Solution to DC Optimal Power Flow Problems with Demand Response via Distributed Asynchronous Primal-Dual Algorithms

Eiji Masuda *, Yusuke Matsuda *, Yuji Wakasa *, and Ryosuke Adachi *

Abstract: As distributed energy resources increase, distributed optimization for power system operation and demand response (DR) have been intensively investigated in recent years. This paper presents a distributed solution for a DC optimal power flow problem with DR, which can cope with communication delays and update delays of some subsystems by applying a distributed asynchronous primal-dual (DAPD) algorithm. Furthermore, we propose an accelerated DAPD algorithm that accelerates the DAPD algorithm based on the work of Nesterov. The effectiveness of the proposed algorithms is illustrated through simulation results.

Key Words: distributed optimization, distributed asynchronous algorithm, DC optimal power flow, demand response.

1. Introduction

Optimal power flow (OPF) problems are optimization problems that minimize the total power generation cost while satisfying physical, operational, and technical constraints. The OPF problems are used to determine a daily optimal power operation and develop an expansion plan of a power system. A DC-OPF problem is an approximation of an AC-OPF problem for obtaining the optimal real power dispatch solution of the entire power system. Although various OPF problem settings (e.g., a robust AC-OPF problem [1] and a branch-flow OPF problem with step-voltage regulators [2]) have been devised, this paper focuses on a DC-OPF problem with demand response (DR) [3].

DR is to adjust demand by providing economic benefits according to the supply and demand situation. DR plays an important role in smart grid because it is expected to conserve energy by visualizing the amount of electricity usage and level the load by changing the electricity rate according to the time zone [4]. Moreover, it is important to create the demand by DR because surplus power is expected to occur frequently due to an increase in distributed energy resources (e.g., photovoltaic and wind power generation).

In recent years, research to optimize power systems in a distributed manner has received significant attention from the viewpoint of complicated situations of power systems. Distributed optimization algorithms are roughly divided into synchronous algorithms and asynchronous algorithms. In distributed synchronous algorithms, computation at each subsystem is performed after receiving information from adjacent subsystems, and therefore, communication delays and update delays of some subsystems may lead to degradation of convergence speed of the whole system. By contrast, the distributed asynchronous algorithms can solve distributed optimization problems under various asynchronous situations, and need not wait for updated results of adjacent subsystems. Therefore, distributed asynchronous algorithms overcome the disadvantage of distributed synchronous algorithms.

A distributed algorithm based on an alternating direction method of multipliers (ADMM) has been applied to a DC-OPF problem with DR in [3]. To accelerate the distributed ADMM, an accelerated ADMM has been presented in [3] by aggregating computation results at all subsystems into one central computer and computing an accelerated factor. Moreover, a fully distributed accelerated ADMM has been proposed in [5],[6] to avoid the requirement of the centralized calculation in the accelerated ADMM. However, since these algorithms are distributed synchronous algorithms, they cannot cope with communication and update delays.

On the other hand, a distributed asynchronous primal-dual (DAPD) algorithm has been presented in [7]. The DAPD algorithm is based on the so-called ADMM+ [7], which is a general type of the ADMMs and has good features such as no centralized computation and relatively fast convergence.

In this paper, by applying the DAPD algorithm to the DC-OPF problem with DR, we present a distributed solution which can cope with communication and update delays. Moreover, we propose an accelerated DAPD algorithm, which accelerates the DAPD algorithm based on the work of Nesterov [8]. As an earlier version [9] of this paper, we have presented this approach and validated its effectiveness for a DC-OPF problem without DR. This paper extends the results into the DC-OPF problem with DR. We illustrate the effectiveness of the proposed method through simulation results.

2. DC-OPF Problem with Demand Response

2.1 Problem Formulation

The DC-OPF problem with DR which minimizes the total power generation cost of generators and maximizes the utilization while satisfying various constraints is formulated as follows [3],[10]:

\[
\begin{align*}
\max_{P, \theta, d} \quad & \sum_{r \in F} \sum_{e \in I_G} u_r(d_e) - \sum_{r \in F} \sum_{e \in I_G} c_r(P_{r,e}) \\
\text{s.t.} \quad & \theta_{ref,r} = 0 \text{ for } r \in T,
\end{align*}
\]

(1)
\[ \sum_{j \in J} \theta_{ij} - \theta_{ij} \frac{X_{ij}}{X_{ij}} = P_{ij} - D_{ij} \forall i \in I \forall t \in T, \quad (3) \]

\[ P_{ij} = \begin{cases} P_{ij}, & \forall i \in I_G \forall t \in T, \\ 0, & \forall i \in I \setminus I_G \forall t \in T, \end{cases} \quad (4) \]

\[ D_{ij} = \begin{cases} d_{ij}, & \forall i \in I_D \forall t \in T, \\ 0, & \forall i \in I \setminus I_D \forall t \in T, \end{cases} \quad (5) \]

\[ \frac{\theta_{ij} - \theta_{ij}}{X_{ij}} \leq F_{ij} \forall i \in I \forall j \in I, \quad (6) \]

\[ P_{ij}^{\min} \leq P_{ij} \leq P_{ij}^{\max}, \quad (7) \]

\[ P_{ij} - P_{ij-1} \leq P_{ij}^{\text{up}}, \quad (8) \]

\[ P_{ij-1} - P_{ij} \leq P_{ij}^{\text{down}}, \quad (9) \]

\[ d_{ij}^{\min} \leq d_{ij} \leq d_{ij}^{\max}, \quad (10) \]

\[ \sum_{t \in T} d_{ij} = d_{ij}^{\text{total}} \forall i \in I_D, \quad (11) \]

\[ c_i(P_{ij}) = a_i^0 + a_i^1 P_{ij}^2 + a_i^2 P_{ij}^3 \forall i \in I_G \forall t \in T, \quad (12) \]

\[ u_i(d_{ij}) = \sum_{k \in K} p^i_{kj} h^i_{kj} \forall i \in I_D \forall t \in T, \quad (13) \]

\[ d_{ij} = \sum_{k \in K} h^i_{kj} \forall i \in I_D \forall t \in T, \quad (14) \]

\[ 0 \leq h^i_{kj} \leq h^i_{kj} \forall k \in K \forall i \in I_D \forall t \in T, \quad (15) \]

Table 1 shows a notation of the DC-OPF problem with DR. Note that a generator and a DR load connected to bus \( i \) are called generator \( i \) and DR load \( i \). The objective (1) is to maximize the social welfare (i.e., maximize the utility function \( u_t \) and minimize the generation cost function \( \kappa(t) \)). Equation (2) determines a reference bus. Equation (3) expresses that an active power flow into bus \( i \) equals a difference between the active power injection and withdrawal at bus \( i \), where \( (\theta_{ij} - \theta_{ij})/X_{ij} \) represents an active power flow from bus \( i \) to bus \( j \) in time period \( t \). Equations (4) and (5) mean whether or not bus \( i \) is equipped with a generator and a DR load. The right complement of \( I_G \) with respect to set \( I \) is denoted by \( I \setminus I_G \), which indicates the set of elements in \( I \) but not in \( I_G \). Equations (6) and (7) are capacities of transmission lines and generators. Equations (8) and (9) enforce ramp up and down limits of generators. Equation (10) ensures lower and upper limits of DR loads. Equation (11) guarantees that the sum of active power loads in each time period equals the total load required. The quadratic function (12) is used to calculate the fuel cost of generator \( i \). The utility function \( u_t \) in (13) is given by the price-elastic demand curve, which is a step-wise function as shown in Fig. 1. The price \( P_{ij}^{\text{up}} \) is strictly decreasing with \( \kappa \), and therefore, DR load is preferentially consumed when \( P_{ij}^{\text{up}} \) is large. Since the step-wise function is calculated by (14) and (15), we have (16) when \( \sum_{k=1}^{a_i} p^i_{kj} < d_{ij} \leq \sum_{k=1}^{a_i+1} p^i_{kj} \) holds for a certain step \( s_0 \).

\[ h^i_{kj} = \begin{cases} p^i_{kj} & \text{if } s < s_0, \\ d_{ij} - \sum_{k=1}^{s-1} p^i_{kj} & \text{if } s = s_0, \\ 0 & \text{if } s > s_0. \end{cases} \quad (16) \]

2.2 Distributed DC-OPF Problem with DR

In this research, we consider a power system divided into several subsystems as shown in Fig. 2 and aim to solve the DC-OPF problem with DR in a distributed asynchronous manner.
Therefore, the coupling variables in subsystem 1 is denoted by subsystem 3 are duplicated as local variables in subsystem 1. In a 3-bus system shown in Fig. 3,pling variables include voltage phase angles of boundary buses systems. Then the constraint (17) is necessary to guarantee that the adjacent subsystems. To bring (3) and (6) into a suitable form for distributed optimization, voltage phase angles of boundary buses in each subsystem are duplicated in its adjacent subsystems. Then the constraint (17) is necessary to guarantee that the original voltage phase angles are equal to the duplicated voltage phase angles:

\[
\hat{x}_n = T_{nn}\tilde{x}_n \quad \forall n \in \mathcal{N}, \forall m \in \mathcal{N}_n,
\]  

(17)

where \( \mathcal{N} \) and \( \mathcal{N}_n \) are sets of all subsystems and subsystems adjacent to subsystem \( n \), respectively. In (17), \( \hat{x}_n \) called coupling variables include voltage phase angles of boundary buses in subsystem \( n \) and duplicated voltage phase angles of boundary buses in adjacent subsystems in time period \( t \). For instance, in a 3-bus system shown in Fig. 3, \( \hat{\theta}_b \) in subsystem 2 and \( \hat{\theta}_b \) in subsystem 3 are duplicated as local variables in subsystem 1. Therefore, the coupling variables in subsystem 1 is denoted by \( \hat{x}_1 = (\hat{\theta}_b, \theta_1^a, \theta_1^b)^T \), where \( \theta_1^a \) is the duplicated voltage phase angle of bus \( i \) in subsystem \( n \). Likewise, the coupling variables in subsystems 2 and 3 are \( \hat{x}_2 = (\theta_2^a, \theta_2^b)^T \) and \( \hat{x}_3 = (\theta_3^a, \theta_3^b)^T \). Since the equality constraints with the adjacent subsystem that subsystem 2 satisfies are only \( \theta_2^a = \theta_1 \) and \( \theta_2^b = \theta_1^b \), they can be written as \( \hat{x}_2 = T_{21}\hat{x}_1 \), where

\[
T_{21} = \begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0
\end{pmatrix}.
\]

As described above, the DC-OPF problem with DR (1)–(15) can be reformulated in a separable form that is suitable for the distributed optimization. For the sake of discussion, the variables of subsystem \( n \) are expressed as \( \hat{x}_n = (\hat{x}_n^a, \hat{x}_n^b)^T \), where \( \hat{x}_n \) called local variables include active power outputs, voltage phase angles except boundary buses, and active power loads of DR loads in subsystem \( n \).

For the distributed DC-OPF problem with DR, the objective of the optimization problem (1)–(15) can be represented as the summation of the objectives for individual subsystems:

\[
\min_{\mathbf{x}} \sum_{n \in \mathcal{N}} W_n(x_n) + \iota(\mathbf{x})
\]

s.t. \( x_n \in \mathcal{X}_n \quad \forall n \in \mathcal{N}, \)

(18)

where \( \mathbf{x} = (x_n)_{n \in \mathcal{N}} \) is a vector in which \( x_n \) for all \( n \in \mathcal{N} \) are arranged, and \( \iota(\mathbf{x}) \). In (18), \( W_n(x_n) = -\sum_{s \in \mathcal{D}_n} \sum_{t \in \mathcal{D}_n} u_t(d_{st}) + \sum_{s \in \mathcal{G}_n} c_s(P_s) \) represents the objective function of subsystem \( n \), where \( \mathcal{D}_n \) and \( \mathcal{G}_n \) are sets of buses connected to DR loads and generators in subsystem \( n \), respectively. The vector \( x_n \) satisfying the constraints (2)–(15) in subsystem \( n \) is represented as \( x_n \in \mathcal{X}_n \) in (19). An indicator function \( \iota(\mathbf{x}) \) for constraint (17) is expressed as

\[
\iota(\mathbf{x}) = \begin{cases}
0 & \text{if } \hat{x}_n = T_{nn}\tilde{x}_n \quad \forall n \in \mathcal{N} \quad \forall m \in \mathcal{N}_n, \\
\infty & \text{otherwise}.
\end{cases}
\]

(20)

3. DAPD Algorithm

In this section, we first explain the standard DAPD algorithm [7]. We next reformulate the DAPD algorithm for groups of agents. Then, we present the DAPD algorithm applicable to the DC-OPF problem with DR. We lastly propose an accelerated DAPD algorithm by adding simple calculation steps.

3.1 Standard DAPD Algorithm

We consider an undirected graph \( \mathcal{G} = (\hat{\mathcal{I}}, \hat{\mathcal{E}}) \), where \( \hat{\mathcal{I}} = \{1, \ldots, \hat{I}\} \) and \( \hat{\mathcal{E}} \) are sets of agents and edges, respectively. Note that \( \mathcal{G} \) is connected and has no self-loops. The standard DAPD algorithm in [7] can solve the following distributed convex optimization problem on \( \mathcal{G} \) asynchronously:

\[
\min_{\mathbf{\xi} \in \mathbb{R}^I} \sum_{i \in \hat{\mathcal{I}}} (f_i(\xi_i) + g_i(\xi_i)) + \iota(\mathbf{\xi}),
\]

(21)

where \( \mathbb{R}^I \) is an \( \hat{I} \)-dimensional Euclidean space, \( \mathbf{\xi} = (\xi_i)_{i \in \hat{\mathcal{I}}} \) is a vector of primal variables, and \( f_i \) and \( g_i \) are two private functions available at agent \( i \). An indicator function \( \iota(\mathbf{\xi}) \) is expressed as

\[
\iota(\mathbf{\xi}) = \begin{cases}
0 & \text{if } \xi_i = \xi_j \quad \forall i \in \hat{\mathcal{I}} \quad \forall j \in \hat{\mathcal{I}}, \\
\infty & \text{otherwise},
\end{cases}
\]

(22)

where \( \hat{\mathcal{I}}_j \) is the set of agents adjacent to agent \( i \). We make here the following assumption for the problem (21):

Assumption 1 For each \( i \in \hat{\mathcal{I}}, \)

(i) \( f_i \) is a convex differentiable function on \( \mathbb{R}^I \).

(ii) \( \nabla f_i \) is \( L \)-Lipschitz continuous on \( \mathbb{R}^I \).

(iii) \( g_i \) is a proper lower semi-continuous convex function on \( \mathbb{R}^I \rightarrow \mathbb{R} \).

(iv) \( \cap_{i \in \hat{\mathcal{I}}} \text{ri dom } g_i \neq \emptyset \).

Assumption 1–(ii) means that a constant \( L \geq 0 \) satisfying the following inequality exists:

\[
\|\nabla f_i(a) - \nabla f_i(b)\| \leq L \|a - b\| \quad \forall a, b \in \mathbb{R}^I.
\]

(23)
Algorithm 1 Standard DAPD algorithm [7]

1: Initialize:
   $\xi^0_i, \lambda^0_i, \forall i \in \mathcal{I}$
2: for all agent $i \in \mathcal{I}_{act}$ do
3: Update $\xi^{k+1}_i$ and $\lambda^{k+1}_i$ via (24) and (25).
4: Send $\xi^{k+1}_i$ and $\lambda^{k+1}_i$ to $j \in \mathcal{I}_i$.
5: end for
6: for all agent $i \in \mathcal{I}_{act}$ do
7: $\xi^{k+1}_i = \xi^k_i, \lambda^{k+1}_i = \lambda^k_i$.
8: end for
9: Increment $k$.

We denote by dom $g_i$ the domain of a function $g_i$ and by $r_i C$ the relative interior of a set $C \subseteq \mathbb{R}^l$, respectively.

The $k$-th iteration of the DAPD algorithm is represented as follows:

$$\xi^{k+1}_i = \arg \min_{\xi_i} \left[ g_i(\xi_i) + \frac{\Delta_i}{2\tau} \left( (1 - \frac{\tau}{\rho}) \xi^k_i + \frac{\tau}{\rho} \xi^k_i \right)^2 \right],$$

(24)

$$\lambda^{k+1}_{i,j} = \frac{\lambda^k_{i,j}}{2} + \frac{\lambda^k_{i,j} - \xi^k_j}{2\rho},$$

(25)

where $\lambda^k_{i,j}$ and $\Delta_i$ are the dual variable at agent $i$ for agent $j$ and the number of neighbors of agent $i$, respectively. Since each term on the right-hand side in (24) is a convex function, it can be expressed as follows:

$$\xi^{k+1}_i = \arg \min_{\xi_i} \left[ G_i(\xi_i) + \frac{1}{\Delta_i} \sum_{j \in \mathcal{I}_i} \left( (1 - \frac{\tau}{\rho}) \xi^k_i + \frac{\tau}{\rho} \xi^k_i \right)^2 \right].$$

(28)

Remark 1 The DAPD algorithm is based on the ADMM+ [7]. The ADMM+ includes the ADMM and the forward-backward algorithm as its special cases. Let $\tau = \rho$ in the ADMM+, then the method is equivalent to the ADMM, and $1/\rho$ corresponds to the penalty parameter. The relationship between the equality constraint (22) and the second term in (25) also shows that $1/\rho$ has the same meaning in the DAPD algorithm. On the other hand, $\tau$ is a stepsize in the forward-backward algorithm, and the same applies to the DAPD algorithm.

The procedure of the standard DAPD algorithm is summarized in Algorithm 1. The set $\mathcal{I}_{act}$ is a set of activated agents that update the variables at iteration $k$. Let $(\hat{I}_{act})_{k=0}^\infty$ be a sequence of random variables, and we make the following assumption.

Assumption 2 The probability that each agent becomes active at iteration $k = 1$ is positive, and the sequence $(\hat{I}_{act})_{k=0}^\infty$ is independent and identically distributed.

Note that, under Assumption 2, each agent updates the variables with a positive probability at each iteration. Let Assumptions 1 and 2 hold, and let the setting parameters satisfy

$$\frac{1}{\tau} - \frac{1}{\rho} > \frac{L}{2\Delta_{\min}},$$

(26)

where $\Delta_{\min} = \min_{i \in \mathcal{I}} \Delta_i$. Then the convergence of the DAPD algorithm is guaranteed [7].

3.2 Reformulation of the DAPD Algorithm for Groups of Agents

We rewrite the problem (21) as an optimization problem for groups of agents as follows:

$$\min_{\xi \in \mathbb{R}^l} \sum_{i \in \mathcal{I}} (F_i(\xi_i) + G_i(\xi_i)) + \bar{i}(\xi),$$

(27)

where $F_i(\xi_i) = \sum_{j \in \mathcal{J}_i} f_j(\xi_j)$ and $G_i(\xi_i) = \sum_{j \in \mathcal{J}_i} g_j(\xi_j)$. Note that $\mathcal{N} \cup \mathcal{I}_{act} \cup \mathcal{I}_{act} \cap \mathcal{N}$ is a set of sets of groups and agents in group $\mathcal{N}$, respectively. Since each term on the right-hand side in (24) is a convex function, it can be expressed as follows:

$$\xi^{k+1}_i = \arg \min_{\xi_i} \left[ G_i(\xi_i) + \frac{1}{\Delta_i} \sum_{j \in \mathcal{I}_i} \left( (1 - \frac{\tau}{\rho}) \xi^k_i + \frac{\tau}{\rho} \xi^k_i \right)^2 \right].$$

(28)

We suppose that each group updates the variables of all its own agents (i.e., the agents in each group are synchronized). Then the convergence of the DAPD algorithm for groups of agents is guaranteed because Assumption 2 still holds by replacing $\hat{I}_{act}$ with $\hat{I}_{act}^k$, where $\hat{I}_{act}^k$ is a set of groups that update the variables at iteration $k$.

3.3 DAPD Algorithm for DC-OPF Problems

We discuss the DAPD algorithm for DC-OPF problem with DR (18) and (19). We can make the distributed optimization problem (27) correspond to the DC-OPF problem (18) and (19). When $F_i(\xi_i) = 0$ and $G_i(\xi_i) = W_i(x_i)$, Assumption 1–(i) and (ii) are clearly satisfied. In Assumption 1–(iii) and (iv), $g_i$ corresponds to $c_i$ in the DC-OPF problem with DR. Since the cost function $c_i$ is a continuous quadratic function in $\mathbb{R}$, Assumption 1–(iii) and (iv) are justly satisfied. Let the agent and the group in the previous subsection correspond to the boundary bus and the set of boundary buses in each subsystem in the DC-OPF problem, respectively. Then the update formulas (25) and (28) are represented as follows:

$$x^{k+1}_n = \arg \min_{x_n \in \mathcal{X}} \left[ W_n(x_n) + \frac{1}{\Delta_{\min}} \sum_{i \in \mathcal{I}_n} \left( (1 - \frac{\tau}{\rho}) \theta^k_i + \frac{\tau}{\rho} \theta^k_i \right)^2 \right],$$

(29)

$$x^{k+1}_{r,n} = \frac{x^{k+1}_{r,n}}{2} + \frac{x^{k+1}_{r,n} - T_{min,k,n}}{2\rho},$$

(30)

where $\mathcal{I}_n$ and $I_n$ are sets of boundary buses in subsystem $n$ and boundary buses adjacent to bus $i$ in the neighbors of the subsystem containing the bus, respectively. In (29), $\Delta_n$ is the number of elements of $I_n$. Although (28) is unconstrained, we consider the constraint (19) as $x_n \in \mathcal{X}_n$ in (29). In the second term on the right-hand side in (29) and (30), we only need to deal with boundary buses in order to achieve the consensus (17).

The procedure of the DAPD algorithm for the DC-OPF problem is summarized in Algorithm 2. The vector $r^{k+1}_n$ used for a stopping criterion is represented as follows:

$$r^{k+1}_n = \lambda^{k+1}_n - \lambda^k_n,$$

(31)
by the proposed algorithms. When the delays occur, i.e., when some subsystem cannot obtain information from its neighbors, it updates the variables with the previously obtained information. Each subsystem $n \in N^{act}_{k+1}$ updates the variables (Algorithm 2, lines 3–7), but each subsystem $n \notin N^{act}_{k+1}$ does not update the variables and holds the previous values (Algorithm 2, lines 8–10). Thus, it is possible to simulate the situation where some subsystem that cannot obtain information from its neighbors due to the delays updates the variables using the previously obtained information. In other words, it is possible to simulate the asynchronous situations where the random delays occur. Since the convergence of the DAPD algorithm is guaranteed under the assumptions, the presented algorithm can be interpreted as a solution to solve the DC-OPF problems with the delays.

On the other hand, since $L = 0$ holds from $F_0(\xi_n) = 0$, the condition (26) is rewritten as follows:

$$\tau > \rho.$$  (32)

### 3.4 Accelerated DAPD Algorithm

An accelerated DAPD algorithm is based on the work of Nesterov [8], which is a general algorithm for accelerating the gradient descent algorithm when solving convex optimization problems. For the acceleration, we add the following steps to the DAPD algorithm:

$$\beta^{k+1}_n = \frac{1 + \sqrt{1 + 4(\beta^k_n)^2}}{2},$$

$$\alpha^{k+1}_n = \begin{cases} 1 + \frac{\beta^k_n - 1}{\beta^{k+1}_n} & \text{if } \|x^{k+1}_n\| \leq 1, \\ 1 & \text{otherwise,} \end{cases}$$

$$\tilde{x}^{k+1}_n = \alpha^{k+1}_n \tilde{x}^{k}_n + (1 - \alpha^{k+1}_n) x^{k+1}_n,$$

$$x^{k+1}_n = \alpha^{k+1}_n x^{k+1}_n + (1 - \alpha^{k+1}_n) \tilde{x}^{k+1}_n,$$

where $\beta^0_n = 1$ for all $n \in N$. By adding (33)–(36) to the DAPD algorithm, it can be accelerated numerically as shown in Section 4 though there is no theoretical guarantee of convergence. It should be noted that the accelerated ADMM proposed in [3],[11] requires the centralized calculation for the acceleration. However, the accelerated DAPD algorithm requires no centralized calculation because each subsystem updates the parameters $\alpha_n$ and $\beta_n$. The procedure of the accelerated DAPD algorithm is summarized in Algorithm 3.

### 4. Simulation

In this section, the DAPD algorithm and the accelerated DAPD algorithm for the DC-OPF problems are tested on a 6-bus system and the IEEE 118-bus system. For simulations, MATLAB (R2017b) and CVX (Version 2.1) [12] were used on Core i7-7700K 4.2 GHz CPU with 32 GB RAM. The CVX was used to update the primal variables in the proposed algorithm. In simulations, we set the reference bus to bus 1 in (2) and define the set of subsystems $\mathcal{K} = \{1,2,\ldots,5\}$, $p^p_{ij} = 10/k$, and $P^d_{it} = d^p_{it} / 5$, for each $i \in I_0$ and $t \in T$. In addition, each subsystem updates the variables with the probability of 1/2 in order to simulate asynchronous situations.

#### 4.1 6-Bus System

First, we consider the DC-OPF problem with DR for the 6-bus system with 2 generators and 3 DR loads as shown in Fig. 4 [3]. The 6-bus system is composed of 2 subsystems denoted by $S_1 = \{1,2,6\}$ and $S_2 = \{3,4,5\}$ and tested for a 3-hour period (i.e., $T = \{1,2,3\}$). In this case, $T_{in}$ in (17) are expressed as $T_{12} = T_{21} = E_{ij}$ corresponding to $\tilde{x}_1 = (\theta_1 \theta_2 \theta_5 \theta_6)^T$ and $\tilde{x}_2 = (\theta_3 \theta_4 \theta_5 \theta_6)^T$, where $E_{ij}$ is the $i$-dimensional identity matrix. Parameters of the generator, the transmission line, and the load are given in Tables 2–5. In Tables 3 and 5, the parameters $P^p_{max}$, $P^p_{min}$, $P^d_{max}$, $P^d_{min}$, $P^d_{down}$, and $d^p_{max}$ are constant for 3 hours. The value of the parameters $\tau$ and
are set as 0.05 and 2\(\tau\), respectively, so that the convergence condition (32) is satisfied. Let the threshold \(\epsilon\) for the stopping criterion be 10\(^{-4}\).

Figure 5 shows the relationship between the number of iterations and the objective function value in the case of the 6-bus system. The value of the parameters \(\tau\) and \(\rho\) are set as 0.05 and 2\(\tau\), respectively, so that the convergence condition (32) is satisfied. Let the threshold \(\epsilon\) for the stopping criterion be 10\(^{-4}\).

Table 3 Generator constraints.

| Unit | \(P_{\text{min}}^{\text{up}}\) (MW) | \(P_{\text{min}}^{\text{down}}\) (MW) | \(P_{\text{max}}\) (MW) | \(P_{\text{min}}^{\text{up}}\) (MW) |
|------|---------------------------------|---------------------------------|-----------------|---------------------------------|
| G1   | 20                              | 200                             | 200             | 200                             |
| G2   | 50                              | 200                             | 200             | 200                             |

Table 4 Branch information.

| Line number | From | To | \(X_j\) (p.u.) | \(F_{ij}^{\text{max}}\) (MW) |
|-------------|------|----|----------------|-------------------------------|
| 1           | 1    | 2  | 0.6            | 150                           |
| 2           | 1    | 3  | 0.6            | 150                           |
| 3           | 1    | 6  | 0.1            | 150                           |
| 4           | 2    | 4  | 0.1            | 150                           |
| 5           | 3    | 5  | 0.1            | 150                           |
| 6           | 4    | 5  | 0.1            | 150                           |

Table 5 Load information.

| Unit | \(d_{\text{load}}\) (MW) | \(d_{\text{load}}^{\text{up}}\) (MW) | \(d_{\text{load}}^{\text{down}}\) (MW) | \(F_{ij}^{\text{max}}\) (MW) |
|------|--------------------------|-----------------------------------|---------------------------------|-------------------------------|
| L1   | 150                      | 100                               | 40                              | 60                           |
| L2   | 150                      | 100                               | 40                              | 60                           |
| L3   | 10                       | 10                                | 2                               | 4                            |

Figure 5 Relationship between the number of iterations and the objective function value in the case of the 6-bus system.

\(\rho\) is set as 0.05 and 2\(\tau\), respectively, so that the convergence condition (32) is satisfied. Let the threshold \(\epsilon\) for the stopping criterion be 10\(^{-4}\).

Figure 5 shows the relationship between the number of iterations and the objective function value (i.e., value of \(\sum_{x_i \in N} W_i(x_i)\)), where D-ADMM, DAPD, and A-DAPD mean the distributed ADMM [3], the DAPD algorithm, and the accelerated DAPD algorithm, respectively. For the sake of comparison, the distributed ADMM is also simulated in the asynchronous situations by updating the variables with the probability of 1/2 after communication (see [3, Algorithm 2, Steps 4 and 6]). The minimum value is found by centralized computation using the CVX. We can see from Fig. 5 that the DAPD algorithm and accelerated DAPD algorithm converge and provide the minimum value. However, the conventional distributed ADMM does not provide the convergence to the minimum value even in a small scale power system like the 6-bus system.

Remark 2 The distributed ADMM, as its name suggests, updates the primal variables \(x\) and \(z\) alternatively (see, e.g., [3],[13]). In the asynchronous situations, although the primal variables are not always updated alternatively, the equality constraint (17) might be satisfied. As a result, the stopping criteria might be satisfied, but the cost function \(W_i(x_i)\) is not always the optimal value. Such a situation actually occurs in the simulation.

Table 6 Average numbers of iterations and the computation time of 10 trials in the case of the 6-bus system.

| # of iterations | Time (s) |
|-----------------|----------|
| DAPD            | 226.5    |
| A-DAPD          | 189.6    |

Table 7 Average numbers of iterations and computation time of 5 trials in the case of the IEEE 118-bus system.

| # of iterations | Time (s) |
|-----------------|----------|
| DAPD            | 480.6    |
| A-DAPD          | 401.0    |

Table 6 shows the average numbers of iterations and computation time of 10 trials when the algorithms stop due to the stopping criterion. We can see from Table 6 that the accelerated DAPD algorithm has a smaller number of iterations and less computation time on average than the DAPD algorithm.

4.2 IEEE 118-Bus System

Next, the DAPD algorithm and the accelerated DAPD algorithm are tested for the DC-OPF problem with DR on the IEEE 118-bus system. The IEEE 118-bus system is often used as a test case for OPF problems. Details of parameters of the IEEE 118-bus system are described in [14].

In this simulation, the IEEE 118-bus system is partitioned into 3 subsystems. We define the set of time periods \(\mathcal{T} = \{1, 2, \ldots, 24\}\). The value of the parameters \(\tau\) and \(\rho\) are set as 0.001 and 2\(\tau\), respectively, so that the convergence condition (32) is satisfied. Let the threshold \(\epsilon\) for the stopping criterion be 0.1.

Table 7 shows the average numbers of iterations and computation time of 5 trials when the algorithms stop due to the stopping criterion. As in the previous subsection, we can see from Table 7 that the accelerated DAPD algorithm has a smaller number of iterations and less computation time on average than the DAPD algorithm.

Figure 6 shows 24-hour relative errors (REs) for illustrating solution accuracy. The RE in time period \(t\) is expressed as follows:

\[
\text{RE}_t = \left\| \left\{ \sum_{i \in I_G} P_{li} - \sum_{i \in I_G} P_{li}^* \right\} \right\|_2^2 / \left\| \sum_{i \in I_G} P_{li}^* \right\|_2^2 ,
\]

where \(P_{li}\) and \(P_{li}^*\) are obtained power output solutions of generator \(i\) in time period \(t\) from the proposed algorithms and the
centralized computation using the CVX, respectively. It is apparent from Fig. 6 that the proposed algorithms achieve high accuracy in each time period. We can confirm from these results that the stopping criteria of the proposed algorithms related to the numbers of iterations and computation time and the obtained solutions are appropriate.

From the above results, we confirm that the DAPD algorithm can solve the DC-OPF problem with DR asynchronously. In addition, the accelerated DAPD algorithm can solve the DC-OPF problem with DR faster than the DAPD algorithm without requiring any centralized computation.

5. Conclusion

In this paper, we have applied the DAPD algorithm to the DC-OPF problem with DR and proposed an accelerated DAPD algorithm based on the work of Nesterov [8]. In the conventional distributed ADMM, there is a disadvantage that communication and update delays cannot be coped with. On the other hand, the DAPD algorithm can overcome the disadvantage of the conventional method. Furthermore, the accelerated DAPD algorithm can accelerate the convergence of the DAPD algorithm. We have conducted numerical simulations to illustrate the effectiveness of the proposed algorithms.

References

[1] R. Louca and E. Bitar: Robust AC optimal power flow, IEEE Trans. Power Systems, Vol. 34, No. 3, pp. 1669–1681, 2019.
[2] M. Bazrafshan, N. Gatsis, and H. Zhu: Optimal power flow with step-voltage regulators in multi-phase distribution networks, arXiv: 1901.04566, 2019.
[3] Y. Wang, L. Wu, and S. Wang: A fully-decentralized consensus-based ADMM approach for DC-OPF with demand response, IEEE Trans. Smart Grid, Vol. 8, No. 6, pp. 2637–2647, 2017.
[4] M.H. Albadi and E.F. El-Saadany: A summary of demand response in electric markets, Electric Power Systems Research, Vol. 78, pp. 1989–1996, 2018.
[5] Y. Matsuda and Y. Wakasa: Solution to DC optimal power flow problems via a fully-distributed accelerated ADMM, IEEE Trans. EIS, Vol. 139, No. 6, pp. 725–731, 2019 (in Japanese).
[6] Y. Matsuda, Y. Wakasa, and E. Masuda: Fully-distributed accelerated ADMM for DC optimal power flow problems with demand response, Proc. ASCC 2019, pp. 740–745, 2019.
[7] P. Bianchi, W. Hachem, and F. Iutzeler: A coordinate descent primal-dual algorithm and application to distributed asynchronous optimization, IEEE Trans. Automatic Control, Vol. 61, No. 10, pp. 2947–2957, 2016.
[8] Y. Nesterov: A method of solving a convex programming problem with convergence rate $O(1/k^2)$, Soviet Math. Doklady, Vol. 27, No. 2, pp. 372–376, 1983.
[9] E. Masuda, Y. Matsuda, and Y. Wakasa: Solution to DC optimal power flow problems via distributed asynchronous primal-dual algorithms, Proc. SICE Annual Conference 2019, pp. 1408–1411, 2019.
[10] C. Zhao, J. Wang, J.-P. Watson, and Y. Guan: Multi-stage robust unit commitment considering wind and demand response uncertainties, IEEE Trans. Power Systems, Vol. 28, No. 3, pp. 2708–2717, 2013.
[11] T. Goldstein, B. O’Donoghue, S. Setzter, and R. Baraniuk: Fast alternating direction optimization methods, SIAM J. Imag. Sci., Vol. 7, No. 3, pp. 1588–1623, 2014.
[12] M.C. Grant and S.P. Boyd: The CVX Users’ Guide, Release 2.1, http://cvxr.com/cvx/doc/CVX.pdf, accessed 2019.
[13] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein: Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends Mach. Learn., Vol. 3, No. 1, pp. 1–122, 2011.
[14] L. Wu: https://people.clarkson.edu/~lwu/data/IEEE118_ADMM/, accessed 2019.

Eiji Matsuda (Student Member)

He received his B.S. degree from Yamaguchi University, Japan, in 2019. He is currently an M.S. student at Yamaguchi University. His research interests include distributed optimization in power systems.

Yuusuke Matsuda (Member)

He received his B.S. degree from Yamaguchi University, Japan, in 2018. He is currently an M.S. student at Yamaguchi University. His research interests include distributed optimization in power systems. He is a student member of ISCIE.

Yuji Wakasa (Member)

He received the B.S. and M.S. degrees in engineering and the degree of doctor of informatics all from Kyooto University, Japan, in 1992, 1994, and 2000, respectively. He was a Research Associate at Okayama University from 1994 to 1998 and at Kyoto University from 1998 to 2002. Since October 2002, he has been with Yamaguchi University, where he is currently a Professor. His current research interests include control system design via mathematical programming and its applications. He is a member of IEEJ, ISCIE, and IEE.

Ryosuke Adachi (Member)

He received his B.E., M.S.I., and Ph.D. degrees from Hokkaido University, Japan, in 2014, 2016, and 2019, respectively. Since 2019, he has been an Assistant Professor at the Graduate School of Sciences and Technology for Innovation at Yamaguchi University. His research interests include the estimation of distributed control systems. He is a member of ISCIE and IEE.