energies

Article

Distributed Absorption and Half-Search Approach for Economic Dispatch Problem in Smart Grids

Bo Li [1], Panpan Zhang [1], Xiangjun Li [2,3,*] and Shengxian Cao [1]

1 School of Automation Engineering, Northeast Electric Power University, Jilin 132012, China; libo@neepu.edu.cn (B.L.); panpanzhangneepu@126.com (P.Z.); csxlb_jl@163.com (S.C.)
2 State Key Laboratory of Control and Operation of Renewable Energy and Storage Systems, Energy Storage and Electrical Engineering Department, China Electric Power Research Institute, Beijing 100192, China
3 Contemporary Amperex Technology Limited (Qinghai), Xining 810021, Qinghai, China
* Correspondence: li_xiangjun@126.com; Tel.: +86-133-7012-0059

Received: 23 March 2019; Accepted: 18 April 2019; Published: 23 April 2019

Abstract: The economic dispatch problem (EDP) is a significant class of optimization issues in the power system, which works on minimizing the total cost when generating a certain amount of power. A novel distributed approach for EDP is proposed in this paper. The presented approach consists of two steps. The first step, named absorption search, is to simplify the network structure through absorption searching. A flooding-based consensus approach is applied in the first step, which can be used to achieve consensus information among nodes. After the first step, only the generation nodes are kept in the network. The data collection can be completed by local computation and communication between neighbors. The first step can be considered as the stage of gathering information. In the second step, a distributed half-search algorithm makes the nodes obtain the final optimal solution in a distributed way. The results on three case studies demonstrate that the proposed approach is highly effective for solving the EDP.

Keywords: economic dispatch problem; distributed consensus; energy management system; half-search algorithm; absorption searching; smart grids

1. Introduction

For power generation, smart grids exploit intelligent controls and developed technology to control the power generation combination composed of renewable resources [1]. Because of advantages such as environmental friendliness and sustainable development, cost savings, electricity market and shared economy, smart grid and its related research topics have attracted much attention of scientific researchers. Ref. [2] surveyed the enabling technologies for Smart Grid. The authors of [3–5] analyzed the stability of nonlinear power systems based on backstepping control approaches. The optimal control and management for a large-scale battery energy storage system with wind and photovoltaic power station is introduced in [6–8]. Ref. [9] discussed optimization of sustainable microgrid considering cost analysis, carbon emission and availability of energy resources. In the wake of development and expansion of the scale of smart grid, energy management systems (EMS) [10,11], especially online EMS [12,13], are becoming important research subjects. As an essential research direction of EMS, the economic dispatching problem (EDP) has been deeply studied due to its significant benefit of economical efficiency for smart grid. The EDP is a resource allocation problem, which minimizes the total generation cost while meeting the load demands. Several classical techniques such as Lambda-iteration method [14], Gradient methods [15], and Newton's method [16] have been developed to solve EDP whose cost function is convex. From the converge procedure point of view, Lambda-iteration method can offer rapid results. There are many studies on solving EDP when cost functions are non-convex. The authors of [17,18] introduced a hybrid evolutionary algorithm based on shuffle frog leaping and particle swarm optimization (PSO) to address multi-objective EDP. The authors of [19–21] used PSO and its improved algorithms to solve EDP with non-smooth or non-convex cost function. Other optimization algorithms such as self-adaptive differential evolution (DE) [22], real coded genetic algorithm (GA) [23], biogeography-based optimization (BBO) algorithm [24], firefly algorithm (FA) [25], and grey wolf optimization (GWO) [26] are presented to solve this problem.

The algorithms mentioned above are, usually, performed by centralized controllers. The central controller needs to communicate with each unit in the power system and implement the optimal solution. As the power system becomes bigger or more complex on the generation or demand side, the centralized units need to perform huge computing and communicating tasks in real time, and they must be equipped with powerful servers to handle these data, which increase the cost of power generation [27]. In addition, a single-point-failure may cause the centralized algorithm to be redesigned [28]. At the same time, with many distributed generators (DG) entering the network and diversified user needs, centralized algorithms are having more and more difficulty with economic dispatch problems [29]. Besides, another important reason is that the centralized approach does not adapt to the plug-and-play characteristics of modern smart grids. To make up for the defects of this algorithm, many scalable and robust distributed algorithms have been proposed and applied in EDP. Ref. [30] presented a decentralized, non-hierarchical framework, in which economic dispatch analysis can be completed by calculating proper weighted averages of interest variables. Ref. [31] focused on a novel consensus based algorithm to address EDP in a distributed manner, in which the mismatch between demand and total power supply by generators is learned collectively, and then used as a feedback mechanism to adjust the power generation by each generator. The authors of [32] introduced a hybrid GWO-PSO algorithm to solve multi-objective energy management. Ref. [33] proposed an approach to solve the distribution feeder reconfiguration problem in smart grid environment with flexible electricity price. Ref. [34] gave a mathematical formulation based on incremental cost consensus for EDP, in which a distributed control algorithm is carried out by selecting the parameter of incremental cost for each generator. The authors of [35] put forward a distributed algorithm based on two-consensus approach. The first-order consensus protocol focuses on the local mismatch and ensures supply–demand equality, and the second-order consensus strategy, called most up-to-date information, estimates the power mismatch in the system, the two consensus approach runs in parallel. The authors of [36] proposed a distributed algorithm based on consensus-like iterative, and adopted bisection search approach to obtain optimal incremental cost parameter in distributed way. A full decentralized algorithm for EDP is presented in [37], in which a flooding-based consensus method is used to obtain total power demand by communicating with neighbors of each node in the power system. The authors of [38] proposed an optimization algorithm based on auction techniques and consensus protocols to solve EDP with non-convex cost function. Considering the distribution losses in EDP, the authors of [39] gave a distributed approach to collect active power loss in power system, and the solution is reached by local computations.

In the smart grid environment, the approaches for solving EDP are changing from centralized to distributed. In distributed algorithms, data storage and processing can be performed on local nodes, which can avoid the risk of collapse of the whole algorithm due to the single-node-failure. Moreover, distributed algorithm can better meet the plug-and-play characteristics of the smart grid.

In view of the merits of distributed algorithm, this paper proposes a distributed approach to solve economic dispatch problem for smart grids. The present algorithm consists of two stages: data collection and optimal solution. An absorption search and a flooding-based consensus approach are applied successively in the data collection stage. A distributed half-search algorithm is used to obtain power values of generation units as final result of EDP. The contributions of this article are as follows:

1. Simplified network structure by absorption search: In the network simplification stage, nodes in the system can share information and communicate with neighbors. Each time information is shared, the generation nodes (nodes in which the generator unit is connected) will update their status information by recording the current information of their neighbors, and the load nodes (nodes in which no generator unit is connected) whose information is recorded by their generation neighbor will disappear from the network at the current stage.
2. Flooding-based consensus (FBC) approach for collection information: A flooding-based algorithm is used to gather information, and each node (agent) can obtain consistent information about its neighbors in the power network; this process of collecting data is carried out in a distributed way.
3. Distributed half-search algorithm to obtain optimal solution in distributed manner: When only generation nodes are included in the network, each agent in the network can be competent for the result of EDP. A fully decentralized system based on half-search is used to solve EDP.

The remainder of this paper is organized as follows. Section 2 introduces the basic theory used in this article, including graph theory and consensus protocols. In Section 3, the EDP and the optimal solution are discussed. Section 4 presents the key network simplification and distributed half-search algorithm. Section 5 presents the analysis of the numerical simulation of a series of case studies. In Section 6, we conclude the paper.

2. Preliminary

In this section, basic preliminaries including some concepts in graph theory and consensus algorithm are introduced, which is needed in the following discussion.

2.1. Graph Theory

A (directed) graph [40,41] G is composed of vertices and edges, and is denoted by ordered sets $G = \{V(G), E(G)\}$, where $V(G) = \{1, 2, ..., n\}$ and $E(G) \subseteq V(G) \times V(G)$ are the finite nonempty sets of the vertices and the edges, respectively. Define a symbolic relationship between ordered vertices in directed graph G, that is, $F(e) = (\mu, \omega)$, which means that (μ, ω) denotes a path from node μ to node ω. μ and ω are the tail and head of e, respectively. An undirected graph G is a special form of the directed graph with a bidirectional path by exchanging the direction of each edge (μ, ω) in G.

Define an adjacency matrix $A(G) = (a_{ij})_{n \times n}$ that associates with G (directed or undirected), and the index set of generation nodes in the smart grid can be represented by $V(G)$. The directed path $(i, j) \in E(G)$ shows that node i can send information to node j, and node j can receive information from node i. In this paper, we assume each node $i \in V(G)$ does not belong to the neighbors of itself, that is, there is no self-loop in a directed graph. However, node i is allowed to receive its own information. Let $\mathcal{N}_i^+ = \{j \in V(G) \mid (j, i) \in E(G)\}$ and $\mathcal{N}_i^- = \{j \in V(G) \mid (i, j) \in E(G)\}$ denote the in-neighbor set and out-neighbor set of ith node, respectively. Physically, it implies that a node $i \in V(G)$ can send information to any node $j, (j \in \mathcal{N}_i^-)$ and receive information from any node $j, (j \in \mathcal{N}_i^+)$. Let $d_i^{\uparrow} = \mid \mathcal{N}_i^+ \mid$ and $d_i^{\downarrow} = \mid \mathcal{N}_i^- \mid$ represent the in-degree and out-degree of node i, respectively, where $\mid \cdot \mid$ denotes the cardinality of a set; it is the number of elements in set "$\cdot$". If there is a path, (i, j) between any two nodes (i and j) in G, the graph can be treated as strongly connected. The connection between the smart grid nodes constitutes the strong connected graph, distinctly, $d_i^{\uparrow} \neq 0$ and $d_i^{\downarrow} \neq 0$ for each node $i \in V(G)$ in a strongly connected graph. Further, considering undirected graphs, it can be concluded that $d_i^{\uparrow} = d_i^{\downarrow} = d_i = \mid \mathcal{N}_i \mid$, where $\mid \mathcal{N}_i \mid = \{j \in V(G) \mid (i, j) \in E(G)\}$ and the number of neighbors of node i is represented by d_i.

Based on the characteristics of power system, the concepts of directed and undirected graphs are used to study power flow and communication between nodes in power networks. Especially in solving consistency based on multi-agent system [42,43], network computing is indispensable.

2.2. Consensus Algorithm

For $G = (V, E)$ that is strongly connected, define a non-negative adjacency matrix $\mathcal{Q} \subseteq \mathbb{R}^{n \times n}$ as follows:

$$\mathcal{Q} = \{q_{ij}\} = \begin{cases} \dfrac{1}{\mid \mathcal{N}_j^+ \mid +1} & \text{if } (j,i) \in E, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

It is not difficult to prove that $[\mathcal{Q}]_{ij} > 0$ for $(j,i) \in E$, and we can verify that the sum of column entries of $\mathcal{Q}$ is 1, i.e., $\sum_{i=1}^{N} q_{ij} = 1$. This shows that the matrix $\mathcal{Q}$ is a column stochastic matrix associated with graph G. Similarly, $\mathcal{Q}^T$ is a row stochastic matrix. According to the properties of the stochastic matrix, we have $\mathcal{Q}^T\mathbf{1} = 1\mathbf{1}$, where $\mathbf{1} = [1, 1, ..., 1]^T$.

We consider all nodes in the smart grid work together as a team of N autonomous agents; it is assumed that each agent can define a subjective probability distribution for itself but does not know the subjective probability distribution of other agents. Label them from 1 to n and they can achieve a common goal value with certain constraint conditions. To explore the feasibility of this problem, we can learn from DeGroot's model of consensus [44]. In this model, the opinions of each agent are gathered into an opinion pool and each agent adjusts the subjective probability distribution of itself by merging its own subjective probability distribution and learning the subjective probability distribution of other agents. Finally, the subjective probability distribution of all agents can reach a common value, and a certain subjective probability distribution parameter θ is formed on the basis, where θ can be considered as any discretionary variable whose value is a subset of the abstract parameter space Ψ. The subjective probability distribution assigned by each monomer i to parameter θ is represented by $\Phi_i \in \mathbb{R}$, and the subjective probability distributions of all agents is denoted by $\Phi = [\Phi_1, \Phi_2, ..., \Phi_n]^T \in \mathbb{R}^n$ on space Ψ. It is assumed that the subjective probability distributions of all agents in different backgrounds on parameter space θ are denoted by $\mathcal{Q}_i = [q_{i1}, q_{i2}, ..., q_{in}]$, and $\sum_{i=1}^{n} q_{ij} = 1$, $q_{ij} \geq 0$. The stochastic $\mathcal{Q}$ associated with G denotes the $n \times n$ matrix consisting of the elements q_{ij}, and we assume that it is feasible for each monomer i to update its subjective probability distribution from $\Phi(t)$ to $\Phi(t+1)$. The linear iteration process for the update is as follows:

$$\Phi(t+1) = \mathcal{Q}\Phi(t) = ... = \mathcal{Q}^{(t+1)}\Phi(0) \tag{2}$$

The iteration index of the linear iteration is denoted by $t = 0, 1, 2, ...$, and the initial value is denoted by $\Phi(0)$. For the column stochastic matrix $\mathcal{Q}$ associated with a undirected graph G, the element $\Phi(t+1)$ shown in Equation (2) can be expressed as:

$$\Phi_i(t+1) = q_{ii}\Phi_i(t) + \sum_{j \in \mathcal{N}_i} q_{ij}\Phi_j(t), \quad \forall i = 1, ..., n \tag{3}$$

where q_{ii} and q_{ij} are the elements located on the diagonal and the ith row, jth column in matrix $\mathcal{Q}$, respectively.

Before studying the asymptotic consensus performance shown in Equation (2), we give two related theorems on the consensus problem.

Theorem 1. *[45] Given a non-negative matrix $\mathcal{Q}$ associated with a strongly connected graph G, the $\mathcal{Q}$ is a primitive matrix if G is aperiodic.*

Theorem 2. *[46] If matrix $\mathcal{Q} \in \mathbb{R}^{n \times n}$ is a primitive matrix and $q_{ij} \geq 0$, which is the element of $\mathcal{Q}$, then,*

$$\lim_{t \to \infty} [\rho(\mathcal{Q})^{(-1)}\mathcal{Q}]^t = \mathcal{U}\mathcal{V}^T > 0$$

where $\rho(\mathcal{Q})$ is the spectral radius of $\mathcal{Q}$. $\mathcal{U}$ and $\mathcal{V}$ are the right and left Perron vectors of $\mathcal{Q}$, respectively.

Considering Theorem 1, we have

$$\lim_{t\to\infty} \mathcal{Q}^t = \lim_{t\to\infty}((\mathcal{Q}^T)^t)^T = (\mathbf{1}\zeta^T)^T = \zeta\mathbf{1}^T \tag{4}$$

where $\mathbf{1} = [1, 1, ..., 1]^T$ and $\zeta = [\zeta_1, \zeta_2, ..., \zeta_n]^T$ is the right eigenvector with the character $\zeta_i > 0$ corresponding to the matrix $\mathcal{Q}$ at the eigenvalue 1. For all nodes i, there exists a properties equation, that is, $\mathbf{1}^T\zeta = 1$.

Considering Theorem 2, let $\rho(\mathcal{Q}) > 0$ denote a specific eigenvalue of $\mathcal{Q}$ (spectral radius). Consider the matrix $\mathcal{Q}$, which is nonnegative and stochastic, such that $\rho(\mathcal{Q}) = 1$. $\mathcal{U} > 0$ and $\mathcal{V} > 0$ are the right Perron vector and left Perron vector of $\mathcal{Q}$ with the following properties: $\mathcal{Q}\mathcal{U} = \rho(\mathcal{Q})\mathcal{U}$, $\mathcal{V}^T\mathcal{Q} = \rho(\mathcal{Q}\mathcal{V}^T$ and $\mathcal{U}^T\mathcal{V} = 1$.

From Equations (2) and (4), the consensus algorithm satisfies the following property:

$$\Phi^* = \lim_{t\to\infty} \Phi(t) = \zeta\mathbf{1}^T\Phi(0) = (\sum_{i=1}^{n} \Phi_i(0))\zeta \tag{5}$$

For any initial state $\Phi(0)$, if there exists $\Phi^* \in \mathbb{R}$, such that $\lim_{t\to\infty} \Phi_i(t) = \Phi^*$, $\forall i = 1, 2, ..., n$, the system state (shown in Equation (3)) has a stable value after t times iteration.

Based on the above mentioned theorems, we can draw a conclusion: the system (shown as in Equation (2)) state of each monomer i can converge to a stable value, and the convergence performance is determined by the initial value $\Phi(0)$ and the strongly connected topology $\mathcal{Q}$.

3. Problem Formulation

In this paper, we only consider that EDP with a quadratic cost function [47]. The basic problem of economic operation of power system with quadratic cost function can be considered as a convex optimization problem. It is widely used in the traditional EDP. It can be solved analytically without any approximations. Furthermore, the duality theory can be used to solve this problem.

3.1. Economic Dispatch Problem

We first establish a traditional power system model with M buses in which N generation units are included. Considering that the actual power system network may not contain a generation unit on every bus, we can prescribe a limit to $M > N$. The EDP can be expressed as the following form:

$$\min \quad \sum_{g=1}^{N} C_g(P_{Gg}) \tag{6}$$

$$s.t. \quad \underline{P}_{Gg} < P_{Gg} < \overline{P}_{Gg} \tag{7}$$

$$\sum_{j=1}^{M} P_{Dj} = \sum_{g=1}^{N} P_{Gg} = P_0 \tag{8}$$

where $C_g(P_{Gg})$ is generation cost function. P_{Gg} and P_{Dj} denote the power generated and load demand by generator g and bus j, $(g = 1, 2, ..., N; j = 1, 2, ..., M)$, respectively. From inequality in Equation (7), we can know that the power generated P_{Gg} is restricted in the corresponding maximum $\overline{P}_{Gg}$ and minimum $\underline{P}_{Gg}$ bounds. The total load demand on all buses is denoted by P_0 and the generation cost function is denoted by $C_i(P_{Gg})$, which is approximated as the follow quadratic function:

$$C_i(P_{Gg}) = a_i P_{Gg}^2 + b_i P_{Gg} + c_i \tag{9}$$

where $a_i > 0$, $b_i \geq$ and $c_i \geq 0$ are the parameters of cost function. To simplify the expression, Equation (9) can be rewritten as:

$$C_i(P_{Gg}) = \frac{(P_{Gg} - \alpha_i)^2}{2\beta_i} + \gamma_i \tag{10}$$

where $\beta_i > 0$, $\alpha_i \leq 0$. Obviously, the necessary and sufficient condition for the above problems to be feasible is

$$\sum_{g=1}^{N} \underline{P}_{Gg} \leq P_0 \leq \sum_{g=1}^{N} \overline{P}_{Gg} \tag{11}$$

The model of optimization problem shown as above is applied in this paper to solve EDP.

3.2. Solution

It can easily be proven that generation cost function $C_g(P_{Gg})$ is strictly convex, and all the constraints of the EDP are linear, therefore the EDP can be treated as s convex-quadratic optimization problem. In other words, such a strict convex function can guarantee the establishment of the strong duality theorem, and then, we can set up the Lagrange dual problem of the original problem shown as Equations (6)–(8), and the dual optimal solution obtained is also the solution of the original problem. According to the inverse form of the dual problem, the Lagrange dual problem is established as a non-differentiable concave function with the following structure:

$$\max\left(\sum_{g=1}^{n} C_g^{\mathcal{L}}(\lambda) + \lambda P_0\right) \tag{12}$$

where λ denote the Lagrangian multiplier and $C_g^{\mathcal{L}}(\lambda)$ denotes the the Lagrange dual cost function. The incremental cost rates of the generator unit g is denoted by

$$u_g(P_{Gg}) = \frac{\partial C_g(P_{Gg})}{\partial P_{Gg}} = \frac{P_{Gg} - \alpha_i}{\beta_i}, \forall g \in V_n, \tag{13}$$

where $\lambda \in \mathbb{R}$ is the Lagrange multiplier and

$$C_g^{\mathcal{L}}(\lambda) = \begin{cases} C_g(\underline{P}_{Gg}) - \lambda \underline{P}_{Gg}, & \lambda < u_g(\underline{P}_{Gg}), \\ -\lambda(\alpha_i + \dfrac{\lambda\beta_i}{2}), & u_g(\underline{P}_{Gg}) \leq \lambda < u_g(\overline{P}_{Gg}), \\ C_g(\overline{P}_{Gg}) - \lambda \overline{P}_{Gg}, & u_g(\overline{P}_{Gg}) \leq \lambda. \end{cases} \tag{14}$$

The above-mentioned Lagrangian dual problem can be established depending on whether the following equation or inequality condition can be satisfied, which includes power generated upper and lower bounds constraints and power generated balance constraints. The constraints mentioned above are written as,

$$\begin{cases} \dfrac{\partial C_g(P_{Gg})}{\partial P_{Gg}} = u_g(P_{Gg}) \\ \underline{P}_{Gg} \leq P_{Gg} \leq \overline{P}_{Gg} \\ \sum_{j=1}^{M} P_{Dj} = \sum_{g=1}^{N} P_{Gg} \end{cases} \tag{15}$$

Based on the above analysis, we can define a mapping as follows

$$\tau_g(\lambda) = \frac{\partial C_g^{\mathcal{L}}(\lambda)}{\partial \lambda} = \begin{cases} -\underline{P}_{Gg}, & \lambda < u_g(\underline{P}_{Gg}), \\ -(\alpha_i + \lambda \beta_i), & u_g(\underline{P}_{Gg}) \leq \lambda < u_g(\overline{P}_{Gg}), \\ -\overline{P}_{Gg}, & u_g(\overline{P}_{Gg}) \leq \lambda. \end{cases} \tag{16}$$

Based on the above discussion, we can conclude that, under the restriction of power generated and load demand balance, if the primal problem shown as Equations (6)–(8) is solvable, we can easily get λ^* as the unique optimal solution, which satisfies

$$P_0 = -\sum_{g=1}^{n} \tau_g(\lambda^*)$$

According to the property of Lagrange's dual theorem and the strong duality theorem, we can verify that there is no duality gap between the solution of the primal problem and the dual problem, which means that the primal problem in Equations (6)–(8) exists where the unique optimal primal P_{Gg}^* and optimal dual $-\tau_g(\lambda^*)$ are equivalent in function value. That is, $P_{Gg}^* = -\tau_g(\lambda^*)$, $g = 1, 2, ..., n$, i.e.,

$$P_{Gg}^* = \begin{cases} \underline{P}_{Gg}, & \lambda^* < u_g(\underline{P}_{Gg}), \\ \alpha_i + \lambda^* \beta_i, & u_g(\underline{P}_{Gg}) \leq \lambda^* < u_g(\overline{P}_{Gg}), \\ \overline{P}_{Gg}, & u_g(\overline{P}_{Gg}) \leq \lambda^*. \end{cases} \tag{17}$$

We can discover from the optimal solution of the EDP that the constant coefficient γ_i has no effect on the cost increment. From Equation (17), we can conclude that, once the value of parameter λ^* is determined, the optimal solution of the EDP is determined.

4. Fully Distributed Solution for EDP

We divide the proposed approach into two stages. The first stage is network simplification, in which the load connected to each bus is incorporated into the nodes with generator units. A distributed approach based on FBC is used as the way of information dissemination. In the second stage, we propose a distributed solution for the optimal solution of EDP. The algorithm is implemented in the platform of Java agent development framework (JADE), which is an effective middleware for the development of multi-agent systems.

4.1. Absorption Search

To solve the EDP, we need to collect the load power demand in the first phase. However, it is not easy to calculate the load demand $P_0 = \sum_{j=1}^{M} P_j$ with a distributed method. We complete data collection with two steps: shrinking the size of network and exchanging information based on FBC. The concept of flooding algorithms [48,49] are applied by FBC, which is used in data networks for broadcasting. In this paper, this communication method is used between adjacent agents.

Each node in the power network may only contain pure generator unit, pure load unit, or both. Assume that an agent is embedded in each node to share information. It means that agents can send (receive) messages to (from) their neighbors in the power system. We create agents in JADE; each agent has a unique identification (ID) that consists of the order number and the node type. There are two kinds of node types in this paper, generating node (the node with generator unit) and load node (pure load connected to the node), and they are represented by $(ID)_N$ and $(ID)_M$, respectively, where (ID) is the order number, and subscripts N and M denote generation node and load node, respectively. Assume that each agent knows its neighbor agent's ID and has the sufficient ability of computation. Each node i updates information by collecting and recording the information which includes the power data of the neighbor nodes and the ID of all neighbors of the neighbor.

At the first step of absorption search, load agents hold the message shown as follow,

$$msg\_j = \{ID_M, pm_{id}, [ID_{Mneibs}], [ID^*_{Mneibs}]\}$$

where msg_j is the set of information, ID_M is the ID of the current load node, pm_{id} and ID_{Mneibs} are the load demand and the ID set of its neighbors, and $[ID^*_{Mneibs}]$ is alternative neighbor set, whose specific meaning is given below. Load agent will log off after successfully sending his information to the adjacent generation agent. Figure 1 gives the life cycle of load agents.

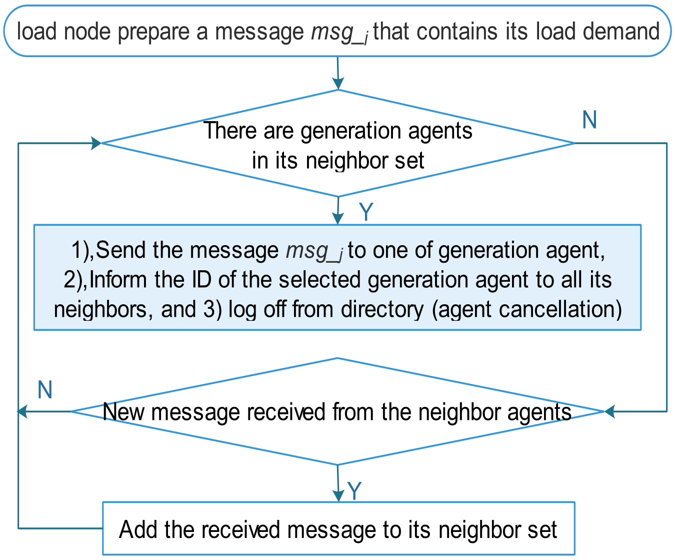

Figure 1. Life cycle of load agent.

We create generation agent with tuple,

$$msg\_i = \{[ID_N, (pm_{id}), MSG_{VA}], [< ID_M > (ID_{Mneibs})], [ID_{Nneibs}], [ID^*_{Nneibs}]\}$$

where ID_N represents the ID of the current generation node, and pm_{id} is a parameter set that includes the load demand P_{Di}, the upper bound $(\overline{P}_{Gg})$ and lower bound $(\underline{P}_{Gg})$ of the generator unit that is connected to the current node. ID_M is the ID set of absorbed nodes at current iteration. ID_{Mneibs} is the set of neighbors ID that connect to the absorbed nodes, $[ID_{Nneibs}]$ is the set of neighbors ID of the generation node, and $[ID^*_{Nneibs}]$ is the set of alternative neighbors, which is designed for the generation agent that contains only one neighbor, and the neighbor corresponds to the type of load. Alternative neighbor sets prevent isolated nodes in the process of network simplification. When a generation agent receives a massage from its load neighbors, it updates its message. Furthermore, define a virtual agent (VA) in JADE, and each generation agent sends a message MSG_{VA} to VA after each iteration step. $MSG_{VA} = 1$ if the message sender receive new message from his neighbors, otherwise $MSG_{VA} = 0$. All generation agents can receive feedback information from VA. If the received messages from generation agents are equal to zero in one iteration step, then VA will send message $MSG_{VA} = 0$ as feedback information to all generation agent. Here, we obtain the final simplified network without load nodes.

The process of simplifying network is shown in Figure 2.

To illustrate the process of simplifying network, we take IEEE 14-bus system as an example, and the multi-agent system based on IEEE 14-bus system is shown in Figure 3. The blue dashed line connecting agents $8N$ and $4M$ represents that $4M$ is an element of 8^*_{Nneibs}.

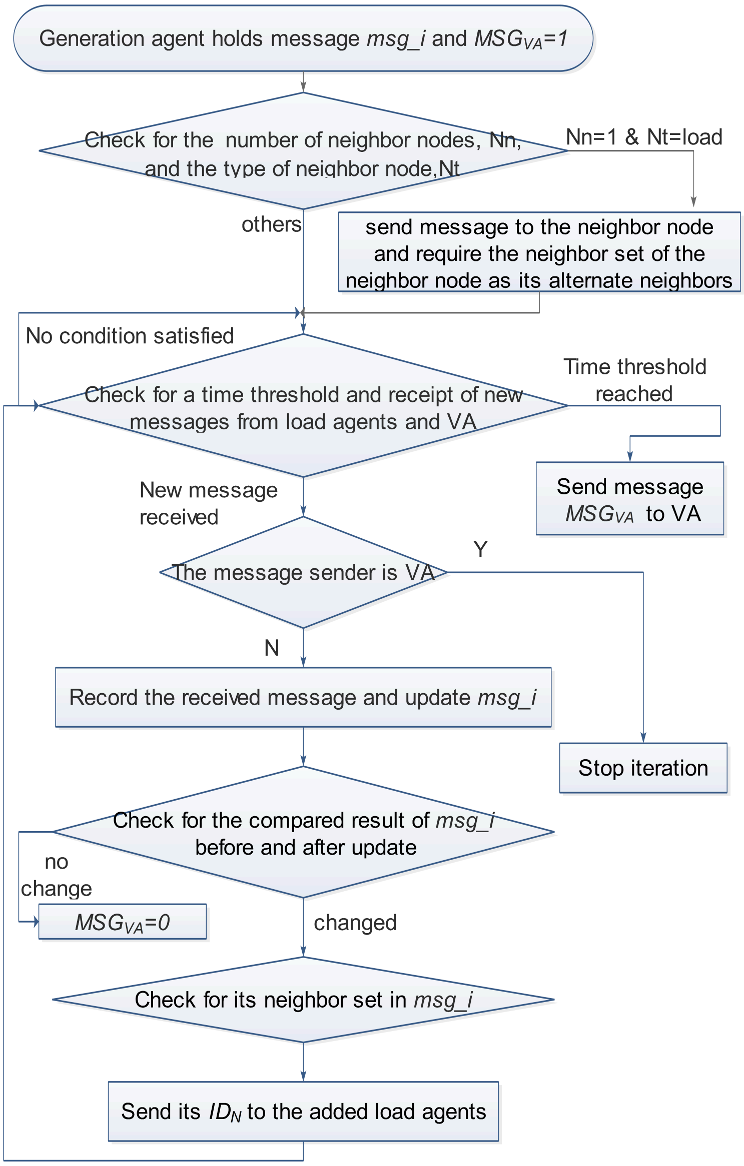

Figure 2. Flow chart of simplifying network.

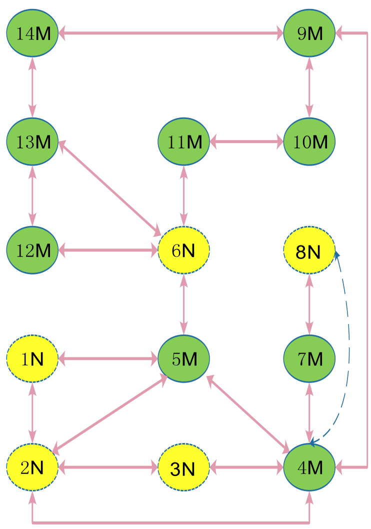

Figure 3. Multi-agent system for IEEE 14 bus system.

Before iteration, each agent holds the following messages:

$ST = 0$

$\{[1_N, (pm_1), MSG_{VA} = 1], [<> ()], [2_N, 5_M], []\}$ $\{[2_N, (pm_2), MSG_{VA} = 1], [<> ()], [1_N, 3_N, 4_M, 5_M], []\}$

$\{[3_N, (pm_3), MSG_{VA} = 1], [<> ()], [2_N, 4_M], []\}$ $\{[4_M, pm_4], [2_N, 3_N, 5_M, 7_M, 9_M], [\mathbf{8_N}]\}$

$\{[5_M, pm_5], [1_N, 2_N, 3_N, 4_M, 6_N], []\}$ $\{[6_N, (pm_6), MSG_{VA} = 1], [<> ()], [5_M, 11_M, 12_M, 13_M], []\}$

$\{[7_M, pm_7], [4_M, 8_N]\}$ $\{[9_M, pm_9], [4_M, 10_M, 14_M], []\}$ $\{[8_N, (pm_8), MSG_{VA} = 1], [<> ()], [7_M], [\mathbf{4_M}]\}$

$\{[10_M, pm_{10}], [9_M, 11_M], []\}$ $\{[11_M, pm_{11}], [6_N, 10_M], []\}$ $\{[12_M, pm_{12}], [6_N, 13_M], []\}$

$\{[14_M, pm_{14}], [9_M, 13_M], []\}$ $\{[13_M, pm_{13}], [6_N, 12_M, 14_M], []\}$

In the first iteration, we take the load agent $5M$ as an example; its neighbor set contains five elements, and four of them are generation agents. Agent $5M$ randomly selects an generation agent as message receiver, and send its message to the selected agent. When the information is successfully transmitted, agent $5M$ informs all its neighbors the generation agent ID which it selected, and logs off from the current agent system. This process can be called absorption search: the generation agent receives information from load agent and updates its own information according the received message, then the data from the load agent are recorded by generation agent, and at this moment the log off of the load agent does not affect the final results of network computing. In this way, the network size is reduced.

During the first iteration cycle, the agent's information changes as follows, and the corresponding multi-agent system network is shown in Figure 4a.

$ST = 1$

$\looparrowright$ *Receive message from load agent*

$\{[1_N, (pm_1)], [< 5_M > (1_N, 2_N, 4_M, 6_N)], [2_N, 5_M]\}$ $\{[2_N, (pm_2)], [<> ()], [1_N, 3_N, 4_M, 5_M]\}$

$\{[3_N, (pm_3)], [< 4_M > (2_N, 3_N, 5_M, 7_M, 9_M, \mathbf{8_N})], [2_N, 4_M]\}$

$\{[6_N, (pm_6)], [< 11_M, 12_M, 13_M > (6_N, 10_M, 14_M)], [5_M, 11_M, 12_M, 13_M]\}$

$\{[8_N, (pm_8)], [< 7_M > (4_M, 8_N)], [7_M, \mathbf{4M}]\}$ $\{[9_M, (pm_9)], [4_M, 10_M, 14_M)\}$

$\{[10_M, (pm_{10})], [9_M, 11_M]\}$ $\{[14_M, (pm_{14})], [9_M, 13_M]\}$

$\looparrowright$ *Update information*

$\{[1_N, (pm_1, pm_5)], [<> ()], [2_N, \mathbf{6_N}]\}$ $\{[2_N, (pm_2)], [<> ()], [1_N, 3_N]\}$

$\{[3_N, (pm_3, pm_4)], [<> ()], [2_N, \mathbf{9_M}, \mathbf{8N}]\}$ $\{[6_N, (pm_6, pm_{11}, pm_{12}, pm_{13})], [<> ()], [\mathbf{10_M}, \mathbf{14_M}]\}$

$\{[8_N, (pm_8, pm_7)], [<> ()], [\mathbf{3_N}]\}$ $\{[9_M, (pm_9)], [\mathbf{3_N}, 10_M, 14_M)\}$

$\{[10_M, (pm_{10})], [9_M, \mathbf{6_N}]\}$ $\{[14_M, (pm_{14})], [9_M, \mathbf{6_N}]\}$

Obviously, each load agent selects one of its neighbors among generation agent in a random way, thus the structure of simplified network is not unique. After the second iteration, there is no load agent in the multi-agent system, and the message of each agent is shown as follow.

$ST = 2$

......

$\{[1_N, (pm_1, pm_5)], [<> ()], [2_N, 6_N]\}$ $\{[2_N, (pm_2)], [<> ()], [1_N, 3_N]\}$

$\{[3_N, (pm_3, pm_4, pm_9)], [<> ()], [2_N, 8N, \mathbf{6N}]\}$ $\{[8_N, (pm_8, pm_7)], [<> ()], [\mathbf{3_N}]\}$

$\{[6_N, (pm_6, pm_{11}, pm_{12}, pm_{13}, pm_{10}, pm_{14})], [<> ()], [\mathbf{3_N}]\}$

Figure 4 shows the absorption search approach for IEEE 14 bus.

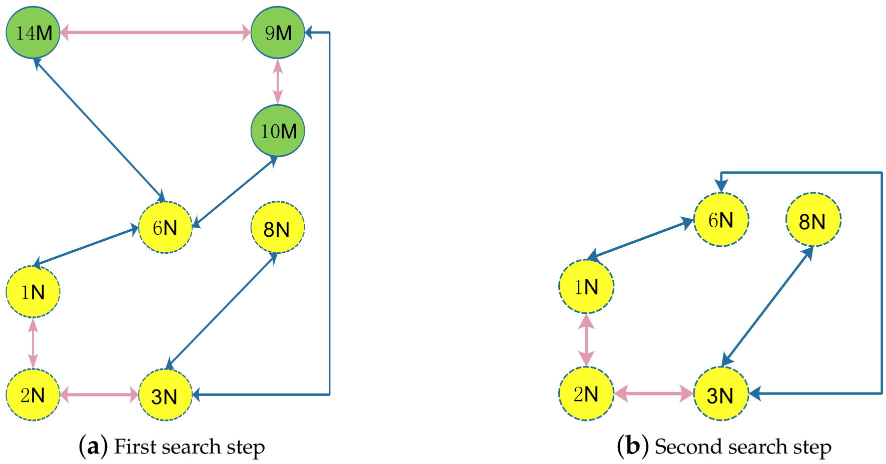

Figure 4. Absorption search for IEEE 14 bus.

4.2. Distributed Half-Search Algorithm

After the first stage of information collection, the structure of the original communication network G_M is reduced to the communication network $G_N = (V_N, E_N)$. For every node $i \in V_n$, we can get the summation of load demand P_i collected by each generation node. Since all load nodes act as the dependency nodes of the bus where the generation node is determined, we now propose a fully distributed algorithm in the communication network G_N to solve EDP. For the needs of the algorithm, we define an adjacency matrix $A = \{a\}_{ij}$ associated with G_N.

$$A = \{a_{ij}\} = \begin{cases} \dfrac{1}{|\mathcal{N}_{n,j}^{+}| + 1} & if(j,i) \in E_n \\ 0 & otherwise. \end{cases} \tag{18}$$

Now, it is reasonable to assume that the non-negative matrix A can be used as the state matrix of each generation node in the topological G_N. For $i \in V_N$, a corresponding variable $g_i(t)$ is established for each generation node as the carrier variable of the power load carried by the generation node. And the total power collected by each generation node from the neighboring node is taken as the initial value of the variable, that is $g_i(0) = P_i$. Then, the following iterative algorithm is applied to reallocate the load power of each generation node.

$$g_i(t+1) = a_{ii} g_i(t) + \sum_{j \in N_{n,i}} a_{ij} g_j(t), \tag{19}$$

where $g_i(t)$ is the carrier variable of the power load carried by the generation node, and $j \in N_{n,i}$ denotes all neighbors of node i in G_N. For any node $i \in V_n$, the computation in Equation (19) applied to each generation node will converge after t-time iterations. That is, $g^* = \lim_{t \to \infty} g(t)$. The following equation is derived from Equation (5).

$$g_i^* = \left(\sum_{j=1}^{n} g_j(0) \right) \zeta_i = P_0 \zeta_i \tag{20}$$

where $\zeta = [\zeta_1, \zeta_2, ..., \zeta_n]^T$ is the right eigenvector with the character $\zeta_i > 0$ corresponding to the matrix A at the eigenvalue 1, and g_i^* is the final load information obtained by each generation node in G_N.

Define the basic variables $\lambda(k)$, $\lambda^{\uparrow}(k)$, $\lambda^{\downarrow}(k)$ required by the algorithm, where $\lambda(k)$ represents the Lagrange multiplier for each iteration, and the upper and lower bounds of $\lambda(k)$ are denoted by $\lambda^{\uparrow}(k)$ and $\lambda^{\downarrow}(k)$, respectively. The step index of the half-search algorithm is denoted by $k \geq 0$. The initializations of $\lambda^{\uparrow}(k)$ and $\lambda^{\downarrow}(k)$ may take any values as long as $\lambda^{\uparrow}(k)$ is large enough and $\lambda^{\downarrow}(k)$ is small enough. To make the interval as tight as possible, we initialize with the following equation:

$$\begin{bmatrix} \lambda^{\downarrow}(0) \\ \lambda^{\uparrow}(0) \end{bmatrix} = \begin{bmatrix} \min_{i\in V_n} u_i(\underline{P}_{Gi}) \\ \max_{i\in V_n} u_i(\overline{P}_{Gi}) \end{bmatrix}$$

This means that the closer are $\lambda^{\downarrow}(0)$ and $\lambda^{\uparrow}(0)$ to the optimal Lagrangian λ^*, the fewer iterations are needed. There are multiple ways to define and solve for variable $\lambda(k)$, and for simplification, $\lambda(k)$ as an approximation of the optimal Lagrangian λ^*, that is

$$\lambda(k) = \frac{1}{2}[\lambda^{\uparrow}(k) + \lambda^{\downarrow}(k)]. \tag{21}$$

In the communication network graph $G_N = (V_n, E_n)$, for all nodes $i \in V_n$, the output of each generation node is assigned as follows:

$$P_{Gi}(k) = -\tau_i(\lambda(k)) \tag{22}$$

Then , for all nodes $i \in V_n$, define an auxiliary variable $\mathcal{X}_i(t)$ as the power generated carried by each generation node and initialized by $\mathcal{X}_i(0) = P_{Gi}(k)$. Then, the following iterative algorithm is applied to compute $\mathcal{X}_i(t)$.

$$\mathcal{X}_i(t+1) = a_{ii}\mathcal{X}_i(t) + \sum_{j\in N_{n,i}} a_{ij}\mathcal{X}_j(t). \tag{23}$$

where $\mathcal{X}_i(t)$ is the power generated carried by node. Let us define $\mathcal{X}^* = \lim_{t\to\infty} \mathcal{X}_i(t)$, and then, we get the convergent generators output variable $\mathcal{X}^*$ corresponding to the current $\lambda(k)$ according to Equation (5):

$$\mathcal{X}_i^* = \left(\sum_{j=1}^{n} P_{Gj}(k)\right)\zeta_i, \quad \forall i \in V_n \tag{24}$$

Now, the half-search algorithm is proposed. In the communication topology graph G_N, generation node i updates the values of the current upper bound $\lambda^{\uparrow}(k+1)$ and lower bound $\lambda^{\downarrow}(k+1)$ of the Lagrange multiplier $\lambda(k)$ by comparing the magnitude of the local load information g^* and the local output $\mathcal{X}^*$ as follows:

$$\begin{bmatrix} \lambda^{\uparrow}(k+1) \\ \lambda^{\downarrow}(k+1) \end{bmatrix} = \begin{bmatrix} \lambda(k) \\ \lambda^{\downarrow}(k) \end{bmatrix} \quad for \mathcal{X}^* > g^*, \tag{25}$$

$$\begin{bmatrix} \lambda^{\uparrow}(k+1) \\ \lambda^{\downarrow}(k+1) \end{bmatrix} = \begin{bmatrix} \lambda^{\uparrow}(k) \\ \lambda(k) \end{bmatrix} \quad for \mathcal{X}^* \le g^*. \tag{26}$$

From Equations (21), (25) and (26), it is easy to get that $\lambda^* = \lim_{k\to\infty} \lambda(k)$, and then, the output of each generation node can get a local optimal solution from Equation (22), that is

$$P_{Gi}^* = -\tau_i(\lambda^*), \quad \forall i \in V_n \tag{27}$$

The distributed half-search algorithm is summarized in Algorithm 1.

Algorithm 1 Distributed half-search algorithm.

Input: $P_i, i = 1, 2, ..., N$:load demand for each generation agent;
P_{Gi}^{min}: lower limit for generation unit i;
P_{Gi}^{max}: upper limit for generation unit i;
a_i, b_i or α_i, β_i: cost coefficient; k=0.
ϵ: a sufficiently small positive number greater than 0

Output: P_{Gi}^{*}:generating power for each generation unit

Initialize the Lagrange multiplier $\lambda^{\uparrow}(0)$ and $\lambda^{\downarrow}(0)$
repeat
  Obtain $\lambda(k)$ by Equation(21)
  Carry out $P_{Gi}(k)$ by Equations (16) and (22)
  Calculate Equation (24)
  Complete calculation as shown in Equations (25) and (26)
  k=k+1
until $\lambda^{\uparrow}(k) - \lambda^{\downarrow}(k) \leq \epsilon$
return $P_{Gi}^{*} = P_{Gi}(k)$

4.3. *Algorithm Analysis*

We divide the proposed algorithm into two parts for analysis. First, we need to prove that the algorithm is convergent asymptotically; and second, we need to formulate a stop criterion for the algorithm.

4.3.1. Proof of Convergence

Before proving the convergence of the proposed algorithm, the following two remarks are given, which are used in the following proof.

Remark 1. *If there exists an optimal solution to the original EDP (shown as in Equations (6)–(8), then there exists a positive k that, when $k \to \infty$, the proposed algorithm can converge to the globally unique optimal solution asymptotically.*

Remark 2. *If a continuous function $f(x)$ is strictly monotonic in a bounded interval $[a, b]$, then the function $f(x)$ converges in a bounded interval $[f(a), f(b)]$.*

Proof. From Equation (13), for any node $i \in V_n$, we can get the incremental cost rates of the generation node i $u_i(P_{Gi})$. $\lambda^{+} = \max(u_i(\overline{P}_{Gi}))$ and $\lambda^{-} = \min(u_i(\underline{P}_{Gi}))$ denote the upper and lower bounds of λ, respectively. Since the original EDP is feasible, we can get $\lambda^{-} \leq \lambda \leq \lambda^{+}$.

From Equation (16), for any node $i \in V_n$, we can easily get that the function $\tau_i(P_{Gi})$ is about monotonous continuous decrementing of λ. Therefore, $-\sum_{i=1}^{n} \tau_i(P_{Gi})$ is strictly monotonically increasing in interval $\lambda \in [\lambda^{-}, \lambda^{+}]$. From Equation (22), $P_{Gi}(k) = -\tau_i(\lambda(k))$, for any node $i \in V_n$, thus we can get $\sum_{i=1}^{n} P_{Gj}(k)$ is also strictly monotonically increasing in interval $\lambda \in [\lambda^{-}, \lambda^{+}]$.

According to Remark 2, we can easily get that the distributed algorithm we proposed is convergent. In this paper, the EDP is constructed as astrictly convex optimization problem, therefore the distributed algorithm proposed can converge to a unique global optimal solution. □

4.3.2. Stopping Criterion

For the half-search algorithm proposed in this paper, we give the algorithm stop criterion. We know from the discussion of the above algorithm that the lower and upper bounds of $\lambda(k)$ as the search interval of the algorithm will be reduced by half after each search. Therefore, $\lambda(k)$ in this algorithm can rapidly converge to the unique optimal solution λ^*. At the same time, the optimal solution of the original power system EDP can be directly obtained as $P^*_{Gi} = -\tau_i(\lambda^*)$ by solving the Lagrange multiplier optimal solution λ^*. Theoretically, as long as the preset step size k value can ensure the convergence of the algorithm, the stopping criterion can be set to stop when the algorithm runs to the preset k value. However, this stopping criterion does not satisfy the flexibility of the algorithm. Now, a stopping criterion with algorithm change is given as follow:

$$\left[k(|\lambda^{\uparrow}(k) - \lambda^{\downarrow}(k)|) - \delta\right] \leq 0$$

where k is the iteration step size, and δ is a preset positive number small enough. By applying this stopping criterion, the algorithm can be stopped after convergence to unique optimal solution.

5. Simulation Results

Three simulation cases were studied to verify the effectiveness of the proposed algorithm. Firstly, we studied the simulation results of the algorithm based on IEEE14-bus data with generators constraints. Secondly, on the basis of simulation Case 1, we verified the plug-and-play performance of the proposed algorithm by removing load nodes of IEEE14-bus and adding a generation unit. Lastly, a case study based on IEEE57-bus was used to test the performance of the proposed algorithm in a wide range of power system.

5.1. Case 1: With Generator Constraints

We first studyied the case with generation constraints based on the IEEE 14-bus system. In the course of the study, all research data were derived from IEEE14-bus data. In our study, buses $\{1, 2, 3, 6, 8\}$ were chosen as the generation buses and the load buses were $\{2, 3, 4, 5, 6, 9, 10, 11, 12, 13, 14\}$. Note that bus 7 was selected as neither a generation bus nor a load bus since the power generated and the load demand of bus 7 were equal to zero. The local power load were: $P_{D2} = 21.7$ MW, $P_{D3} = 94.2$ MW, $P_{D4} = 47.8$ MW, $P_{D5} = 7.6$ MW, $P_{D6} = 11.2$ MW, $P_{D9} = 29.5$ MW, $P_{D10} = 9$ MW, $P_{D11} = 3.5$ MW, $P_{D12} = 6.1$ MW, $P_{D13} = 13.5$ MW, and $P_{D14} = 14.9$ MW. We calculated the summation of the load demand as $P_0 = 259$ MW, which was unknown to each individual bus node. Five generation nodes were located on buses 1, 2, 3, 6, 8, and the generator parameters of each generation unit are shown in Table 1. We took $\delta = 0.0001$ as the stopping criterion.

Table 1. Generator parameters in IEEE 14-bus system.

Bus	a ($/MW^{2}h)	b ($/MWh)	P_{Gi}^{min} (MW)	P_{Gi}^{max} (MW)
1	0.0430293	20	0	332.4
2	0.25	20	0	140
3	0.01	40	0	100
6	0.01	40	0	100
8	0.01	40	0	100

Each generator unit was initialized with the sum of the load power collected in the first stage: $P_{G1} = 332.4$ MW, $P_{G2} = 70$ MW, $P_{G3} = 100$ MW, $P_{G6} = 100$ MW, and $P_{G8} = 100$ MW. We set the lower and upper bounds of Lagrange multiplier as: $\lambda^{\downarrow}(0) = 2$ \$/MW and $\lambda^{\uparrow}(0) = 72$ \$/MW. The numerical simulation results with generators constraints are shown in Figure 5. To clearly analyze the simulation results of this case, we show the Lagrange multiplier, $\lambda(k)$, in Figure 5 (top), the evolution of $\sum P_j$ $(j = 1, 2, 3, 6, 8)$ in Figure 5 (middle), and the generator output P_{Gg} $(g = 1, 2, 3, 6, 8)$ in Figure 5 (bottom). we subjectively set the iteration step to $k = 20$, while the stopping condition was satisfied at $k = 12$.

The Lagrangian multiplier $\lambda(k)$ gradually approached an optimum stable value $\lambda^* = 5.4165$ \$/MW at iteration step $k = 12$, which affected the output of each generator unit. As $\lambda(k)$ was asymptotically stable, the total power generated by all generators units was gradually adapted to the total static load demand. When $\lambda(k)$ converged asymptotically to the optimal λ^*, the combination of generators output of each generator unit at the optimal λ^* was optimized, and the output powers of the generator units were: $P_{G1}^* = 39.70$ MW, $P_{G2}^* = 6.84$ MW, $P_{G3}^* = 70.82$ MW, $P_{G4}^* = 70.82$ MW, and $P_{G5}^* = 70.82$ MW, and $\sum P_j = 259$ MW $(j = 1, 2, 3, 6, 8)$. The optimal generators output P_g^* and the optimal Lagrangian multiplier λ^* satisfied the generator constraints. At the same time, the total power generated equaled the total load demand. Since the generator constraints and the power generated cost coefficient of generator units 3, 4, and 5 in the IEEE14-bus data were identical, the output of each generator unit was completely uniform.

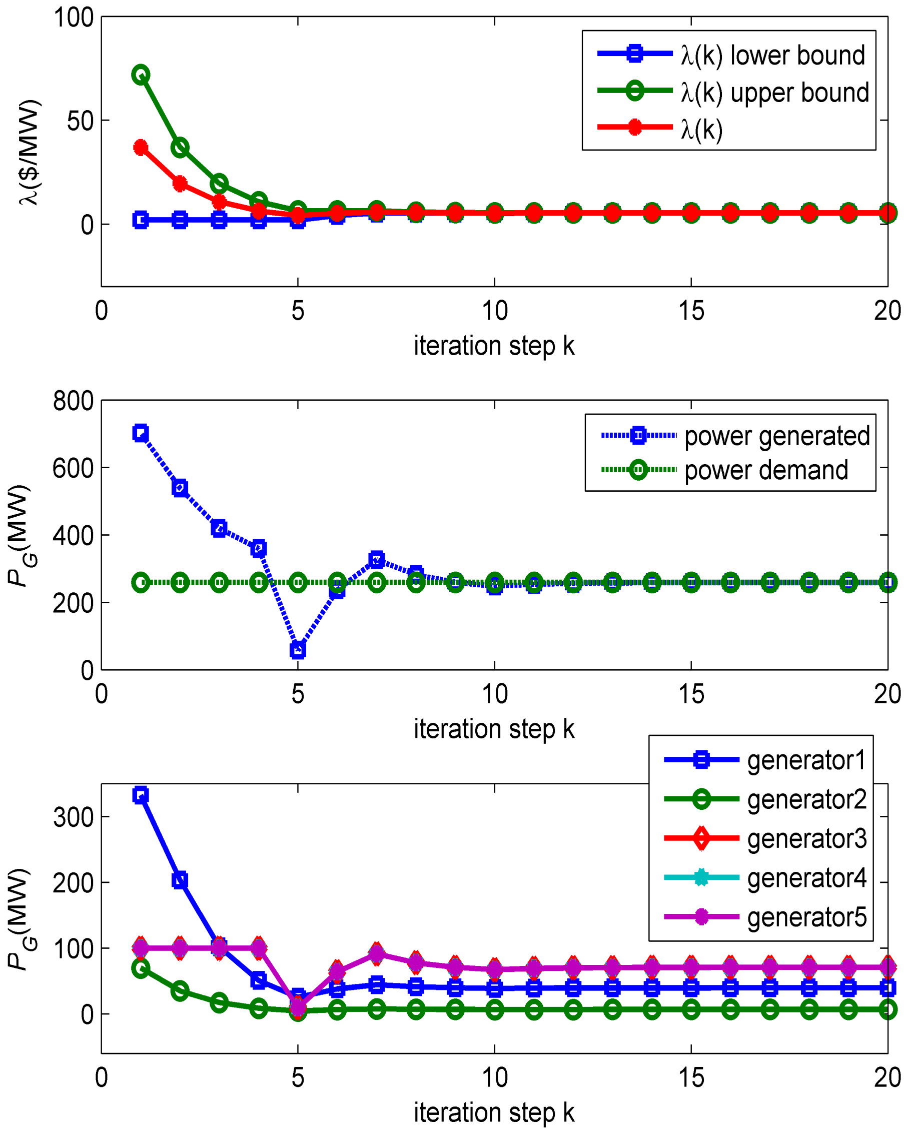

Figure 5. Case 1: with constrains for power generator.

5.2. Case 2: Plug and Play Capability

The characteristics of plug and play reflect the flexibility of power supply and demand of smart grid. In this case, we slightly changed the IEEE14-bus power system structure in Case 1 to study the plug-and-play performance of the proposed half-search algorithm in power systems. In this case, all research data were the same as in Case 1 until the system was changed. To design a more realistic power supply and demand scenario experiment, we first removed a load power based on Case 1, which caused a sharp reduction in the total load power of the system. The plug-and-play performance of the proposed half-search algorithm was effectively tested when the total load power was sharply reduced. Besides, to verify the plug-and-play performance of the algorithm under different generator conditions, we added a generator unit to change the generator conditions of the system for a period of time after the system converged again. Note that, whether removing a load to reduce the total power of the system or adding a generator unit to change the system's generator conditions, the total power generated must equal the total power demand.

At iteration step $k = 23$, the load demand on bus 4 was removed, and the situation of sharply reduced power demand was simulated. At iteration step $k = 43$, a generator unit was added between the 13th and 14th buses. The constraints of the generator units were $\overline{P}_{G6} = 200$ MW and $\underline{P}_{G6} = 50$ MW, and the power generated cost coefficient was the same as the power generated coefficient of the second generator unit. That is, the power generated cost coefficient of the added generator unit was: $a = 0.25, b = 2, c = 0$. The results of the case study are shown in Figure 6.

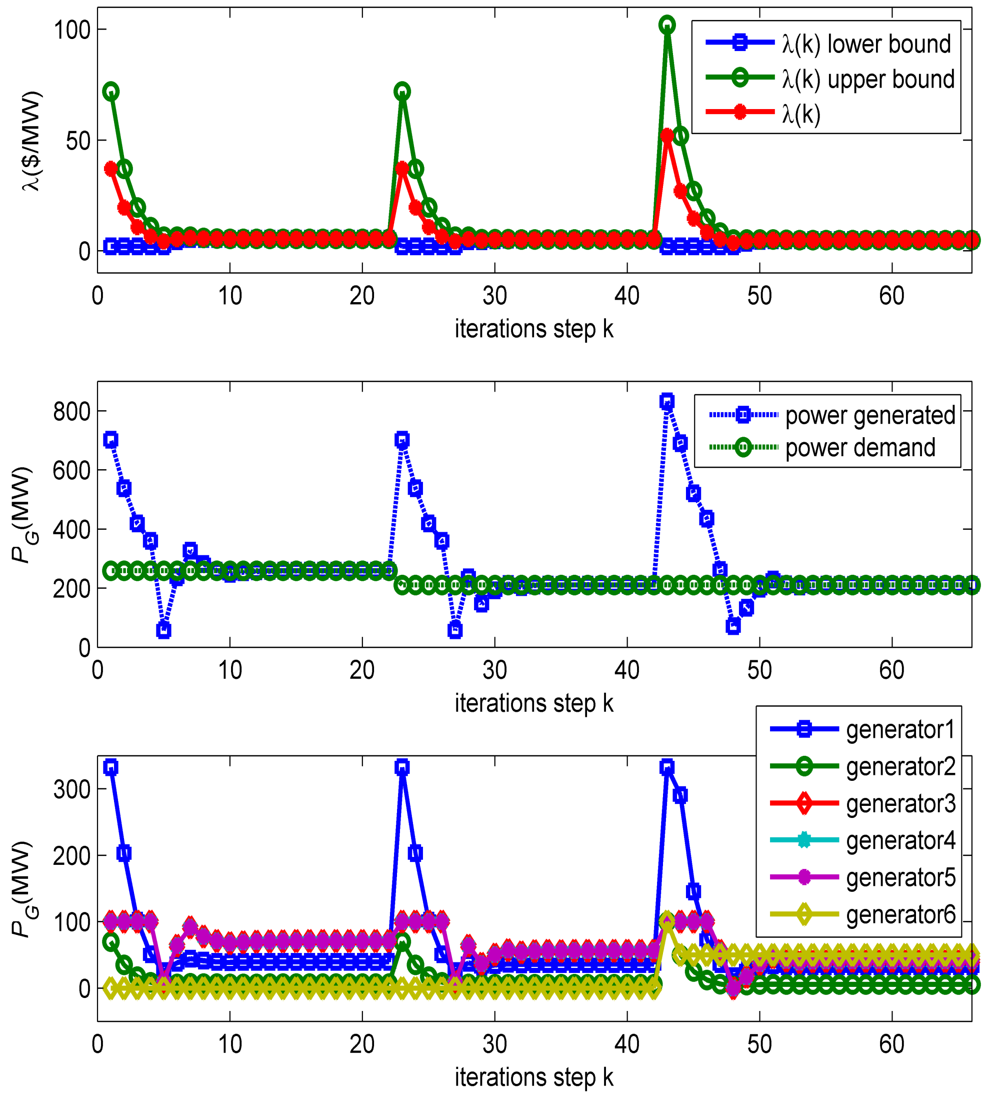

Figure 6. Case 2: Plug and play capability.

Since this case study was based on Case 1, the simulation data were the same as the simulation data of Case 1 at the beginning. Therefore, before the reduction of the total power demand, the power generated by each generator unit reached a steady state value. At iterative step $k = 23$, the total load demand decreased by 47.8 MW. After another 13 iterations of the proposed algorithm, the Lagrangian multiplier $\lambda(k)$ could approach a steady state value $\lambda^* = 5.124$ \$/MW again. At the same time, the total power generated and the total load demand were balanced again, and the power generated by each generator unit was also stabilized at a steady state value. The power generated by the generator units were: $P^*_{G1} = 36.20$ MW, $P^*_{G2} = 6.24$ MW, $P^*_{G3} = 56.22$ MW, $P^*_{G4} = 56.22$ MW, and $P^*_{G5} = 56.22$ MW. At iterative step $k = 43$, since a new generator unit was added, the generator conditions were changed. At this time, the total load demand was reallocated among six generator units by applying the proposed half-search algorithm. Then, the algorithm gradually converged to the Lagrange multiplier optimal value $\lambda^* = 4.819$ \$/MW again under the new generator conditions. Under the result of the balance between the total power generated and the total load demand, the optimized output powers of the generator units were: $P^*_{G1} = 32.77$ MW, $P^*_{G2} = 5.64$ MW, $P^*_{G3} = 40.93$ MW, $P^*_{G4} = 40.93$ MW, $P^*_{G5} = 40.93$ MW, and $P^*_{G6} = 50.00$ MW. Simulation results show that the half-search algorithm could meet the demand of plug and play in current power system.

5.3. Case 3: Implementation on IEEE 57-Bus System

In this case, based on IEEE57-bus system, we studied the application performance of the proposed half-search algorithm in large-scale power system. The research data were derived from IEEE57-bus data. IEEE57-bus system contains seven generators units located on buses $\{1, 2, 3, 6, 8, 9, 12\}$. Besides, we selected the buses with load power as the load buses, which are all buses except for buses $\{4, 7, 11, 21, 22, 24, 26, 34, 36, 37, 39, 40, 45, 46, 48\}$ in the IEEE57-bus system. By analyzing the load power data of IEEE57-bus, the summation of the power load could be calculated as $P_0 = 1250.8$ MW, which was unknown to each individual node. The generator parameters of each generator unit are shown in Table 2. We took $\delta = 0.00001$ for the stopping criterion.

Table 2. Generator parameters in IEEE 57-bus system.

Bus	*a* (\$/MW^{2}h)	*b* (\$/MWh)	P^{min}_{Gi} (MW)	P^{max}_{Gi} (MW)
1	0.077579519	20	0	575.88
2	0.01	40	0	100
3	0.25	20	0	140
6	0.01	40	0	100
8	0.022222222	20	0	550
9	0.01	40	0	100
12	0.0322580645	20	0	410

Compared with IEEE14-bus system, IEEE57-bus system has more bus nodes to test the application performance of the proposed half-search algorithm in large-scale power system. After the proposed half-search algorithm was applied in IEEE57-bus system, all of the generators nodes reached consensus about the solution and simulation results are shown in Figure 7. In Figure 7 (top), the upper and lower bounds of the Lagrangian multiplier $\lambda(k)$ are initialized to $\lambda^{\uparrow}(0) = 91.35$ \$/MW, and $\lambda^{\downarrow}(0) = 2$ \$/MW. Note that the power mismatch between the total power generated and the total load power equaled zero, as shown in Figure 7 (middle). Figure 7 (bottom) shows the generators output P_{Gg} $(g = 1, 2, 3, 6, 8, 9, 12)$. Each generation unit was initialized with the sum of the load power collected in the first stage, that is, $P_{G1} = 287.94$ MW, $P_{G2} = 100$ MW, $P_{G3} = 89.353$ MW, $P_{G6} = 100$ MW, $P_{G8} = 550$ MW, $P_{G9} = 100$ MW, and $P_{G12} = 410$ MW. The half-search algorithm had the advantage of rapid convergence, which could asymptotically reach a steady-state values within a finite iterative step in IEEE57-bus system with generator constraints.

Firstly, the Lagrangian multiplier $\lambda(k)$ could asymptotically reach a steady state value of $\lambda^* = 22.472$ \$/MW at $k = 13$. In the asymptotic stabilization process of $\lambda(k)$, the total power generated and the total power demand in the power system were gradually balanced: $\sum P_{Gj} = P_0$ $(j = 1, 2, 3, 6, 8, 9, 12)$. The optimal output power combination of the seven power generation units were as follows: $P^*_{G1} = 131.94$ MW, $P^*_{G2} = 100.00$ MW, $P^*_{G3} = 40.943$ MW, $P^*_{G6} = 100.00$ MW, $P^*_{G8} = 460.614$ MW, $P^*_{G9} = 100.00$ MW and $P^*_{G12} = 317.312$ MW. The optimal Lagrangian multiplier λ^* and the optimal solution P^*_{Gi} stayed within the generator constraints, and $\sum P_j = 1250.8$ MW

$(j = 1, 2, 3, 6, 8, 9, 12)$. The case simulation of the IEEE57-bus power system verified the availability of the proposed half-search algorithm in large-scale power system.

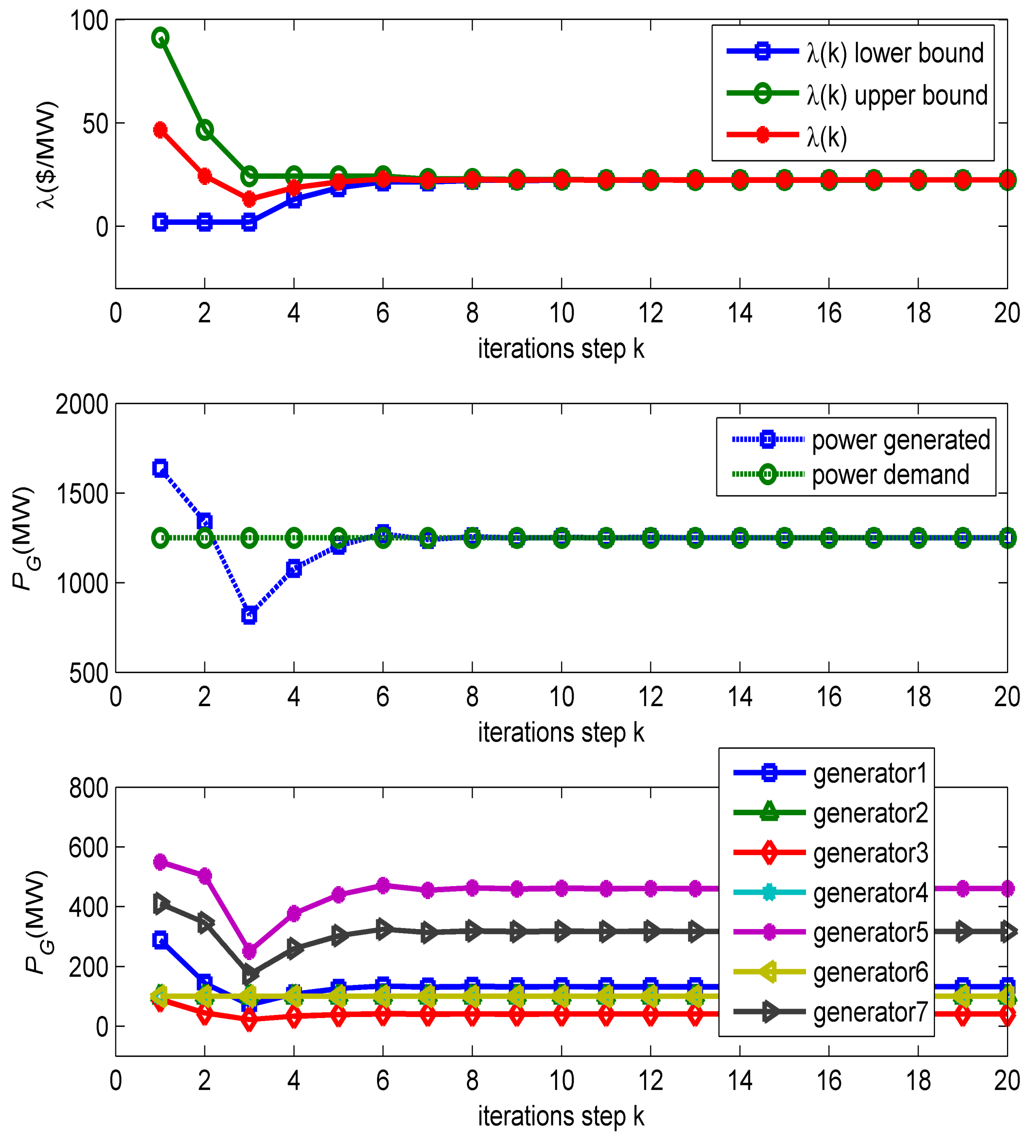

Figure 7. Case 3: IEEE 57-bus test system.

6. Conclusions

This paper proposes a distributed algorithm based on an absorption and half-search approach. The algorithm consists of two stages. In the first stage, absorption search, a popular broadcasting manner known as flooding is used as the mode of agent communication. The size of agent network is reduced and only the generation agents are retained in the multi-agent network, which reduces subsequent computing and agent communication. The process of network simplification is completely distributed. In the second stage, named optimal solution stage, a distributed half-search algorithm is proposed, which runs in a simple and straightforward manner. The idea of network simplification proposed in this paper can be applied to other multi-agent systems, and other economic dispatching problems for smart grids. For our future work, we will concentrate on expanding the presented approach to solve EDP with non-convex cost functions.

Author Contributions: X.L. and S.C. proposed the subject of this article. P.Z. and B.L. designed the study cases in the simulation part, and completed the first draft of the paper. All authors participated in the revision of the first draft.

Funding: This work was supported in part by National Natural Science Foundation of China, (NSFC Grant Nos. 61703091 and 61873057), in part by State Grid Technology Program "Research and Demonstration of Key Technologies of 'Generation–Storage–Load' Coordinative Operation in Commercial Districts" (Grant No. 52094016000F), and in part by Science Research of Education Department of Jilin Province (Grant Nos: JJKH20180441KJ and JJKH20170106KJ).

Acknowledgments: The author Xiangjun Li would like to thank "High-end Innovation Thousand Talents Plan", Learning Talents project funding of Qinghai Province, China. The authors would like to thank the anonymous reviewers and the associate editors for their helpful remarks that improved the presentation of the paper

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

EMS	Energy management systems
EDP	Economic dispatch problem
FBC	Flooding-based consensus
DE	Differential evolution
GA	Genetic algorithm
BBO	Biogreography-based optimization
FA	Firefly algorithm
GWO	Grey wolf optimization
DG	Distributed generation
JADE	Java Agent DEvelopment Framework
VA	Virtual agent in JADE
G	Directed or undirected graph
$V(G)/E(G)$	The finite nonempty set of vertices/edges
$F(e)$	The relationship between ordered vertices of an edge
$A(G)$	An adjacency matrix associates with graph G
$\mathcal{N}_i^+/\mathcal{N}_i^-$	In-neighbor/Out-neighbor set for node i
$d_i^{\uparrow}/d_i^{\downarrow}$	In-degree/Out-degree of node i
$\lvert \cdot \rvert$	The cardinality of a set
P_{Gi}	Power generated by generator i
P_{Dj}	Load demand in bus j
P_0	Total load demand
$\alpha_i, \beta_i, \gamma_i$	Parameters about the cost of generator i
$\overline{P}_{Gi}/\underline{P}_{Gi}$	Maximum/Minimum power generated by generator i
msg_i	The message that load agent i holds
MSG_{VA}	The information from each load agent to virtual agent
id_i	Unique identifier of agent i
ID_M	The identification of load node
ID_N	The identification of generating node
ID^*_{Mneibs}	The alternative neighbor set
ID_{Mneibs}	The ID set of current load node neighbors
pm_{id}	The load demand of current load node

References

1. Momoh, J. *Smart Grid: Fundamental of Design and Analysis*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2012; pp. 1–15.
2. Fang, X.; Misra, S.; Xue, G.; Yang, D. Smart grid—The new and emproved power grid: A survey. *IEEE Commun. Surv. Tutor.* **2011**, *14*, 944–980. [CrossRef]

3. Su, Q.; Quan, W.; Cai, G.; Li, J. Improved adaptive backstepping sliding mode control for generator steam valves of non-linear power systems. *IET Control Theory A* **2017**, *11*, 1414–1419. [CrossRef]
4. Su, Q.; Quan, W.; Quan, W.; Cai, G.; Li, J. Improved robust adaptive backstepping control approach on STATCOM for nonlinear power systems. *IET Gener. Transm. Dis.* **2017**, *11*, 3428–3437. [CrossRef]
5. Li, J.; Zhang, D. Backstepping and sliding-mode techniques applied to distributed secondary control of islanded microgrids. *Asian J. Control* **2017**, *20*, 1288–1295. [CrossRef]
6. Li, X.; Yao, L.; Hui, D. Optimal control and management of a large-scale battery energy storage system to mitigate fluctuation and intermittence of renewable generations. *J. Mod. Power Syst. Clean Energy* **2016**, *4*, 593–603. [CrossRef]
7. Zhu, Y.; Zhao, D.; Li, X.; Wang, D. Control-limited adaptive dynamic programming for multi-battery energy storage systems. *IEEE Trans. Smart Grid* **2018**. [CrossRef]
8. Li, X.; Yan, H. Fuzzy Logic-based Coordinated Control Method for Multi-type Battery Energy Storage Systems. *Artif. Intell. Rev.* **2018**, *49*, 227–243. [CrossRef]
9. Iqbal, F.; Siddiqui, A.S. Optimal configuration analysis for a campus microgrid—A case study. *Prot. Control Mod. Power Syst.* **2017**, *2*, 245–256. [CrossRef]
10. Byun, J.; Hong, I.; Kang, B.; Park, S. A smart energy distribution and management system for renewable energy distribution and context-aware services based on user patterns and load forecasting. *IEEE Trans. Consum. Electron.* **2011**, *57*, 436–444. [CrossRef]
11. Tushar, W.; Chai, B.; Yuen, C.; Smith, D.; Wood, K.; Yang, Z.; Poor, H.V. Three-party energy management with distributed energy resources in smart grid. *IEEE Trans. Ind. Electron.* **2015**, *62*, 2487–2498. [CrossRef]
12. Luna, A.C.; Meng, L.; Diaz, N.L.; Vasquez, J.C.; Guerrero, J.M.; Graells, M. Online energy management systems for microgrids: Experimental validation and assessment framework. *IEEE Trans. Power Electron.* **2018**, *33*, 2201–2215. [CrossRef]
13. Shareef, H.; Al Hassan, E.; Ahmed, M.S.; Mohamed, A. Review on home energy management system considering demand responses, smart technologies, and intelligent controllers. *IEEE Access* **2018**, *6*, 24498–24509. [CrossRef]
14. Dogra, R.; Gupta, N.; Saroa, H. *Economic Load Dispatch Problem and Matlab Programming of Different Mothods*; ICARI: Delhi, India, 2014; pp. 202–207.
15. Wood, A.J.; Wollenberg, B.F. Economic dispatch of thermal units and methods of solution. In *Power Generation, Operation, and Control*, 2nd ed.; John Wiley & Sons, Inc.: New York, NY, USA, 1996; Chapter 3, Section 4, pp. 43–47.
16. Wood, A.J.; Wollenberg, B.F. Economic dispatch of thermal units and methods of solution. In *Power Generation, Operation, and Control*, 2nd ed.; John Wiley & Sons, Inc.: New York, NY, USA, 1996; Chapter 3, Section 5, pp. 47–49.
17. Naderi, E.; Azizivahed, A.; Narimani, H.; Fathi, M.; Narimani, M.R. A comprehensive study of practical economic dispatch problems by a new hybrid evolutionary algorithm. *Appl. Soft Comput.* **2017**, *61*, 1186–1206. [CrossRef]
18. Narimani, H.; Razavi, S.E.; Azizivahed, A.; Naderi, E.; Fathi, M.; Ataei, M.H.; Narimani, M.R. A multi-objective framework for multi-area economic emission dispatch. *Energy* **2018**, *154*, 126–142. [CrossRef]
19. Lu, H.; Sriyanyong, P.; Song, Y.H.; Dillon, T. Experimental study of a new hybrid PSO with mutation for economic dispatch with non-smooth cost function. *INT J. Electr. Power* **2010**, *32*, 921–935. [CrossRef]
20. Qin, Q.; Chu, X.; Cheng, S.; Lei, X.; Shi, Y. Solving non-convex/non-smooth economic load dispatch problems via an enhanced particle swarm optimization. *Appl. Soft Comput.* **2017**, *59*, 229–242. [CrossRef]
21. Basu, M. Modified particle swarm optimization for nonconvex economic dispatch problems. *INT J. Electr. Power* **2015**, *69*, 304–312. [CrossRef]
22. Zaman, M.F.; Elsayed, S.M.; Ray, T.; Sarker, R.A. Evolutionary algorithms for dynamic economic dispatch problems. *IEEE Trans. Power Syst.* **2016**, *31*, 1486–1495. [CrossRef]
23. Amjady, N.; Nasiri-Rad, H. Nonconvex economic dispatch with AC constraints by a new real coded genetic algorithm. *IEEE Trans. Power Syst.* **2009**, *24*, 1489–1502. [CrossRef]
24. Bhattacharya, A.; Chattopadhyay, P.K. Biogeography-based optimization for different economic load dispatch problems. *IEEE Trans. Power Syst.* **2010**, *25*, 1064–1077. [CrossRef]
25. Abdullah, M.N.; Ali, A.A.; Ariff, M.A.M. Firefly algorithm for solving non-convex economic load dispatch problem. In Proceedings of the CEAT, Kuala Lumpur, Malaysia, 14–15 November 2016. [CrossRef]

26. Moradi, M.; Badri, A.; Ghandehari, R. Non-convex constrained economic dispatch with valve point loading effect using a grey wolf optimizer algorithm. In Proceedings of the CTPP, Tehran, Iran, 19–20 January 2016. [CrossRef]
27. Karavas, C.; Kyriakarakos, G.; Arvanitis, K.G.; Papadakis, G. A multi-agent decentralized energy management system based on distributed intelligence for the design and control of autonomous polygeneration microgrids. *Energy Convers. Manag.* **2015**, *103*, 166–179. [CrossRef]
28. Xu, Y.; Liu, W. Novel multiagent based load restoration algorithm for microgrids. *IEEE Trans. Smart Grid* **2011**, *2*, 152–161. [CrossRef]
29. Huang, A.Q.; Crow, M.L.; Heydt, G.T.; Zheng, J.P.; Dale, S.J. The future renewable electric energy delivery and management (FREEDM) system: The energy internet. *Proc. IEEE* **2010**, *99*, 133–148. [CrossRef]
30. Loia, V.; Vaccaro, A. Decentralized economic dispatch in smart grids by self-organizing dynamic agents. *IEEE Trans. Syst. Man Cybern.* **2014**, *44*, 397–408. [CrossRef]
31. Yang, S.; Tan, S.; Xu, J. Consensus based approach for economic dispatch problem in a smart grid. *IEEE Trans. Power Syst.* **2013**, *28*, 4416–4426. [CrossRef]
32. Azizivahed, A.; Naderi, E.; Narimani, H.; Fathi, M.; Narimani, M.R. A new bi-objective approach to energy management in distribution networks with energy storage systems. *IEEE Trans. Sustain. Energy* **2018**, *9*, 56–64. [CrossRef]
33. Azizivahed, A.; Narimani, H.; Fathi, M.; Naderi, E.; Safarpour, H.R.; Narimani, M.R. Multi-objective dynamic distribution feeder reconfiguration in automated distribution systems. *Energy* **2018**, *147*, 896–914. [CrossRef]
34. Zhang, Z.; Chow, M.Y. Convergence analysis of the incremental cost consensus algorithm under different communication network topologies in a smart grid. *IEEE Trans. Power Syst.* **2012**, *27*, 1761–1768. [CrossRef]
35. Binetti, G.; Naso, D.; Turchiano, B.; Davoudi, A.; Lewis, F.L. Distributed consensus-based economic dispatch with transmission losses. *IEEE Trans. Power Syst.* **2014**, *29*, 1711–1720. [CrossRef]
36. Xing, H.; Mou, Y.; Fu, M.; Lin, Z. Distributed bisection method of economic power dispatch in smart grid. *IEEE Trans. Power Syst.* **2015**, *30*, 3024–3035. [CrossRef]
37. Li, B.; Wang, Y.; Li, J.; Cao, S. A fully distributed approach for economic dispatch problem of smart grid. *Energies* **2018**, *11*, 1993. [CrossRef]
38. Binetti, G.; Naso, D.; Turchiano, B.; Davoudi, A.; Lewis, F.L. A distributed auction-based algorithm for the nonconvex economic dispatch problem. *IEEE Trans. Ind. Inform.* **2014**, *10*, 1124–1132. [CrossRef]
39. Lysikatos, I.K.; Hatziargyriou, N. Fully distributed economic dispatch of distributed generators in active distribution networks considering losses. *IET Gener. Transm. Dis.* **2017**, *11*, 627–636. [CrossRef]
40. Bondy, J.A.; Murty, U.S.R. Graphs and subgraphs. In *Graph Theory with Applications*; Elsevier Science Publishing Co., Inc.: New York, NY, USA, 1976; Chapter 1, pp. 1–3.
41. Diestel, R. The basics. In *Graph Theory*, 5th ed.; Springer: Berlin, Germany, 2017; Chapter 1, pp. 1–30.
42. Ji, L.H.; Liao, X.F. Consensus analysis of multi-agent system with multiple time delays. *Acta Phys. Sin.* **2012**, *61*, 150202(1)–150202(9). [CrossRef]
43. Xia, H. Group consensus of multi-agent systems with communication delays. *Neurocomputing* **2016**, *171*, 1666–1673. [CrossRef]
44. Degroot, M.H. Reaching a Consensus. *J. Am. Stat. Assoc.* **1974**, *69*, 118–121. [CrossRef]
45. Lin, Z. Nonnegative matrices and graphs. In *Distributed Control and Analysis of Coupled Cell Systems*; VDM Publishing: Saarbrücken, Germany, 2008; Chapter 3, pp. 25–33.
46. Horn, R.A.; Johnson, C.R. Positive and nonnegative matrices. In *Matrix Analysis*; Cambridge University Press: New York, NY, USA, 2013; Chapter 8, pp. 517–553.
47. Madrigal, M.; Quintana, V.H. An analytical solution to the economic dispatch problem. *IEEE Power Eng. Rev.* **2000**, *20*, 52–55. [CrossRef]
48. Saligrama, V.; Alanyali, M. A token-based approach for distributed computation in sensor networks. *IEEE J. Sel. Top. Signal. Process.* **2011**, *5*, 817–832. [CrossRef]
49. Bertsekas, D.; Gallager, R. Broadcasting routing information: coping with link failures. In *Date Networks*, 2nd ed.; Prentic Hall: Englewood Cliffs, NJ, USA, 1992; Chapter 5, Section 3, pp. 418–432.

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).