1-1-2012

A multi-agent solution to distribution system management by considering distributed generators

Fenghui Ren
University of Wollongong, fren@uow.edu.au

Minjie Zhang
University of Wollongong, minjie@uow.edu.au

Darmawan Sutanto
University of Wollongong, soetanto@uow.edu.au

A multi-agent solution to distribution system management by considering distributed generators

Abstract

A traditional distribution network carries electricity from a central power resource to consumers, and the power dispatch is controlled centrally. Distributed generators (DGs) emerge as an alternative power resource to distribution networks at a smaller and distributed scale, which will bring benefits such as reduced voltage drop and loss. However, because most of high penetration DGs are not utility owned and characterized by high degree of uncertainty such as solar and wind, the distribution network may perform differently from the conventionally expected behaviors. How to dynamically and efficiently manage the power dispatch in a distribution network to balance the supply and demand by considering the variability of DGs and loads becomes a significant research issue. In this paper, a multi-agent system (MAS) was proposed to solve this problem through introducing five types of autonomous agents, the electricity management mechanisms, the agent communication ontology, and the agent cooperation strategy. The simulation of the MAS by using InterPSS, JADE and JUNE well demonstrates the performance of the system on dynamic supply and demand balance by considering both efficiency and economy.

Keywords

Disciplines

Engineering | Science and Technology Studies

Publication Details

A Multi-Agent Solution to Distribution System Management by Considering Distributed Generators

Fenghui Ren, Minjie Zhang and Danny Sutanto

Abstract—A traditional distribution network carries electricity from a central power resource to consumers, and the power dispatch is controlled centrally. Distributed generator (DG) emerge as an alternative power resource to distribution networks at a smaller and distributed scale, which will bring benefits such as reduced voltage drop and loss. However, because most of high penetration DGs are not utility owned and characterized by high degree of uncertainty such as solar and wind, the distribution network may perform differently from the conventionally expected behaviors. How to dynamically and efficiently manage the power dispatch in a distribution network to balance the supply and demand by considering the variability of DGs and loads becomes a significant research issue. In this paper, a multi-agent system (MAS) was proposed to solve this problem through introducing five types of autonomous agents, the electricity management mechanisms, the agent communication ontology, and the agent cooperation strategy. The simulation of the MAS by using InterPSS, JADE and JUNE well demonstrates the performance of the system on dynamic supply and demand balance by considering both efficiency and economy.

Index Terms—Distribution network, electricity dispatch, distributed generator, multi-agent system

I. INTRODUCTION

A distribution network (DN) is the final stage in the delivery of electricity to end users [1]. Typically, the bulk generation is the only energy resource to a DN, and the direction of the power flow is strictly from the central generation to downstream electric components [2]. Conventionally, the DN is centrally monitored and controlled, and the classical control techniques focus more on average load and demand management [3], [4].

Distributed Generator (DG) emerges as an alternative power resources to a DN at a smaller and distributed scale, and generally located close to the loads [5], [6]. The introduction of DGs has both advantages and disadvantages [7]. On one hand, DGs can supply power to the network near the loads without needing the transmission system, so as to significantly decrease the power loss and cost, and share the loads with the central generation. Also, DGs can provide supports on voltage drop and loss to a DN during peak load [8]. On the other hand, most DGs can only provide intermittent power to the network due to the intermittent nature of the distributed energy resources such as wind and sun. Also, the utility usually does not own the DGs, and has difficulty in controlling their outputs. Therefore, with an increasing level of DG penetration,

F. Ren and M. Zhang are with the School of Computer Science and Software University of Wollongong, Australia, email:{fren,minjie}@uow.edu.au

D. Sutanto is with the School of Electrical, Computer and Telecom University of Wollongong, Australia, email:soetanto@uow.edu.au

the DN may behave quite differently from the conventional operations. For example, in PV applications during noon time, when the PV is producing its highest output, the residential load may be at its lowest, which results in power excess and is transmitted back to the substation leading to voltage rise. Also, the cloud passing issue in PV applications will lead to fast changes of PV output with high ramp rate. Furthermore, with high penetration of DGs, the power flow in a distributed network may change from a single directional flow to bi-directional flow [9]. Therefore, how to dynamically and efficiently manage the electricity dispatch in a DN to balance the energy supply and demand by considering the variability of DGs and loads becomes an important research issue in power engineering. The conventional method may no longer be suitable for doing this due to its limitations on flexibility, communication, cooperation, and decision making [3], [8], [10].

Multi-Agent Systems (MASs) have been employed to solve many power engineering problems in recent years, and are being developed for a range of applications including system monitoring and fault diagnostics [11], [12], system restoration [13], [14], system simulation [15], [16], and system control [17], [18]. In this paper, a decentralized MAS is proposed to solve the dynamic electricity dispatch problem in DN by considering the variabilities of DG's electricity supply and load's demand. The contributions of this paper are that: (1) it proposes five types of agents to model the DN in an autonomous and decentralized way; (2) it proposes several mechanisms to control the DGs and other components to dynamically balance the electricity supply and demand in the DN by considering both efficiency and economy; and (3) it describes the development and implement of the proposed MAS by using InterPSS (Internet Technology Based Power System Simulator), JADE (Java Agent Development Framework), and JUNG (Java Universal Network/Graph Framework), and demonstrates the simulation result.

The organization of this paper is as follows. Section II introduces the principle and the objectives of the proposed MAS. Section III introduces three major mechanisms used in the MAS. Section IV shows how to develop the MAS by using JADE. Section V demonstrates the performance of the proposed MAS through a simulation, and Section VI compares the proposed MAS with some related work. Finally, the conclusion is given in Section VII.

II. PRINCIPLE AND OBJECTIVE

A. Principle

Different modeling strategies could be used to model a DN [19], [20]. Since this paper focuses on the electricity dispatch problem in a DN by considering DGs, five types of agents are proposed to simulate the five key electric components in a DN, i.e., the substation, the busbar, the feeder, the load, and the DG. The rationale of our design is to employ multiagent technology to model each of the key components, and provide them with the communication and decision making abilities. Then these components can make individual decisions on local power management based on local information, and collective decisions on regional/global power management through agent communication and cooperation. The proposed MAS contains three layers, i.e., the power system layer, the multiagent layer and the interface layer. As shown in Fig. 1, the lower power system layer presents the physical electric components in a DN, the middle multi-agent layer handles the communication, decision making, and cooperation between the control agents of electric components, and the upper interface layer illustrates the MAS. The major characteristics of each type of agents are introduced below.

Substation Agent (SA): A SA represents a secondary substation. It monitors both the magnitude and the direction of electricity through the substation. The SA handles the power dispatch requests from other neighbor agents by considering the capacity of the substation and the cable's limit constraint in-between them.

Bus Agent (BA): A BA represents a physical busbar that conducts electricity between electric components. The BA records information on the connected electric components, such as electricity magnitude and direction, cable's limit constraint, and the component's capacity on power supply or consumption. The BA decides whether an power dispatch request can be satisfied based on the information it has and order operations on switches to fulfill its decision.

Feeder Agent (FA): A FA represents a physical feeder which delivers electricity to downstream components. The FA records the cable's limit constraint, and monitors the current magnitude and direction on the cable. The FA checks the cable's transmission ability to decide whether the required power can be delivered. In case the electricity direction needs to be changed, the FA needs to handle such a procedure by communicating with other agents along the feeder.

Load Agent (LA): A LA represents a physical load which consumes electricity from the network. Each LA is assigned a priority to indicate its significance. If the load needs to be changed, the LA will contact the upstream component to apply the power consumption change. The LA's priority is considered by the upstream BA in determining how the LA's request is answered.

Generation Agent (GA): A GA represents a physical DG in the network, and is also assigned a priority. The GA handles the connections and disconnections of a DG to the network through communicating with upstream BA. Also, the GA monitors the power supply of the DG, and modifies the power supply under requests by considering the DG's capacity and
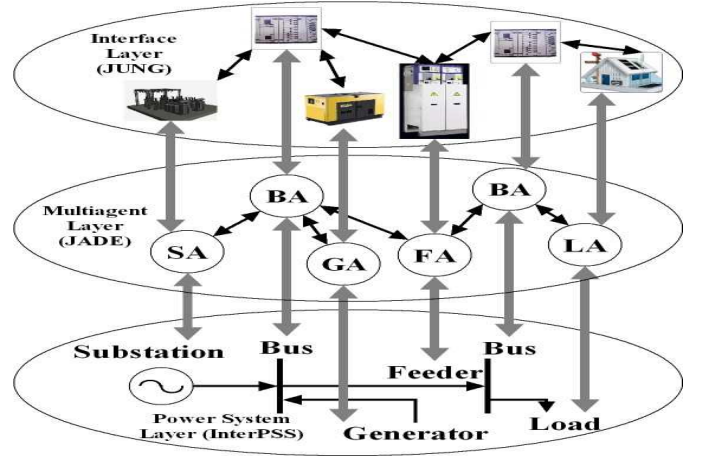

Fig. 1. The three layers view of the proposed MAS.

priority.

The proposed MAS has the following features by comparing with classic centralized power management systems: (i) the MAS employs the *distributed system structure*, which means that agents are assigned to different components at different locations in a DN. Therefore, they are not integrated, but separated from each other; (ii) the *decentralized management* is employed by the MAS, which means that there is no a central controller in the MAS, and agents work automatically based on the information they receive from the corresponding electric components and neighbor agents. No agent in the system can preset the global information of the whole DN; (iii) agents can only notice and communicate with their neighbor agents. Non-adjacent agents do not know the existence of each other, but the information can still be exchanged among them through other agents in-between; and (iv) the MAS requires no dependency between agents, and the system architecture is extendable. Agents play as a "plug and operate" component.

B. Objectives

The multiobjective approach [21] is employed to represent the objectives of the MAS, which are to (i) dynamically balance the power supply and demand in a DN; (ii) maximize the power usage from DGs; and (iii) minimize the power usage cost in a DN. The restrictions such as cables' limits, electric components' limits, and generators' power supply capacities are also considered. Some related MASs also take the similar considerations into account, such as [22] considers the cost minimization, [23] considers the voltage and current limitation, and [14] considers the load maximization. In this paper, we combine these considerations together, and also take the system balance and optimization of DGs into account.

(i) Balance Objective: For each component i in a DN, we must ensure that the power delivering to and from i are balanced,

$$p_i^+ + \sum_{j \in \mathbf{J}^+} p_{j \to i} = p_i^- + \sum_{k \in \mathbf{J}^-} p_{i \to k} \qquad (1)$$

where p_i^+ is the power generated by component i, p_i^- is the power consumed by component i, $p_{j \to i}$ is the power delivered to i from a neighbor component j, and $p_{i \to k}$ is the power delivered from i to another neighbor component k.

(ii) Usage Objective: In order to decrease the burden on the bulk generations, we want to maximize the power usage from DGs.

$$\max \sum_{i=1}^{m} p_i^+ \qquad (2)$$

where m is the total number of DGs in a DN.

(iii) Cost Objective: We also want to minimize the power usage cost for a DN. Firstly, component i's cost on consuming an unit electricity is defined as follows:

$$c_i = \frac{\sum_{j \in \mathbf{J}^+} p_{j \to i} \times c_{j \to i}}{\sum_{j \in \mathbf{J}^+} p_{j \to i}} \qquad (3)$$

where $c_{j \to i}$ is the unit power price from component j. If component i needs to further deliver power to another component k, then the cost is $c_{i \to k} = c_i + c_{i,k}^{cb}$, where $c_{i,k}^{cb}$ is the cost on the cable between components i and k. Then the cost objective is formulated as:

$$\min \sum_{i=1}^{n} c_i \times p_i^- \qquad (4)$$

where n is the total number of components in the network.

The fulfillment of the objectives should not lead to violation of operating limits of the components; hence, several constraints are reinforced.

(i) Limit on Cable: The current between components i and k should be not greater than the cable's carrying capacity rating to ensure that the upper thermal current limit will not be exceeded.

$$I_{i \to k} \leq I_{i,k}^{max} \qquad (5)$$

(ii) Limit on Generator: For each generator, the supplied power should be not greater than its capacity.

$$p_i^+ \leq p_i^{+max} \qquad (6)$$

(iii) Limit on Busbar: Each bus's voltage must be always within its voltage limits.

$$V_i^{min} \leq |V_i| \leq V_i^{max} \qquad (7)$$

(iv) Limit on Component: Each component also has current rating limit.

$$\sum_{j \in \mathbf{J}^+} I_{j \to i} \leq I_i^{max} \qquad (8)$$

III. Agent Mechanisms

In this subsection, we introduce three mechanisms used in the proposed MAS, i.e., connection mechanism, disconnection mechanism, and power management mechanism.

A. Connection Mechanism

When a new electric component i needs to connect to a DN, a corresponding agent a_i will be firstly generated to represent the new component. Let a_i be represented by a five-tuple $< AID_i, I_i^{max}, S_i^{max}, C_i^{max}, P_i >$, and $n_{i,j} = < AID_j, S_{i,j}^{max}, C_{i,j}^{max}, I_{i,j}^{max}, A_{i,j}, I_{i,j}, P_{i,j} >$ be a_i's record on its neighbor a_j (the explanation of notations is given by Table I). Then if the component i needs to connect to the DN through component j (represented by a_j), the connection process is given as follows (see Fig. 2 for the UML diagram):

TABLE I
THE EXPLANATION ON NOTATIONS.

| Notation	Explanation
AID_i	Agent a_i's unique identification
I_i^{max}	the maximum current allowed by the electric component i
S_i^{max}	the maximum power that component i can supply
C_i^{max}	the maximum power that component i can consume
P_i	the unit electricity price a_i supply
AID_j	neighbor agent a_j's identification
$S_{i,j}^{max}$	the maximum power that a_i can get through a_j
$C_{i,j}^{max}$	the maximum power that a_j can take from a_i
$I_{i,j}^{max}$	the maximum current on the cable between components i and j
$A_{i,j}$	the existing power transferring between components i and j
$I_{i,j}$	the existing current on the cable between components i and j
$P_{i,j}$	the cost on delivering unit power from components j to i
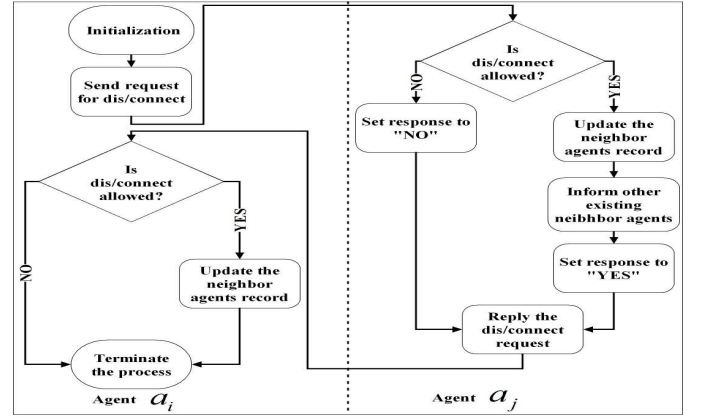
Fig. 2. The UML diagram for dis/connection mechanism

Step (i): a_i is created to represent the electric component i, and is initialized according to component i.

Step (ii): a_i sends the connection request and its information (i.e., $< AID_i, S_i^{max}, C_i^{max}, P_i >$) to a_j, and waits for a_j's response.

Step (iii): a_j receives the request. If the connection is not allowed, a_j denies the connection, and the procedure goes to **Step (v)**. Otherwise, the procedure goes to **Step (iv)**.

Step (iv): Firstly, a_j creates a new neighbor agent record according to the information sent by a_i, i.e., $n_{j,i} = < AID_i, S_i^{max}, C_i^{max}, I_{j,i}^{max}, 0, 0, P_i + L_{j,i} >$ (where $L_{j,i}$ indicates the loss in the cable between components i and j) and $\mathbf{N_j} \leftarrow \{n_{j,i}\} \cap \mathbf{N_j}$. Secondly, a_j informs other existing neighbor agents about its update on power supply and demand by sending (S_i^{max}, C_i^{max}). Thirdly, the neighbor agents update their record on a_j's electricity supply and demand, i.e., $S_{k,j}^{max} \leftarrow S_{k,j}^{max} + S_i^{max}$ and $C_{k,j}^{max} \leftarrow C_{k,j}^{max} + C_i^{max}$. Lastly, a_j replies a_i's with its information, i.e., $< AID_j, S_j^{max} + \sum_{k \in \mathbf{N_j}, k \neq i} S_{j,k}^{max}, C_j^{max} + \sum_{k \in \mathbf{N_j}, k \neq i} C_{j,k}^{max}, P_j >$.

Step (v): If a_i receives a refuse response, the process is terminated. Otherwise, a_i creates a new neighbor agent record $n_{i,j}$ according to the information sent by a_j, i.e., $\mathbf{N_i} \leftarrow \{n_{i,j}\} \cap \mathbf{N_i}$, and then a_i connects to the DN.

B. Disconnection Mechanism

An existing electric component may also need to be disconnected from a DN. Let suppose that agent a_i wants to disconnect from the DN, and agent a_j is the upstream

component, then the disconnection process is given as follows (see Fig. 2 for the UML diagram):

Step (i): a_i initializes the process, sends the disconnection request to a_j, and waits for a_j's response.

Step (ii): a_j receives the request. If the disconnection is not allowed, a_j denies the disconnection, and the procedure goes to **Step (iv)**. Otherwise, the procedure goes to **Step (iii)**.

Step (iii): Firstly, a_j informs other neighbor agents about its update on power supply and demand by sending $(-S_{j,i}^{max}, -C_{j,i}^{max})$. Secondly, the neighbor agents update their record on a_j's power supply and demand, i.e., $S_{k,j}^{max} \leftarrow S_{k,j}^{max} - S_{j,i}^{max}$ and $C_{k,j}^{max} \leftarrow C_{k,j}^{max} - C_{j,i}^{max}$. Thirdly, a_j deletes the record of a_i from its record set, i.e., $\mathbf{N_j} \leftarrow \mathbf{N_j}/n_{j,i}$. Lastly, a_j replies a_i with an agreement.

Step (iv): If a_j denies the disconnection request, the process is terminated. Otherwise, a_i will deletes the record of a_j from its record set, i.e., $\mathbf{N_i} \leftarrow \mathbf{N_i}/n_{i,j}$, and then a_i disconnects from components j.

C. Power Management Mechanism

If any power unbalance occurs on any component, the power management mechanism will be activated to balance the power supply and demand by considering all the objectives and constraints mentioned in Subsection II-B.

Let a_k be the agent which firstly notices an unbalance. Let $\mathbf{N_k^+}$ ($\subseteq \mathbf{N_k}, \forall n_{k,m} \in \mathbf{N_k^+} \Rightarrow I_{k,m} \geq 0$) be the set of a_k's neighbor agents who get power from a_k, $\mathbf{N_k^-}$ ($\subseteq \mathbf{N_k}, \forall n_{k,m} \in \mathbf{N_k^-} \Rightarrow I_{k,m} < 0$) be the set of a_k's neighbor agents who send power to a_k, and $req_k (\neq 0)$ be the amount of power that a_k wants to change, then there are two possible scenarios, i.e., $req_k < 0$ and $req_k > 0$. Fig. 3 illustrates the UML diagram for the management process, and the detail process is introduced as follows.

1) When $req_k < 0$: this scenario indicates that a_k needs more req_k power. In order to satisfy the *balance objective*, two possible solutions are considered by a_k, i.e., **(a)** seeking extra power from existing suppliers, or **(b)** decreasing output to existing consumers (i.e., load shedding). In order to minimize the impact on consumers, a_k will seek extra amount power req_k from its existing suppliers firstly, and then perform the load shedding when necessary. Based on this rule, the following procedures are performed:

Step (i): In order to satisfy the *cost objective*, the suppliers with lower price will be considered firstly by a_k to seek the extra req_k power. a_k ranks all suppliers according to the ascending order of their supply prices. Let $\mathbf{N_k^{-r}}$ be the ranked suppliers and agent a_i be the first agent in the rank. Firstly, a_k estimates a_i's power supply ability as $S_{k,i}^r = S_{k,i}^{max} - S_{k,i}$. If $S_{k,i}^r > |req_k|$, the request $req_{k,i} \leftarrow (req_k, pri_k)$ (pri_k is a_k's priority) will be sent to a_i; otherwise, the request $req_{k,i} \leftarrow (-S_{k,i}^r, pri_k)$ will be sent to a_i and leave the remaining $S_{k,i}^r - |req_k|$ for the next supplier. Then a_k will remove a_i from the set $\mathbf{N_k^{-r}}$ and wait for a_i's response. However, if a_k fails to receive any response from a_i within a predefined time period, a_k will abandon getting power from a_i by sending a cancelation, and reallocate the power req_k to the next supplier.

Step (ii): a_i may receive many requests from different agents. Therefore, a_i will rank all requests in queue Q_i by considering the priority of the requestor and the time they received. A request with the higherest priority will be put in the front of the queue. For requests with a same priority, the earlier received request will be put in the front. For requests with same priority and time label, they will be inserted into the queue randomly. Therefore, once a_i receives the request $req_{k,i}$ from a_k, a_i will firstly insert $req_{k,i}$ into the queue. Let us suppose that a_i already completes all requests in front of $req_{k,i}$. If agent a_i is a GA or SA, a_i can make decision on the request $req_{k,i}$ without contacting other agents. In order to do that, a_i firstly calculates its remaining supply ability to a_k as $S_{i,k}^r = S_i^{max} - \sum_{j \in \mathbf{N_i^+}} S_{i,j}$, and replies a_k to indicate the amount that a_i can actually supply, i.e., $rsp_{i,k} = \min(S_{i,k}^r, |req_{k,i}|)$, then the procedure goes to **Step (iii)**. However, if a_i is a BA or FA, a_i needs to contact other neighbors for a_k's request. To do that, a_i will initialize another management process by seeking $|req_{k,i}|$ amount of power from its neighbors. Obviously, such a procedure will be repeated until a GA or a SA is found (see Subsection III-D on the discussion on the scalability and complexity). After a_i receives all responses from its neighbor agents, a_i will generate a response to a_k by combing all the responses, i.e., $rsp_{i,k} = \sum_{m \in \mathbf{N_i}} rsp_{m,k}$. However, if a_i receives a cancelation request from a_k before the response $rsp_{i,k}$ can be generated, a_i will also send the cancelation to its neighbors.

Step (iii): a_k receives a_i's response. If a_k's request can be fully satisfied by a_i, i.e., $rsp_{i,k} = |req_k|$, then the procedure is terminated and all related agents adjust their power according to their agreements. Otherwise, a_k will seek for the remaining power $req_k^r \leftarrow rsp_{i,k} - |req_k^r|$ (where req_k^r is initialized as req_k) from next available supplier in $\mathbf{N_k^{-r}}$. Then steps **(i)** to **(iii)** will be repeated until a_k's request is fully satisfied (i.e., $req_k^r = 0$) or the ranked list $\mathbf{N_k^{-r}}$ becomes empty. For the latter case, the procedure goes to **Step (iv)**.

Step (iv): Because a_k cannot get enough power from its existing suppliers, it has to decrease output to existing consumers (i.e., load shedding). The load shedding strategy employed in this paper is simple, i.e., performing load shedding according to the neighbor agent's priority. Firstly, a_k ranks all consumers according to the ascending order of consumers' priorities. Let $\mathbf{N_k^{+r}}$ be the ranked consumers and a_j be the first agent. If a_k's priority is higher than a_j's, then a_k will shed a_j's power by sending request $req_{k,j} \leftarrow (-\min(A_{k,j}, |req_k^r|), pri_k)$ to a_j. Then a_k removes a_j from set $\mathbf{N_k^{+r}}$, and the procedure goes to **Step (v)**. However if the waiting period is longer than a predefined period, a_k will shed a_j's load without the response.

Step (v): a_j will insert the request from a_k to its request queue Q_j. Let us assume that $req_{k,j}$ is proceeded by a_j now. If a_j is a LA, the load will be shed on a_j directly and the procedure goes to **Step (vi)**. Otherwise, if a_j is a BA or FA, a_j needs to contact other neighbors for a_k's request. In order to do that, a_j will initialize another management process by seeking the $|req_k|$ amount of power from its neighbors. After a_j receives all responses from its neighbor agents, a_j will reply a_k with the response $rsp_{j,k} = \sum_{m \in \mathbf{N_j}} rsp_{m,k}$, and the procedure goes to **Step (vi)**. However, if a_j's load was shed by
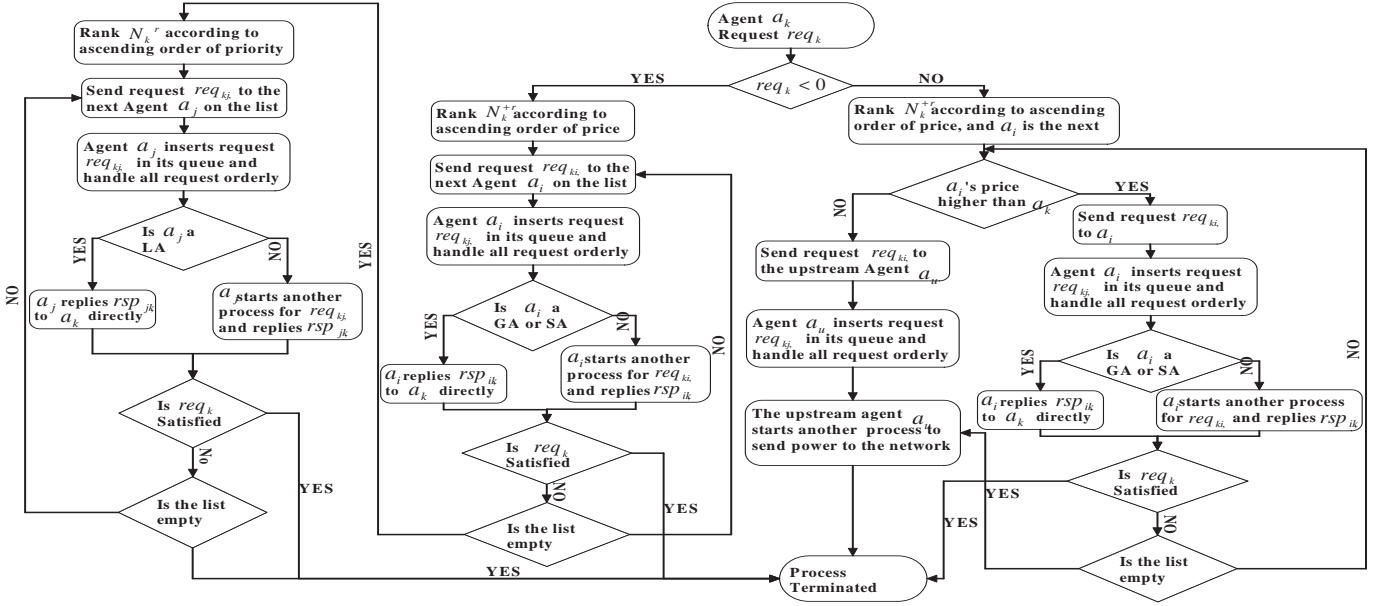
Fig. 3. The UML diagram for power management

a_k before the response $rsp_{j,k}$ can be generated, a_j will shed the same amount of load on its consumers by considering their priorities.

Step (vi): a_k receives a_j's response. If a_k's request can be fully satisfied by a_j, i.e., $rsp_{j,k} = |req_k|$, then the procedure is terminated and all related agents adjust their power according to their agreements. Otherwise, a_k will seek the remaining power $req_k^r \leftarrow rsp_{j,k} - |req_k^r|$ from next available consumer in $\mathbf{N_k^{+r}}$. Then steps **(iv)** to **(vi)** will be repeated until a_k's request is fully satisfied (i.e., $req_k^r = 0$) or no agent's priority in the ranked list is lower than a_k's priority.

2) When $req_k > 0$: this scenario indicates a_k has extra power. Let P_k be the unit power price providing by a_k. In order to satisfy the *balance objective*, two possible solutions are considered, i.e., **(a)** decreasing power input from existing suppliers, or **(b)** delivering the extra power back to the upstream components. By considering both the *usage and cost objective*, the two options should be combined, i.e., firstly trying to decrease the power from suppliers whose price is higher than P_k, and then delivering the remaining extra power back to the upstream components. The procedures are as follows:

Step (i): a_k ranks its suppliers according to the descending order of suppliers' prices. Let $\mathbf{N_k^{-r}}$ be the ranked suppliers and a_i be the first agent. If a_i's price is higher than a_k's, a_k will send the request $req_{k,i} \leftarrow (\min(req_k, A_{i,k}), pri_k)$ to a_i. Then a_k removes a_i from the list and the procedure goes to **Step (ii)**. Otherwise, if a_i's price is not higher than a_k's price, the procedure goes to **Step (iv)**. However, if a_k cannot receives a_i's response within a predefined period, a_k will cancel the request to a_i and contact the next supplier.

Step (ii): Once a_i receives the request to decrease its power output to a_k, a_i will insert the request to its queue Q_i. Let us assume that request $req_{k,i}$ is processed now. If a_i is a GA or a SA, it can make the decision without contacting other neighbor agents. The actual deduction on output amount will be sent back to a_k as a response, i.e., $rsp_{i,k}$. Then

the procedure goes to **Step (iii)**. However, if a_i is a BA or FA, a_i needs to contact its neighbors for a_k's request by initializing another management process. Such a procedure will be repeated until a GA or a SA is found. After a_i receives all responses from its neighbor agents, a_i will send a request $rsp_{i,k} = \sum_{m \in \mathbf{N_i}} rsp_{m,k}$ to a_k. However, if a_i receives a cancelation request from a_k before the response can be generated, a_i will also send the cancelation to its neighbors.

Step (iii): a_k receives a_i's response. If a_k's request can be fully satisfied by a_i, i.e., $rsp_{i,k} = req_k$, then the procedure is terminated and all related agents adjust their power according to their agreements. Otherwise, a_k will deduce the remaining power $req_k^r \leftarrow req_k^r - rsp_{i,k}$ (req_k^r is initialized as req_k) from next supplier in $\mathbf{N_k^{-r}}$. Then **Step (i)** to **Step (iii)** will be repeated until a_k's request is fully satisfied (i.e., $req_k^r = 0$) or the first supplier, whose price is not higher than P_k, is met. For the latter case, the procedure goes to **Step (iv)**.

Step (iv): a_k will send the request $req_{k,i} \leftarrow (req_k^r, pri_k)$ to its upstream agent, i.e., a_u. Then a_u will start another power management procedure to meet a_k's request by repeating **Step (i)** to **Step (iv)**. We assume that the extra power can be sent back to the substation eventually. Therefore, a_u can always fully satisfy a_k's request.

D. Discussion on Scalability and Complexity

It can be seen that the proposed *power management mechanism* employs a recursive strategy to search for power sources and dispatch the appropriate power accordingly. By comparing with the conventional centralized management, the proposed mechanism does not require knowledge of the DN as a whole, and can make quick response based on local information. However, the proposed mechanism will have a higher requirement on agents' communication ability. This will not be significant for small radial distribution systems, but further work needs to be carried out to investigate its impact on very long distribution systems.

In general, the proposed recursive mechanism will complete when (i) the power resources are found and the power supply and demand are balanced; or (ii) the procedure is abandoned because of a time restriction. Usually, by considering a normal radial DN, the number of times of recursion to send a request from a load/DG to a secondary bus is three (i.e., load/DG $\Rightarrow$ local bus $\Rightarrow$ feeder $\Rightarrow$ secondary bus), then the secondary bus will take an extra recursion (i.e., secondary bus $\Rightarrow$ substation) to get a response from a substation, or three extra recursions (i.e., secondary bus $\Rightarrow$ feeder $\Rightarrow$ local bus $\Rightarrow$ DG) to get a response from another DG on a different feeder. For an extreme complex case when multiple DGs and the unbalanced load are located on different feeders, in order to fulfill the *usage and cost objectives*, the secondary bus agent will contact all DG agents, which have not reached their maximum power outputs, before it contacts the substation agent. Therefore, the total number of recursion will be the three times of the feeder's number in the DN. Of course, if we consider to share DGs from different substations, the number of recursion will also increase exponentially in order to get cheaper power from other areas. Therefore, theoretically these is no limitation on the scalability of the proposed mechanism. But if we consider the practical issues such as restrictions on time and communication, the efficiency of the proposed mechanism may be impacted when the system scale is large. However, such practical issues in a large scale DN will also be of concerned to other modern and conventional power management mechanisms. The complexity of the proposed mechanisms are given below.

According to the UML diagram in Fig. 2, for both the *dis/connection mechanism*, because a_i needs to spend time on initializing (one pass), exchanging messages with a_j (two passes) and updating a_j's information (one pass); a_j needs to spend time on exchanging messages with a_i (two passes), updating a_i's information (one pass) and broadcasting its updating to other agents (n passes for the worst case); and other agents need to spend time on updating their own information (n passes), so the complexity of the *dis/connection mechanism* is $T(n) = 2n + 7 = O(n)$.

According to the UML diagram in Fig. 3, for the *power management mechanism*, the complexity of the case $req_k < 0$ is more complex than the case $req_k > 0$, therefore the complexity of the whole mechanism depends on the complexity of the case $req_k < 0$. Let n^s be the total power source's number (including all DGs and the substation), n^l be the total load's number, and n^{bf} be the total number of buses and feeders in a DN, then we should have $n = n^s + n^l + n^{bf}$. According to the mechanism when $req_k < 0$, an agent performs two steps, i.e., (i) seeking extra power and (ii) performing load shedding. For the first step, the agent will firstly spend one pass on initializing the procedure. Although the exact information of the agent's neighbors is unknown (including neighbor's number and type), each bus or feeder cannot immediately make the final decision and needs to initialize other procedure. Therefore, there will be another n^{bf} passes (the worst situation) for initialization. Also, for all the involved bus, feeder, substation and DG agents, each of them will spend three passes (i.e., receiving a message,

TABLE II
ONTOLOGY FOR AGENT COMMUNICATION

Name (Type)	Slot Name (Type)
Power (C)	from-power (Float), to-power (Float)
Cost (C)	from-cost (Float), to-cost (Float)
Limit (C)	limit-power (Power)
Line (C)	id (String), limit (Limit), power (Power), cost (Float)
Neighbor (C)	name (String), type (String), level (String), line (Line), cost (Cost), limit (Limit)
Dis/Connect (A)	type (String), level (String), line (Line), cost (Cost), limit (Limit)
UpdateCost (A)	cost (Cost)
UpdateLimit (A)	limit (Limit)
ChangePower (A)	change (Float), cost (Float)

making a decision, and replying the message) in order to response a request. Therefore, there will be another $3(n^s+n^{bf})$ passes (the worst situation), which results in the complexity of step one to $1 + 4n^{bf} + 3n^s$. For the load shedding step, the analysis is similar as the first step, but we just replace the DGs and the substations with the loads. Therefore, the total passes for the load shedding step (the worst situation) is $1+4n^{bf}+3n^l$. Eventually, the complexity of the mechanism is $T(n) = (1+4n^{bf}+3n^s)+(1+4n^{bf}+3n^l) = 2+3n+5n^{bf} \leq 2 + 8n = O(n)$.

IV. MAS Development

A. Java Agent Development Framework (JADE)

We employ the Java Agent Development(JADE) Framework to implement the proposed MAS. The communication among agents in JADE is carried out according to FIPA-specified Agent Communication Language (ACL) [24].

B. Agent Development

In order to fulfill the communication between adjacent agents, we define the ontology that agents must follow during their communications in Table II to ensure that one agent's messages are understandable by others. Five concept (C) ontologies, e.g. *Power*, *Cost*, *Limit*, *Line* and *Neighbor*, and five action (A) ontologies e.g. *Connect/Disconnect*, *Update-Cost*, *Update-Limit*, and *Change-Power*, are defined. The actual work that an agent does is carried out in behaviors. We define six agents' behaviors, e.g. *Connect*, *Disconnect*, *Update Cost*, *Update Limit*, *Change Power Supply*, and *Change Power Consumption* in Table III. A detailed explanation on how to develop agents by using JADE can be found in [24].

V. Simulation

In this section, we demonstrate the performance of the proposed MAS through a case study. The simulation contains three layers, i.e., (i) we employed InterPSS (Internet technology based Power System Simulator) to simulate a DN [25]. InterPSS is an open-source Java-based development project to enhance power system design, analysis, diagnosis, and operation; (ii) we employed JADE on top of InterPSS to implement the proposed MAS, and it is linked with InterPSS to monitor and control the DN; and (iii) we employed JUNG (Java Universal Network/Graph Framework) [26] on top of

TABLE III
DESCRIPTION OF BEHAVIORS ASSOCIATED WITH AGENTS

| Behaviors	Send Message	Receive Message	Description
Connect	performative "REQUEST", ID "connect-parent", sender "Downstream agent", receiver "Upstream agent", content "(type, 'child', line, cost, limit)"	performative "AGREE/REFUSE", ID "connect-parent", sender "Upstream agent", receiver "Downstream agent", content "(type, 'parent', line, cost, limit)"	Sends connection request to an upstream agent; receives "AGREE/REFUSE" for a successful/unsuccessful result; exchanges information in predefined format.
Disconnect	performative "REQUEST", ID "disconnect-parent", sender "Downstream agent", receiver "Upstream agent", content "(type, 'child', line, cost, limit)"	performative "AGREE/REFUSE", ID "disconnect-parent", sender "Upstream agent", receiver "Downstream agent", content "(type, 'parent', line, cost, limit)"	Sends disconnection request to an upstream agent; receives "AGREE/REFUSE" for a successful/unsuccessful result; exchanges information in predefined format.
Update Cost	performative "INFORM", ID "update-cost", sender "Neighbor agent", receiver "Neighbor agent", content "(cost)"	performative "INFORM", ID "update-cost", sender "Neighbor agent", receiver "Neighbor agent", content "(cost)"	Sends/receives/forwards price change inform to/from/to other neighbor agents.
Update Limit	performative "INFORM", ID "update-limit", sender "Neighbor agent", receiver "Neighbor agent", content "(limit)"	performative "INFORM", ID "update-limit", sender "Neighbor agent", receiver "Neighbor agent", content "(limit)"	Sends/receives/forwards limit change inform to/from/to other neighbor agents.
Change Power Supply	performative "AGREE/REFUSE", ID "change-power-supply", sender "the agent", receiver "a neighbor agent", content "(power, cost)"	performative "REQUEST", ID "change-power-supply", sender "a neighbor agent", receiver "the agent", content "(power, cost)"	receives change power supply "REQUEST" from a neighbor agent; replies "AGREE" if can fully satisfy the request and "REFUSE" if cannot; replies actual power supply change and the new price.
Change Power Consumption	performative "AGREE/REFUSE", ID "change-power-consumption", sender "the agent", receiver "a neighbor agent", content "(power, cost)"	performative "REQUEST", ID "change-power-consumption", sender "a neighbor agent", receiver "the agent", content "(power, cost)"	receives change power consumption "REQUEST" from a neighbor agent; replies "AGREE" if can fully satisfy the request and "REFUSE" if cannot; replies actual power consumption change and the new price.
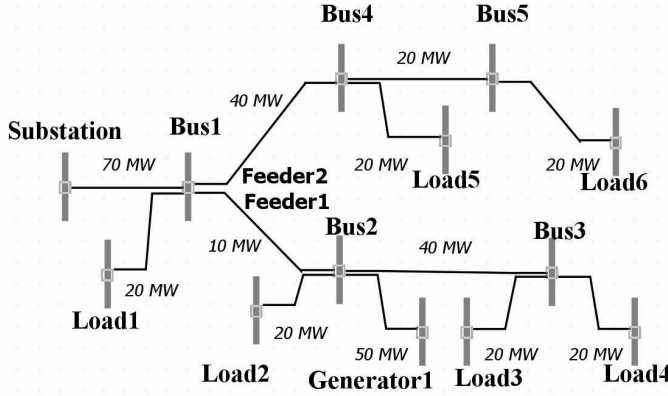
Fig. 4. An InterPSS output showing the one-line diagram of a distribution network.

JADE to implement the interface of the proposed MAS with InterPSS. JUNG is a free software library that provides a common and extendible language for the modeling, analysis, and visualization of data that can be represented as a graph or network.

In Fig. 4, a DN is simulated by using InterPSS. The DN contains one substation, two feeders, five buses, six loads, and one generator. The limits of power flow for substation, bus and feeder are set to 100 MW. The maximum power supply ability for the substation is set to 100 MW, and is connected to a 50 MW generator resource. The power consumption for each load is set to 20 MW, and we assume that all loads' consumption and the DG's supply can change dynamically in an unexpected way. It is also assumed that a DG's power price is cheaper than a substation's power price, and we set those two prices to $10 and $20, respectively. We set the length of all cables to 1 km, and assume that the cost for delivering 1 MW power on a cable is $1. Because all loads and DGs are dynamically changed, the power dispatched by the DN is also changed accordingly. Fig. 4 shows a moment when the Generator1 provides 50 MW to Bus2, and the Substation1

provides the remaining power (i.e., 70 MW) to the network. It is assumed that another Generator2, wants to connect to the network through Bus5 at this moment. The Generator2 is rated at 50 MW. In the following, we demonstrate the simulation of Generator2's connection by using our MAS.

In Fig. 5 (Messages (M) from 224 to 235), the communications between agents for implementing the Generator2's connection are displayed, and are explained in below.

(M224) GA2 sends a request and self's information to BA5 for connection. **(M225)** BA5 replies GA2's with an agreement and also sends self's information to GA2. **(M226-235)** BA5 informs FA2 and LA6 about this update. Then FA2 and LA2 further inform this update to their neighboring agents. Finally, this information is broadcasted in the network.

After GA2 was connected, it requests to supply 50 MW to the network through BA5 immediately. In Fig. 5 (messages from M236 to M248), the communications between agents for implementing the power dispatch are displayed. The explanations are given below.

(M236) GA2 requests to deliver 50 MW to BA5. **(M237)** BA5 receives GA2's request and performs the power management, i.e., BA5 will use GA2's power to replace the power from FA2 to support LA6 (20 MW), and deliver the remaining power (30 MW) back to FA2 (i.e., the power flow direction changes). So, BA5 further sends a request to FA2. **(M238)** FA2 receives BA5's request and performs the power management, i.e., FA2 will use BA5's power to replace the power from BA1 to support BA4 (20 MW), and deliver the remaining power (10 MW) back to BA1 (i.e., the power flow direction changes). So, FA2 sends a request to BA1. **(M239)** BA1 receives FA2's request and performs the power management, i.e., BA1 will stop the power supplying (40 MW) to FA2, and use the extra power (10 MW) from FA2 to support FA1 and LA1. Therefore, BA1 will decrease the power from SA1 by 50 MW, and BA1 sends this request to SA1. **(M240)** SA1 receives BA1's request and replies BA1 with an agreement to decrease its power supply by 50 MW. **(M241)** BA1 receives SA1's
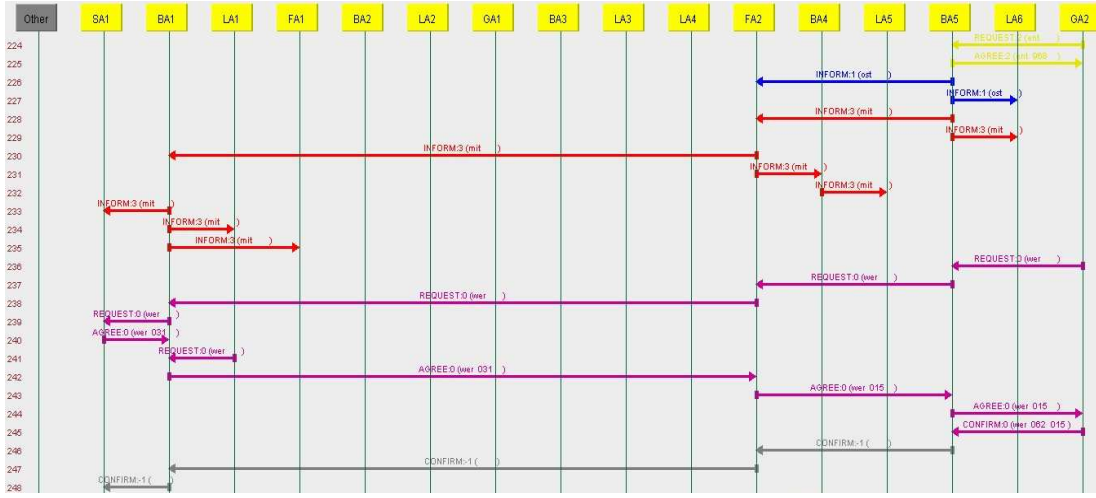
Fig. 5. The agents' communications when GA2 is connected
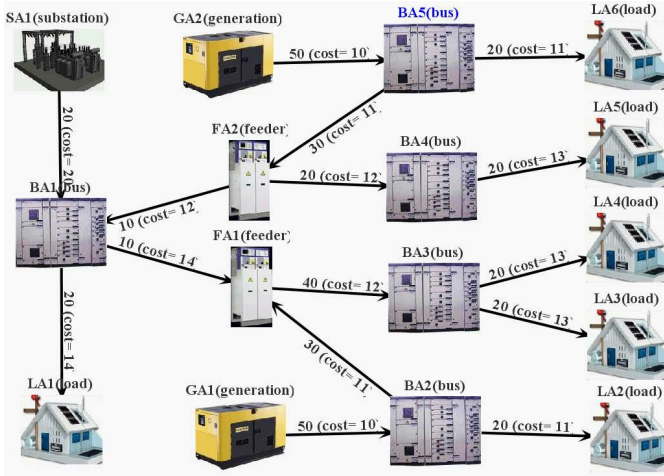

Fig. 6. The simulated distribution network after G2's connection.

response, and replies FA2 with an agreement to take the extra 10 MW. **(M243)** FA2 receives BA1's response, and replies BA5 with an agreement to take the extra 30 MW. **(M244)** BA5 receives FA2's response, and replies GA2 with an agreement to take the 50 MW. **(M245-248)** GA2 starts to supply power to the network through BA5, and all related agents arrange their power according the agreements mentioned above. The simulation of the network after GA2's connection by using JUNE is illustrated in Fig. 6.

The results from the demonstration show that the proposed MAS is effective in dynamically executing power management in a DN when the power supply and/or demand changes. We also employed the classical centralized management [4] on the same network to repeat the above connection and power dispatch procedures. According to [4], the centralized management scheme should contain a distribution management system controller, which accepts equipment status measurements at selected locations in a distribution network. The controller would then use the global information of the network to estimate the 'best' dispatching solution by considering each equipment's specification. In this paper, we also follow this well accepted scheme in the simulation of the centralized management. The central controller is assumed to be located at

the substation, and is implemented by a controller agent using JADE. All loads and DGs are selected as the measurement points, and their statuses are reported to the controller agent directly by local monitor agents. The communication scheme between the central controller agent and all local monitor agents also follows FIPA's ACL scheme. Comparing with the proposed distributed bus agent, the central controller agent acts more likes a super bus agent with the global information and control ability. For comparison purpose, the objectives and constraints of the central controller agent are set to be the same as the proposed MAS (see formulas (1)-(8)). The optimal solution for the objectives can be calculated directly by the central controller agent with the global information, and then the controller agent adjusts the substation's and the DGs' power output accordingly.

We keep both the decentralized and centralized systems on running for 48 hours after GA2's connection, and let all loads and DGs dynamically and randomly modify their consumptions and supplies, respectively. In Fig. 7, we display the historical record monitored by InterPSS on Bus2. The fluctuation displayed in Fig. 7 is caused by the random change of loads' consumption and DGs' supply. We can see that the proposed MAS can always balance the supply and demand by considering the economy.

Then we summarize the time (which includes the time of information exchange between InterPSS and JADE) spent by the two systems to respond to all loads and DGs' requests in Table IV. It can be seen that the average response time spent by the proposed decentralized system is only 12.3% of the time spent by the centralized system. That is because the centralized system needs to deliver all messages to the central controller agent for decision making (i.e., large amount of time is consumed on communication and computation), while the proposed decentralized system can make the decision locally. However, the response time spent by the proposed decentralized system on each agent's request is impacted by the agent's location and nearby components, and is not as consistent as the centralized system. Generally, the decentralized system spent less time on answering the agents, who have energy resources nearby, such as LA1, LA2 and LA6, but more time
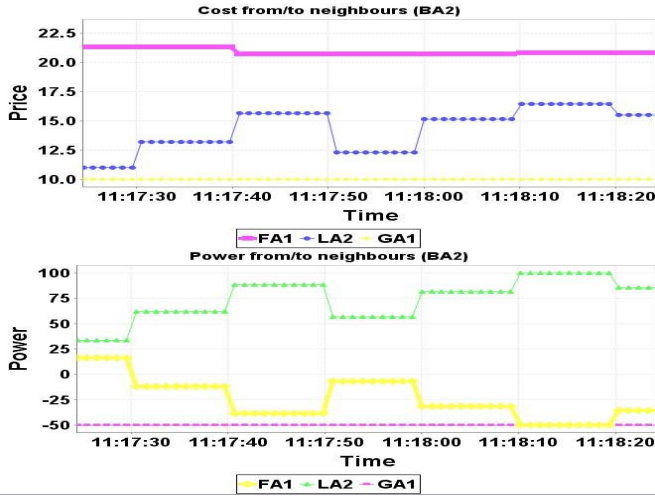
Fig. 7. The historical records of BA2.

TABLE IV
TIME NEEDED OF THE PROPOSED MAS

Agent	Centralized Time (Sec)			Decentralized Time (Sec)		
	Min	Max	Avg	Min	Max	Avg
LA1	0.015	1.322	0.201	0.002	0.030	0.020
LA2	0.018	1.317	0.250	0.004	0.059	0.021
LA3	0.020	1.511	0.289	0.007	0.949	0.028
LA4	0.020	1.584	0.216	0.006	0.863	0.026
LA5	0.021	1.403	0.286	0.004	0.771	0.025
LA6	0.020	1.378	0.245	0.003	0.055	0.020
GA1	0.022	1.661	0.235	0.016	0.742	0.058
GA2	0.024	1.828	0.320	0.020	0.747	0.054

on answering other agents, who do not have energy resources nearby, such as LA3, LA4 and LA5. Also, the decentralized system spent longer time on answering DGs' requests than loads' requests on average. That is because the DGs' requests have higher chances to impact the substation's power supply than the the loads' requests.

VI. RELATED WORK

Some agent-based systems were already implemented by researchers to provide supports on power system control and management by considering DGs. To list few of them, Pezeshki et al. [3] reviewed the use of multiagent systems to model the impacts of high levels of photovoltaic system in Perth Solar City project, and investigated the potential problems caused by voltage regulation though agent-based simulation. The simulation results confirmed that the usage of DGs could cause problems such as over-voltage, short circuit, protection desensitization, and incorrect operation in a DN. Finally, the paper suggested to employ more agents to monitor and control the loads, the DGs, the feeders, and the power quality in a DN, which is also consistent with our motivation.

Farag and El-Saadany [8] studied the voltage regulation in unbalanced distribution feeders with high penetration of DGs. Their research showed that the traditional control techniques was no longer suitable for voltage regulation when DGs are considered, and could cause overvoltage and/or undervoltage problem because of incorrect operations on voltage regulators. The reasons behind such incorrect operations were mainly because the regulator only employs one-way communication with other components and lacks of DGs information. Finally,

a multiagent systems with three types of agents, i.e., regulator agent, capacitor agent, and DG agent, were proposed to solve this incoordination problem through agent communication and cooperation. The power management problem studied by our paper has similar motivation as Farag and El-Saadany's work. But our proposed MAS considers more electric components, and eventually optimized the power management results by considering both efficiency and economy.

Miller et al. [27] modeled the electricity balance problem between DGs and loads as an optimal dispatch problem, and solved the problem through minimizing CO_2 emissions of a DN. Firstly, a factor graph was created based on the structure of a DN by considering only DGs and loads, then the optimal solution was calculated to set the electricity output for each generator by targeting the satisfaction of all loads and the minimization of all DGs' CO_2 emissions. However, their work only focused on the balance problem between DGs' supply and load demands, but other practical issues such as economy, electricity delivery, and variabilities of DGs and loads, are not considered. Also, it is not clear how their proposed MAS was implemented and operated to fulfil the systemic goals.

Kok et al. [10] proposed a MAS to match the electricity supply and demand in electricity network by considering the economical optimization. The electricity supply and consumption devices were modeled by agents, and they communicated with the SD-Matcher agent to bid the electricity price. Through the cooperations between SD-Matcher agents in different hierarchical levels, the price agreements between electricity suppliers and consumers were created. Finally, a contract network was constructed based on physical connections between the electric devices, and the price agreements were achieved. However, their work focused more on the electricity market and tried to achieve the economical optimization. Our work presented more details on electric components management, agents communication, agents cooperation and MAS implementation.

Huang et al. [22] proposed a market-based MAS to perform the automatical reconfiguration of radial electric shipboard power system. The issues such as power cost and limitation on components were also considered. Through communication with neighbor agents, the behaviors such as switching between power suppliers and modifying power supplying could be automatically performed. However their MAS focused on a shipboard power system with a fixed electricity network, while our MAS pays attentions on the DN by considering DGs' variabilities.

VII. CONCLUSION

DG has been considered as a supplemental energy resource to existing electricity network, and can provide additional supports to solve the voltage drop and energy loss problems. However, the uncertainty of DG's location and the variability of DG's electricity output has caused a serious systemic management problem to the existing control techniques. The conventional control techniques were no longer suitable to control an electricity network with DGs due to their limitations on flexibility, communication, cooperation, and decision

making. In this paper, a multiagent solution was proposed to solve the power dispatching problem by considering the variabilities of DGs and loads. Five types of agent were proposed to model the key electric components in a DN, and three system objectives were formulated to guarantee the system balance as well as to optimize the system's efficiency and economy. Through the communication and cooperation between the proposed agents, unexpected imbalances in the DN were effectively resolved by considering all objectives. We also introduced the details on the MAS implementation by using InterPSS, JADE and JUNG, and demonstrated the good system performance through the simulation.
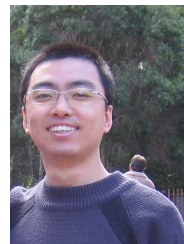
ACKNOWLEDGEMENT

REFERENCES

[1] I. Baxevanos and D. Labridis, "Implementing Multiagent Systems Technology for Power Distribution Network Control and Protection Management," *IEEE Trans. on Power Delivery*, vol. 22, no. 1, pp. 433–443, 2007.

[2] M. Nordman and M. Lehtonen, "Distributed Agent-Based State Estimation for Electrical Distribution Networks," *IEEE Trans. on Power Systems*, vol. 20, no. 2, pp. 652–658, 2005.

[3] H. Pezeshki, P. J. Wolfs, and M. Johnson, "Multi-Agent Systems for Modeling High Penetration Photovoltaic System Impacts in Distribution Networks," in *Proc. of Innovative Smart Grid Technologies Asia*, pp. 1–8, 2011.

[4] P. Vovos, A. Kiprakis, A. Wallace, and G. Harrison, "Centralized And Distributed Voltage Control: Impact on Distributed Generation Penetration," *IEEE Trans. on Power Systems*, vol. 22, no. 1, pp. 476–483, 2007.

[5] H. Wan, K. Li, and K. Wong, "Multi-Agent Application of Substation Protection Coordination with Distributed Generators," *European Trans. on Electrical Power*, vol. 16, no. 5, pp. 495–506, 2006.

[6] T. Ackermann, G. Andersson, and L. Söder, "Distributed Generation: A Definition," *Electric Power Systems Research*, vol. 57, no. 3, pp. 195–204, 2001.

[7] P. Barker and R. Mello, "Determining the Impact of Distributed Generation on Power Systems. I. Radial Distribution Systems," in *Power Engineering Society Summer Meeting*, vol. 3, pp. 1645–1656, 2000.

[8] H. Farag and E. El-Saadany, "Voltage Regulation in Distribution Feeders with High DG Penetration: From Traditional to Smart," in *Proc. of IEEE Power and Energy Society General Meeting*, pp. 1–8, 2011.

[9] M. Doyle, "Reviewing the Impacts of Distributed Generation on Distribution System Protection," in *Power Engineering Society Summer Meeting*, vol. 1, pp. 103–105, 2002.

[10] J. Kok, C. Warmer, and I. Kamphuis, "PowerMatcher: Multiagent Control in the Electricity Infrastructure," in *Proc. of the fourth Int. conf. on Autonomous agents and multiagent systems*, pp. 75–82, 2005.

[11] R. Rayudu, "A Knowledge-based Architecture for Distributed Fault Analysis in Power Networks," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 4, pp. 514–525, 2010.

[12] I. Baxevanos and D. Labridis, "Software Agents Situated in Primary Distribution Networks: A Cooperative System for Fault and Power Restoration Management," *IEEE Trans. on Power Delivery*, vol. 22, no. 4, pp. 2378–2385, 2007.

[13] F. Ren, M. Zhang, D. Soetanto, and X. Su, "Conceptual Design of A Multi-Agent System for Interconnected Power Systems Restoration," *IEEE Trans. on Power Systems*, vol. 27, no. 2, pp. 732–740, 2012.

[14] J. Solanki, S. Khushalani, and N. Schulz, "A Multi-Agent Solution to Distribution Systems Restoration," *IEEE Trans. on Power Systems*, vol. 22, no. 3, pp. 1026–1034, 2007.

[15] M. Nordman and M. Lehtonen, "An Agent Concept for Managing Electrical Distribution Networks," *IEEE Trans. on Power Delivery*, vol. 20, no. 2 Part 1, pp. 696–703, 2005.

[16] C. Lin, C. Chen, T. Ku, C. Tsai, and C. Ho, "A Multiagent-Based Distribution Automation System for Service Restoration of Fault Contingencies," *European Trans. on Electrical Power*, vol. 21, no. 1, pp. 239–253, 2011.

[17] S. McArthur, E. Davidson, V. Catterson, A. Dimeas, N. Hatziargyriou, F. Ponci, and T. Funabashi, "Multi-Agent Systems for Power Engineering Applications‡Part I: Concepts, Approaches, and Technical Challenges," *IEEE Trans. on Power Systems*, vol. 22, no. 4, pp. 1743–1752, 2007.

[18] M. Pipattanasomporn, H. Feroze, and S. Rahman, "Multi-Agent Systems in A Distributed Smart Grid: Design and implementation," in *Power Systems Conf. and Exposition*, pp. 1–8, 2009.

[19] A. Saleem, K. Heussen, and M. Lind, "Agent Services for Situation Aware Control of Power Systems with Distributed Generation," in *IEEE Power and Energy Society General Meeting*, pp. 1–8, 2009.

[20] M. Baran and I. El-Markabi, "A Multiagent-based Dispatching Scheme for Distributed Generators for Voltage Support on Distribution Feeders," *IEEE Trans. on Power Systems*, vol. 22, no. 1, pp. 52–59, 2007.

[21] M. Abido, "Environmental/Economic Power Dispatch Using Multiobjective Evolutionary Algorithms," *IEEE Trans. on Power Systems*, vol. 18, no. 4, pp. 1529–1537, 2003.

[22] K. Huang, S. Srivastava, D. Cartes, and L. Sun, "Market-based Multiagent System for Reconfiguration of Shipboard Power Systems," *Electric Power Systems Research*, vol. 79, no. 4, pp. 550–556, 2009.

[23] K. Huang, S. Srivastava, D. Cartes, and M. Sloderbeck, "Intelligent Agents Applied to Reconfiguration of Mesh Structured Power Systems," in *Int. Conf. on Intelligent Systems Applications to Power Systems*, pp. 1–7, 2007.

[24] F. Bellifemine, G. Caire, and D. Greenwood, *Developing Multi-Agent Systems with JADE*, vol. 5. Wiley, 2007.

[25] M. Zhou and S. Zhou, "Internet, Open-source and Power System Simulation," in *IEEE Power Engineering Society General Meeting*, pp. 1–5, 2007.

[26] J. OMadadhain, D. Fisher, P. Smyth, S. White, and Y. Boey, "Analysis and Visualization of Network Data Using JUNG," *Journal of Statistical Software*, vol. 10, no. 2, pp. 1–25, 2005.

[27] S. Miller, S. Ramchurn, and A. Rogers, "Optimal Decentralised Dispatch of Embedded Generation in the Smart Grid," in *Proc. of the Nineteenth Int. conf. on Autonomous Agents and Multiagent Systems*, pp. 281–288, 2012.

BIOGRAPHIES

Dr. Fenghui Ren received his MCompSc-Res. and Ph.D. from the University of Wollongong in 2006 and 2010, respectively. He received his BCompSc from the Xidian University in 2003. Currently, he is an Research Fellow in the School of Computer Science and Software Engineering at the University of Wollongong. Dr. Ren is an active researcher and published 35 research papers. His research interests include agent-based modeling, simulation, reasoning and learning, agent coordination, negotiation and optimization.

A/Prof. Minjie Zhang received her BSc degree from Fudan University, China in 1982, and her PhD degree from the University of New England, Australia in 1996. Currently, she is the Director of Intelligent System Research Group in the Faculty of Informatics, at University of Wollongong. Dr. Zhang has published over 130 research papers. Her research interests include multi-agent systems, agent-based simulation and modeling in complex domains, agent-based mart grids and knowledge discovery and data mining.

Prof. Danny Sutanto received his B.Eng. and Ph.D. from the University of Western Australia in 1978 and 1981 respectively. In 2006, he joined the School of Electrical, Computer and Telecommunications Engineering at the University of Wollongong as the Professor of Power Engineering. Dr. Sutanto has published over 170 research papers. His main areas of research are power system analysis, power system economics, voltage stability, harmonics, power electronics and computer aided education. He is currently the IEEE industry applications society area chair for region ten.