A Big Data Scale Algorithm for Optimal Scheduling of Integrated Microgrids

Hung Khanh Nguyen, Amin Khodaei, *Senior Member, IEEE,* and Zhu Han, *Fellow, IEEE*

***Abstract*—The capability of switching into the islanded operation mode of microgrids has been advocated as a viable solution to achieve high system reliability. This paper proposes a new model for the microgrids optimal scheduling and load curtailment problem. The proposed problem determines the optimal schedule for local generators of microgrids to minimize the generation cost of the associated distribution system in the normal operation. Moreover, when microgrids have to switch into the islanded operation mode due to reliability considerations, the optimal generation solution still guarantees for the minimal amount of load curtailment. Due to the large number of constraints in both normal and islanded operations, the formulated problem becomes a large-scale optimization problem and is very challenging to solve using the centralized computational method. Therefore, we propose a decomposition algorithm using the alternating direction method of multipliers (ADMM) that provides a parallel computational framework. The simulation results demonstrate the efficiency of our proposed model in reducing generation cost as well as guaranteeing the reliable operation of microgrids in the islanded mode. We finally describe the detailed implementation of parallel computation for our proposed algorithm to run on a computer cluster using the Hadoop MapReduce software framework.**

***Index Terms*—ADMM, big data, Hadoop, integrated microgrid, islanded operation, load curtailment, MapReduce.**

I. Introduction

Microgrids have been proposed as one of the key components for grid modernization, which operate as single controllable entities to supply a group of interconnected loads [1], [2]. With the capability of integrating renewable energy resources and energy storage devices, microgrids are expected to reduce large capital investment by meeting increased energy demand using locally generated power [3]–[5]. In addition, by deploying power generation close to end users, microgrids are becoming promising solution to improve reliability and power quality [6]–[9].

Despite microgrids has been advocated as a viable technology to ensure grid reliability, several research challenges remain that need to be addressed [10]. Recent research has studied problems of generation schedule for microgrids in the most cost-effective way. In [11], the authors propose a distributed algorithm to obtain the global optimal solution of optimal power flow for microgrids, with the objective of minimizing either the power distribution losses or the cost of power drawn from the substation. The work in [12] provides a distributed approach to solve the economic dispatch problem for microgrids. A distributed energy management strategy for the optimal operation of microgrids is proposed in [13]. However, most of the mentioned works focus on the operation of microgrids without considering reliability requirements in the proposed models.

In addition to the aforementioned characteristics, one of the salient features of microgrids is islanded operation, which is defined as the capability to disconnect from the main distribution network and locally supply their loads [14]. By rapidly disconnecting from the main grid, microgrids can protect their components from upstream disturbances or voltage fluctuations. More importantly, the islanded operation mode allows microgrids to ensure energy supply for critical loads by increasing the generation output of local generators when the main distribution network is faulty. This capability has been advocated as a viable solution to achieve high system resiliency during major outages [15]. However, due to the limitation of ramping ability of local generators in increasing generation output from the operating point in the normal operation before switching into the islanded operation, microgrids may not be able to satisfy all load demand for local customers. In order to overcome supply deficiency, the work in [16] investigates the real-time pricing for a power grid operator to incentivize aggregators to reschedule energy consumption when experiencing contingency. Another approach is to propose a power management algorithm for islanded microgrids using energy storage and demand response program [17]. However, since these works either focus on the operation of microgrids in the islanded mode solely or tackle the problem when contingencies have already been occurred, the proposed models present limitations in maintaining proper level of reliability that can be improved by incorporating certain requirements of the islanded operation into the optimal scheduling problem for microgrids.

The work presented in this paper is to fill the gap in considering the operation for integrated microgrids in both grid-connected and islanded modes. More specifically, the optimal scheduling of local generators in the normal operation needs to take into account the requirement for satisfying critical loads when switching into the islanded operation. The objective is to minimize the generation cost of the associated distribution system in the normal operation while ensure the minimal amount of load curtailment when microgrids switch into the islanded operation. Due to the large number of constraints incorporated into the model, the formulated problem

This work was supported in part by the U.S. National Science Foundation under grant CMMI-1434789, in part by Electric Power Analytics Consortium.

H. K. Nguyen and Z. Han are with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77004 USA (e-mail: hknguyen9@uh.edu, zhan2@uh.edu).

A. Khodaei is with the Department of Electrical and Computer Engineering, University of Denver, Denver, CO 80210 USA (e-mail: amin.khodaei@du.edu).

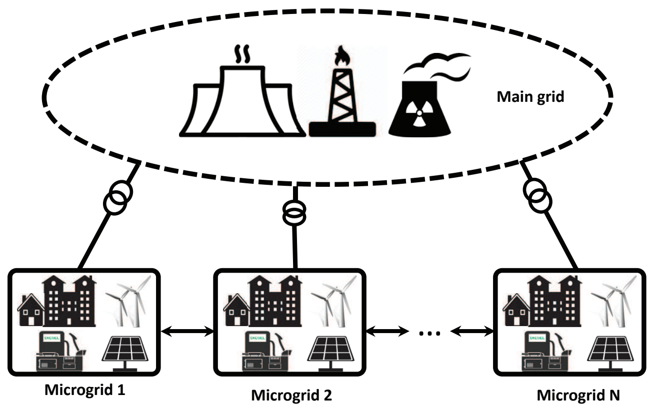

Fig. 1. The model of microgrids with islanding operation.

becomes a large-scale optimization problem, and may not be scalable to solve by the centralized method. Therefore, we apply the alternating direction method of multipliers (ADMM) decomposition technique [18] to efficiently solve the problem. By jointly tackling the above discussed challenges, our main technical contributions can be summarized as follows:

- *Minimal Load Curtailment Modeling in Islanded Operations*: We formulate a microgrid optimal scheduling problem in the normal operation mode. Moreover, we also incorporate ℓ_1-norm into the objective function to obtain the minimal amount of load curtailment when microgrids are disconnected from the main grid. The model demonstrates that only a sparse number of microgrids have to curtail load when switching into the islanded operation.
- *Parallel Algorithm*: The formulated problem consists of a large number of constraints and becomes a large-scale optimization problem. Therefore, we propose a parallel algorithm using the ADMM decomposition technique to efficiently solve the optimization problem.
- *Big Data Framework Implementation*: We provide a detailed implementation of the parallel computing for our proposed algorithm using the Hadoop MapReduce software framework to run on a computer cluster to reduce the computational complexity.

The remainder of this paper is organized as follows. The optimal scheduling problem with the islanded operation constraints is formulated in Section II. Section III provides the decomposition algorithm using ADMM. Simulation results are presented in Section IV. Detailed implementation of parallel computing for the proposed algorithm on a computer cluster using the Hadoop MapReduce is described in Section V, and Section VI concludes the paper.

II. System Model

In this section, we describe the model of the microgrid system, and formulate the optimal scheduling problem of integrated microgrids.

A. System Description

Consider a distribution network consisting a set $\mathcal{N} \triangleq \{1, 2, \ldots, N\}$ of microgrids, which are connected to the main power grid as in Fig. 1. Each microgrid $i \in \mathcal{N}$ is required to serve a group of customers having demand D_i. To fully satisfy demand requested from users, each microgrid can locally generate power using its generator and/or acquires power from the main grid. Moreover, each microgrid has direct connections with a group of other microgrids and can exchange power locally.

Let $\mathcal{N}_i$ be the set of neighboring microgrids connected to microgrid i through transmission lines (including the microgrid i itself), with cardinality $|\mathcal{N}_i| = N_i$. Define $\boldsymbol{x}_i^o = \{x_{ij}^o\}_{j\in\mathcal{N}_i}$ be the vector of power generation that microgrid i generates to exchange with its neighbors, where x_{ii}^o is the amount of power that microgrid i generates to supply its own customers. We use superscript o for all variables in the normal operation to differentiate with variables in the islanded operation, which will be defined later. Then the total amount of power that microgrid i has to generate during the normal operation using its local generation is $\mathbf{1}^T\boldsymbol{x}_i^o = \sum_{j\in\mathcal{N}_i} x_{ij}^o$, where $\mathbf{1} = [1, 1, \ldots, 1]^T$ is a column vector of ones.

In order to satisfy demand for its users, microgrid i acquires y_i^o amount of power from the main grid. Any power transfer between a microgrid and the main grid is accompanied with the loss of power over the distribution lines. The power loss due to the power exchange between the main power grid and microgrid i, $P_{loss,i-o}^o$, can be calculated as [19]

$$P_{loss,i-o}^o = \frac{R_{oi}(y_i^o)^2}{V_o^2} + \alpha y_i^o, \tag{1}$$

where R_{oi} is resistance of the transmission line connecting microgrid i and the main grid, α is the power loss constant due to transformer, and V_o is operating voltage between microgrid i and the main grid. Similarly, the power loss due to power exchange between two microgrids i and j, $P_{loss,i-j}^o$, can be calculated as

$$P_{loss,i-j}^o = \frac{R_{ij}(x_{ij}^o)^2}{V_1^2}, \tag{2}$$

where R_{ij} is the resistance of the transmission line connecting microgrid i and j, and V_1 is the operating voltage between microgrids. The difference between (1) and (2) is that there is no power loss due to transformer in (2) since all microgrids operate at the same voltage level.

Then the total power that a microgrid obtains to supply its customers includes: its local generation, power acquired from the main grid and/or from its neighbors. We have the following power balance constraint in the normal operation

$$x_{ii}^o + (y_i^o - P_{loss,i-o}^o) + \sum_{j\in\mathcal{N}_i, j\neq i} (x_{ji}^o - P_{loss,j-i}^o) = D_i, \forall i. \tag{3}$$

B. Islanded Operation

When microgrid k is disconnected from the main grid, the power that it obtains to supply its customers can only be generated locally from its generator and/or from the power exchange with its neighbors. However, it may be possible that the total amount of available power when a microgrid is islanded cannot fully satisfy the demand for its customers.

Therefore, the load demand that microgrid k is able to serve will be reduced. Let ϵ^k denote the fraction amount of load demand that microgrid k has to curtail if it is islanded from the main grid (we use superscript k to denote all variables in islanded operation case k, which is corresponding to situation that microgrid k is islanded from the main grid). Since microgrids serve different types of demand, they typically have different requirements for load satisfaction when switching into the islanded operation. Therefore, we have the following constraint for load curtailment of microgrid k

$$0 \leq \epsilon^k \leq \epsilon^k_{max}, \tag{4}$$

where ϵ^k_{max} is the predefined maximum allowable fraction of load curtailment. The amount of load demand that microgrid k has to satisfy in the islanded operation mode is $(1-\epsilon^k)D_k$. Then we have the following constraint for microgrid k

$$x^k_{kk} + \sum_{j \in \mathcal{N}_k, j \neq k} (x^k_{jk} - P^k_{loss,j-k}) \geq (1-\epsilon^k)D_k. \tag{5}$$

Note that, in (5), there is no variable y^k_k since microgrid k is islanded and cannot obtain the power from the main grid. Moreover, we assume that each microgrid is located at a different geographical area, and therefore, when the upstream disturbance or voltage fluctuation happens, only one microgrid is disconnected from the main grid at any given time instance. The more general case of more than one microgrid are islanded from the main grid can be directly applied without changing the structure of the paper by constructing and adding constraint (5) into the problem for any microgrids switch into the islanded operation. All remaining microgrids $i \neq k$ that are still connected to the main grid must fully satisfy their users demand as the following power balance constraint

$$x^k_{ii} + (y^k_i - P^k_{loss,i-o}) + \sum_{j \in \mathcal{N}_i, j \neq i} (x^k_{ji} - P^k_{loss,j-i}) = D_i. \tag{6}$$

When microgrid k is islanded from the main grid, all microgrids in the network have to adjust the power generation output compared to the normal operation mode to help satisfy power demand for users. However, the adjustment of power generation output must be constrained by the ramping limit

$$|\mathbf{1}^T \boldsymbol{x}^k_i - \mathbf{1}^T \boldsymbol{x}^o_i| \leq \Delta^{max}_i, \forall i \in \mathcal{N}, \tag{7}$$

where Δ^{max}_i is the maximum ramping rate of local generator at microgrid i. Similarly, the main grid also needs to reschedule its generation output when a microgrid is disconnected from the main distribution network. Therefore, we have the following ramping constraint

$$|\mathbf{1}^T \boldsymbol{y}^k_i - \mathbf{1}^T \boldsymbol{y}^o_i| \leq \Delta^{max}_o, \tag{8}$$

where $\boldsymbol{y}^k = [y^k_1, y^k_2, \ldots, y^k_N]$ denotes the power flow vector from the main grid to microgrids when microgrid k is islanded, and Δ^{max}_o is the maximum ramping rate of generators at the main grid.

Note that, all constraints (5), (6), (7), and (8) must be constructed for all possible islanded operation cases by considering each microgrid is disconnected from the main grid one at a time.

C. Optimal Scheduling Problem of Integrated Microgrids

In this subsection, we formulate an optimization problem to obtain the optimal scheduling for all microgrids in the normal operation. In addition, we want the optimal solution to guarantee the reliable operation of microgrids as well. Specifically, when a microgrid is disconnected from the main grid, the local generator at each microgrid is able to adjust its generation output to a new operation point compared to that in the normal operation so that to minimize the amount of load curtailment.

Let $\mathcal{C}_i(x)$ be the convex cost function for generating x amount units of power at microgrid i, or at the main grid if $i = 0$. We also define $\boldsymbol{\epsilon_D} = [\epsilon^1 D_1, \epsilon^2 D_2, \ldots, \epsilon^k D_k, \ldots, \epsilon^N D_N]^T$ as the load curtailment vector of microgrids. Then the optimal scheduling problem for microgrids and the main grid can be formulated as follow

$$\min \quad \mathcal{C}_o(\boldsymbol{y}^o) + \sum_{i=1}^{N} \mathcal{C}_i(\boldsymbol{x}^o_i) + \tau \|\boldsymbol{\epsilon_D}\|_1 \tag{9}$$

$$\text{s.t.} \quad x^o_{ii} + (y^o_i - P^o_{loss,i-o}) + \sum_{j \in \mathcal{N}_i, j \neq i} (x^o_{ji} - P^o_{loss,j-i}) = D_i, \forall i, \tag{10}$$

$$-FL^{max}_{i-j} \leq x^o_{ij} + x^o_{ji} \leq FL^{max}_{i-j}, \forall i,j, \tag{11}$$

$$-FL^{max}_i \leq y^o_i \leq FL^{max}_i, \forall i, \tag{12}$$

$$x^k_{kk} + \sum_{j \in \mathcal{N}_k, j \neq k} (x^k_{jk} - P^k_{loss,j-k}) \geq (1-\epsilon^k)D_k, \forall k, \tag{13}$$

$$x^k_{ii} + (y^k_i - P^k_{loss,i-o}) + \sum_{j \in \mathcal{N}_i, j \neq i} (x^k_{ji} - P^k_{loss,j-i}) = D_i, \forall i \neq k, \forall k, \tag{14}$$

$$-FL^{max}_{i-j} \leq x^k_{ij} + x^k_{ji} \leq FL^{max}_{i-j}, \forall i,j, \forall k, \tag{15}$$

$$-FL^{max}_i \leq y^k_i \leq FL^{max}_i, \forall i, \forall k, \tag{16}$$

$$0 \leq \epsilon^k \leq \epsilon^k_{max}, \forall k, \tag{17}$$

$$|\mathbf{1}^T \boldsymbol{x}^k_i - \mathbf{1}^T \boldsymbol{x}^o_i| \leq \Delta^{max}_i, \forall i, \forall k, \tag{18}$$

$$|\mathbf{1}^T \boldsymbol{y}^k - \mathbf{1}^T \boldsymbol{y}^o| \leq \Delta^{max}_o, \forall k, \tag{19}$$

where τ is a positive weighted parameter to capture the trade-off between generation cost minimization and minimal amount of load curtailment. The third term in (9), $\|\boldsymbol{\epsilon_D}\|_1$, is ℓ_1-norm of vector $\boldsymbol{\epsilon_D}$, which determines the amount of load curtailment in the islanded operation mode

$$\|\boldsymbol{\epsilon_D}\|_1 \stackrel{\text{def}}{=} \sum_k |\epsilon^k D_k|.$$

Incorporating ℓ_1-norm, $\|\boldsymbol{\epsilon_D}\|_1$, into the objective function (9) allows us to obtain the optimal solution that minimizes the amount of load curtailment when microgrids switch into islanded operation mode. The constraints (11), (12), (15), and (16) are the line flow limit where FL^{max} is the flow limit.

The optimization problem in (9)-(19) is convex and can be solved in a centralized fashion to obtain the global optimal solution. However, since a large number of constraints are coupled over the normal operation and the islanded operation, the centralized computation scheme is not scalable.

III. Decomposition Algorithm based on ADMM

In this section, we first provide an overview of the ADMM method to solve a convex optimization problem. Then a parallel algorithm for the optimal scheduling of microgrids is proposed.

A. An Introduction to ADMM

Consider an optimization problem with the general form as

$$\begin{aligned} \min \quad & f(x) + g(z) \\ \text{s.t.} \quad & Ax + Bz = c. \end{aligned} \tag{20}$$

The augmented Lagrangian function of the problem in (20) is written as [18]

$$\begin{aligned} \mathcal{L}(x, z, \lambda) = f(x) + g(z) + \lambda^T(Ax + Bz - c) \\ + \frac{\rho}{2}\|Ax + Bz - c\|^2, \end{aligned} \tag{21}$$

where λ is the Lagrangian multiplier, and ρ is a penalty parameter. The iterative procedure of ADMM to solve the problem in (20) can be expressed as follows

$$x[t+1] := \arg\min_{x} \mathcal{L}(x, z[t], \lambda[t]), \tag{22}$$

$$z[t+1] := \arg\min_{z} \mathcal{L}(x[t+1], z, \lambda[t]), \tag{23}$$

$$\lambda[t+1] := \lambda[t] + \rho\left(Ax[t+1] + Bz[t+1] - c\right), \tag{24}$$

where in each iteration, the augmented Lagrangian function is minimized over x and z in an alternating fashion.

B. Parallel Algorithm for Microgrid Scheduling

The problem in (9)-(19) contains a large number of constraints and becomes a large-scale optimization problem. However, we realize that constraints are separable into the different islanded operation cases for different microgrids. In order to make the problem in (9)-(19) to be more compact and ready for using ADMM, we define the feasible set of power generation vectors in the normal operation case and in the islanded operation cases as follow

$$\begin{aligned} &\mathcal{F}_o = \{(\boldsymbol{x}^o, \boldsymbol{y}^o) | (10), (11), (12)\}, \\ &\mathcal{F}_k = \{(\boldsymbol{x}^k, \boldsymbol{y}^k, \epsilon^k) | (13), (14), (15), (16), (17)\}, \forall k. \end{aligned}$$

We further introduce auxiliary variables $\boldsymbol{y}^{o,k}$ and $\boldsymbol{x}_i^{o,k}$ as the local copies of $\boldsymbol{y}^o$ and $\boldsymbol{x}_i^o$ in the normal operation at each islanded operation case

$$\begin{aligned} \boldsymbol{y}^{o,k} &= \boldsymbol{y}^o, \forall k, \\ \boldsymbol{x}_i^{o,k} &= \boldsymbol{x}_i^o, \forall i, k. \end{aligned}$$

Then, the problem in (9)-(19) can be reformulated as follow

$$\begin{aligned} \min \quad & \mathcal{C}_o(\boldsymbol{y}^o) + \sum_{i=1}^{N} \mathcal{C}_i(\boldsymbol{x}_i^o) + \tau \sum_{k=1}^{N} \epsilon^k D_k \\ \text{s.t.} \quad & (\boldsymbol{x}^o, \boldsymbol{y}^o) \in \mathcal{F}_o, \\ & (\boldsymbol{x}^k, \boldsymbol{y}^k, \epsilon^k) \in \mathcal{F}_k, \forall k = 1, \ldots, N, \\ & -\Delta_o^{max} \le \mathbf{1}^T\boldsymbol{y}^k - \mathbf{1}^T\boldsymbol{y}^{o,k} \le \Delta_o^{max}, \forall k, \\ & -\Delta_i^{max} \le \mathbf{1}^T\boldsymbol{x}_i^k - \mathbf{1}^T\boldsymbol{x}_i^{o,k} \le \Delta_i^{max}, \forall i, k, \\ & \boldsymbol{y}^{o,k} = \boldsymbol{y}^o, \forall k, \\ & \boldsymbol{x}_i^{o,k} = \boldsymbol{x}_i^o, \forall i, k. \end{aligned} \tag{25}$$

The augmented Lagrangian function of the problem in (25) is given by [18]

$$\begin{aligned} \mathcal{L} =& \mathcal{C}_o(\boldsymbol{y}^o) + \sum_{i=1}^{N} \mathcal{C}_i(\boldsymbol{x}_i^o) + \tau \sum_{k=1}^{N} \epsilon^k D_k \\ &+ \sum_{k=1}^{N} (\boldsymbol{\lambda}^k)^T(\boldsymbol{y}^{o,k} - \boldsymbol{y}^o) + \sum_{k=1}^{N}\sum_{i=1}^{N} (\boldsymbol{\mu}_i^k)^T(\boldsymbol{x}_i^{o,k} - \boldsymbol{x}_i^o) \\ &+ \frac{\gamma}{2}\sum_{k=1}^{N} \|\boldsymbol{y}^{o,k} - \boldsymbol{y}^o\|^2 + \frac{\gamma}{2}\sum_{k=1}^{N}\sum_{i=1}^{N} \|\boldsymbol{x}_i^{o,k} - \boldsymbol{x}_i^o\|^2 \\ =& \mathcal{C}_o(\boldsymbol{y}^o) + \sum_{i=1}^{N} \mathcal{C}_i(\boldsymbol{x}_i^o) \\ &+ \sum_{k=1}^{N} \Big[\tau\epsilon^k D_k + (\boldsymbol{\lambda}^k)^T(\boldsymbol{y}^{o,k} - \boldsymbol{y}^o) \\ &\quad + \sum_{i=1}^{N} (\boldsymbol{\mu}_i^k)^T(\boldsymbol{x}_i^{o,k} - \boldsymbol{x}_i^o) \\ &\quad + \frac{\gamma}{2}\|\boldsymbol{y}^{o,k} - \boldsymbol{y}^o\|^2 + \frac{\gamma}{2}\sum_{i=1}^{N} \|\boldsymbol{x}_i^{o,k} - \boldsymbol{x}_i^o\|^2 \Big], \end{aligned} \tag{26}$$

where $\boldsymbol{\lambda}, \boldsymbol{\mu}$ are the Lagrangian multipliers, and γ is a penalty parameter.

Define the primal variables $\boldsymbol{z} = (\boldsymbol{y}^o, \{\boldsymbol{x}_i^o\}_{\forall i})$, which is the decision variable vector in normal operation, and $\boldsymbol{w} = (\{\boldsymbol{w}^k, \epsilon^k\}_{\forall k})$, where $\boldsymbol{w}^k = (\boldsymbol{y}^k, \{\boldsymbol{x}_i^k\}_{\forall i})$ is the decision variable vector in islanded operation case k. Then the ADMM decomposition technique can be used to solve the problem in (25) in an iterative procedure. Specifically, at the t^{th} iteration, the primal variables and dual variables are updated sequentially as

$$\boldsymbol{z}[t+1] = \arg\min \mathcal{L}(\boldsymbol{z}, \boldsymbol{w}[t], \boldsymbol{\lambda}[t], \boldsymbol{\mu}[t]), \tag{27}$$

$$\boldsymbol{w}[t+1] = \arg\min \mathcal{L}(\boldsymbol{z}[t+1], \boldsymbol{w}, \boldsymbol{\lambda}[t], \boldsymbol{\mu}[t]), \tag{28}$$

$$\boldsymbol{\lambda}^k[t+1] = \boldsymbol{\lambda}^k[t] + \gamma\left(\boldsymbol{y}^{o,k}[t+1] - \boldsymbol{y}^o[t+1]\right), \forall k, \tag{29}$$

$$\boldsymbol{\mu}_i^k[t+1] = \boldsymbol{\mu}_i^k[t] + \gamma\left(\boldsymbol{x}_i^{o,k}[t+1] - \boldsymbol{x}_i^o[t+1]\right), \forall i, k. \tag{30}$$

Based on the Lagrangian function in (26), we decompose the problem in (25) into the following $N+1$ optimization problems. The first problem is associated with variables in the normal operation mode only and corresponding to the primal variables update in (27)

$$
\begin{aligned}
\min \quad & \mathcal{C}_o(\boldsymbol{y}^o) + \sum_{i=1}^{N} \mathcal{C}_i(\boldsymbol{x}_i^o) - (\boldsymbol{y}^o)^T \sum_{k=1}^{N} \boldsymbol{\lambda}^k \\
& - \sum_{k=1}^{N}\sum_{i=1}^{N} (\boldsymbol{\mu}_i^k)^T \boldsymbol{x}_i^o + \frac{\gamma}{2}\sum_{k=1}^{N} \|\boldsymbol{y}^{o,k} - \boldsymbol{y}^o\|^2 \\
& + \frac{\gamma}{2}\sum_{k=1}^{N}\sum_{i=1}^{N} \|\boldsymbol{x}_i^{o,k} - \boldsymbol{x}_i^o\|^2 \qquad (31) \\
\text{s.t.} \quad & (\boldsymbol{y}^o, \{\boldsymbol{x}_i^o\}_{\forall i}) \in \mathcal{F}_o.
\end{aligned}
$$

By fixing the values of $\{\boldsymbol{x}_i^{o,k}\}_{\forall i}, \boldsymbol{y}^{o,k}, \boldsymbol{\lambda}, \boldsymbol{\mu}$, and then solving the problem in (31), we obtain the optimal solution for $(\boldsymbol{y}^o, \{\boldsymbol{x}_i^o\}_{\forall i})$.

The remaining N problems are associated with variables in the islanded operation cases and corresponding to the primal variables update in (28). For each islanded operation case, we decompose into the following problem

$$
\begin{aligned}
\min \quad & \tau\epsilon^k D_k + (\boldsymbol{\lambda}^k)^T \boldsymbol{y}^{o,k} + \sum_{i=1}^{N} (\boldsymbol{\mu}_i^k)^T \boldsymbol{x}_i^{o,k} \\
& + \frac{\gamma}{2}\|\boldsymbol{y}^{o,k} - \boldsymbol{y}^o\|^2 + \frac{\gamma}{2}\sum_{i=1}^{N} \|\boldsymbol{x}_i^{o,k} - \boldsymbol{x}_i^o\|^2 \qquad (32) \\
\text{s.t.} \quad & (\boldsymbol{x}^k, \boldsymbol{y}^k, \epsilon^k) \in \mathcal{F}_k, \\
& -\Delta_o^{max} \le \mathbf{1}^T \boldsymbol{y}^k - \mathbf{1}^T \boldsymbol{y}^{o,k} \le \Delta_o^{max}, \\
& -\Delta_i^{max} \le \mathbf{1}^T \boldsymbol{x}_i^k - \mathbf{1}^T \boldsymbol{x}_i^{o,k} \le \Delta_i^{max}, \\
\text{variables:} \quad & \{\boldsymbol{x}_i^k\}_{\forall i}, \boldsymbol{y}^k, \epsilon^k, \{\boldsymbol{x}_i^{o,k}\}_{\forall i}, \boldsymbol{y}^{o,k}.
\end{aligned}
$$

By fixing $(\boldsymbol{y}^o, \{\boldsymbol{x}_i^o\}_{\forall i}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ and then solving the problem in (32), we obtain the optimal solution for $\left(\{\boldsymbol{x}_i^k\}_{\forall i}, \boldsymbol{y}^k, \{\boldsymbol{x}_i^{o,k}\}_{\forall i}, \boldsymbol{y}^{o,k}, \epsilon^k\right)$.

Algorithm Implementation: The whole procedure for solving the problem in (25) using ADMM is described in Algorithm 1. First, a master computing node solves the optimization problem in (31) to obtain the optimal solution $(\boldsymbol{y}^o, \{\boldsymbol{x}_i^o\}_{\forall i})$. Then it broadcasts the optimal solution in the normal operation mode to all distributed computing nodes. Each distributed computing node solves the optimization problem in (32) to obtain the optimal solution $\left(\{\boldsymbol{x}_i^k\}_{\forall i}, \boldsymbol{y}^k, \{\boldsymbol{x}_i^{o,k}\}_{\forall i}, \boldsymbol{y}^{o,k}, \epsilon^k\right)$ in the islanded operation. Finally, based on the local values of $\left(\boldsymbol{y}^{o,k}, \{\boldsymbol{x}_i^{o,k}\}_{\forall i}\right)$, and $(\boldsymbol{y}^o, \{\boldsymbol{x}_i^o\}_{\forall i})$, the dual variables can be updated as in line 15 in Algorithm 1. Note that, N optimization problems associated with the islanded operation cases are decoupled and can be solved in a parallel fashion at different computing nodes without affecting the others. This parallel implementation reduces the computation time for the proposed Algorithm 1.

The amount of information exchange between the master node and distributed computing nodes is depicted in Fig. 2. The master node broadcasts the same solution in the normal operation to all distributed nodes. Each distributed node needs to send the local information $\left(\boldsymbol{\lambda}, \boldsymbol{\mu}_i, \boldsymbol{y}^{o,k}, \boldsymbol{x}_i^{o,k}\right)$ to the master. Note that in the proposed algorithm, all microgrids are required to exchange information with the master computer

Algorithm 1 ADMM Decomposition

1: Initialization: $t = 0, \boldsymbol{\lambda} = 0, \{\boldsymbol{\mu}_i = 0\}_{\forall i}$
2: **repeat**
3: **At master computer:**
4: **repeat**
5: wait
6: **until** receive updates $\boldsymbol{\lambda}, \boldsymbol{\mu}_i, \boldsymbol{y}^{o,k}, \boldsymbol{x}_i^{o,k}$ from all distributed computers
7: **solve** (31) for optimal solution $(\boldsymbol{y}^o[t+1], \{\boldsymbol{x}_i^o[t+1]\}_{\forall i})$
8: **broadcast** $(\boldsymbol{y}^o[t+1], \{\boldsymbol{x}_i^o[t+1]\}_{\forall i})$ to all distributed computers
9: ———————————————
10: **At each distributed computer** k
11: **repeat**
12: wait
13: **until** receive updates $(\boldsymbol{y}^o[t+1], \{\boldsymbol{x}_i^o[t+1]\}_{\forall i})$ from master computer
14: **solve** (32) for the optimal solution $\{\boldsymbol{x}_i^k\}_{\forall i}, \boldsymbol{y}^k, \{\boldsymbol{x}_i^{o,k}\}_{\forall i}, \boldsymbol{y}^{o,k}, \epsilon^k$
15: **update** dual variables:
$$\boldsymbol{\lambda}^k[t+1] = \boldsymbol{\lambda}^k[t] + \gamma\left(\boldsymbol{y}^{o,k}[t+1] - \boldsymbol{y}^o[t+1]\right)$$
$$\boldsymbol{\mu}_i^k[t+1] = \boldsymbol{\mu}_i^k[t] + \gamma\left(\boldsymbol{x}_i^{o,k}[t+1] - \boldsymbol{x}_i^o[t+1]\right), \forall i$$
16: **send** $\left(\boldsymbol{\lambda}, \boldsymbol{\mu}_i, \boldsymbol{y}^{o,k}, \boldsymbol{x}_i^{o,k}\right)$ to the master computer
17: ———————————————
18: $t \leftarrow t + 1$
19: **until** a stopping criterion is met

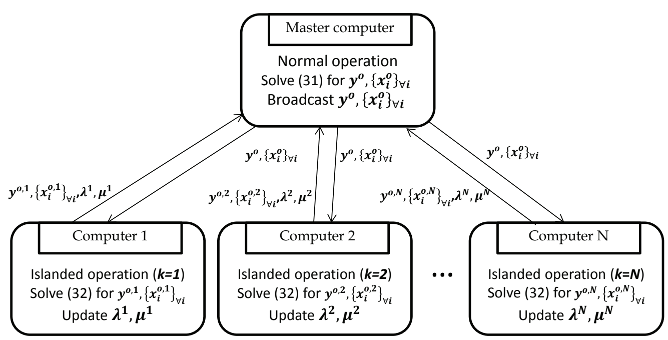

Fig. 2. The illustration of information exchange between the normal operation subproblem and islanded operation subproblems.

to solve subproblems. This can be performed via certain entities that are designed to operate distribution networks such as distribution system operators (DSOs). Besides the responsibility of controlling and operating the distribution grids, DSOs will play a role as information hubs to facilitate for data exchange as well as data aggregation [20]–[23].

IV. Simulation Results

In this section, we use computational experiments to evaluate the performance of our proposed algorithm. We use two modified IEEE 9-bus and 14-bus power systems [24] to obtain the physical connections as indicators for communication lines between microgrids, in which each bus is considered as a

TABLE I
DEMAND OF MICROGRIDS

Microgrid	1	2	3	4	5
Demand (MW)	108	97	180	74	71
Microgrid	6	7	8	9	10
Demand (MW)	136	125	171	175	195
Microgrid	11	12	13	14	
Demand (MW)	265	194	317	100	

microgrid. The operation voltage between the main grid and microgrids, $V_0 = 50$ kV, while the operation voltage between microgrids is $V_1 = 22$ kV [25]. The power loss constant $\alpha = 0.02$ [25]. The generation capacity of power generator for each microgird is generated as 10% greater than its total demand, and is able to adjust up to 10% of its maximum output capacity when switching to the islanded operation mode. To avoid the infeasibility of the problem, we set the maximum allowable fraction amount of load curtailment, $\epsilon^k_{max} = 1$ for all microgrids. We select $\tau = 10$ in all simulations, unless otherwise stated. The power demand of microgrids is given in Table I. We assume the convex cost function for main grid and microgrids as $\mathcal{C}(x) = ax^2 + bx$, where a and b are generated randomly from a uniform distribution $a \in [0.01, 0.5]\left(\$/MW^2\right)$, $b \in [10, 40]\,(\$/MW)$. All tests are conducted on a Windows 7 64-bit personal computer with Intel i7-4770 3.4 GHz CPU and 16GB of RAM using Matlab. Each sub-problem in our proposed algorithm is solved using CVX [26].

To demonstrate the advantage of the ADMM decomposition, we show the number of iterations required for the proposed algorithm to converge in Fig. 3. For the system with 9 microgrids, the relative error approaches to 10^{-4} in about 40 iterations, while 14 microgrids system needs 60 iterations to yield the same relative error. This is due to the fact that a larger system leads to a greater number of constraints in the islanded mode, and consequently produces more sub-problems when using ADMM decomposition. Further, notice that the computational time required for each iteration is varied for different systems. The average computational time for each iteration in the system with 9 microgrids is 8.2 seconds, while it is 41.3 seconds for the system with 14 microgrids.

It is an important task to find an approximate value for the parameter τ in problem (9)-(19). In general, higher values of τ increase the weight for ℓ_1-norm term in the objective function, and the resultant optimal solution achieves smaller amount of the load curtailment in the islanded operation. Particularly, Fig. 4 plots the average percentage of load curtailment as a function of τ, and the result shows that we can significantly reduce the amount of load shedding by setting a higher value for τ, which can achieve about 1% of the total load of the system.

Even though a higher value of τ leads to a smaller amount of load shedding, it may incur in general a higher generation cost in normal operation. To investigate the impact of τ on the operational cost of the system, we plot the generation cost versus τ in Fig. 5. It shows that the system total generation cost does not increase too much when we increase τ. Based on the results from Fig. 4 and Fig. 5, we can select an appropriate value of τ to satisfy the design criteria for power systems. This

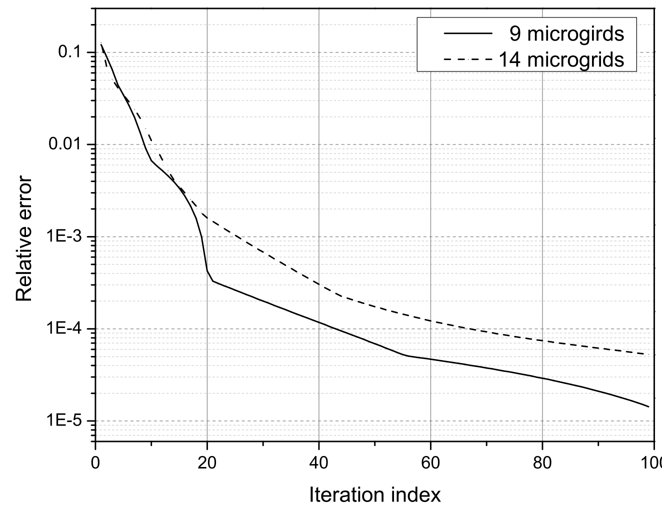

Fig. 3. The convergence performance of the proposed algorithm.

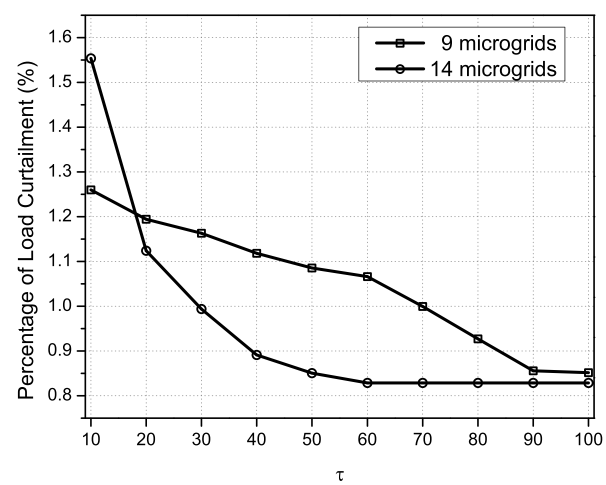

Fig. 4. The effect of τ on the percentage of load curtailment when microgrids switch into the islanded operation mode.

selection will obtain the trade-off between the generation cost in the normal operation and the amount of load shedding in the islanded mode.

To further study the effectiveness of our model in improving the reliability of power systems, we report the generation cost and the amount of load shedding as in Table II, with and without ℓ_1-norm in the objective function. The column with ℓ_1-norm denotes the results when we incorporate ℓ_1-norm into the objective function. It shows that with ℓ_1-norm in the model does not increase too much generation cost in the normal operation while still obtains large reduction on the percentage of load shedding compared to the model without ℓ_1-norm. This

TABLE II
TOTAL GENERATION COST AND LOAD CURTAILMENT COMPARISON

	No ℓ_1-norm		With ℓ_1-norm	
	Cost($)	Cut(%)	Cost($)	Cut(%)
9 microgrids	33236	9.57	33291	1.26
14 microgrids	57000	6.3	57402	1.55

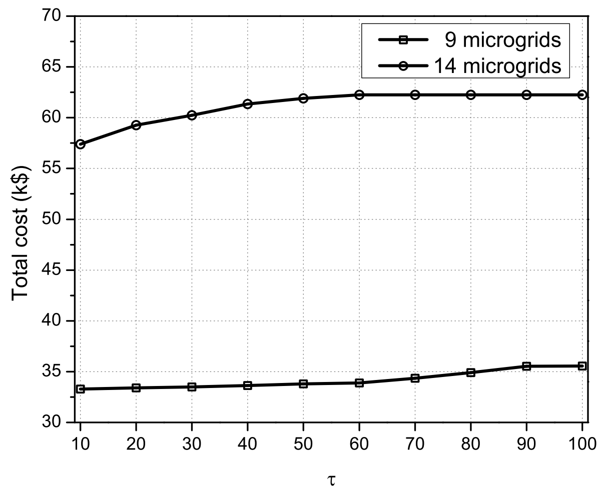

Fig. 5. The effect of τ on the generation cost of the system in the normal operation.

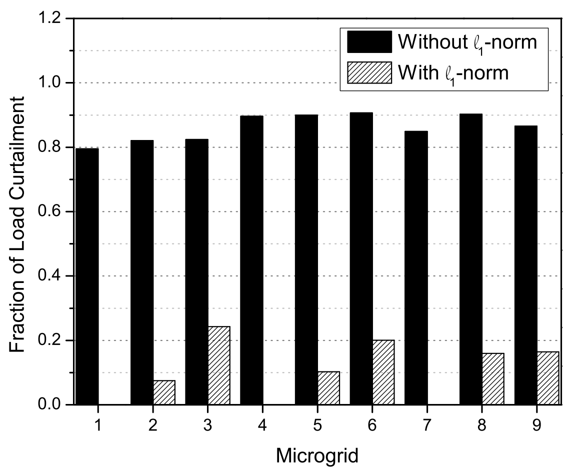

Fig. 6. The fraction of load shedding of each individual microgird in the islanded operation mode for the 9-microgrid system.

indicates that incorporating ℓ_1-norm in our model can improve the power system reliability without significantly increasing the generation cost in the normal operation. Moreover, our model not only reduces the percentage of load shedding, but also produces a sparse solution, which means that only several microgrids have to reduce their loads. To demonstrate this, Fig. 6 plots the fraction of load shedding for each individual microgrid in the system with 9 microgrids. It can be noticed that only microgrids 2, 3, 5, 6, 8, 9 will reduce their loads when switching into the islanded operation.

V. Hadoop MapReduce Implementation

In this section, we introduce an overview of MapReduce programming model and describe the detailed implementation of ADMM Algorithm 1 using Hadoop MapReduce framework.

A. MapReduce Programming Model

MapReduce is a programming model for distributed processing of very large datasets using a large cluster of commodity machines [27]. It has been widely used to perform

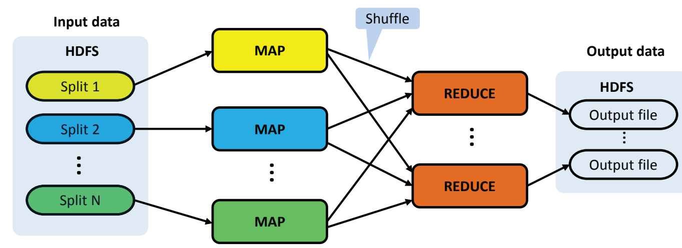

Fig. 7. The flow chart of MapReduce programming model.

special-purpose computations both in industry and academia [28]. A MapReduce computation consists of a set of Map tasks and Reduce tasks. The input data will be split into independent blocks and processed by the Map tasks in a completely parallel manner to produce a set of intermediate key-value pairs. Then, all outputs of the mapping operation that share the same intermediate key will be grouped together and passed to the same Reducer. The MapReduce work flow is shown in Fig. 7. Generally, the Map and Reduce steps can be conceptually expressed as [27]

$$\begin{array}{lll} \text{map} & (k_1, v_1) & \longrightarrow list(k_2, v_2) \\ \text{reduce} & (k_2, list(v_2)) & \longrightarrow list(v_3). \end{array}$$

Apache Hadoop is an open-source software framework written in Java for easily writing application to process massive amount of data on computer clusters in reliable, fault-tolerant manner [29]. The core of Hadoop consists of a storage part, which provides the Hadoop Distributed File System (HDFS) architecture, and a processing part which implements the MapReduce computation paradigm. The HDFS manages the storage of data across an entire cluster of machines by splitting files into blocks and distributing them amongst the nodes in the cluster. Then, the data at each node is divided into fixed-size piece called splits. Each split of data is processed in the Map tasks based on the user-defined Map function to produce a list of key-value pairs. The process of sorting key-value pairs of map tasks and sending them to reducers is handling internally by Hadoop. This allows Hadoop to reduce many complexities such as data partitioning, scheduling tasks across many machines, handling machine failures and performing inter-machine communication [30].

B. ADMM Implementation using Hadoop MapReduce

Each iteration of ADMM Algorithm 1 can be represented as a MapReduce job as illustrated in Fig. 8. The parallel computations for islanded operation sub-problems in (32) are performed by Map tasks, and the normal operation sub-problem computation in (31) is performed by a Reduce task. We have total N Mappers, one for each islanded operation sub-problem. Each Mapper solves the optimization problem in (32) to obtain $\boldsymbol{y}^{o,k}, \{\boldsymbol{x}_i^{o,k}\}_{\forall i}$. However, solving the problem in (32) for $\boldsymbol{y}^{o,k}[t+1], \boldsymbol{x}_i^{o,k}[t+1]$ on iteration $t+1$ needs to use $\boldsymbol{y}^o[t], \boldsymbol{x}_i^o[t]$ and $\boldsymbol{\lambda}^k[t], \boldsymbol{\mu}^k[t]$ from the previous iteration. Since

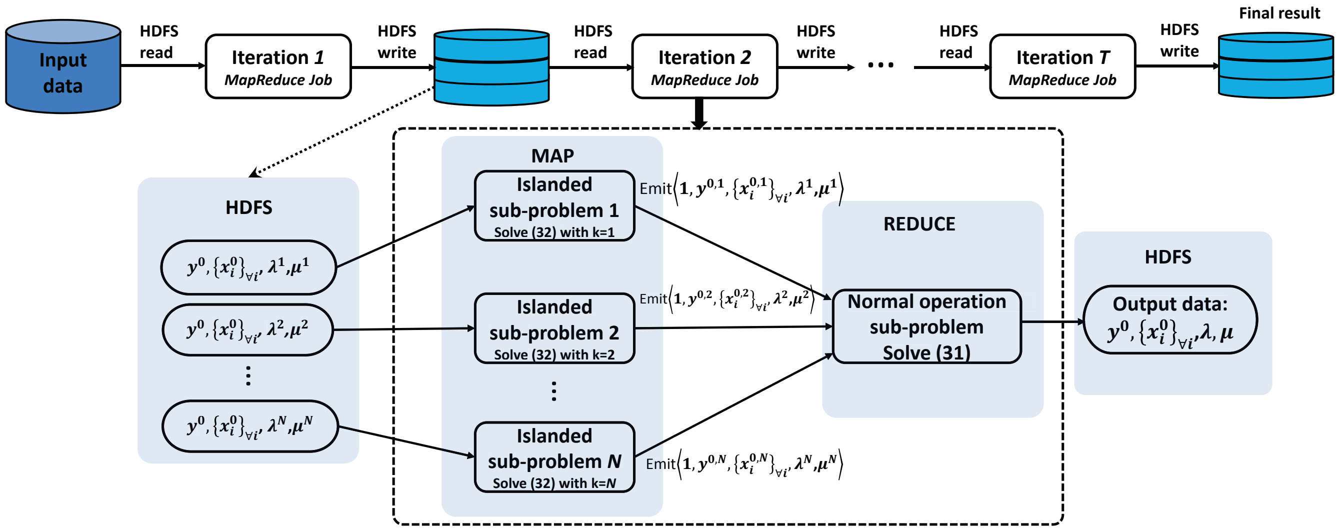

Fig. 8. Data sharing for iterative ADMM using Hadoop MapReduce and the detailed illustration of Map tasks and Reduce task in each MapReduce job in each iteration.

MapReduce is not designed to support iterative applications, we facilitate iterative computation for Algorithm 1 by writing the output data at each iteration to the HDFS, which will be used as the input data for Mappers in the next iteration. Particularly, each Mapper uses *splitID* provided by Hadoop to identify which islanded problem is and loads the corresponding $\boldsymbol{\lambda}^k[t], \boldsymbol{\mu}^k[t]$ from HDFS in the previous iteration, the $\boldsymbol{y}^o[t], \boldsymbol{x}_i^o[t]$ are the same for all Mappers.

After solving the optimization problem, each Mapper updates values for $\boldsymbol{\lambda}^k$ and $\boldsymbol{\mu}^k$ using (29) and (30), respectively. Then each Mapper emits an intermediate key-value pair, which is $\langle 1, \{\boldsymbol{y}^{o,k}, \boldsymbol{x}_i^{o,k}, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k\}\rangle$ to the Reducer. Since in our problem, there is a single Reducer, which plays the role of performing the normal operation computation, all the keys in all Map tasks are selected as 1 to force all information from the Mappers is sent to a unique Reducer. Based on all information received from the Mappers, the Reducer solves (31) to obtain $\boldsymbol{y}^o, \{\boldsymbol{x}_i^o\}_{\forall i}$. The values of $\boldsymbol{y}^o, \{\boldsymbol{x}_i^o\}_{\forall i}, \boldsymbol{\lambda}, \boldsymbol{\mu}$ are written out to HDFS directly by the Reducer, which will be used as the input data for MapReduce job in the next iteration. The detailed Map tasks and Reduce task in each iteration is illustrated in Fig. 8. The pseudo code for implementing Algorithm 1 using Hadoop MapReduce is described in Algorithm 2.

Algorithm 2 ADMM using Hadoop MapReduce

```
function MAP(islanded_ID, inputData)
    Load data of previous iteration from HDFS corresponding to islanded_ID
    Solve islanded sub-problem (32)
    Update λ^k, μ^k using (29) and (30)
    EMIT ⟨1, {y^{o,k}, x_i^{o,k}, λ^k, μ^k}⟩
end function

function REDUCE(key, Data from Mappers)
    Concatenate {y^{o,k}, x_i^{o,k}, λ^k, μ^k}
    Solve normal operation sub-problem (31) for y^o, x^o
    EMIT ⟨{y^o, x^o, λ, μ}⟩
end function

function MAIN(inputPath, outputPath)
    Initialization
    while ( notConverged and k < maxIterations ) do
        run MapReduceJob (inputPath, outputPath)
        t ← t + 1
    end while
end function
```

C. Performance Results

We build a Hadoop cluster with 8 computers in which each computer has a 2.33GHz Intel processor, 4GB of RAM. Algorithm 2 is written in Java. Each Mapper and Reducer solve optimization problem using Gurobi, which provides interface to construct and solve optimization problem in Java programming language [31]. We run Algorithm 2 on the cluster with different total number of microgrids in the system and report the running time as in Table III. We can see that the running time for one iteration, which is determined as the time for reading input data, processing Map and Reduce tasks, and writing out data, on the cluster does not increase significantly when the number of microgrids increase. Specifically, the time for Map and Reduce tasks, which solve subproblems, is relatively small for three different systems. However, the total time for convergence increases since systems with large number of microgrids need more iterations to converge to the optimal solution.

VI. Conclusions

In this paper, we propose a new model for the microgrid generation schedule problem with the islanded operation constraints. The proposed problem produces an optimal generation schedule with a minimal amount of load curtailment when

TABLE III
RUNNING TIME ON THE CLUSTER

	9 Microgrids	14 Microgrids	30 Microgrids
One iteration (sec.)	24	25	27.7
MAP (sec.)	10.7	12.2	14.1
REDUCE (sec.)	4.2	4.8	5.4
Total time (min.)	19	24	32

microgrids have to switch into the islanded operation. To achieve this, we incorporate the ℓ_1-norm into the objective function of the problem. We apply the ADMM-based decomposition technique to decompose the large-scale centralized optimization problem into multiple sub-problems in which each sub-problem corresponds to the optimization problem in each islanded case and can be solved simultaneously at different computing nodes. Numerical results are conducted to demonstrate the convergence performance of our proposed algorithm. Moreover, the results also show that our model reduces the generation cost in the normal operation and achieves the minimal load curtailment when microgrids switch to the islanded mode. Finally, we describe the detailed implementation of parallel computing for the proposed algorithm using the Hadoop MapReduce software framework to run on a computer cluster.

REFERENCES

[1] E. Hossain, Z. Han, and V. Poor, *Smart grid communications and networking*. UK: Cambridge University Press, 2012.

[2] D. Olivares, A. Mehrizi-Sani, A. Etemadi, C. Canizares, R. Iravani, M. Kazerani, A. Hajimiragha, O. Gomis-Bellmunt, M. Saeedifard, R. Palma-Behnke, G. Jimenez-Estevez, and N. Hatziargyriou, "Trends in microgrid control," *IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 1905–1919, Jul. 2014.

[3] A. Khodaei, S. Bahramirad, and M. Shahidehpour, "Microgrid planning under uncertainty," *IEEE Transactions on Power Systems*, vol. 30, no. 5, pp. 2417–2425, Sep. 2015.

[4] A. Khodaei and M. Shahidehpour, "Microgrid-based co-optimization of generation and transmission planning in power systems," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 1582–1590, May 2013.

[5] A. Khodaei, "Provisional microgrid planning," *IEEE Transactions on Smart Grid*, vol. PP, no. 99, pp. 1–1, 2015.

[6] S. Sun, M. Dong, and B. Liang, "Joint supply, demand, and energy storage management towards microgrid cost minimization," in *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Venice, Italy, Nov. 2014, pp. 109–114.

[7] K. Rahbar, J. Xu, and R. Zhang, "Real-time energy storage management for renewable integration in microgrid: An off-line optimization approach," *IEEE Transactions on Smart Grid*, vol. 6, no. 1, pp. 124–134, Jan. 2015.

[8] M. Khodayar, M. Barati, and M. Shahidehpour, "Integration of high reliability distribution system in microgrid operation," *IEEE Transactions on Smart Grid*, vol. 3, no. 4, pp. 1997–2006, Dec. 2012.

[9] A. Khodaei, "Microgrid optimal scheduling with multi-period islanding constraints," *IEEE Transactions on Power Systems*, vol. 29, no. 3, pp. 1383–1392, May 2014.

[10] S. Parhizi, H. Lotfi, A. Khodaei, and S. Bahramirad, "State of the art in research on microgrids: A review," *IEEE Access*, vol. 3, pp. 890–925, 2015.

[11] E. Dall'Anese, H. Zhu, and G. Giannakis, "Distributed optimal power flow for smart microgrids," *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1464–1475, Sep. 2013.

[12] G. Hug, S. Kar, and C. Wu, "Consensus + innovations approach for distributed multiagent coordination in a microgrid," *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 1893–1903, Jul. 2015.

[13] W. Shi, X. Xie, C.-C. Chu, and R. Gadh, "Distributed optimal energy management in microgrids," *IEEE Transactions on Smart Grid*, vol. 6, no. 3, pp. 1137–1146, May 2015.

[14] A. Khodaei, "Resiliency-oriented microgrid optimal scheduling," *IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 1584–1591, Jul. 2014.

[15] B. Kroposki, R. Lasseter, T. Ise, S. Morozumi, S. Papatlianassiou, and N. Hatziargyriou, "Making microgrids work," *IEEE Power and Energy Magazine*, vol. 6, no. 3, pp. 40–53, May 2008.

[16] M. Vedady Moghadam, R. Zhang, and R. Ma, "Demand response for contingency management via real-time pricing in smart grids," in *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Venice, Nov. 2014, pp. 632–637.

[17] S. Pourmousavi, M. Nehrir, and R. Sharma, "Multi-timescale power management for islanded microgrids including storage and demand response," *IEEE Transactions on Smart Grid*, vol. 6, no. 3, pp. 1185–1195, May 2015.

[18] 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, Jan. 2011.

[19] A. J. Wood and B. F. Wollenberg, *Power Generation, Operation, and Control*, 2nd ed. Wiley-Interscience, 1996.

[20] E. Martinot, L. Kristov, and J. D. Erickson, "Distribution system planning and innovation for distributed energy futures," *Current Sustainable/Renewable Energy Reports*, vol. 2, no. 2, pp. 47–54, 2015.

[21] J. Taft and A. Becker-Dippmann, "Grid architecture," Jan. 2015. [Online]. Available: http://energy.gov/sites/prod/files/2015/04/f22/QER Analysis - Grid Architecture_0.pdf

[22] F. Rahimi and S. Mokhtari, "From iso to dso: imagining a new construct– an independent system operator for the distribution network," *Public Utilities Fortnightly*, vol. 152, no. 6, pp. 42–50, Dec. 2014.

[23] L. Kristov and P. De Martini, "21st century electric distribution system operations," *Caltech*, 2014. [Online]. Available: http://resnick.caltech.edu/docs/21st.pdf

[24] R. Zimmerman, C. Murillo-Sanchez, and R. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, Feb. 2011.

[25] C. Wei, Z. Md Fadlullah, N. Kato, and I. Stojmenovic, "A novel distributed algorithm for power loss minimizing in smart grid," in *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Venice, Italy, Nov. 2014, pp. 290–295.

[26] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[27] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan 2008.

[28] P. Lubell-Doughtie and J. Sondag, "Practical distributed classification using the alternating direction method of multipliers algorithm," in *2013 IEEE International Conference on Big Data*, Oct. 2013, pp. 773–776.

[29] "Apache hadoop," https://hadoop.apache.org/.

[30] T. White, *Hadoop: The Definitive Guide*, 1st ed. O'Reilly Media, Inc., 2009.

[31] I. Gurobi Optimization, "Gurobi optimizer reference manual," 2015. [Online]. Available: http://www.gurobi.com

Hung Khanh Nguyen received the B.S. degree from the Department of Electrical and Electronic Engineering, Ho Chi Minh City University of Technology, Vietnam, in 2010, and the M.S. degree from the Department of Electronics and Radio Engineering, Kyung Hee University, Korea, in 2012. He is currently working towards his Ph.D. degree at University of Houston, TX, USA. His current research interests are resource allocation and game theory, distributed and parallel optimization, large scale data processing in smart grid and wireless network.

Amin Khodaei (SM'14) received the Ph.D. degree in electrical engineering from the Illinois Institute of Technology, Chicago, in 2010. He was a visiting faculty (2010-2012) in the Robert W. Galvin Center for Electricity Innovation at Illinois Institute of Technology. He joined the University of Denver, Denver, CO, in 2013 as an Assistant Professor. Dr. Khodaei is the Technical Co-Chair of the 2016 IEEE PES T&D Conference and Exposition, Technical Co-Chair of the 2016 North American Power Symposium, and Chair of the IEEE Microgrid Taskforce. His research interests include power system operation, planning, computational economics, microgrids, and smart electricity grids.

Zhu Han (S'01-M'04-SM'09-F'14) received the B.S. degree in electronic engineering from Tsinghua University, in 1997, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, in 1999 and 2003, respectively.

From 2000 to 2002, he was an R&D Engineer of JDSU, Germantown, Maryland. From 2003 to 2006, he was a Research Associate at the University of Maryland. From 2006 to 2008, he was an assistant professor in Boise State University, Idaho. From 2006 to 2008, he was an assistant professor at Boise State University, Idaho. Currently, he is a Professor in the Electrical and Computer Engineering Department as well as in the Computer Science Department at the University of Houston, Texas. His research interests include wireless resource allocation and management, wireless communications and networking, game theory, wireless multimedia, security, and smart grid communication. Dr. Han received an NSF Career Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the EURASIP Best Paper Award for the Journal on Advances in Signal Processing in 2015, and several best paper awards in IEEE conferences. Currently, Dr. Han is currently an IEEE Communications Society Distinguished Lecturer.