulation results with load plug-and-play. Estimates of (a) all agents and (b) agent 1. (c) Total cost.

in Fig. 6(c) that the total cost increases to $C = 5960.7\$$ due to the absence of generator 1. It further indicates that generator 1 is more economic efficient compared to the average of other generators. Then at the time step $l = 60 \times 10^3$, generator 1 is connected to the system again and all the results converge to those of the previous ones.

2) Load Plug-and-Play: The results of this subcase are shown in Fig. 8. The initial condition is the same as that in subcase 1). At the time step $l = 30 \times 10^3$, load 6 (agent 6) is disconnected from the power system. The other agents detect this change, update the communication graph as well as the total load demand $P_d = 400$ MW. Then the remaining agents ensure the convergence to new optimized power outputs $P_1{}^\star = 173.56$ MW, $P_2{}^\star = 100.0785$ MW, $P_3{}^\star = 50.0784$ MW, and $W_1{}^\star = 76.2811$ MW with the total cost $C = 4024.7\$$. At the time step $l = 60 \times 10^3$, load 6 is connected to the system again and all the values are back to the previous values before load 6 is disconnected.

Both subcases 1) and 2) show that the proposed method is fully distributed and has the plug-and-play property.

TABLE VI
GENERATOR PARAMETERS IN IEEE 30-BUS

Generator No.	α_i $[\$/MW^2h]$	β_i $[\$/MWh]$	γ_i $[\$/h]$	$P_i{}^{\min}$ $[MW]$	$P_i{}^{\max}$ $[MW]$
1	0.00375	2	0	50	200
2	0.0175	1.75	0	20	80
5	0.0625	1.0	0	15	50
8	0.00834	3.25	0	10	35
11	0.025	3.0	0	10	30
13	0.025	3.0	0	12	40

TABLE VII
LOAD PARAMETERS IN IEEE 30-BUS

Load No.	$P_i{}^d$ $[MW]$	Load No.	$P_i{}^d$ $[MW]$	Load No.	$P_i{}^d$ $[MW]$
1	0.00	11	0.00	21	17.5
2	21.7	12	11.2	22	0.00
3	2.40	13	0.00	23	3.20
4	7.60	14	6.20	24	8.70
5	94.2	15	8.20	25	0.00
6	0.00	16	3.50	26	3.50
7	22.8	17	9.00	27	0.00
8	30.0	18	3.20	28	0.00
9	0.00	19	9.50	29	2.40
10	5.80	20	2.20	30	10.6

TABLE VIII
SIMULATION RESULTS OF IEEE 30-BUS TEST SYSTEM

Total Demand P_d $[MW]$	Optimized Power Output P_i $[MW]$					
	Gen 1	Gen 2	Gen 5	Gen 8	Gen 11	Gen 13
283.4	185.03	47.11	19.20	10.02	10.02	12.02
368.42	200.03	66.13	24.57	35.03	21.33	21.33
226.72	140.71	37.49	16.52	10.00	10.00	12.00

C. Case Study 3: Implementation on IEEE 30-Bus Test System

In order to test the effectiveness of the proposed method for a large network, the IEEE 30-bus system is chosen as a test system. The generator and load bus parameters are adopted from [15], which are also listed in Tables VI and VII, respectively. First, the communication graph can be chosen as the same as the physical connections. Then the total load demand $P_d = \sum_{i=1}^{30} P_i^d = 283.4$ MW can be easily discovered by applying proposed distributed FACA. The optimized power allocation can be obtained by applying DPGM. Suppose that at the time steps $l = 30 \times 10^3$ and $l = 60 \times 10^3$ the load demand is increased by 30% and reduced by 20%, respectively. The simulation results are shown in Fig. 9 and Table VIII. These simulation results illustrate the effectiveness of the proposed method applied on a large network.

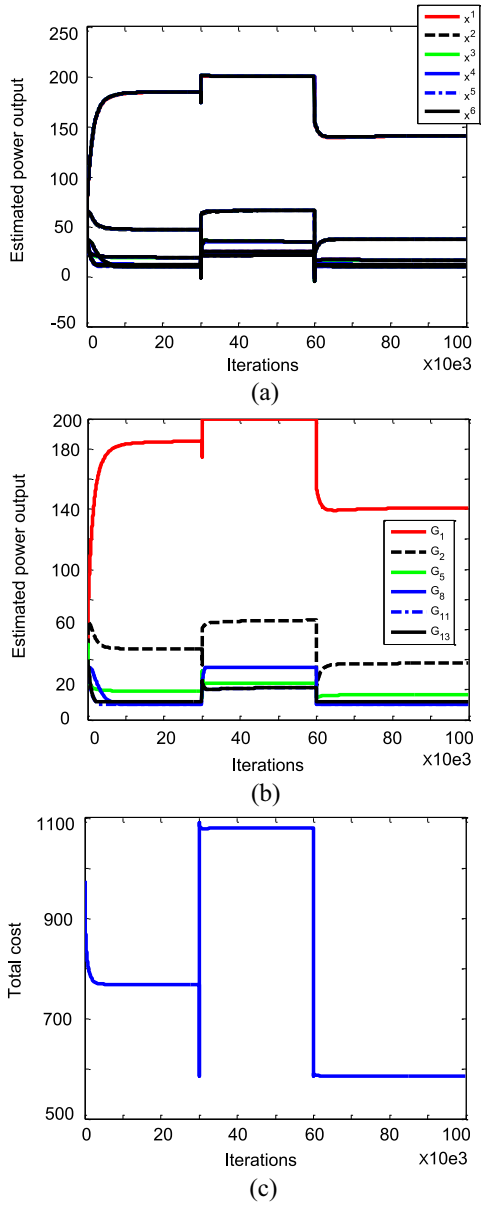


Fig. 9. Simulation results of IEEE 30-bus system. Estimates of (a) all agents and (b) agent 1. (c) Total cost.

TABLE IX
PARAMETERS OF GA METHOD

Parameter	Value	Parameter	Value
Population Size	80	Crossover Fraction	0.8
Migration Interval	20	Migration Interval Fraction	0.2
Maximum Generations	60	Penalty Factor	100
Stall Time Limit	200	Function Tolerance	1e-6

D. Case Study 4: Comparison With Heuristic Search Method

In this section, we apply one popular heuristic search method, i.e., genetic algorithm proposed in [31], to our proposed ED problem for comparison. The parameters of the test system are the same as those in case study 1. Some key parameters of the GA method are listed in Table IX. The fitness function is chosen as the total cost in (1) plus some penalty functions constraining the variables according to [31]. The evolution of the fitness value is shown in Fig. 10. The best fitness value for a population is the smallest fitness value among all the individuals in the population, while the mean fitness value is the average of their fitness values [31]. After 51 generations, the mean fitness approaches to the best fitness value. The GA stops when the average relative change in the fitness value is less than the predefined function tolerance, which is listed in Table IX. The final power outputs are obtained from the best individual of the last generation as $P_1{}^\star = 349.06$ MW, $P_2{}^\star = 103.5257$ MW, $P_3{}^\star = 50.0306$ MW, and $W_1{}^\star = 97.3836$ MW with the total cost $C = 5614.7\$$. Comparing to the results in case study 1 ($P_1{}^\star = 351.7814$ MW, $P_2{}^\star = 100.0248$ MW, $P_3{}^\star = 50.0248$ MW, and $W_1{}^\star = 98.1691$ MW with a total cast $C = 5614.4\$$), they are about the same. The consistency of these results further illustrates and verifies the effectiveness of our scheme.

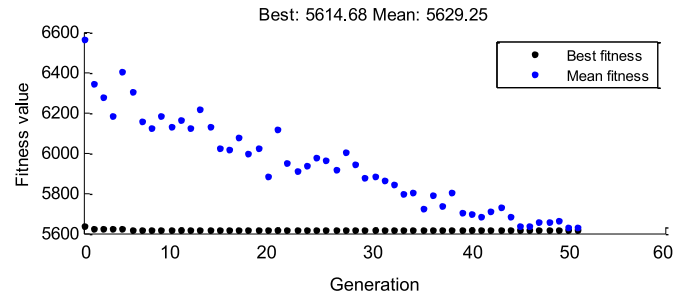


Fig. 10. Evolution of fitness values in GA method.

However, such similar results are achieved with three major differences between GA method and our proposed approach. First, similar to most heuristic search methods, the GA method is a centralized one while ours is fully distributed. Second, the GA method does not have the plug-and-play property. Third, the operation and implementation costs are different.

VI. CONCLUSION

In this paper, a fully distributed ED algorithm based on FAC and projected gradient is proposed for smart grid systems with random wind power. By allowing each agent to communicate with their neighbor agents, the total cost of the whole system can be minimized by the proposed distributed ED algorithm while satisfying both equality and inequality constraints. Compared to the existing methods, no private confidential gradient or incremental cost information exchange is needed and the objective function is not required to be quadratic. What is more, the initial estimate values of our proposed method can be chosen arbitrarily by each agent individually. The effectiveness of the proposed scheme has been validated by several case studies including without generator constraints, with generator constraints, plug-and-play of generators and loads and a large IEEE 30-bus test system. The results show good performance of the proposed method.

As an alternative approach, how to develop a distributed ED strategy based on stochastic programming method is an interesting topic worthy of consideration as a future work.

Appendix A
Cost Penalty Related Function of Wind Turbine

The underestimation cost can be expressed as the penalty cost for not using all the available wind power, which is linear to the mean of random variable $Y_{ue}(= W_{j,\mathrm{av}} - W_j)$. The expression of $E(Y_{ue,j})$ is derived in [21] as

$$\begin{aligned} E(Y_{ue,j}) = {} & (W_r - W_j)\left[\exp\left(-\frac{v_r^\kappa}{c^\kappa}\right) - \exp\left(-\frac{v_{\mathrm{out}}^\kappa}{c^\kappa}\right)\right] \\ & + \left(\frac{W_r v_{\mathrm{in}}}{v_r - v_{\mathrm{in}}} + W_j\right)\left[\exp\left(-\frac{v_r^\kappa}{c^\kappa}\right) - \exp\left(-\frac{v_j^\kappa}{c^\kappa}\right)\right] \\ & + \frac{W_r c}{v_r - v_{\mathrm{in}}}\left\{\Gamma\left[1 + \frac{1}{\kappa}, \left(\frac{v_j}{c}\right)^\kappa\right]\right. \\ & \left. - \Gamma\left[1 + \frac{1}{\kappa}, \left(\frac{v_r}{c}\right)^\kappa\right]\right\} \end{aligned} \tag{14}$$

where W_r is the rated wind power, v_r, v_{in}, and v_{out} are the rated, cut-in and cut-out wind speeds, κ and c are the scale factor and shape factor of the Weibull distribution of wind, $\Gamma(a, x)$ is a standard incomplete gamma function, and v_j is a intermediary variable, which is given as

$$v_j = v_{\mathrm{in}} + \frac{(v_r - v_{\mathrm{in}})W_j}{W_r}. \tag{15}$$

Note that in order to simplify the notation, we have dropped the subscript j in the above parameters.

Similarly, the overestimation cost is due to the available wind power being less than the scheduled wind power so needs to get some power from other source, e.g., ESS, which is expressed as $C_{rwj}E(Y_{oe,j})$, where $E(Y_{oe,j})$ is given as

$$\begin{aligned} E(Y_{oe,j}) = {} & W_j\left[1 - \exp\left(-\frac{v_{\mathrm{in}}^\kappa}{c^\kappa}\right) + \exp\left(-\frac{v_{\mathrm{out}}^\kappa}{c^\kappa}\right)\right] \\ & + \left(\frac{W_r v_{\mathrm{in}}}{v_r - v_{\mathrm{in}}} + W_j\right)\left[\exp\left(-\frac{v_{\mathrm{in}}^\kappa}{c^\kappa}\right) - \exp\left(-\frac{v_j^\kappa}{c^\kappa}\right)\right] \\ & + \frac{W_r c}{v_r - v_{\mathrm{in}}}\left\{\Gamma\left[1 + \frac{1}{\kappa}, \left(\frac{v_j}{c}\right)^\kappa\right]\right. \\ & \left. - \Gamma\left[1 + \frac{1}{\kappa}, \left(\frac{v_{\mathrm{in}}}{c}\right)^\kappa\right]\right\}. \end{aligned} \tag{16}$$

Appendix B
Projection Operation

In the formulated ED problem, if the generator constraint is ignored, the constraint set for each agent $X_k{}'$ is identical, e.g., $\sum_{k=1}^N x_i = P_d$. This can be treated as a "N-dimension" plane in the Hillbert space with the normal vector $\vec{n} = \begin{bmatrix}1 & \cdots & 1\end{bmatrix}^T \in \mathbb{R}^{N\times 1}$. The projection operation for a given point $p_0 = [x_1, x_2, \ldots, x_N]^T$ to this plane can be easily obtained as

$$P_{X_k{}'}[p_0] = p_0 - \frac{\vec{n}^T p_0 - P_d}{N}\vec{n}, \quad k = 1, \ldots, N. \tag{17}$$

Obviously, if $p_0 \in X_k{}'$, then $P_{X_k{}'}[p_0] = p_0$.

If the generator constraint, i.e., $P_k{}^{\min} \le x_k \le P_k{}^{\max}$ is imposed, then the projection operation should consider the boundary constraint. Let $p_1 = P_{X_k{}'}[p_0]$, if $(p_1)_k > P_k{}^{\max}$ or $(p_1)_k < P_k{}^{\min}$, then set $(p_1)_k = P_k{}^{\max}$ or $(p_1)_k = P_k{}^{\min}$, respectively, where $(.)_i$ denotes the ith component of the vector. Let $p_2 \in \mathbb{R}^{(N-1)\times 1}$ be the remaining vector by removing $(p_1)_k$, i.e., $p_2 = [x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_N]^T$. Then project p_2 onto the new constrain set $X_k{}''$, i.e., $\sum_{i=1}^N x_i - x_k = P_d - P_k{}^{\max}$ or $\sum_{i=1}^N x_i - x_k = P_d - P_k{}^{\min}$ using the following operations:

$$P_{X_k{}''}[p_2] = \begin{cases} p_2 - \dfrac{\vec{n}'^T p_2 - P_d + P_k{}^{\max}}{N-1}\vec{n}', & (p_1)_k > P_k^{\max} \\ p_2 - \dfrac{\vec{n}'^T p_2 - P_d + P_k{}^{\min}}{N-1}\vec{n}', & (p_1)_k < P_k^{\min} \end{cases} \tag{18}$$

where $\vec{n}' = \begin{bmatrix}1 & \cdots & 1\end{bmatrix}^T \in \mathbb{R}^{(N-1)\times 1}$. Let $p_3 = P_{X_k{}''}[p_2]$, the final projection result of p_0 with consideration of boundary constraint can be obtained by inserting $(p_1)_k$ into p_3 in the kth place, that is

$$P_{X_k}[p_0] = \left[(p_3)_1, \ldots, (p_3)_{k-1}, (p_1)_k, (p_3)_k, \ldots, (p_3)_{N-1}\right]^T. \tag{19}$$

References

[1] D. E. Olivares *et al.*, "Trends in microgrid control," *IEEE Trans. Smart Grid*, vol. 5, no. 4, pp. 1905–1919, Jul. 2014.

[2] J.-Y. Fan and L. Zhang, "Real-time economic dispatch with line flow and emission constraints using quadratic programming," *IEEE Trans. Power Syst.*, vol. 13, no. 2, pp. 320–325, May 1998.

[3] C. E. Lin, S. T. Chen, and C. L. Huang, "A direct Newton-Raphson economic dispatch," *IEEE Trans. Power Syst.*, vol. 7, no. 3, pp. 1149–1154, Aug. 1992.

[4] T. Guo, M. Henwood, and M. van Ooijen, "An algorithm for combined heat and power economic dispatch," *IEEE Trans. Power Syst.*, vol. 11, no. 4, pp. 1778–1784, Nov. 1996.

[5] H. Liang, B. J. Choi, A. Abdrabou, W. Zhuang, and X. Shen, "Decentralized economic dispatch in microgrids via heterogeneous wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 6, pp. 1061–1074, Jul. 2012.

[6] J. A. P. Lopes, C. L. Moreria, and A. G. Madureira, "Defining control strategies for microgrids islanded operation," *IEEE Trans. Power Syst.*, vol. 21, no. 2, pp. 916–924, May 2006.

[7] F. Guo, C. Wen, J. Mao, and Y.-D. Song, "Distributed secondary voltage and frequency restoration control of droop-controlled inverter-based microgrids," *IEEE Trans. Ind. Electron.*, vol. 62, no. 7, pp. 4355–4364, Jul. 2015.

[8] W. Yu, G. Wen, X. Yu, Z. Wu, and J. Lu, "Bridging the gap between complex networks and smart grids," *J. Control Decis.*, vol. 1, no. 1, pp. 102–114, 2014.

[9] Z. Fan, "A distributed demand response algorithm and its application to PHEV charging in smart grids," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1280–1290, Sep. 2012.

[10] Z. Zhang and M. 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.

[11] 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.

[12] G. Binetti, A. Davoudi, F. K. Lewis, D. Naso, and B. Turchiano, "Distributed consensus-based economic dispatch with transmission losses," *IEEE Trans. Power Syst.*, vol. 29, no. 4, pp. 1711–1720, Jul. 2014.

[13] 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.

[14] W. Zhang, W. Liu, X. Wang, L. Liu, and F. Ferrese, "Online optimal generation control based on constrained distributed gradient algorithm," *IEEE Trans. Power Syst.*, vol. 30, no. 1, pp. 35–45, Jan. 2015.

[15] C. Li, X. Yu, and W. Yu, "Optimal economic dispatch by fast distributed gradient," in *Proc. 13th Int. Conf. Control Automat. Robot. Vis.*, Singapore, 2014, pp. 571–576.

[16] J. Chen, J. Chen, and C. Gong, "On optimizing the aerodynamic load acting on the turbine shaft of PMSG-based direct-drive wind energy conversion system," *IEEE Trans. Ind. Electron.*, vol. 61, no. 8, pp. 4022–4031, Aug. 2014.

[17] J. Chen, J. Chen, and C. Gong, "New overall power control strategy for variable-speed fixed pitch wind turbines within the whole wind velocity range," *IEEE Trans. Ind. Electron.*, vol. 60, no. 7, pp. 2652–2660, Jul. 2013.

[18] X. Liu and W. Xu, "Economic load dispatch constrained by wind power availability: A here-and-now approach," *IEEE Trans. Sustain. Energy*, vol. 1, no. 1, pp. 2–9, Apr. 2010.

[19] X. Liu, "Economic load dispatch constrained by wind power availability: A wait-and-see approach," *IEEE Trans. Smart Grid*, vol. 1, no. 3, pp. 347–355, Dec. 2010.

[20] J. Hetzer, D. C. Yu, and K. Bhattarai, "An economic dispatch model incorporating wind power," *IEEE Trans. Energy Convers.*, vol. 23, no. 2, pp. 603–611, Jun. 2008.

[21] X. Liu and W. Xu, "Minimum emission dispatch constrained by stochastic wind power availability and cost," *IEEE Trans. Power Syst.*, vol. 25, no. 3, pp. 1705–1713, Aug. 2010.

[22] Y. Zhang, N. Gatsis, and G. B. Giannakis, "Robust energy management for microgrids with high-penetration renewables," *IEEE Trans. Sustain. Energy*, vol. 4, no. 4, pp. 944–953, Oct. 2013.

[23] J. P. Barton and D. G. Infield, "Energy storage and its use with intermittent renewable energy," *IEEE Trans. Energy Convers.*, vol. 19, no. 2, pp. 441–448, Jun. 2004.

[24] A. J. Wood, B. F. Wollenberg, and G. B. Sheble, *Power Generation, Operation and Control*. 3rd ed. New York, NY, USA: Wiley, 2012.

[25] C. Godsil and G. Royle, *Algebraic Graph Theory*. New York, NY, USA: Springer, 2001.

[26] A. Y. Kibangou, "Graph Laplacian based matrix design for finite-time distributed average consensus," in *Proc. Amer. Control Conf.*, Montreal, QC, Canada, 2012, pp. 1901–1906.

[27] R. Olfati-Saber, J. A. Fax, and R. M. Murrary, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

[28] R. Aragues, G. Shi, D. V. Dimarogonas, C. Sagues, and K. H. Johansson, "Distributed algebraic connectivity estimation for adaptive event-triggered consensus," in *Proc. Amer. Control Conf.*, Montreal, QC, Canada, 2012, pp. 32–37.

[29] G. Li, C. Wen, W. Zheng, and Y. Chen, "Identification of a class of nonlinear autoregressive models with exogenous inputs based on kernel machines," *IEEE Trans. Signal Process.*, vol. 59, no. 5, pp. 2146–2159, May 2011.

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

[31] D. C. Walters and G. B. Sheble, "Genetic algorithm solution of economic dispatch with valve point loading," *IEEE Trans. Power Syst.*, vol. 8, no. 3, pp. 1325–1332, Aug. 1993.

Fanghong Guo received the B.Eng. degree in automation science from Southeast University, Nanjing, China, in 2010, and the M.Eng. degree in automation science and electrical engineering from Beihang University, Beijing, China, in 2013. He is currently pursuing the Ph.D. degree with the Energy Research Institute@NTU, Interdisciplinary Graduate School, Nanyang Technological University, Singapore.

His current research interests include distributed cooperative control, microgrid systems, distributed optimization, and smart grid.

Changyun Wen (F'10) received the B.Eng. degree in automatic control from Xi'an Jiaotong University, Xi'an, China, in 1983, and the Ph.D. degree from the University of Newcastle, Callaghan, NSW, Australia, in 1990.

From 1989 to 1991, he was a Postdoctoral Fellow with the University of Adelaide, Adelaide, SA, Australia. Since 1991, he has been with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, where he is currently a Full Professor. His current research interests include control systems and applications.

Prof. Wen is an Associate Editor of a number of journals, including *Automatica*, the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, and the IEEE CONTROL SYSTEMS MAGAZINE. He is the Executive Editor-in-Chief of the *Journal of Control and Decision*. He has served as an Associate Editor for the IEEE TRANSACTIONS ON AUTOMATIC CONTROL from 2000 to 2002. He was a Distinguished Lecturer of the IEEE Control Systems Society from 2010 to 2013.

Jianfeng Mao (M'04) received the B.E. and M.E. degrees in automation science from Tsinghua University, Beijing, China, in 2001 and 2004, respectively, and the Ph.D. degree in systems engineering from Boston University, Boston, MA, USA, in 2009.

He is currently an Assistant Professor with the Division of Systems and Engineering Management, School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore. His current research interests include modeling and optimization of discrete event, and hybrid systems with applications to transportation systems, air traffic management, energy management, and healthcare systems.

Yong-Duan Song (SM'10) received the Ph.D. degree in electrical and computer engineering from Tennessee Technological University, Cookeville, TN, USA, in 1992.

He was a Full Professor with North Carolina A&T State University, Greensboro, NC, USA, from 1993 to 2008, and a Langley Distinguished Professor with the National Institute of Aerospace (NIA), Hampton, VA, USA, from 2005 to 2008. He is currently the Dean of the School of Automation, and the Founding Director of the Institute of Smart Systems and Renewable Energy, Chongqing University, Chongqing, China. He was one of the six Langley Distinguished Professors and the Founding Director of Cooperative Systems with NIA. His current research interests include intelligent systems, guidance navigation and control, bio-inspired adaptive and cooperative systems, rail traffic control and safety, and smart grid.

Prof. Song has served as an Associate Editor/Guest Editor for several prestigious scientific journals.