Electric Power Systems Research 110 (2014) 45–54

Contents lists available at ScienceDirect

Electric Power Systems Research

journal homepage: www.elsevier.com/locate/epsr

Multi-agent control of community and utility using Lagrangian relaxation based dual decomposition

Vahid R. Disfani, Lingling Fan*, Lakshan Piyasinghe, Zhixin Miao

Department of Electrical Engineering, University of South Florida, Tampa, FL 33620, USA

ARTICLE INFO

Article history:
Received 24 October 2013
Received in revised form 4 January 2014
Accepted 6 January 2014
Available online 30 January 2014

Keywords:
Multi-agent system
Distributed algorithm
Dual decomposition
AC optimal power flow
Lagrangian relaxation

ABSTRACT

In multi-agent based demand response program, communities and a utility make decisions independently and they interact with each other with limited information sharing. This paper presents the design of multi-agent based demand response program while considering ac network constraints. This project develops two types of information sharing and iterative decision making procedures for the utility and communities to reach Nash equilibrium. The distributed algorithms of decision making are based on Lagrangian relaxation, duality, and the concept of upper and lower bounds. The first algorithm is subgradient iteration based distributed decision making algorithm and the second algorithm is based on lower bound and upper bound switching. The two algorithms require different information flow between the utility and communities. With the adoption of distributed algorithms, the utility solves optimal power flow at each iteration while considering ac network constraints, and the communities also conduct optimization. Through information sharing, the utility and the communities update their decisions until convergence is reached. The decision making algorithms are tested against three test cases: a distribution network IEEE 399 system, two meshed networks (IEEE 30-bus system and IEEE 300-bus system). Fast convergence is observed in all three cases, which indicates the feasibility of the demand response design.

1. Introduction

Multi-agent control has been applied in microgrid or demand side interactions with utility [1–4]. Micrograds or demand side make their own decisions while exchanging limited information with the grid. For example, in [1], an optimal demand response is designed for the demand sides to bid the amount of load shedding as a supply function of price. The utility collects the bids from all demand sides and update the price. In [2], the demand sides send the utility the information on their demands, and the utility sets the prices. The demand sides update their demand requests upon receiving the price.

In all above mentioned references, distribution networks are either represented in a simplified way or not represented at all. The objective of this paper is to implement multi-agent control of demand sides and utility while considering ac network constraints including line flow limits and bus voltage limits.

Implementing multi-agent control requires distributed algorithms. For optimization problems, there are ways to decompose and construct distributed optimization algorithms [5]. In the field of communication layering problems, primal decomposition, dual decomposition, and primal-dual decomposition can be applied in different scenarios [6]. In power systems optimization problems, due to the decoupled cost function structure and coupled constraints, Lagrangian relaxation based dual decomposition is commonly used. Example applications can be found in aggregated PHEV control considering global constraint [7], and distributed voltage control [8].

In optimization decomposition, an original problem is separated into a master problem along with many subproblems with small sizes. After each subproblem is solved, the main problem is solved adopting iterative methods such as subgradient update. Subgradient algorithm based on Lagrangian relaxation has been applied by Luh *et al* for manufacturing job scheduling [9]. Zero or small duality gap can prove that the solution is optimal or very close to optimal. In game theory, iteration means each agent in the system is exchanging information and learning to reach a Nash equilibrium.

Not all distributed algorithms have the information exchange structure suitable for multi-agent control [10]. In this research, distributed algorithm and learning methods suitable for multi-agent based microgrid and utility interaction will be examined.

Subgradient update based distributed algorithms are popular as seen in the literature. One shortcoming of subgradient method is its slow convergence speed. Scaling factors of Lagrangian multipliers need to be updated to enhance convergence. The update is dependent on specific problems. An improved Lagrangian multiplier or

* Corresponding author. Tel.: +1 813 974 2031; fax: +1 813 974 5250.
E-mail address: linglingfan@usf.edu (L. Fan).

0378-7796/$ – see front matter © 2014 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.epsr.2014.01.009

price update scheme is presented in this paper to improve convergence. In addition, this paper proposes an alternative algorithm based on lower bound and upper bound switching to have a faster convergence speed. The philosophy of the bound switching algorithm is similar as the philosophy of Bender's decomposition where lower bound and upper bound are computed. Details and refinement of the bound switching algorithm and the requirement of information exchange structure for demand response program are presented in this paper. Both algorithms are tested against multiple case studies: a radial network IEEE 399 system and two meshed networks (IEEE 30-bus system and IEEE 300-bus system).

The rest of the paper is organized as follows. Section 2 describes Lagrangian relaxation, Lagrangian dual problem and the concept of upper and lower bounds. Section 3 describes the decomposition of the utility and community optimization problems and the two demand response programs based on subgradient update and a bound switching algorithm. Section 4 presents numerical results and remarks. Section 5 concludes this paper.

2. Lagrangian relaxation based dual decomposition

2.1. Lagrangian relaxation

An optimization problem is generally defined as

$$\begin{aligned} f_0^* = f_0(\mathrm{x}^*) = \min_{\mathbf{x}} \quad & f_0(\mathbf{x}) \\ \text{subject to} \quad & f_i(\mathbf{x}) \le 0 \quad 1 \le i \le m \\ & h_j(\mathbf{x}) = 0 \quad 1 \le j \le p \end{aligned} \tag{1}$$

where $\mathbf{x}^*$ is the optimal solution of the decision variable vector $\mathbf{x}$. $f_0(\mathbf{x})$ is the objective function. $f_i(\mathbf{x})$ is an inequality constraint and $h_j(\mathbf{x})$ is an equality constraint.

Lagrangian relaxation technique relaxes the minimization problem by transferring constraints to objective function in the form of weighted sum as shown in (2).

$$L(\mathbf{x}; \lambda, \mu) = f_0(\mathbf{x}) + \sum_{i=1}^{m} \mu_i f_i(\mathbf{x}) + \sum_{j=1}^{p} \lambda_i h_j(\mathbf{x}) \tag{2}$$

where λ_i and μ_i are Lagrangian multipliers or weights. μ_i should be greater or equal to zero.

Considering the Langrange dual function, $g(\lambda, \mu)$, as the greatest lower bound of $L(\mathbf{x}\,; \lambda, \mu)$, the Lagrange dual problem is defined as (3).

$$g^* = \max_{\{\lambda,\mu\}} g(\lambda, \mu) = \max_{\{\lambda,\mu\}} \{\inf_{\mathbf{x}} L(\mathbf{x}; \lambda, \mu)\} \tag{3}$$

According to the weak duality theorem, for any feasible solution (λ, μ) of the dual problem (3) and any feasible solution $\mathbf{x}$ of the original problem (1), the following relationship is true.

$$g(\lambda, \mu) \le g^* \le f_0(\mathrm{x}^*) \le f_0(\mathbf{x}) \qquad \forall \mu \in \mathbb{R}_+^m, \lambda \in \mathbb{R}^p \tag{4}$$

Therefore, any feasible solution of the dual problem can result in a lower bound of the optimal value of the original problem (1).

2.2. Lower bound, upper bound, and gap

The definition of the upperbound (*UB*) and lowerbound (*LB*) must guarantee that $UB \ge f_0^*$ and $LB \le f_0^*$, respectively.

Indicated from the previous subsection, the cost corresponding to any feasible solution for the dual problem (3) is a **Lower Bound**.

On the other hand, since the optimal solution $\mathbf{x}^*$ for original problem (1) leads to the minimum cost, the resulting cost of any feasible solution $\mathbf{x}$ is an **Upper Bound**.

The difference between an upper bound and a lower bound which indicates the efficiency of the solution sought is called *Duality Gap* or *Gap*. (*Gap=UB–LB*).

3. System model and algorithms

Consider a power network consisting of a set N of buses and a set B of branches. The utility is responsible to operate the power grid, its generation units and transactions with transmission systems. Some community microgrids are connected to the network and behave as autonomous agents. The connected buses belong to a set A. The buses that belong to utility belong to a set $N - A$.

The communities share with the utility only limited information, which implies the following situations:

- Due to privacy issues, a community does not fully share information to the grid.
- Due to computing burden, the energy management center of a utility has no ability to collect every piece of information from customers. Instead, it is more feasible to have aggregated loads.
- The utility and the communities all behave as autonomous agents.

3.1. Lagrange relaxation and decomposition of optimal power flow problems

Optimal Power Flow (OPF), the well-known problem in power system operation, is defined in (5). It is obvious that an AC OPF takes care of not only active and reactive power balance constraints but also the other constraints such as voltage constraints, power line capacities, maximum and minimum limits of generators, etc.

$$\begin{aligned} \min \quad & \sum_{i \in N} C_i(P_{g_i}) \\ \text{subject to} \quad & \forall i \in N, \quad \forall j \in B \\ & P_{g_i} - P_{L_i} - P_i(V, \theta) = 0 \\ & Q_{g_i} - Q_{L_i} - Q_i(V, \theta) = 0 \\ & V_i^m \le V_i \le V_i^M \\ & P_{g_i}^m \le P_{g_i} \le P_{g_i}^M \\ & Q_{g_i}^m \le Q_{g_i} \le Q_{g_i}^M \\ & S_j(V, \theta) - S_j^M \le 0 \end{aligned} \tag{5}$$

where $C(\cdot)$ is the cost function, superscripts M and m denote upper and low bounds. Subscript i refers to the variables corresponding to bus i. P_g, Q_g, P_L and Q_L are the vectors of bus real and reactive power injection, and real and reactive loads. $P(V, \theta)$ and $Q(V, \theta)$ are the power injection expressions in terms of bus voltage magnitude and phase angles. $S(V, \theta)$ is the vector of line complex power flow.

Let us define two subscripts $(\cdot)_{imp_i}$ and $(\cdot)_{\exp_i}$ which are used in Fig. 1. The subscript $(\cdot)_{imp_i}$ denotes the utility's power import from a community connected to bus i while $(\cdot)_{\exp_i}$ denotes the same community's power export to utility. Hereafter in the paper, the community connected to bus i will be called community i for simplicity. In order to meet the power balance constraint, (6) must be fulfilled for all $i \in A$.

$$\begin{aligned} P_{imp_i} &= P_{\exp_i} \\ Q_{imp_i} &= Q_{\exp_i} \end{aligned} \tag{6}$$

In order to decompose the OPF problem between utility and communities, joint constraints (6) can be relaxed using Lagrange relaxation. In both utility's and communities' optimization problems, these constraints are not considered explicitly, but rather are

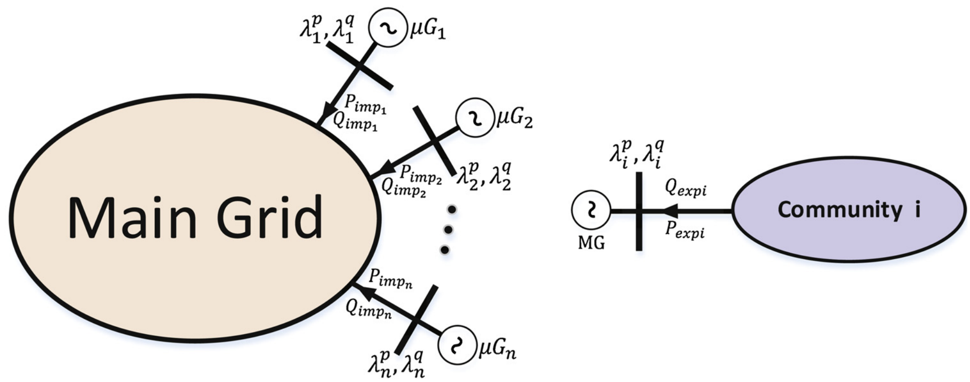

Fig. 1. Power networks in utility and community optimization problems.

used as a terminating conditions. Applying Lagrangian relaxation to (5) leads to (7).

$$\begin{array}{ll} \min & \sum_{i\in A} C_i(P_{g_i}) + \sum_{i\in N-A} C_i(P_{g_i}) + \sum_{i\in A} \lambda_i^p(P_{imp_i} - P_{\exp_i}) + \sum_{i\in A} \lambda_i^q(Q_{imp_i} - Q_{\exp_i}) \\ \text{subject to} & \text{Constraints of}\,(5) \end{array} \tag{7}$$

where λ_i^p and λ_i^q are Lagrangian multipliers which also determine the marginal prices for active and reactive power exchange between the utility and community i.

The decomposition is illustrated in Fig. 1 and explained as follows.

Decomposition of (7) creates agent-based optimization problems. In these agent-based subprograms, the Lagrangian multipliers λ_i^p and λ_i^q are given. The utility level decides utility generator active and reactive power outputs (P_{g_i}, Q_{g_i}) and power import levels (P_{imp_i}, Q_{imp_i}). The problem is formulated in (8).

$$\begin{array}{lll} \min & \sum_{i\in N-A} C_i(P_{g_i}) + \sum_{i\in A} [\lambda_i^p P_{imp_i} + \lambda_i^q Q_{imp_i}] & \\ \text{subject to} & \text{Constraints in}\,(5) & \\ & P_{imp_i}^m \le P_{imp_i} \le P_{imp_i}^M & \forall i \in A \\ & Q_{imp_i}^m \le Q_{imp_i} \le Q_{imp_i}^M & \forall i \in A \end{array} \tag{8}$$

Each community will solve an optimization problem (9), where the power prices are given and the decision variables are the community system generator output power levels (P_{g_i} and Q_{g_i}) and exporting power levels ($P_{\exp_i}$ and $Q_{\exp_i}$). In this paper, it is assumed that each community has only a generator and a defined power load. Thus, no AC network constraints appear in community optimization problem (9).

$$\begin{array}{ll} \min & C_i(P_{g_i}) - \lambda_i^p P_{\exp_i} - \lambda_i^q Q_{\exp_i} \\ \text{subject to} & P_{g_i} - P_{L_i} - P_{\exp_i} = 0 \\ & Q_{g_i} - Q_{L_i} - P_{\exp_i} = 0 \\ & P_{g_i}^m \le P_{g_i} \le P_{g_i}^M \\ & Q_{g_i}^m \le Q_{g_i} \le Q_{g_i}^M \end{array} \tag{9}$$

3.2. Subgradient algorithm

The subgradient method is an iterative method in which active and reactive power price signals are first specified by the Price Update Center (PUC) as illustrated in Fig. 2. The price signals are used by the utility and communities to define their power import and export levels. Any mismatch between corresponding power import and export levels is then used to update the price values for the next iteration. Fig. 2 also depicts the information flow between these three blocks through the algorithm. The utility is assumed to know the maximum and minimum power export level of each community and they are kept constant during the iterations.

3.2.1. Price updating

The Price Update Center is responsible for updating the electricity price at each bus in the power grid through comparing the power import and export levels announced by the utility and communities, respectively. The update procedure agrees the well-known economic rule: the excess demand always leads to increase the price while excess supply causes to reduce it.

3.2.1.1. Conventional subgradient method. Subgradient method based price update is given as follows.

$$\begin{aligned} \lambda_i^p(k+1) &= \lambda_i^p(k) + \alpha_i^p(k)\cdot(P_{imp_i} - P_{\exp_i}) \\ \lambda_i^q(k+1) &= \lambda_i^q(k) + \alpha_i^q(k)\cdot(Q_{imp_i} - Q_{\exp_i}) \end{aligned} \tag{10}$$

where k is the index of step, $\alpha_i^p(k)$ and $\alpha_i^q(k)$ denote the step sizes to update active and reactive power prices of bus i at kth iteration. The values $P_{imp_i} - P_{\exp_i}$ and $Q_{imp_i} - Q_{\exp_i}$ are the gradients.

The α values are required to get updated (reduced) during the algorithm run to prevent the algorithm to fluctuate over the final solution. It is reported in the literature that the convergence of algorithm to the optimal solution is guaranteed for a decreasing step-size rule $\alpha_i(k) = (1+m)/(k+m)$ and $m \ge 0$, if the gradient/subgradient $P_{err_i} = (P_{imp_i} - P_{\exp_i})$ and $Q_{err_i} = Q_{imp_i} - Q_{\exp_i}$ are bounded [6].

3.2.1.2. Modified subgradient algorithm. A shortcoming of subgradient method is its slow convergence rate. The conventional update algorithm reduces the convergence speed and needs to be modified for two reasons.

i It updates all alpha values even if any update is not needed, which make the algorithm meet $P_{err_i} = 0$ and $Q_{err_i} = 0$ in more iterations.
ii Its effectiveness in updating α, which can be indicated by its derivative $d\alpha_i(k)/dk = -(1+m)/(k+m)^2$, decreases significantly for big number of iterations.

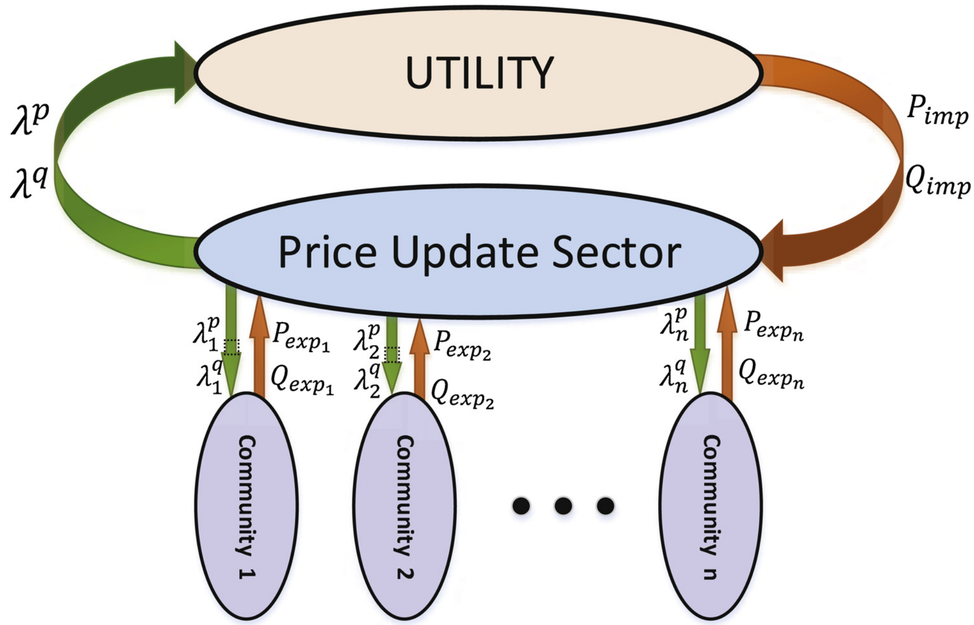

Fig. 2. Information flow of subgradient algorithm.

Therefore a modified algorithm (11) is proposed in this paper.

$$\alpha_i^p(k) = \begin{cases} \dfrac{\alpha_i^p(k)}{2} & \text{if } P_{err_i}(k)\cdot P_{err_i}(k-1) \le 0 \\ \alpha_i^p(k) & \text{otherwise} \end{cases}$$
$$\alpha_i^q(k) = \begin{cases} \dfrac{\alpha_i^q(k)}{2} & \text{if } Q_{err_i}(k)\cdot Q_{err_i}(k-1) \le 0 \\ \alpha_i^q(k) & \text{otherwise} \end{cases} \tag{11}$$

The two major changes in the modified algorithm compared to the conventional method are:

i Instead of updating all α values at each iteration, only those α values which require updating will change.
ii The method proposed reduces α values only when the sign of their corresponding gradients changes. Otherwise, there is no change.

3.2.2. Utility optimization

After utility receives the price profile from PUC, it runs OPF assuming that the communities are traders selling their power with a linear cost function or a fixed price equal to the updated price. The OPF determines how much active and reactive power the utility is willing to purchase or import from each community. These values create the power import vector P_{imp} and Q_{imp}, which can be either positive or negative. The utility announces P_{imp} and Q_{imp} to the PUC. MATPOWER 4.1 [11] is employed to solve the utility optimization problem.

3.2.3. Community optimization

In order to define its power export level, each community must perform its optimization based on the given price from the PUC. In this paper, it is assumed that each community has only a generator and a defined power load. The optimization problem corresponding to the community connected to bus $i \in A$ is presented in (9).

The community receives the updated price, solves its own subproblem (9) and determines the amount of active and reactive power to sell or to export to the main grid ($P_{\exp_i}$ and $Q_{\exp_i}$) as shown in Fig. 2. Then the values of $P_{\exp_i}$, $Q_{\exp_i}$ are announced to PUC.

Algorithm 1 presents the subgradient method based algorithm.

Algorithm 1. Subgradient Method

```
initialize vectors λ^p and λ^q
while Convergence is not met do
  Solve (8) given (λ^p, λ^q) to define (P_imp, Q_imp)
  for all i ∈ A do
    Solve (9) given (λ_i^p, λ_i^q) to define (P_exp_i, Q_exp_i)
  end for
  Check Convergence
  Update vectors α^p and α^q using (11)
  Update vectors λ^p and λ^q for the next step using (10)
end while
```

3.3. Lower-Upper-Bound Switching (LUBS) algorithm

Algorithms such as Branch-and-Bound and Bender's decomposition rely on finding lower bound and upper bound iteratively [12]. Once lower and upper bounds converge, the optimal solution is found.

We apply this philosophy in developing a new algorithm. Here the information flow between the utility and communities are shown in Fig. 3: the communities send the bid prices to the utility; the utility determines how much power to import from each community based on the bid price; and the communities update the bid price according to the utility decision.

The utility's role is exactly the same as that in the subgradient algorithm. However, there is no Price Update Center to update the power price vectors. In this structure, communities are responsible to determine the price vectors. The utility is assumed to know the maximum and minimum power export level of each community and they are kept constant during the iterations.

Mathematically speaking, when the communities announce prices and the utility determines power import level considering the prices, there is one common price vector shared by the communities and the utilities. As the solution found is a feasible one for the dual problem (7), aggregating the costs of all communities and utility leads to a *lower bound*.

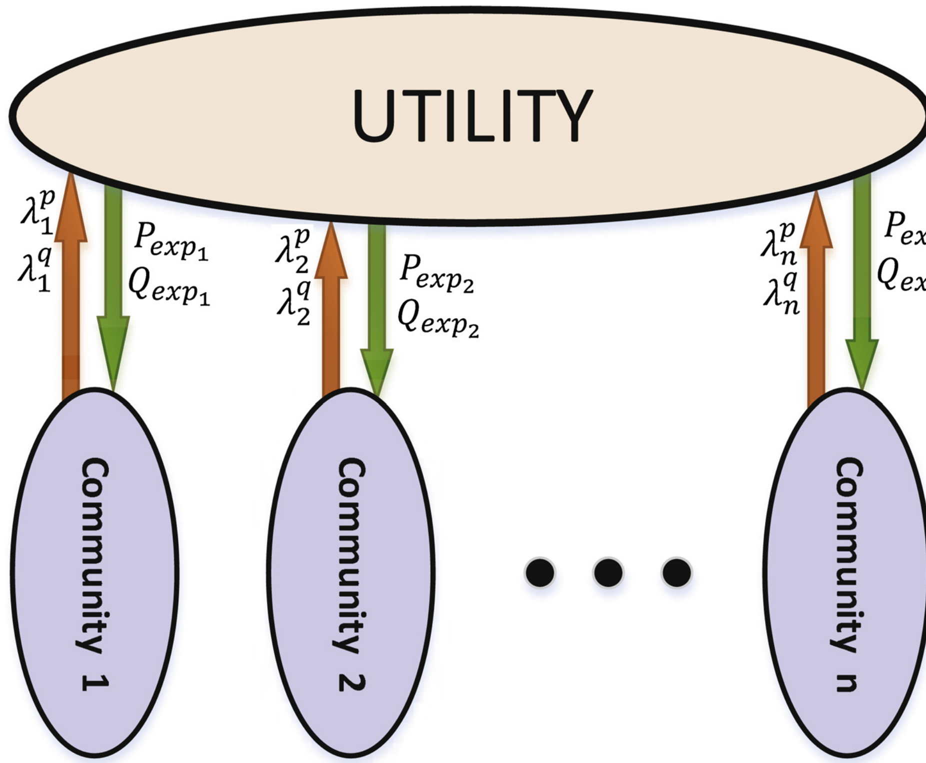

Fig. 3. Information flow of LUBS algorithm.

On the other hand, when the utility announces power import levels and the communities determine prices based on the power import levels, the coupling constraints (6) are complied. As the decision variables are in the feasible region of the master optimization problem (5), aggregating the costs leads to an *upper bound*.

The algorithm relies on iteration and the converging of the lower and upper bounds.

3.3.1. Utility optimization

At iteration k, the utility solves (8) given price vectors $(\lambda_i^p(k), \lambda_i^q(k))$ at the first iteration or updated price vectors at the next iterations to determine the power import vector (P_{imp}, Q_{imp}). Then, the vectors P_{imp} and Q_{imp} are announced to the communities. The procedure to solve utility OPF problem is the same as that in the subgradient method.

The utility also computes the utility cost $C_u(k)$ at kth iteration.

3.3.2. Community optimization and price updating

Once the communities receive the utility's power import vector, they must determine their local generator outputs and announce their price signals back to the utility.

As general, the community connected to the bus $i \in A$ seeks the Lagrangian multipliers related to the equality constraints $(\lambda_i^{p^*}, \lambda_i^{q^*})$ in (12). When the limits are not binding, the prices are same as the marginal prices of the generators.

$$
\begin{aligned}
\min \quad & C_i(P_{g_i}) \\
\text{subject to} \quad & P_{g_i} - P_{L_i} = P_{imp_i} \\
& Q_{g_i} - Q_{L_i} = Q_{imp_i} \\
& P_{g_i}^m \le P_{g_i} \le P_{g_i}^M \\
& Q_{g_i}^m \le Q_{g_i} \le Q_{g_i}^M
\end{aligned}
\tag{12}
$$

In a special case where the communities only have generation units and fixed loads, $(\lambda_i^{p^*}, \lambda_i^{q^*})$ are their marginal costs corresponding to $(P_{imp,i}, Q_{imp,i})$. For example, if the cost function of a generation unit is quadratic $C_i(P_{g_i}) = 0.5\alpha_i P_{g_i}^2 + \beta_i P_{g_i} + \gamma_i$, then the active power price is equal to $\lambda_i^{p^*} = \alpha_i(P_{imp,i} + P_{L_i}) + \beta_i$.

The cost of the community i at kth iteration, denoted as $C_{ci}(k)$, is computed by the community and announced to the utility.

3.3.3. Lower and upper bound computation

The lower bound can be found when the utility and the communities share the same price vector.

$$LB = C_u(k) + \sum C_{ci}(k-1) \tag{13}$$

The upper bound can found when the utility and the communities share the same power exchange.

$$UB = C_u(k) + \sum C_{ci}(k) \tag{14}$$

Duality gap is then calculated to decide whether to stop the iterations or continue.

3.3.3.1. Modification. In some cases, especially for meshed power networks, prices shuffle and have difficulty to converge. A modified LUBS algorithm is proposed in this paper. A parameter σ is introduced to smoothly update the prices using (15). This technique can prevent the instability of the algorithm although it may slightly decrease the convergence speed.

$$
\begin{aligned}
\lambda_i^p(k+1) &= \sigma \cdot \lambda_i^{p^*}(k) + (1-\sigma) \cdot \lambda_i^p(k) \\
\lambda_i^q(k+1) &= \sigma \cdot \lambda_i^{q^*}(k) + (1-\sigma) \cdot \lambda_i^q(k)
\end{aligned}
\tag{15}
$$

After the price vector is updated, the communities compute their generation cost considering the updated price in lower bound computation.

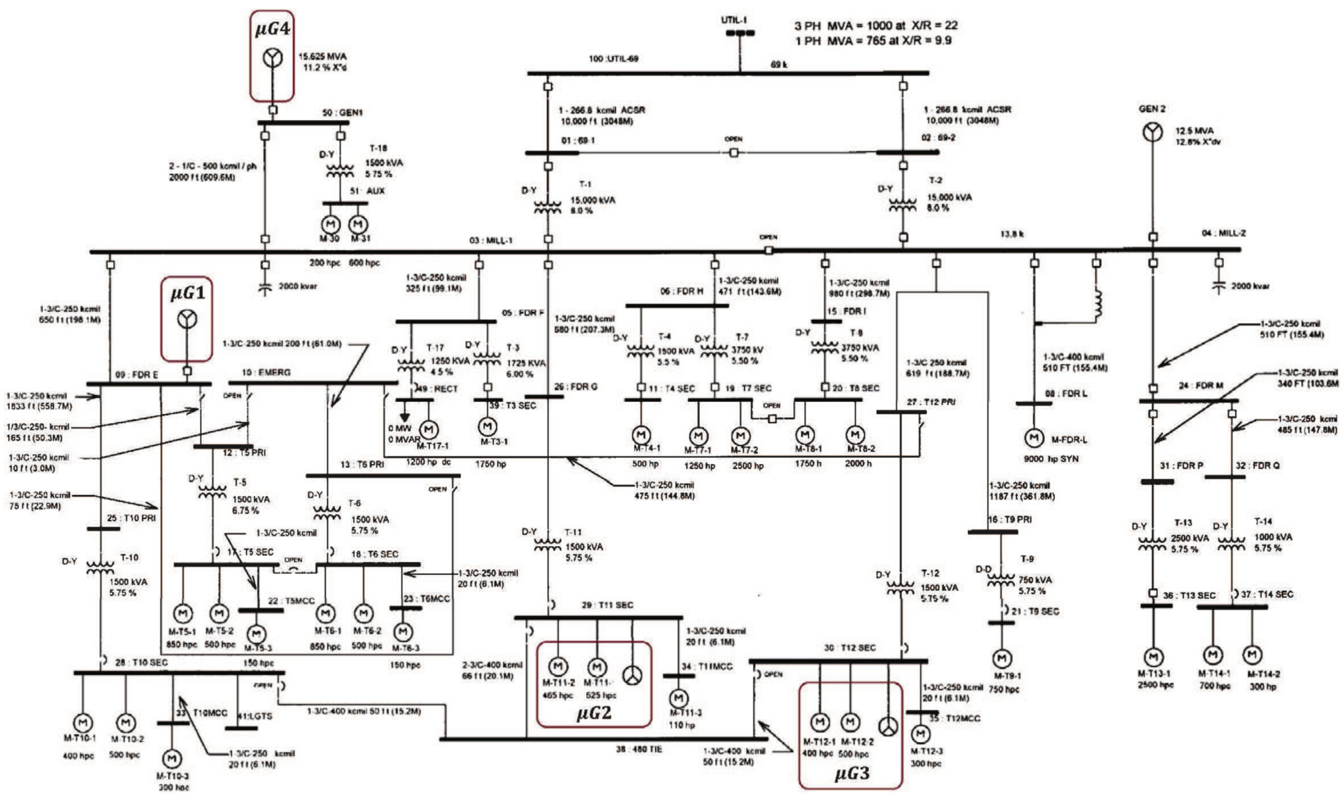

Fig. 4. IEEE Standard 399 [13] radial 42-bus test feeder with four community microgrids.

Algorithm 2 illustrates how LUBS method works.

Algorithm 2. LUBS Method

initialize vectors λ^p and λ^q
while Convergence is not met **do**
  Solve (8) given (λ^p, λ^q) to determine (P_{imp}, Q_{imp}, C_u)
  Compute C_u
  for all $i \in A$ **do**
    find $(\lambda_i^{p^*}, \lambda_i^{q^*})$ by solving optimization problem (12)
    Update vectors λ^p and λ^q using (15)
    Compute C_{ci}
  end for
  Compute **lower bound** using (13)
  Compute **upper bound** using (14)
  Check Convergence
end while

4. Numerical results

To demonstrate the strength of the algorithms, three different test systems are selected. The first one is the radial test feeder introduced in IEEE standard 399 [13], which is presented in Fig. 4. All normally-open switches are assumed to be open to have a redial system. Four community microgrids are connected to buses 9, 29, 30 and 50 of this system. The cost function parameters of all generators are presented in Table 1.

The second test system is IEEE 30-bus test power system [14], shown in Fig. 5 to demonstrate that the algorithms work for meshed power systems. Three community microgrids embedded to the system on buses 2, 13 and 27 are responsible to operate the generators and supply the loads already connected to these buses. The third case is based on IEEE 300-bus test system with three community microgrids integrated to buses 9002, 9051, and 9053.

Both modified subgradient and LUBS methods are applied on these test systems and the results are compared. Convergence criteria of Subgradient method are: $|\lambda(k) - \lambda(k-1)| < \$0.1$/MWh, $|P_{err}(k)| < 10^{-3}$pu and $|Q_{err}(k)| < 10^{-3}$pu for all communities. Convergence criteria of LUBS algorithm are: **gap** < \$0.1, $|\lambda(k) - \lambda(k-1)| <$ \$0.1/MWh, $|P_{imp}(k) - P_{imp}(k-1)| < 10^{-3}$pu, and $|Q_{imp}(k) - Q_{imp}(k-1)| < 10^{-3}$pu for all communities. The final solutions of all cases exactly match the optimal power flow solutions of the test systems, which demonstrates the accuracy of the final results for both algorithms.

4.1. Case 1

Case 1 is related to IEEE 399 system in Fig. 4. Fig. 6 presents the simulation results of both algorithms corresponding to the community microgrid 4 at bus 50. The plots on the left side show utility power import level versus community power export level and active power price at Bus 50 via modified subgradient algorithm. The algorithm converges in 38 iterations. The plots at the right side presents reactive power import, active power import announced by the utility and the active power price of Community 4. They converge to their final values in 9 iterations using Lower-Upper-Bound Switching (LUBS) algorithm with $\sigma = 1$.

The top left plot in Fig. 7 shows the upper bound and lower bound in LUBS algorithms of Case 1. The optimality of solution

Table 1
Parameters and cost functions of generators in Fig. 4 $C(P_g) = 0.5\alpha P_g^2 + \beta P_g + \gamma$.

Bus #	Owner	P_g^m	P_g^M	Q_g^m	Q_g^M	α	β	γ
100	Utility	0	2	−99	99	0.1	55	0
4	Utility	0	8	−2	8	0.3	50	0
9	$\mu G1$	0	1	−0.25	1	0.4	42	0
29	$\mu G2$	0	3	−0.75	3	0.4	35	0
30	$\mu G3$	0	2	−0.5	2	0.2	38	0
50	$\mu G4$	0	11	−2	8	0.2	49	0

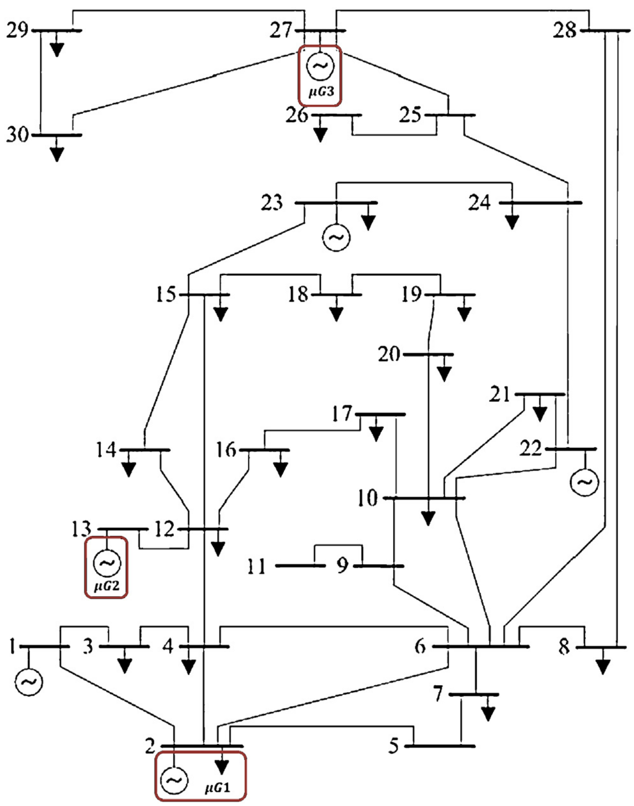

Fig. 5. IEEE 30-bus test system with three community microgrids.

is confirmed through the convergence of upper bound and lower bound.

Both methods (subgradient and LUBS) give the same answers for the final prices and imported/exported power. For the subgradient method, for the community at Bus 50, the price of exported power to the utility and the imported power by the utility is given. Both the utility and the community determine how much power to be imported and be exported. If the two powers have mismatch, then the price will be updated according to the subgradient algorithm in Algorithm 1. The price will be kept the same if the mismatch is zero. For the LUBS method, the community announces the price and the utility makes decision how much real power and reactive power to be imported. The real power and reactive power are the same as the exported power from the community. The community again computes the price based on its exporting powers. After a few iterations, powers and the price converge.

4.2. Case 2

In the second case, three community microgrids are integrated to IEEE 30-bus test system as depicted in Fig. 5. The simulation results of both modified subgradient algorithm and LUBS method for community microgrid 3 connected to bus 27 are presented in Fig. 8.

The left side plots show community's export and utility's import levels of reactive and active power applying Modified Subgradient algorithm. These values converge in 48 iterations. The right side plots are related to LUBS algorithm, showing the utility active and reactive power import level from community 3 as well as active power price. The numerical results demonstrate that the LUBS algorithm with $\sigma = 0.5$ converges in 10 iterations.

According to the top-right graph of Fig. 7, the upper bound and lower bound in LUBS algorithm converge as well.

4.3. Case 3

The third case is based on IEEE 300-bus test system with three community microgrids embedded. The simulation results of

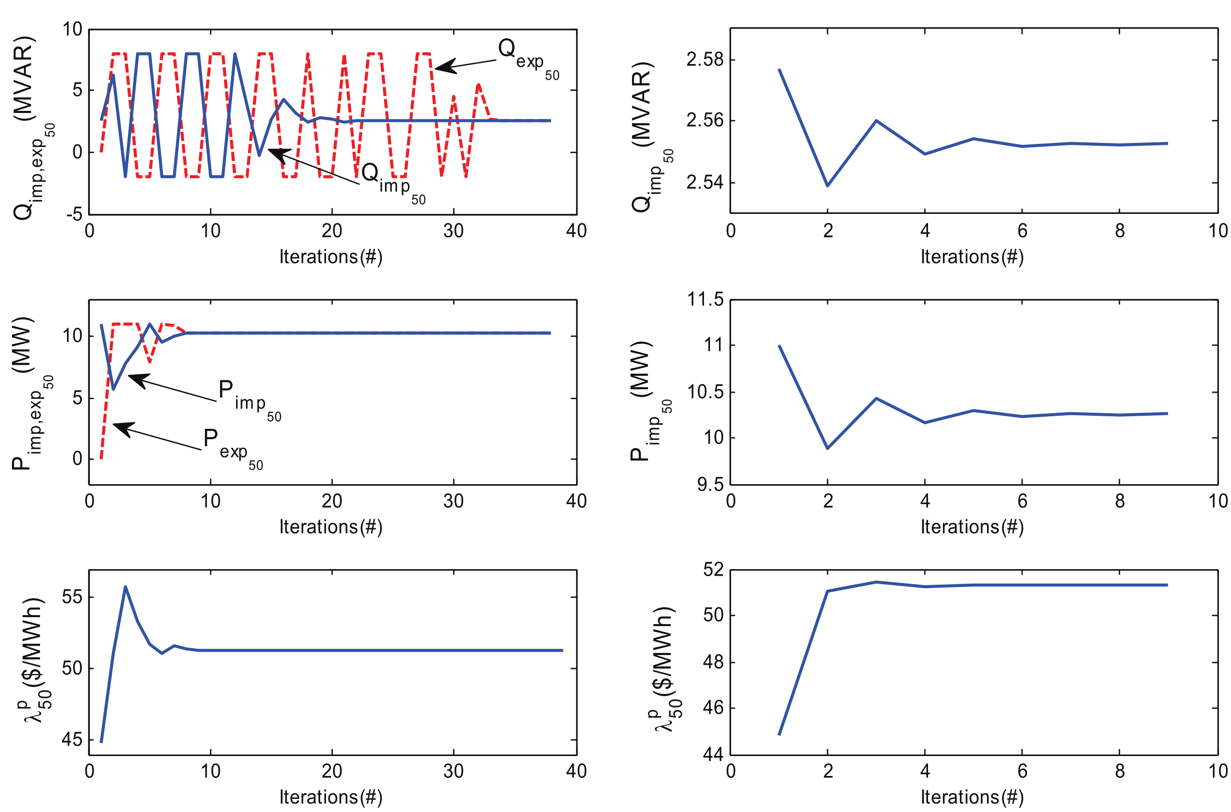

Fig. 6. Case 1 results: reactive power, active power, and active power price of community $\mu G4$. Subgradient method (left) and LUBS algorithm (right).

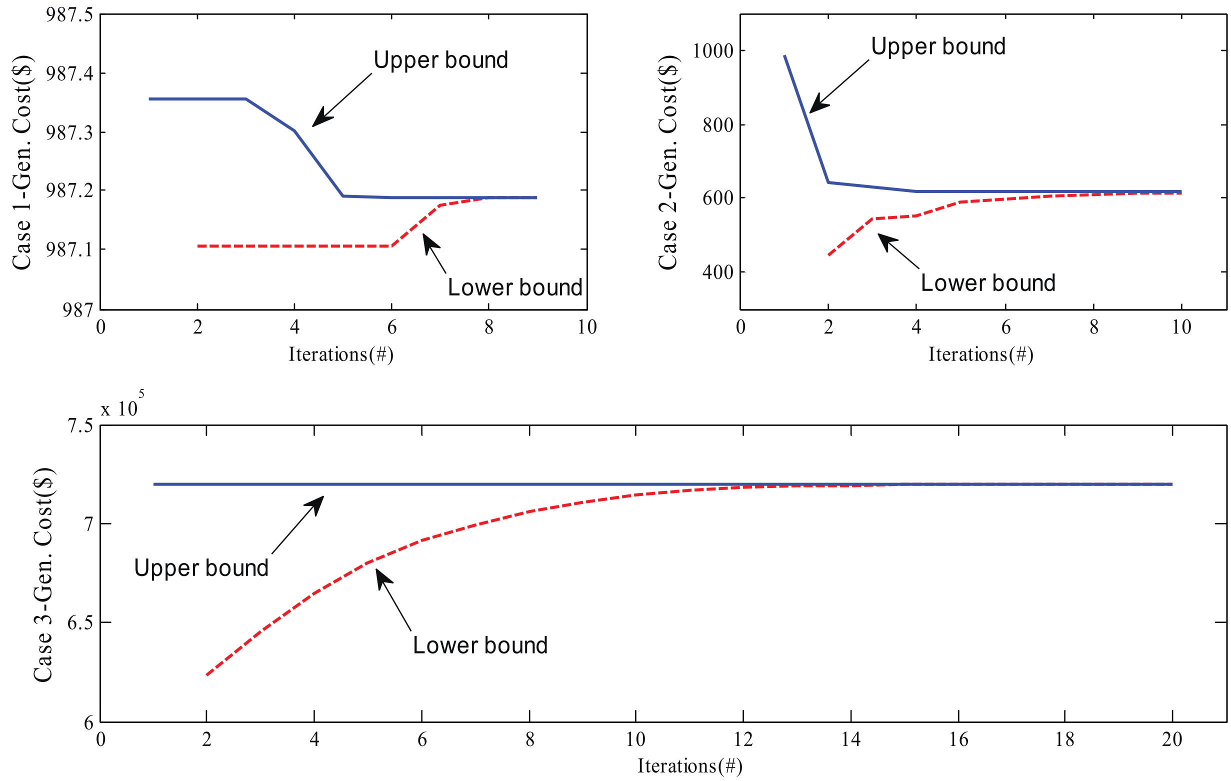

Fig. 7. Convergence of upper bound and lower bound for Case 1 (up-left), Case 2 (up-right), and Case 3 (down).

Modified Subgradient and LUBS algorithms for community micro-grid 2 are shown in Fig 9.

In the left side plots, the utility's import level and community's export level of active and reactive powers reach together in 57 iterations. This convergence is reached in response to the power price levels which are updated by Price Update Center using (10) and (11). The right-hand side graphs illustrated that the convergence of LUBS algorithm for Case 3 with $\sigma = 0.3$ only takes 20

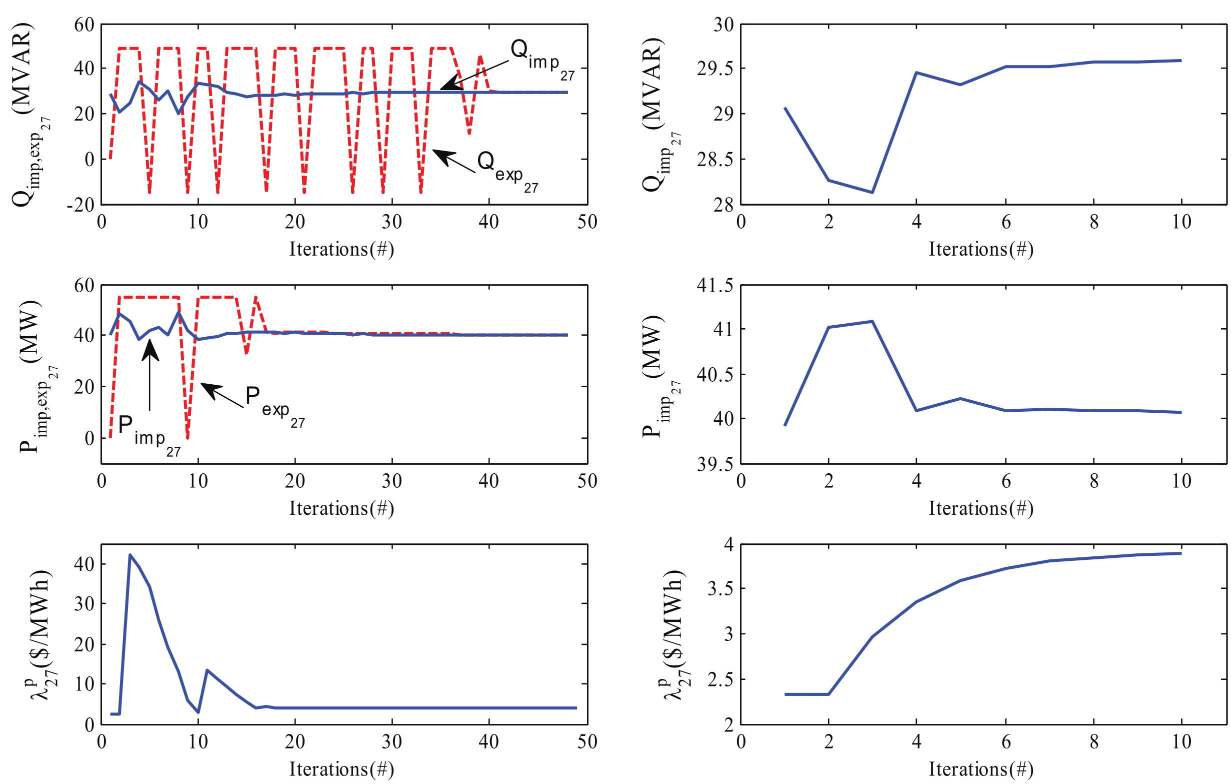

Fig. 8. Case 2 results: reactive power, active power, and active power price of community $\mu G3$. Subgradient method (left) and LUBS algorithm (right).

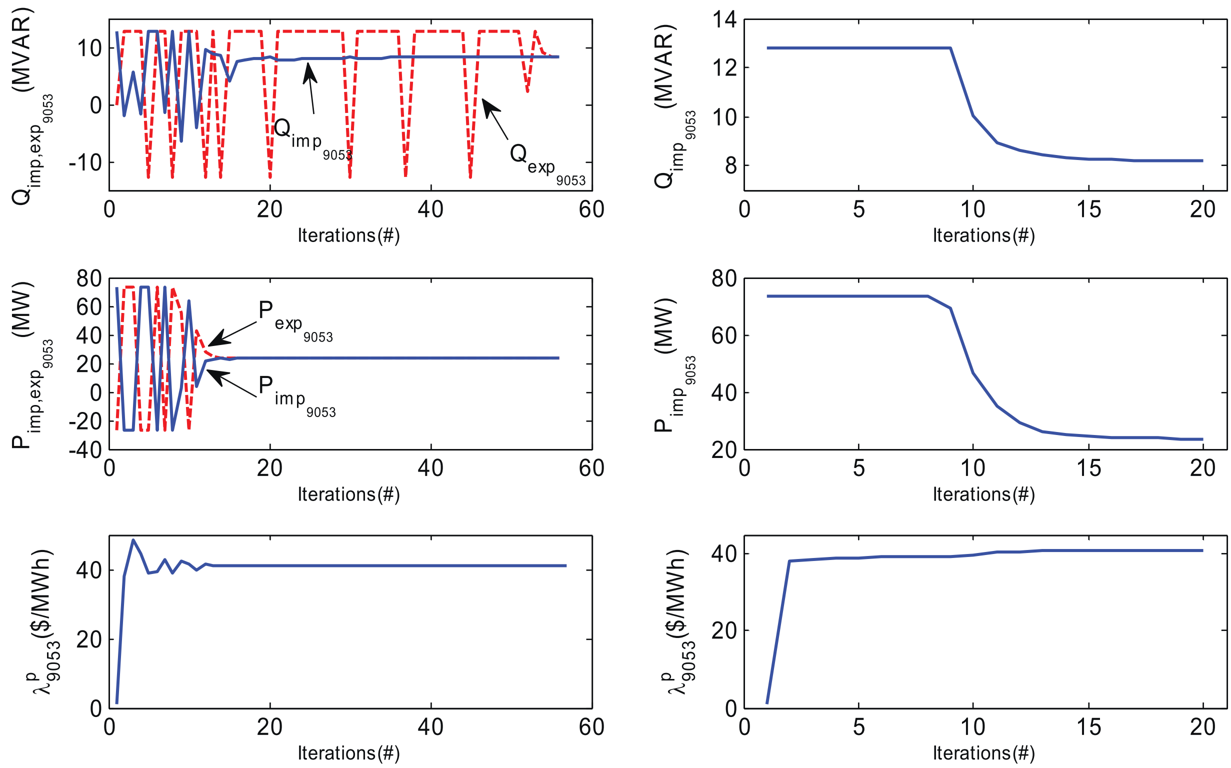

Fig. 9. Case 3 results: reactive power, active power, and active power price of community $\mu G2$. Subgradient method (left) and LUBS algorithm (right).

iterations, and active and reactive power import levels and active power price are settled on their final values finally. The second-row graph of Fig. 7 shows upper bound and lower bound solutions and demonstrates that the optimality gap of the solution found is remarkably low.

The case studies demonstrate the effectiveness of the proposed multi-agent schemes. The main points are as follows.

1. The proposed multi-agent control can enable the utility and a community to autonomously make decision with limited information change.
2. The utility and the community will go through an iterative procedures to reach agreement. The agreement shown in the subgradient method is demonstrated by the convergence of the imported power to the utility and the exported power from the community. The agreement shown in the lower-upper bound switching method is demonstrated by the convergence of the imported power and prices.

5. Conclusion

This paper presents multi-agent control based demand response programs. Based on two different information exchange structures, two distributed algorithms are proposed to facilitate communities and utility interactions to achieve a global cost minimization objective. Each community and the utility behave as an autonomous agent to minimize their own cost. The two algorithms are developed based on Lagrangian relaxation and duality. Our contribution is three-fold: (1) Unlike many demand response papers in the literature where the network model is simplified, this paper considered ac network and formulate ac OPF problems with network constraints considered. (2) Robust distributed algorithms are proposed and tested for multi-agent control of community and utility. (3) Mathematics are presented along with physical meaning of demand response to shed insights on distributed algorithms and optimization decomposition.

Acknowledgement

The authors would like to acknowledge Dr. Bo Zeng for his beneficial discussion on Lagrangian relaxation.

References

[1] L. Chen, N. Li, S.H. Low, J.C. Doyle, Two market models for demand response in power networks, in: First IEEE International Conference on Smart Grid Communications (SmartGridComm), 2010, pp. 397–402.
[2] N. Li, L. Chen, S.H. Low, Optimal demand response based on utility maximization in power networks, in: IEEE Power and Energy Society General Meeting, 2011, pp. 1–8.
[3] A. Mohsenian-Rad, V.W. Wong, J. Jatskevich, R. Schober, A. Leon-Garcia, Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid, IEEE Transactions on Smart Grid 1 (3) (2010) 320–331.
[4] W. Saad, Z. Han, H. Poor, T. Basar, Game-theoretic methods for the smart grid: an overview of microgrid systems, demand-side management, and smart grid communications, IEEE Signal Processing Magazine 29 (5) (2012) 86–105.
[5] D.P. Bertsekas, J.N. Tsitsiklis, Parallel and Distributed Computation, Prentice Hall Inc., Old Tappan, NJ, USA, 1989.
[6] D.P. Palomar, M. Chiang, A tutorial on decomposition methods for network utility maximization, IEEE Journal on Selected Areas in Communications 24 (8) (2006) 1439–1451.
[7] R. Hermans, M. Almassalkhi, I. Hiskens, Incentive-based coordinated charging control of plug-in electric vehicles at the distribution-transformer level, in: IEEE on American Control Conference (ACC), 2012, pp. 264–269.
[8] A.G. Beccuti, T.H. Demiray, G. Andersson, M. Morari, A Lagrangian decomposition algorithm for optimal emergency voltage control, IEEE Transactions on Power Systems 25 (4) (2010) 1769–1779.
[9] P.B. Luh, D.J. Hoitomt, Scheduling of manufacturing systems using the Lagrangian relaxation technique, IEEE Transactions on Automatic Control 38 (7) (1993) 1066–1079.
[10] N. Li, J.R. Marden, Designing games for distributed optimization, in: 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), 2011, pp. 2434–2440.

[11] R.D. Zimmerman, C.E. Murillo-Sánchez, R.J. Thomas, Matpower: steady-state operations, planning, and analysis tools for power systems research and education, IEEE Transactions on Power Systems 26 (1) (2011) 12–19.
[12] G.L. Nemhauser, L.A. Wolsey, Integer and Combinatorial Optimization, vol. 18, Wiley, New York, 1988.
[13] IEEE Recommended Practice for Industrial and Commercial Power System Analysis, IEEE Std 399–1990, 1990, pp. 1–384, http://dx.doi.org/10.1109/IEEESTD.1990.115569.
[14] F. Li, R. Bo, Small test systems for power system economic studies, in: IEEE Power and Energy Society General Meeting, 2010, pp. 1–4, http://dx.doi.org/10.1109/PES.2010.5589973.