A Hierarchical Alternating Direction Method of Multipliers for Fully Distributed Unit Commitment

Jinbao Jian, Chen Zhang, Linfeng Yang, Ke Meng, *Member*, *IEEE*, Yan Xu, *Member*, *IEEE*, Zhaoyang Dong, *Fellow*, *IEEE*

***Abstract*—This paper discusses a hierarchical alternating direction method of multipliers (ADMM) approach for the unit commitment (UC) problem in a fully distributed manner. Decentralized unit commitment operation schemes have several advantages when compared with the traditional centralized management system for smart grid. Specifically, decentralized management is more flexible, less computationally intensive, and easier to implement without relying on communication infrastructure. In this paper, the fully distributed UC approach is used to solve the UC problem of each individual unit in parallel. This approach consists of two layers of ADMM. In outer ADMM, x-update steps and v-update steps can be decoupled for each unit and executed in parallel after we rearranging and grouping the variables and constraints of the UC mixed integer quadratic programming (MIQP) model according to each unit. The z-update steps, which couple all units because of system constraints, were decoupled for each unit by using an inner consensus ADMM to solve their Lagrangian dual problems. The proposed method can be implemented in master-slave distributed and parallel schema. The master node can be deployed in regional center; other computing nodes can be deployed (or installed) on each unit. Each unit keeps its information secret and does its computations in parallel. The simulation results show that the proposed hierarchical ADMM can obtain high-quality solutions in reasonable times and keep the information of units secret. The results in parallel environment show that the proposed method can obtain good speedup and is suit for solving large-scale D-UC problems.**

***Index Terms*—ADMM, full-distributed unit commitment, hierarchical.**

NOMENCLATURE

Indices:

i	Index for unit.
t	Index for time period.

This work was supported by the Natural Science Foundation of China (51407037, 61661004, and 61462006), the Guangxi Natural Science Foundation (2016GXNSFDA380019, 2014GXNSFFA118001, 2014GXNSFBA118274, and 2014GXNSFAA118391), Guangxi Key Laboratory of Power System Optimization and Energy Technology Foundation (15-A-01-11), the Hunan Province 100 Talent Program Fund, and the 2015 Science and Technology Project of China Southern Power Grid (WYKJ00000027).

J.B. Jian is with the School of Electrical Engineering, Guangxi University, China, and also with the School of Mathematics and Information Science, Yulin Normal University Yulin, 537000, China. (e-mail: jianjb@gxu.edu.cn, http://jians.gxu.edu.cn).

C. Zhang is with the School of Electrical Engineering, Guangxi University, Nanning 530004, China. (e-mail: 842982771@qq.com).

L.F. Yang, the corresponding author and joint first authors, is with the School of Computer Electronics and Information, Guangxi University, Nanning 530004, China, and is also with the Guangxi Key Laboratory of Power System Optimization and Energy Technology, Guangxi University. (e-mail: ylf@gxu.edu.cn).

K. Meng is with School of Electrical and Information Engineering, The University of Sydney, NSW 2006, Australia. (e-mail: ke.meng@sydney.edu.au).

Y. Xu is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, 639789. (e-mail: eeyanxu@gmail.com).

Z.Y. Dong is with the School of Electrical Engineering and Telecommunications, The University of NSW, Sydney, NSW 2052, Australia, and also with China Southern Power Grid Research Institute, Guangzhou, 510000, China (emails: zydong@ieee.org).

Operator:

$\{\cdot\}^+$	$\max(0,\cdot)$. When this operator is used for vector, it is carried out element by element.
$\mathrm{bin}\{\cdot\}$	if $\cdot \in [0.5,+\infty)$, then $\mathrm{bin}\{\cdot\}=1$, else $\mathrm{bin}\{\cdot\}=0$. This operator can be used for vector.
$(\cdot)_{p:q}$	get sub vector of $(\cdot)$ containing elements indexed from p to q.

Constants:

N	Total number of units.
T	Total number of time periods.
$\alpha_i, \beta_i, \gamma_i$	Coefficients of the quadratic production cost function of unit i.
$C_{\mathrm{hot},i}$	Hot startup cost of unit i.
$C_{\mathrm{cold},i}$	Cold startup cost of unit i.
$\underline{T}_{\mathrm{on},i}$	Minimum up time of unit i.
$\underline{T}_{\mathrm{off},i}$	Minimum down time of unit i.
$T_{\mathrm{cold},i}$	Cold startup time of unit i.
$\overline{P}_i$	Maximum power output of unit i.
$\underline{P}_i$	Minimum power output of unit i.
$P_{\mathrm{D},t}$	System load demand in period t.
R_t	Spinning reserve requirement in period t.
$P_{\mathrm{up},i}$	Ramp up limit of unit i.
$P_{\mathrm{down},i}$	Ramp down limit of unit i.
$P_{\mathrm{start},i}$	Startup ramp limit of unit i.
$P_{\mathrm{shut},i}$	Shutdown ramp limit of unit i.
$u_{i,0}$	Initial commitment state of unit i (1 if it is online, 0 otherwise).
$T_{i,0}$	Number of periods unit i has been online (+) or offline (–) prior to the first period of the time span (end of period 0).
U_i	$\{\min[T, u_{i,0}(\underline{T}_{\mathrm{on},i} - T_{i,0})]\}^+$
L_i	$\{\min[T, (1-u_{i,0})(\underline{T}_{\mathrm{off},i} + T_{i,0})]\}^+$

Variables:

$u_{i,t}$	Schedule of unit i in period t, binary variable that is equal to 1 if unit i is online in period t and 0 otherwise.
$s_{i,t}$	Startup status of unit i in period t.
$P_{i,t}$	Power output of unit i in period t.
$S_{i,t}$	Startup cost of unit i in period t.

I. INTRODUCTION

THE unit commitment (UC) in power system is one of the most important optimization problem for both daily operation scheduling and planning studies from short term to long term[1]-[2]. In general, the UC problem is formulated as a mixed integer nonlinear programming problem [3]-[4] and comprises two tasks: one is determining the status (on/off) of

the generating units; the other is to dispatch the power distributing for committed units.

In recent years, as power grids grow, the mechanism of energy supply has been greatly changed due to electricity liberalization[5], the introduction of distributed and renewable energy. These changes bring new entities into the electricity market, including renewable generation owners and active customers equipped with self-owned generation resources and advanced metering infrastructures [6]. Generally, these entities want to maximize their own profits without disclosing their actual financial information to system operators. A compelling alternative to solve the aforementioned problem is to distribute, rather than centralize. New distributed operational frameworks will be needed, including distributed economic dispatch [7]-[8], distributed optimal power flow [9]-[10], and distributed UC (D-UC) [11]-[13]. This paper focuses on the fully distributed optimization approach for UC. Comparing to the centralized UC(C-UC), the D-UC has following advantages: (1) Due to the improvement of the system interconnection level, more and more complex factors, such as demand response [15], electrical vehicle [16]-[17], contingency analysis, etc., these has been introduced into UC. However, in C-UC manner, all detailed information need to be collected by center calculator. This manner needs high bandwidth communication infrastructures. In D-UC, all individual subsystems can be computed locally and no information needs to be exchanged with adjacent subsystems during the calculation; (2) In the D-UC, individual subsystems do not need to disclose the secret information to other subsystems and the central controller. Such as characteristics of own electrical power unit. All which are the prerequisite for establishing a fair energy market environment and more detailed modeling and more modest computational requirements; (3) The D-UC is more scalable and flexible with respect to system changes than C-UC. Especially, in the smart grid, the topologies of electric power and communication infrastructure are more dynamic; (4) The D-UC is more robust than C-UC. In C-UC, if the central controller goes offline, the entire optimization computing system will be paralyzed. However, the D-UC can be solved asynchronously via individual controller [14]. That is, even if a local computing node is paralyzed, other nodes can still work, and the whole system is running as usual.

To implement D-UC, some researchers investigate the market dynamics of UC conducted by generating companies only (called self-commitment). In this framework, the independent system operator (ISO) posts hourly energy prices calculated based on the load forecast. Generating companies then conduct UC for their assets as price-takers [23]-[24]. Additional iterations may occur to search for an equilibrium where all the load is served with minimal cost. The existence of such an equilibrium under certain conditions was established in [25] and investigated empirically in [26]. Simulation results in [27] found roughly 4% cost of anarchy for self-commitment, which is a greater cost for nowadays power industry. Challenges related to the nonconvexity, oscillation, high cost, and inefficiency of self-commitment were explored in [27]-[30]. In addition, a number of distributed or agent-based approaches have been investigated for implementing D-UC. Nagata et al. [31] have studied a method of UC scheduling in a decentralized power network based on a multi-agent system. Although they deal with the UC, the decision is made by a mobile agent and the previously fixed order of the generators. Ramachandran et al. [32] presented the application of a hybrid optimization algorithm for the distributed energy resource management of a smart grid in the energy market. Pantoja and Quijano [33] proposed a replicator dynamics strategy for dynamic resource allocation. Kumar Nunna and Doolla [34] discussed distributed energy resource management for multiple microgrids using multi-agent systems. However, in the last three studies, the statuses of units are not considered.

In this paper we wish to investigate new approaches to the UC problem which are completely distributed/decentralized, i.e., most of computations can be done in parallel in each node, and all participators can keep their device characteristics secret. The solution that we propose is nearly based upon a unit-based (individual) optimization process, where a limited amount of information is exchanged between master-node and other slaver-nodes. The solution is based on the alternating direction method of multipliers (ADMM) in the form revisited by [35]. The ADMM is well suited to distributed convex optimization, and in particular to large-scale problems. The method was developed in the 1970s, with roots in the 1950s [36]. It can be viewed as an attempt to blend the benefits of dual decomposition and augmented Lagrangian methods for constrained optimization. The ADMM is readily applicable in many problems, such as in statistical learning, engineering design, or multi-period portfolio optimization. And it is interesting to note that with no tuning, ADMM can be competitive with the best known methods for some problems. After appropriately arranging the variables and constraints of the UC mixed integer quadratic programming (MIQP) model, we at first employ ADMM to solve the rearranged UC model. And two steps of scaled form of ADMM (x-update steps and v-update steps) can be decoupled for each unit and executed in parallel, but the z-update steps couple all units because of system constraints. Then we employ consensus ADMM to solve the Lagrangian dual problem of z-update step, which lets the z-update step to be decoupled for each unit.

The major contributions of this paper are summarized as follows:

1) We propose a fully distributed method to solve UC problem. The proposed method can be implemented in master-slave distributed and parallel schema. The master-node can be deployed in regional center; other computing nodes can be deployed (or installed) on each unit. Most computations for each unit can be done in each node in parallel.
2) We decouple all sub-problems for each unit. This means that, the parameters of each unit can be kept in only one computing node for the purpose of privacy protection or competition.
3) The simulation results show that the proposed hierarchical ADMM (H-ADMM) can obtain high-quality solutions in reasonable times and suit for solving large-scale D-UC problems.

The rest of this paper is organized as follow. Section II introduces the projected 2-bin formulation for UC. Section III discusses the proposed fully discusses UC approaches. Numerical results and analysis are presented in Section IV. Fi-

nally, the conclusions of our study are presented in Section V.

II. FORMULATION FOR UC

Given that the operational costs of thermal units is commonly represented as a quadratic function of the generation level, that is $f_i(P_{i,t}) = \alpha_i u_{i,t} + \beta_i P_{i,t} + \gamma_i (P_{i,t})^2$, the UC problem can be simply formulated as a mixed integer quadratic programming (MIQP) [37]. And after researching more than 5 decades, mixed integer programming (MIP) formulations of UC receive notable attention and obtain remarkable improvements. Many reported models, some of which are approximation models, have been developed for describing and solving the UC problem efficiently [3], [38]-[43].

[44] recently proposed a tight and compact MIP formulation for UC problem based on projection and reformulations by only using two sets of binary variables. In this paper, we will use ADMM to solve this UC formulation fully distributed, and we briefly introduce it as follows,

Let $\tilde{S}_{i,t}$ representing the part of startup cost exceeding $C_{\text{hot},i}$,

$$P_{i,t} = \tilde{P}_{i,t}(\overline{P}_i - \underline{P}_i) + u_{i,t}\underline{P}_i,\ \tilde{P}_{i,t} \in [0,1], \tag{1}$$

$$\tilde{\alpha}_i = \alpha_i + \beta_i \underline{P}_i + \gamma_i (\underline{P}_i)^2, \tag{2}$$

$$\tilde{\beta}_i = (\overline{P}_i - \underline{P}_i)(\beta_i + 2\gamma_i \underline{P}_i), \tag{3}$$

$$\tilde{\gamma}_i = \gamma_i (\overline{P}_i - \underline{P}_i)^2, \tag{4}$$

$$\tilde{P}_{\text{up},i} = \frac{P_{\text{up},i}}{\overline{P}_i - \underline{P}_i}, \tag{5}$$

$$\tilde{P}_{\text{down},i} = \frac{P_{\text{down},i}}{\overline{P}_i - \underline{P}_i}, \tag{6}$$

$$\tilde{P}_{\text{start},i} = \frac{P_{\text{start},i} - \underline{P}_i}{\overline{P}_i - \underline{P}_i}, \tag{7}$$

$$\tilde{P}_{\text{shut},i} = \frac{P_{\text{shut},i} - \underline{P}_i}{\overline{P}_i - \underline{P}_i} \tag{8}$$

The objective of the UC problem is to minimize the total operation cost,

$$\min F_{\text{C}} = \sum_{i=1}^{N} \sum_{t=1}^{T} \left[f_i(\tilde{P}_{i,t}) + C_{\text{hot},i} s_{i,t} + \tilde{S}_{i,t} \right], \tag{9}$$

where F_{C} is the total operation cost, the projected production cost $f_i(\tilde{P}_{i,t}) = \tilde{\alpha}_i u_{i,t} + \tilde{\beta}_i \tilde{P}_{i,t} + \tilde{\gamma}_i (\tilde{P}_{i,t})^2$, and $\tilde{S}_{i,t}$ can be formed as an MILP formulation:

$$\tilde{S}_{i,t} \geq (C_{\text{cold},i} - C_{\text{hot},i}) \left[s_{i,t} - \sum_{\tau=\max\{t - \underline{T}_{\text{off},i} - T_{\text{cold},i} - 1, 1\}}^{t-1} u_{i,\tau} - f_{\text{init},i,t} \right]. \tag{10}$$

$$\tilde{S}_{i,t} \geq 0 \tag{11}$$

where $f_{\text{init},i,t} = 1$ when $t - \underline{T}_{\text{off},i} - T_{\text{cold},i} - 1 \leq 0$ and $\{-T_{i,0}\}^+ < |t - \underline{T}_{\text{off},i} - T_{\text{cold},i} - 1| + 1$, otherwise $f_{\text{init},i,t} = 0$.

According to the formulation proposed in [44], the constraints of the UC formulation are:

1) Unit generation limits:

$$\tilde{P}_{i,t} \leq u_{i,t}, \tag{12}$$

$$\tilde{P}_{i,t} \geq 0. \tag{13}$$

2) Power balance constraint:

$$\sum_{i=1}^{N} \left[\tilde{P}_{i,t} (\overline{P}_i - \underline{P}_i) + u_{i,t} \underline{P}_i \right] - P_{\text{D},t} = 0. \tag{14}$$

3) System spinning reserve requirement:

$$\sum_{i=1}^{N} u_{i,t} \overline{P}_i \geq P_{\text{D},t} + R_t. \tag{15}$$

4) Ramp rate limits:

$$\tilde{P}_{i,t} - \tilde{P}_{i,t-1} \leq u_{i,t} \tilde{P}_{\text{up},i} + s_{i,t} (\tilde{P}_{\text{start},i} - \tilde{P}_{\text{up},i}), \tag{16}$$

$$\tilde{P}_{i,t-1} - \tilde{P}_{i,t} \leq u_{i,t-1} \tilde{P}_{\text{shut},i} + (s_{i,t} - u_{i,t}) (\tilde{P}_{\text{shut},i} - \tilde{P}_{\text{down},i}). \tag{17}$$

5) Minimum up/down time constraints:

$$\sum_{\varpi=\{t-\underline{T}_{\text{on},i}\}^+ + 1}^{t} s_{i,\varpi} \leq u_{i,t},\ t \in [U_i + 1, \dots, T], \tag{18}$$

$$\sum_{\varpi=\{t-\underline{T}_{\text{off},i}\}^+ + 1}^{t} s_{i,\varpi} \leq 1 - u_{i,\{t-\underline{T}_{\text{off},i}\}^+},\ t \in [L_i + 1, \dots, T], \tag{19}$$

6) Initial status of units:

$$u_{i,t} = u_{i,0},\ t \in [1, \dots, U_i + L_i]. \tag{20}$$

7) State constraints:

$$u_{i,t} - u_{i,t-1} \leq s_{i,t}. \tag{21}$$

Then, the UC MIQP formulation is:

$$\min F_{\text{C}} = \sum_{i=1}^{N} \sum_{t=1}^{T} \left[u_{i,t} f_i(\tilde{P}_{i,t}) + C_{\text{hot},i} s_{i,t} + \tilde{S}_{i,t} \right]$$
$$s.\text{t.} \begin{cases} (10) - (21) \\ u_{i,t}, s_{i,t} \in \{0,1\} \end{cases} \tag{22}$$

III. H-ADMM FOR FULLY DISTRIBUTED UC

A. Reformulation of UC Model for ADMM

For convenience, we borrow two symbols in Matlab language, comma ",", as row element separator, and semicolon ";" as array row separator.

Let $u_{i\cdot} = [u_{i,1}; u_{i,2}; \dots; u_{i,T}]$, $\tilde{S}_{i\cdot} = [\tilde{S}_{i,1}; \tilde{S}_{i,2}; \dots; \tilde{S}_{i,T}]$, $\tilde{P}_{i\cdot} = [\tilde{P}_{i,1}; \tilde{P}_{i,2}; \dots; \tilde{P}_{i,T}]$, $s_{i\cdot} = [s_{i,1}; s_{i,2}; \dots; s_{i,T}]^{\text{T}}$, $x_{i\cdot} = [u_{i\cdot}; s_{i\cdot}; \tilde{P}_{i\cdot}; \tilde{S}_{i\cdot}]$, $x = [x_{1\cdot}; x_{2\cdot}; \dots; x_{N\cdot}]$. And let $f(x) = \sum_{i=1}^{N} f_{i\cdot}(x_{i\cdot})$, where $f_{i\cdot}(x_{i\cdot}) = \sum_{t=1}^{T} \left[f_i(\tilde{P}_{i,t}) + C_{\text{hot},i} s_{i,t} + \tilde{S}_{i,t} \right]$.

We define set

$$\chi_1 = \{x | (10)(12)(16)(17)(18)(19)(20)(21)\}. \tag{23}$$

It is obvious that all the constraints in χ_1 can be decoupled according to each unit. So χ_1 can be rewritten as

$$\chi_1 = \left\{ x \left| \begin{bmatrix} A_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & A_N \end{bmatrix} \begin{bmatrix} x_{1\cdot} \\ \vdots \\ x_{N\cdot} \end{bmatrix} - \begin{bmatrix} b_1 \\ \vdots \\ b_N \end{bmatrix} \leq 0 \right. \right\}, \tag{24}$$

where A_i and b_i $(i = 1, \dots, N)$ are corresponding coefficient matrix and constant vector for unit i, respectively.

Similarly, we define set

$$\chi_2 = \{x | (14)(15)\}, \tag{25}$$

then the constraints in χ_2 can be decoupled according to each time period. So χ_2 can be rewritten as

$$\chi_2 = \left\{ x \left| \begin{matrix} \tilde{B}x - \tilde{c} = 0 \\ Bx - c \leq 0 \end{matrix} \right. \right\} \tag{26}$$

where B and $\tilde{B}$ are corresponding coefficient matrix, c and $\tilde{c}$ are corresponding constant vectors, and $B = [B_1, B_2, \dots, B_N]$, $\tilde{B} = [\tilde{B}_1, \tilde{B}_2, \dots, \tilde{B}_N]$.

We define the third set as

$$\chi_3 = \{x | u_{i,t} \in \{0,1\}, s_{i,t} \in \{0,1\}, \tilde{P}_{i,t} \geq 0, \tilde{S}_{i,t} \geq 0\}. \tag{27}$$

Then UC model (22) can be reformulated as

$$\min f(x)$$
$$\text{s.t. } x \in \chi_1 \cap \chi_2 \cap \chi_3 \tag{28}$$

Here it is noticeable that, in our UC model (28), all the information of units are totally included in $f(x)$, A_i, b_i, B, $\tilde{B}$, and x. And for each i, $x_{i\cdot}$, $f_{i\cdot}(x_{i\cdot})$, A_i, b_i, $\tilde{B}_i$, and B_i only include the parameters and variables of unit i.

B. ADMM for UC

ADMM was originally introduced for solving convex problems[36], but practical evidence suggests that it can be an effective method to approximately solve some nonconvex problems as well [45]. ADMM in literature has been well used to solve linearly constrained problem whose objective function is separable into two individual convex functions with non-overlapping variables [35], [46]-[47]. In order to use

ADMM, by introducing vector variable z, we rewrite problem (28) in ADMM form as

$$\begin{aligned} &\min \mathcal{F}(x) + I_{\chi_2}(z) \\ &\text{s.t. } x - z = 0, \end{aligned} \tag{29}$$

where $\mathcal{F}(x) = f(x) + I_{\chi_1 \cap \chi_3}(x)$ and I_χ denotes the indicator function of set χ, defined as,

$$I_\chi(x) = \begin{cases} 0 & \text{if } x \in \chi \\ +\infty & \text{if } x \notin \chi \end{cases}. \tag{30}$$

The corresponding augmented Lagrangian function of (29) is:

$$\begin{aligned} L^1_{\rho_1}(x, z, y) = f(x) + I_{\chi_1 \cap \chi_3}(x) + I_{\chi_2}(z) \\ + y^{\mathrm{T}}(x - z) + \frac{\rho_1}{2}\|x - z\|_2^2 \end{aligned} \tag{31}$$

where ρ_1 is the penalty parameter of the augmented term and $y = [y_1.; y_2.; \dots; y_N.]$, $y_i. = [y_{i,1}; y_{i,2}; \dots; y_{i,T}]$, $i = 1, \dots, N$. This vector y is similar to x in structure. y is a vector of the Lagrangian variables.

Using ADMM to solve problem (29), the iterations are as follows:

$$x^{k+1} = \operatorname{argmin}_x L^1_{\rho_1}(x, z^k, y^k) \tag{32}$$

$$z^{k+1} = \operatorname{argmin}_z L^1_{\rho_1}(x^{k+1}, z, y^k) \tag{33}$$

$$y^{k+1} = y^k + \rho_1(x^{k+1} - z^{k+1}) \tag{34}$$

Obviously, the ADMM iterations can be reduced as:

$$x^{k+1} = \operatorname{argmin}_x \left\{ f(x) + I_{\chi_1 \cap \chi_3}(x) + \frac{\rho_1}{2}\|x - z^k + v^k\|_2^2 \right\} \tag{35}$$

$$z^{k+1} = \operatorname{argmin}_z \left\{ I_{\chi_2}(z) + \frac{\rho_1}{2}\|x^{k+1} - z + v^k\|_2^2 \right\} \tag{36}$$

$$v^{k+1} = v^k + (x^{k+1} - z^{k+1}) \tag{37}$$

where $v^k = \frac{1}{\rho_1} y^k$.

Obviously, v-update steps are totally decoupled, so these updates can be carried out in parallel for each unit, and they can be decoupled according to each period, even for each scale variable as well. We will analyze the solving of update steps (35)-(36) in following subsections.

C. *x-update Step* (35)

The x-update (35) can be rewritten as

$$\begin{aligned} &\min \left\{ f(x) + \frac{\rho_1}{2}\|x - z^k + v^k\|_2^2 \right\} \\ &\text{s.t. } x \in \chi_1 \cap \chi_3 \end{aligned} \tag{38}$$

Observing that (38) is totally decoupled according to unit, then (38) is equivalent to solving the following sub-problems for $i = 1, \dots, N$.

$$\begin{aligned} &\min \left\{ f_i.(x_i.) + \frac{\rho_1}{2}\|x_i. - z_i^k + v_i^k\|_2^2 \right\} \\ &\text{s.t.} \begin{cases} A_i x_i. - b_i \le 0 \\ u_i. \in \{0,1\}^T, s_i. \in \{0,1\}^T \\ \tilde{P}_i. \ge 0, \tilde{S}_i. \ge 0 \end{cases} \end{aligned} \tag{39}$$

(39) is an MIQP and can be carried out independently in parallel for each $i = 1, \dots, N$. Each sub-problem (39) only needs parameters of unit i. And we note that (39) always is a small and scale invariant problem, with only $4 \times T$ variables, for various scale systems.

D. *z-update Step* (36)

z-update step (36) can be rewritten as

$$\begin{aligned} &\min \frac{\rho_1}{2}\|z - x^{k+1} - v^k\|_2^2 \\ &\text{s.t. } z \in \chi_2 \end{aligned} \tag{40}$$

Sub-problem (40) is a QP problem. Its objective function is totally decoupled according to each variable element of z and its constraints can be decoupled according to each period t. So (40) can be decoupled as T sub-problems, and each one for each period. However, (40) and these sub-problems couple all units (See (14) and (15)). In order to implement totally distributed UC solving, we must decoupled all sub problems according to each unit.

In what follows, taking strong duality under consideration [48]-[49], we will use consensus ADMM to solve the Lagrangian dual problem of (40), then all the sub-problems are decoupled according to each unit.

(40) has the same solutions with following problem

$$\begin{aligned} &\min \|z - r\|_2^2 \\ &\text{s.t.} \begin{cases} \tilde{B}z - \tilde{c} = 0 \\ Bz - c \le 0 \end{cases} \end{aligned} \tag{41}$$

where $r = x^{k+1} + v^k$, $\tilde{B}z - \tilde{c} = 0$ is power balance constraint and $Bz - c \le 0$ is system spinning reserve requirement constraint. And, according to the practical significance of these constraints, the corresponding coefficient matrix $\tilde{B}$ and B are row full rank. Thus, Slater condition holds for the problem (41).

The Lagrangian function of (41) is

$$L(z, \lambda, \tilde{\lambda}) = \|z - r\|_2^2 + \lambda^{\mathrm{T}}(Bz - c) + \tilde{\lambda}^{\mathrm{T}}(\tilde{B}z - \tilde{c}) \tag{42}$$

where λ is Lagrangian multiplier vector associated with inequality constraints, and similarly $\tilde{\lambda}$ is Lagrangian multiplier vector associated with equality constraints.

Then the Lagrange dual function of (41) is

$$\begin{aligned} g(\lambda, \tilde{\lambda}) = \inf_z L(z, \lambda, \tilde{\lambda}) = \inf_z \{ \|z - r\|_2^2 + \lambda^{\mathrm{T}}(Bz - c) + \\ \tilde{\lambda}^{\mathrm{T}}(\tilde{B}z - \tilde{c}) \} \end{aligned} \tag{43}$$

Since $L(z, \lambda, \tilde{\lambda})$ is a convex quadratic function of z, we can find the minimizing z from the optimality condition

$$\nabla_z L(z, \lambda, \tilde{\lambda}) = 2(z - r) + B^{\mathrm{T}}\lambda + \tilde{B}^{\mathrm{T}}\tilde{\lambda} = 0$$

which yields

$$z = r - \frac{1}{2}[B^{\mathrm{T}}, \tilde{B}^{\mathrm{T}}] \begin{bmatrix} \lambda \\ \tilde{\lambda} \end{bmatrix} \tag{44}$$

Therefore the dual function is

$$\begin{aligned} &g(\lambda, \tilde{\lambda}) = L\left(r - \frac{1}{2}[B^{\mathrm{T}}, \tilde{B}^{\mathrm{T}}] \begin{bmatrix} \lambda \\ \tilde{\lambda} \end{bmatrix}, \lambda, \tilde{\lambda}\right) = \\ &-\frac{1}{4}\left\| [B^{\mathrm{T}}, \tilde{B}^{\mathrm{T}}] \begin{bmatrix} \lambda \\ \tilde{\lambda} \end{bmatrix} \right\|_2^2 + r^{\mathrm{T}}[B^{\mathrm{T}}, \tilde{B}^{\mathrm{T}}] \begin{bmatrix} \lambda \\ \tilde{\lambda} \end{bmatrix} - [c^{\mathrm{T}}, \tilde{c}^{\mathrm{T}}] \begin{bmatrix} \lambda \\ \tilde{\lambda} \end{bmatrix} \end{aligned} \tag{45}$$

Then the Lagrange dual problem associated with the problem (41) is

$$\begin{aligned} &\max g(\lambda, \tilde{\lambda}) \\ &\text{s.t. } \lambda \ge 0 \end{aligned} \tag{46}$$

That is a QP problem equivalent to

$$\begin{aligned} &\min \frac{1}{4}\left\| [B^{\mathrm{T}}, \tilde{B}^{\mathrm{T}}] \begin{bmatrix} \lambda \\ \tilde{\lambda} \end{bmatrix} \right\|_2^2 - r^{\mathrm{T}}[B^{\mathrm{T}}, \tilde{B}^{\mathrm{T}}] \begin{bmatrix} \lambda \\ \tilde{\lambda} \end{bmatrix} + [c^{\mathrm{T}}, \tilde{c}^{\mathrm{T}}] \begin{bmatrix} \lambda \\ \tilde{\lambda} \end{bmatrix} \\ &\text{s.t. } \lambda \ge 0 \end{aligned} \tag{47}$$

Note the structure of $\tilde{B}$ and B, let $\hat{x} = \begin{bmatrix} \lambda \\ \tilde{\lambda} \end{bmatrix}$, and let

$$f_i^B(\hat{x}) = \frac{1}{4}\left\| [B_i^{\mathrm{T}}, \tilde{B}_i^{\mathrm{T}}] \begin{bmatrix} \lambda \\ \tilde{\lambda} \end{bmatrix} \right\|_2^2 - r_i^{\mathrm{T}}[B_i^{\mathrm{T}}, \tilde{B}_i^{\mathrm{T}}] \begin{bmatrix} \lambda \\ \tilde{\lambda} \end{bmatrix} + \frac{1}{N}[c^{\mathrm{T}}, \tilde{c}^{\mathrm{T}}] \begin{bmatrix} \lambda \\ \tilde{\lambda} \end{bmatrix} \tag{48}$$

Then (47) can be rewritten as

$$\begin{aligned} &\min \textstyle\sum_{i=1}^N f_i^B(\hat{x}) \\ &\text{s.t. } \lambda \ge 0. \end{aligned} \tag{49}$$

Let $\chi_\lambda = \{(\lambda, \tilde{\lambda}) | \lambda \ge 0\}$, we rewrite problem (49) in consensus ADMM form[35] as

$$\begin{aligned} &\min \{ \textstyle\sum_{i=1}^N f_i^B(\hat{x}_i) + I_{\chi_\lambda}(\hat{z}) \} \\ &\text{s.t. } \hat{x}_i = \hat{z}, i = 1, \dots, N \end{aligned} \tag{50}$$

The corresponding augmented Lagrangian function of (50) is:

$$L_{\rho_2}(\hat{x}_1,\dots,\hat{x}_N,\hat{z},\hat{y}_1,\dots,\hat{y}_N)=\sum_{i=1}^{N}\left[f_i^B(\hat{x}_i)+\hat{y}_i^{\mathrm{T}}(\hat{x}_i-\hat{z})+\frac{\rho_2}{2}\|\hat{x}_i-\hat{z}\|_2^2\right]+I_{\chi_\lambda}(\hat{z}) \quad (51)$$

where ρ_2 is the penalty parameter.

To solve problem(50), then scaled form of consensus ADMM consists of the iterations:

$$\hat{x}_i^{\hat{k}+1}=\operatorname{argmin}_{\hat{x}_i}\left\{f_i^B(\hat{x}_i)+\frac{\rho_2}{2}\left\|\hat{x}_i-\hat{z}^{\hat{k}}+\hat{v}_i^{\hat{k}}\right\|_2^2\right\} \quad (52)$$

$$\hat{z}^{\hat{k}+1}=\operatorname{argmin}_{\hat{z}}\left\{I_{\chi_\lambda}(\hat{z})+\frac{N\rho_2}{2}\left\|\hat{z}-(\bar{x}^{\hat{k}+1}+\bar{v}^{\hat{k}})\right\|_2^2\right\} \quad (53)$$

$$\hat{v}_i^{\hat{k}+1}=\hat{v}_i^{\hat{k}}+\hat{x}_i^{\hat{k}+1}-\hat{z}^{\hat{k}+1} \quad (54)$$

where $\hat{v}_i^{\hat{k}}=\frac{1}{\rho_2}\hat{y}_i^{\hat{k}}$, and with the average (over $i=1,\dots,N$) of a vector denoted with an overline, i.e., $\bar{x}^{\hat{k}+1}=\frac{1}{N}\sum_{i=1}^{N}\hat{x}_i^{\hat{k}+1}$, $\bar{v}^{\hat{k}}=\frac{1}{N}\sum_{i=1}^{N}\hat{v}_i^{\hat{k}}$.

Now, we can see that $\hat{x}$-update (52) and $\hat{v}$-update (54) are decoupled, and can be carried out in parallel for each unit.

The $\hat{z}$-update can be rewritten as

$$\hat{z}^{\hat{k}+1}=\Pi_{\chi_\lambda}(\bar{x}^{\hat{k}+1}+\bar{v}^{\hat{k}}) \quad (55)$$

According to the definition of χ_λ, then z-update can be

$$\hat{z}^{\hat{k}+1}=\begin{bmatrix}\left\{\left(\bar{x}^{\hat{k}+1}+\bar{v}^{\hat{k}}\right)_{1:T}\right\}^{+}\\ \left(\bar{x}^{\hat{k}+1}+\bar{v}^{\hat{k}}\right)_{T+1:2T}\end{bmatrix} \quad (56)$$

The $\hat{x}$-update problem (52) is minimizing a convex quadratic function of $\hat{x}_i$, we can find the minimizing $\hat{x}_i^{\hat{k}+1}$ from the optimality condition

$$\frac{1}{2}\begin{bmatrix}B_i\\ \tilde{B}_i\end{bmatrix}\left[B_i^{\mathrm{T}},\tilde{B}_i^{\mathrm{T}}\right]\hat{x}_i^{\hat{k}+1}-\begin{bmatrix}B_i\\ \tilde{B}_i\end{bmatrix}r_i+\frac{1}{N}\begin{bmatrix}c\\ \tilde{c}\end{bmatrix}+\rho_2\left(\hat{x}_i^{\hat{k}+1}-\hat{z}^{\hat{k}}+\hat{v}_i^{\hat{k}}\right)=0. \quad (57)$$

Then, we have

$$\hat{x}_i^{\hat{k}+1}=\left(\frac{1}{2}\begin{bmatrix}B_i\\ \tilde{B}_i\end{bmatrix}\left[B_i^{\mathrm{T}},\tilde{B}_i^{\mathrm{T}}\right]+\rho_2 I\right)^{-1}\left(\rho_2\hat{z}^{\hat{k}}-\rho_2\hat{v}_i^{\hat{k}}+\begin{bmatrix}B_i\\ \tilde{B}_i\end{bmatrix}r_i-\frac{1}{N}\begin{bmatrix}c\\ \tilde{c}\end{bmatrix}\right). \quad (58)$$

We should note that, when a direct method, i.e. matrix inversion, is used to solve (58), one matrix inversion and one matrix multiplication are needed at the first iteration, and only one matrix multiplication is needed in the following iterations, the algorithm calculation is very small, So the $\hat{x}$-update is very easy to execute and quickly solve.

In addition, the variables of the sub-problem (40) are only related to u and $\tilde{F}$. Therefore, when we solve the sub-problem (40), you cannot consider its. So that the number of the columns of coefficient matrix $\tilde{B}$ and B are reduced by half, and, the Lagrange dual problem (47) variables $\hat{x}$ are also reduced by half. This way can improve the efficiency of the ADMM algorithm while reducing the dual variables.

After using consensus ADMM iterations to solve(46), denote $\begin{bmatrix}\lambda\\ \tilde{\lambda}\end{bmatrix}=\hat{z}^{\hat{k}+1}$ as the solution. Then according to strong duality and (44), we can obtain the solution of (41), i.e., z-update step (41) or (36) have been solved. In addition, (44) can be decoupled for unit i as

$$z_i=r_i-\frac{1}{2}\left[B_i^{\mathrm{T}},\tilde{B}_i^{\mathrm{T}}\right]\hat{z}^{\hat{k}+1} \quad (59)$$

To now, we decouple all sub-problems for each unit. This means that, the parameters of each unit can be kept in only one computing node (can be installed on each corresponding unit) for the purpose of privacy protection or competition.

E. Description of the Proposed H-ADMM

We are now in a position to give the H-ADMM for Distributed UC.

Algorithm 1 H-ADMM for Distributed UC

Initialization: $x_{\text{best}}:=\emptyset$, $f_{x_{\text{best}}}:=+\infty$, $M_1>0$, $M_2>0$, $\rho_1>0$, $\rho_2>0$, $\varepsilon^{\text{feasible}}>0$, $\eta\in[0,1]$.
for iteration 0,1,…, M_1
  $v^0=0$, set $z^0=0$ for first iteration and randomly generate $z^0\in\chi_3$ for other iterations.
  for $k=0,1,\dots,M_2$
    x-update:
      for each unit $i=1,\dots,N$:
        get x_i^{k+1} by solving (39)
      end
    z-update:
      $r=x^{k+1}+v^k$.
      $[z^{k+1}]$=sub_DADMM_z(r,ρ_2)
    v-update: (37)
    if $f(x^{k+1})\le f_{x_{\text{best}}}$ and $\varepsilon^{\text{c}}=\frac{\|x^{k+1}-z^{k+1}\|_2}{N}\le\varepsilon^{\text{feasible}}$
      $x_{\text{best}}=x^{k+1}$, $f_{x_{\text{best}}}=f(x^{k+1})$.
    end if
    if $\varepsilon^{\text{c}}\le\eta\varepsilon^{\text{feasible}}$
      break.
    end if
  end for
end for
return x_{best}.

Function 1 $[z]$= sub_DADMM_z1(r,ρ_2)

$v^0=0$, $z^0=0$, $\varepsilon^{\text{pri}}>0$, $\varepsilon^{\text{dual}}>0$.
for $\hat{k}=0,1,\dots,M_3$
  x-update:
    for each unit $i=1,\dots,N$:
      get $\hat{x}_i^{\hat{k}+1}$ from (58)
    end
  $\hat{z}$-update: (56)
  $\hat{v}$-update: (54)
  $p^{\text{r}}=\left[\left(\hat{x}_1^{\hat{k}+1}-\hat{z}^{\hat{k}+1}\right);\dots;\left(\hat{x}_N^{\hat{k}+1}-\hat{z}^{\hat{k}+1}\right)\right]$,
  $d^{\text{r}}=\rho_2\left[\left(\hat{z}^{\hat{k}}-\hat{z}^{\hat{k}+1}\right);\dots;\left(\hat{z}^{\hat{k}}-\hat{z}^{\hat{k}+1}\right)\right]$,
  $\varepsilon^{\text{pri}}=\sqrt{2NT}\varepsilon^{\text{abs}}+\varepsilon^{\text{rel}}\max\left\{\left\|\left(\hat{x}_1^{\hat{k}+1};\dots;\hat{x}_N^{\hat{k}+1}\right)\right\|_2,\sqrt{N}\left\|\hat{z}^{\hat{k}+1}\right\|_2\right\}$
  $\varepsilon^{\text{dual}}=\sqrt{2NT}\varepsilon^{\text{abs}}+\varepsilon^{\text{rel}}\left\|\left(\rho_2\hat{v}_1^{\hat{k}+1};\dots;\rho_2\hat{v}_N^{\hat{k}+1}\right)\right\|_2$
  if $\|p^{\text{r}}\|_2\le\varepsilon^{\text{pri}}$ and $\|d^{\text{r}}\|_2\le\varepsilon^{\text{dual}}$ **break**.
end
for each unit $i=1,\dots,N$:
  (59)
end
return z.

According to the literature [50], Stephen Boyd proposed a simple effective heuristic for embedded mixed-integer quadratic programming which is based on ADMM. According to the algorithm optimization criteria of literature [50], we determine the criterion of iteration termination. Here, we give some explanations for our stop criterion in our H-ADMM. We note that,

in our algorithm, according to the definition of (39), we always have $x^{k+1} \in \chi_1 \cap \chi_3$, that means x^{k+1} is a solution which always meets all the unit physical constraints except two coupling constraints: power balance and spinning reserve constraints. So, while iterating, if $x^{k+1} \in \chi_2$, then $x^{k+1} - z^{k+1} = 0$, i.t., $\varepsilon^{\mathrm{c}} = \frac{\left\|x^{k+1} - z^{k+1}\right\|_2}{N} \le \varepsilon^{\mathrm{feasible}}$, when $\varepsilon^{\mathrm{feasible}}$ is small enough, the x^{k+1} must be a feasible UC solution. So ε^{c} can depicts the feasibility of iteration point x^{k+1}.

The detailed procedure of H-ADMM for fully distributed solving UC problem is illustrated in Fig. 1. As can be seen in this figure, the proposed method can be implemented with master-slave distributed and parallel schema. The main node can be deployed in regional center; other computing node can be deployed (or installed) on each unit. For each unit, for example for unit i, all the physical parameters of unit i (including $f_i(\cdot)$, A_i, b_i, B_i, $\tilde{B}_i$) are preserved in node i, and the other nodes don't know these parameters. Most computations for unit i are done in node i. The master node does not need any information of all units' physical parameters. And the major jobs of the master node include checking the convergent criterion (i.e., in Fig. 1, "⑨" is the convergent criterion for outer ADMM iterations, and "$" is the convergent criterion for inner ADMM iterations). According to Fig. 1, we point out that the proposed H-ADMM method can solve UC problem in fully distributed and parallel schema and all units can taking part in power market competition but keeping their information secret.

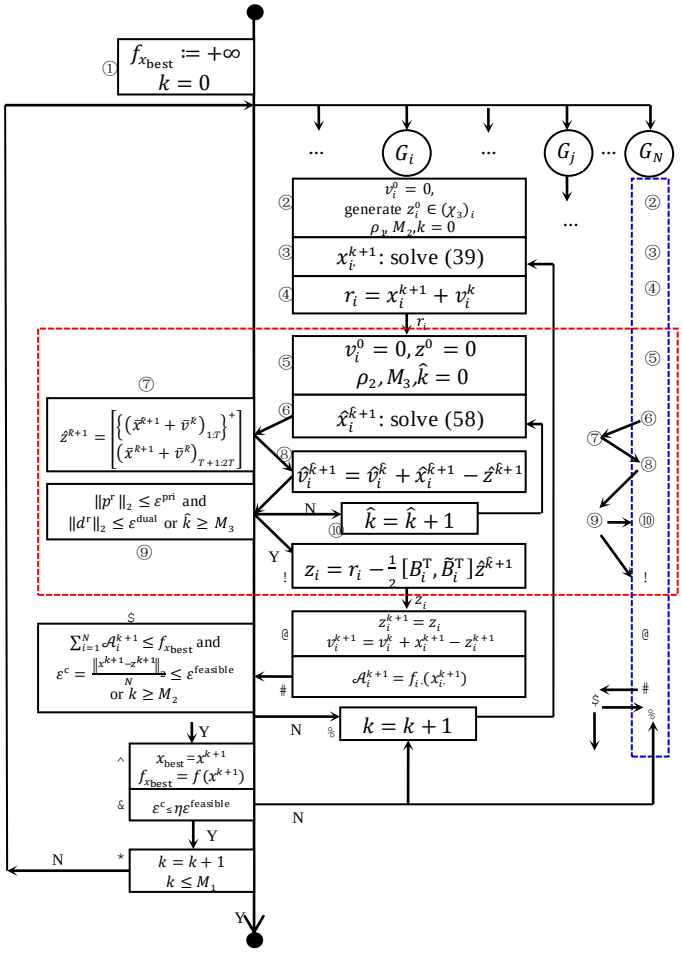

Fig. 1. Illustration of Algorithm 1

IV. Numerical Results and Analysis

In this section we present some numerical results to test the efficiency and effectiveness of the proposed H-ADMM method. The machine on which we perform all of our computations is an Intel i7-4790U 3.6GHz Lenovo desktop with 8 GB of RAM, running MS-Windows 7 and Matlab 2014a. CPLEX 12.6.2 is used to solve MIQP while setting the time limit 3600 s for solver in our numerical experiments. Unless otherwise specified, we used the parameter values $M_1 = 1$, $M_2 = 500$, $\rho_1 = 1.1 \times 10^4$, $\rho_2 = 4 \times 10^4$, $\varepsilon^{\mathrm{feasible}} = 0.01$, $\eta = 0.6$, $M_3 = 50$, $\varepsilon^{\mathrm{abs}} = 10^{-5}$, $\varepsilon^{\mathrm{rel}} = 10^{-4}$, $\varepsilon^{\mathrm{pri}} = \sqrt{2NT}\varepsilon^{\mathrm{abs}} + \varepsilon^{\mathrm{rel}} \max\left\{\left\|\left(\hat{x}_1^{\hat{k}+1}; \ldots; \hat{x}_N^{\hat{k}+1}\right)\right\|_2, \sqrt{N}\left\|\hat{z}^{\hat{k}+1}\right\|_2\right\}$, $\varepsilon^{\mathrm{dual}} = \sqrt{2NT}\varepsilon^{\mathrm{abs}} + \varepsilon^{\mathrm{rel}}\left\|\left(\rho_2\hat{v}_1^{\hat{k}+1}; \ldots; \rho_2\hat{v}_N^{\hat{k}+1}\right)\right\|_2$.

A. The Simulation Results for a 5-units instance

At first, a simple 5-units UC instance has been used to test our proposed H-ADMM method. We randomly chose 5 units (unit 1, 3, 5, 6, 8), including small, medium and large size units, from 10-units instance which can be found in [51] and hourly load demand is assumed to be a half of the corresponding part in original 10-units. And the ramp rate of each unit is selected as $P_{\mathrm{up},i} = P_{\mathrm{down},i} = 0.2\overline{P}_i$, $P_{\mathrm{start},i} = P_{\mathrm{shut},i} = \underline{P}_i$, $P_{i,0} = 0.7\overline{P}_i$.

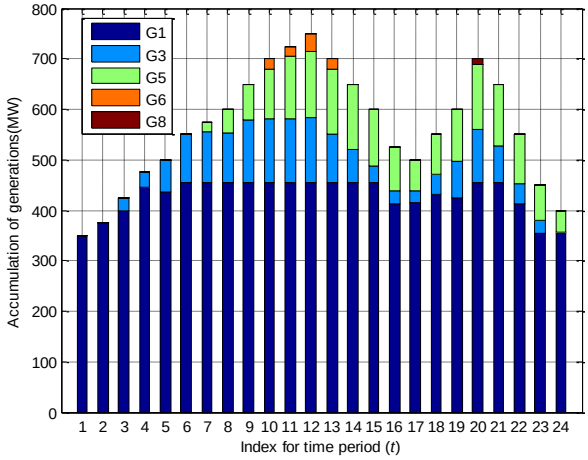

FIG. 2. THE GLOBAL SOLUTION

(22) is an MIQP, and CPLEX can solve small scale MIQP directly. The global solution found by CPLEX generates an objective value of $278937.44 and the objective value with our algorithm was $279872.83. Let $f(x)_{\mathrm{H\_ADMM}}$ denoting the objective value obtained by the proposed H-ADMM and $f(x)_{\mathrm{MIQP}}$ denoting the objective value returned by CPLEX while solving UC MIQP model. And let $p^{\mathrm{opt}} = \frac{f(x)_{\mathrm{H-ADMM}} - f(x)_{\mathrm{MIQP}}}{f(x)_{\mathrm{MIQP}}}$, then $p^{\mathrm{opt}} = 0.33\%$ for this 5-units system, i.e., the solution obtained by H-ADMM is 0.33% suboptimal. The optimal schedule reported by CPLEX is shown in Fig. 2, and the suboptimal schedule returned by H-ADMM is shown in Fig. 3. In these two figures, we see that qualitatively, the optimal schedule and the schedule generated by H-ADMM are very similar. Simulation results show that the proposed method can obtain high-quality solutions for D-UC problems.

All units of UC problem are coupled because of power balance and spinning reserve constraints. Our proposed method decouples these coupling constraints by using consensus

ADMM to solve the dual problem of outer ADMM z-update step. For 5-unit test system, the iterations of consensus ADMM of the third outer ADMM iteration are shown in Fig. 4, Fig. 5, and Fig. 6. Fig. 4 and Fig. 5 show the procedures of the primal and dual residuals converging to primal and dual feasibility tolerances, respectively. And Fig. 6 shows the objective values of (49) while iterating. As can be seen in these figures, after about 10 iterations, the objective value comes to be nearly stable (no more than 0.05% difference), and primal residuals converge after 17 iterations. The total number of iterations for this consensus ADMM z-update step is 25.

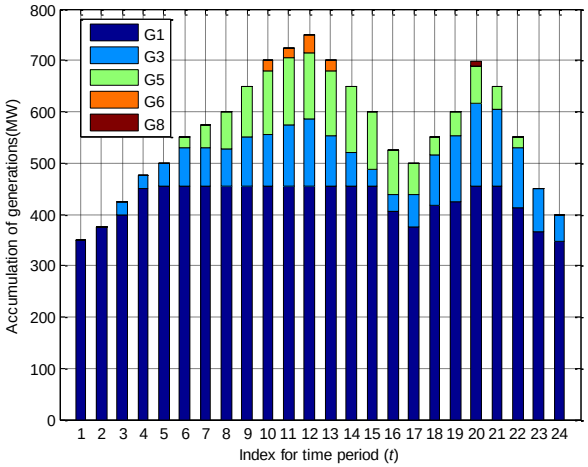

FIG. 3. THE SOLUTION FOUND BY USING H_ADMM

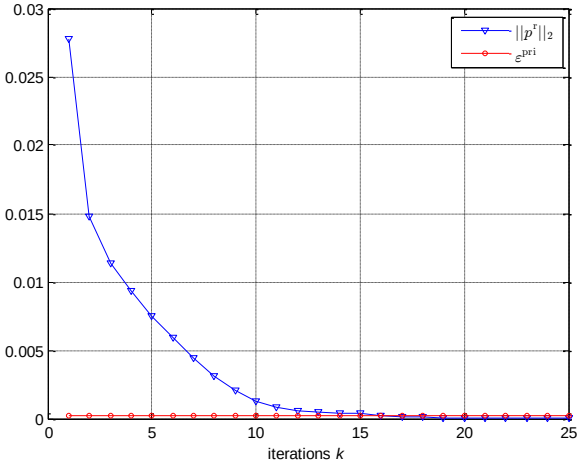

FIG. 4. ITERATIVE VALUES OF $\|p^r\|_2$ AND ε^{pri} FOR INNER ITERATIONS

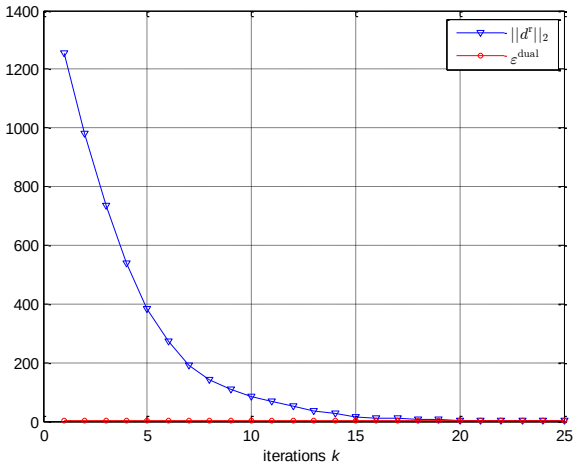

FIG. 5. ITERATIVE VALUES OF $\|d^r\|_2$ AND ε^{dual} FOR INNER ITERATIONS

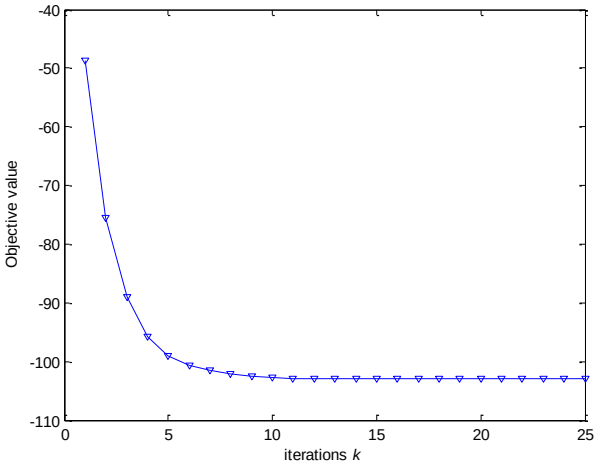

FIG. 6. ITERATIVE VALUES OBJECTIVE FUNCTION OF (49) FOR INNER ITERATIONS

As we point out at the paragraph following the description of sub function 1, the definition of (39) guarantees that $x^{k+1} \in \chi_1 \cap \chi_3$, i.e., x^{k+1} always meets all the unit physical constraints except power balance and spinning reserve constraints. As the method proceeds, the primal residual $r^{k+1} = \|x^{k+1} - z^{k+1}\|_2$ converges to zero. That means $x^{k+1} \in \chi_2$, then $x^{k+1} - z^{k+1} = 0$, i.e. $\varepsilon^c = \frac{\|x^{k+1} - z^{k+1}\|_2}{N} \leq \varepsilon^{feasible}$. When $\varepsilon^{feasible}$ is small enough, the x^{k+1} converges to the UC feasible solution. Values of ε^c in iterations are depicted in Fig. 7. This figure shows that the residual of coupling constraints can be efficiently decreased while outer ADMM iterations proceeding.

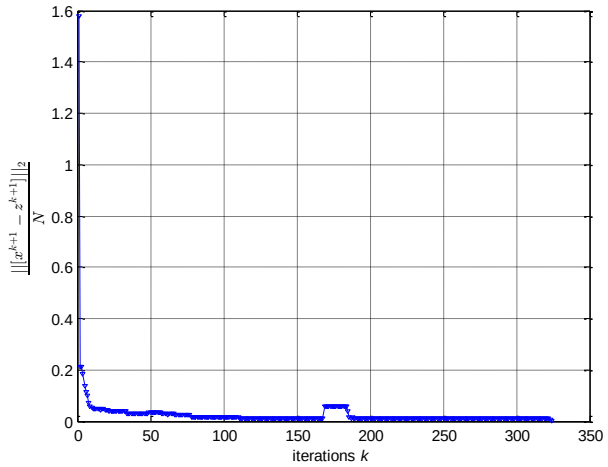

FIG. 7. ILLUSTRATION OF STOP CRITERION FOR OUTER ITERATIONS

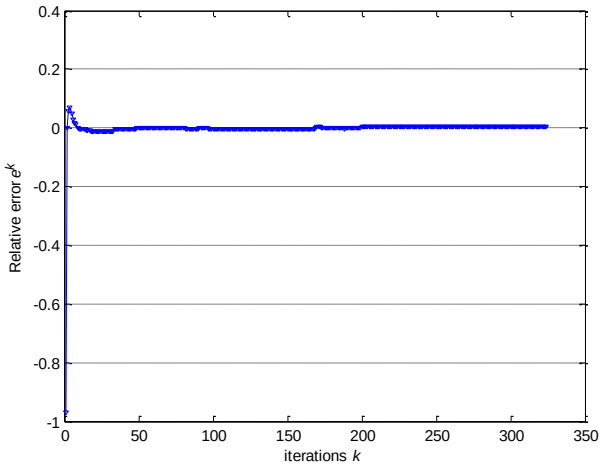

FIG. 8. ILLUSTRATION OF RELATIVE ERROR e^k FOR OUTER ITERATIONS

Suppose $f(x^k)_{\text{H\_ADMM}}$ is the result of value of objective function at the k^{th} iteration. The relative error e^k is defined as $e^k = \frac{f(x^k)_{\text{H-ADMM}} - f(x)_{\text{MIQP}}}{f(x)_{\text{MIQP}}}$. The convergence procedure is shown in Fig. 8. It is shown that, with a moderate number of iterations, the proposed algorithm converges. From Fig. 8, we can see that, after very few iterations, the values of e^k are very close to the "0", which means that the proposed algorithm can yield a good approximation solution in a very short time. Note that there are some points $e^k < 0$, that means the objective value $f(x^k)_{\text{H-ADMM}} < f(x)_{\text{MIQP}}$ for these points. With Fig. 7 and Fig. 8 under consideration simultaneously, at the very start, the very less generating costs are obtained because that these point cannot meet powers balance and spin reserve constraints (which can be seen in Fig. 7). Immediately, with the corrections of algorithm, the feasibility becomes better (i.e. decrease) and generating costs increase. This is due to the intermediate solutions starting to meet the powers balance and spin reserve constraints. In the previous iteration, the value of the algorithm objective function always swings near the CPLEX optimal value in Fig.8. And, from Fig.7 we can see that the curve has a section which is beating. These results are because that the H-ADMM algorithm is continually adjusting the states of the generators. After about 170 iterations, the curve begins to decrease steadily both in Fig.7 and Fig.8. This means the states of the generators in the whole system have been determined. And the following iterations of the algorithm will be used to adjust the generator's active power output.

B. *The Simulation Results for large-scale problems*

In order to present the effectiveness of the proposed H-ADMM for the solution of a relatively complicated and large power system, the proposed method is also tested on another 20 test instances. As presented in [41], an eight-unit data set is replicated to create 20 larger instances and different power-systems mixes. These unit mixes, are shown in Table I. The data of eight units and load demands can be found in [41] and [43]. Spinning reserves are set to be 3% of demand and $P_{i,0} = 0.7\overline{P}_i$. Creating test instances by replicating units introduces symmetry that can result in significantly harder than usual problems [41].

TABLE I
NUMBER OF UNITS PER PROBLEM CASE

No	Unit								Total Units	No.	Unit								Total Units
	1	2	3	4	5	6	7	8			1	2	3	4	5	6	7	8	
1	12	11	0	0	1	4	0	0	28	11	46	45	8	0	5	0	12	16	132
2	13	15	2	0	4	0	0	1	35	12	40	54	14	8	3	15	9	13	156
3	15	13	2	6	3	1	1	3	44	13	50	41	19	11	4	4	12	15	156
4	15	11	0	1	4	5	6	3	45	14	51	58	17	19	16	1	2	1	165
5	15	13	3	7	5	3	2	1	49	15	43	46	17	15	13	15	6	12	167
6	10	10	2	5	7	5	6	5	50	16	50	59	8	15	1	18	4	17	172
7	17	16	1	3	1	7	2	4	51	17	53	50	17	15	16	5	14	12	182
8	17	10	6	5	2	1	3	7	51	18	45	57	19	7	19	19	5	11	182
9	12	17	4	7	5	2	0	5	52	19	58	50	15	7	16	18	7	12	183
10	13	12	5	7	2	5	4	6	54	20	55	48	18	5	18	17	15	11	187

In order to show the quality of solutions obtained by the proposed method, we solve the UC MIQP model (22) directly by setting CPLEX with accuracy 0.5%. The results of CPLEX and the proposed method are listed in Table II. In this table, column "C_{time}" represents the execution time; column "Total cost $f(x)_{\text{MIQP}}$" and "Total cost $f(x)_{\text{H-ADMM}}$" represent the total cost return by CPLEX and the proposed H-ADMM, respectively; "Gap" represents the relative integrality gap reported by CPLEX at the end of solving process; column "p^{opt}" represents relative error of solution obtained by H-ADMM, denoted as $\frac{f(x)_{\text{H-ADMM}} - f(x)_{\text{MIQP}}}{f(x)_{\text{MIQP}}}$; column "$\varepsilon^{\text{c}}$" reports the value of ε^{c} for the H-ADMM solution; column "$N_{\text{it\_out}}$" reports the number of outer ADMM iterations; and column "$N_{\text{it\_in\_avg}}$" reports the average number of inner consensus ADMM itera-

tions.

In Table II, the columns under "MIQP-CPLEX" show that CPLEX can obtain better solutions (Gap less than 0.5%) for most test systems. Take the solutions obtained by CPLEX as references. As can be seen in Table II, H-ADMM can get high quality solutions for these test systems, and the relative errors are less than 1% and for all test systems. And for 12^{th} system, the solution obtained by H-ADMM is better than CPLEX. That is to say the proposed H-ADMM has "good convergence" for our experiments, i.e., it can converge to a point with good objective value. Next, look at the system running time, we can find that the solution time of CPLEX is increased obviously as the scale of the problem is increases. When the generator number reaches 180, the CPLEX solution time is more than 2 minutes. However, H-ADMM solves these large instances, in a serial environment, with no more than 30s. And in parallel environment with four workers, the solution time is less than 15s. The results in Table II show that, in the case of without the disclosure of privacy, our proposed H-ADMM algorithm can quickly obtain high quality solutions.

Now, in order to show the computational efficiency and the quality of the solutions obtained by the proposed method further, we solve the UC MIQP model (22) directly by setting CPLEX with accuracy 0.01% and time limit 1 hour for these 20 systems. The results of CPLEX and the proposed method are listed in Table Ⅲ. In Table III, the columns under "MIQP-CPLEX" show that CPLEX can obtain high quality solutions (Gap less than 0.2%) for all test systems although CPLEX cannot reach preset 0.01% accuracy in one hour. Taking the solutions obtained by CPLEX as references, as can be seen in Table III, H-ADMM can get high quality solutions for these test systems, and the relative errors are about 1%, and for most of the test systems (17 systems), the relative errors are still less than 1%. That is to say the proposed H-ADMM has "good convergence" for our experiments, i.e., it can converge to a point with good objective value.

TABLE II
ILLUSTRATION OF H-ADMM IN SOLVING LARGE SCALE SYSTEMS

No.	MIQP-CPLEX(0.005)			H_ADMM						
	C_{time} (s)	Total cost $f(x)_{\text{MIQP}}$ ($)	Gap	Total cost $f(x)_{\text{H-ADMM}}$ ($)	ε^{c}	p^{opt}	$N_{\text{it\_out}}$	$N_{\text{it\_in\_avg}}$	In serial C_{time}(s)	In Parallel C_{time}(s)
1	1.09	3830181	0.2%	3868032	0.46%	0.98%	26	18	5.74	3.54
2	1.99	4823318	0.4%	4852508	0.47%	0.60%	15	18	3.93	2.25
3	2.53	5135669	0.3%	5151714	0.58%	0.31%	24	17	8.01	4.19
4	1.39	4805839	0.4%	4825959	0.58%	0.41%	20	18	7.37	3.58
5	2.68	5407696	0.3%	5449875	0.59%	0.77%	171	18	76.00	42.01
6	1.70	4425656	0.5%	4430939	0.58%	0.11%	93	20	39.24	18.9
7	1.79	5852746	0.4%	5866804	0.51%	0.24%	20	18	8.63	4.24
8	6.31	5189715	0.3%	5234994	0.52%	0.87%	136	18	57.53	28.71
9	3.18	5640472	0.3%	5683544	0.57%	0.76%	171	17	80.99	46.83
10	3.68	5096676	0.4%	5110531	0.54%	0.27%	76	19	32.39	15.77
11	8.87	15877896	0.5%	15963696	0.55%	0.54%	14	17	16.07	7.37
12	29.95	17242879	0.3%	17234942	0.55%	-0.04%	22	18	28.21	16.09
13	28.21	16936528	0.4%	16950110	0.32%	0.08%	97	18	145.24	69.28
14	117.22	20182255	0.2%	20356607	0.59%	0.83%	13	18	16.32	7.91
15	116.50	17383821	0.1%	17470305	0.33%	0.48%	81	18	140.90	67.13
16	13.20	19570568	0.4%	19583330	0.48%	0.06%	16	18	26.11	11.58
17	129.06	19710424	0.2%	19748998	0.58%	0.19%	18	18	26.37	12.42
18	54.382	19628548	0.2%	19704270	0.58%	0.38%	16	18	25.55	11.49
19	77.00	20137786	0.2%	20282731	0.50%	0.71%	15	18	24.47	10.90
20	138.20	19734927	0.1%	19793775	0.57%	0.29%	17	18	16.99	12.34

TABLE III
ILLUSTRATION OF H-ADMM IN SOLVING LARGE SCALE SYSTEMS

No.	MIQP-CPLEX			H_ADMM		
	C_{time}	Total cost $f(x)_{\text{MIQP}}$ ($)	Gap	C_{time}(s)	Total cost $f(x)_{\text{H-ADMM}}$ ($)	p^{opt}
1	1h	3828408	0.07%	4.78	3868032	1.03%
2	1h	4813859	0.13%	2.25	4852508	0.80%
3	1h	5124010	0.02%	4.19	5151714	0.54%
4	1h	4793846	0.03%	3.58	4825959	0.66%
5	1h	5397428	0.05%	42.01	5449875	0.97%
6	1h	4411234	0.03%	18.9	4430939	0.44%
7	1h	5834861	0.05%	4.24	5866804	0.54 %
8	1h	5180435	0.03%	28.71	5234994	1.05%
9	1h	5629538	0.05%	46.83	5683544	0.95%

10	1h	5082788	0.07%	15.77	5110531	0.54%
11	1h	15824863	0.07%	7.37	15963696	0.87%
12	1h	17218206	0.04%	16.09	17234942	0.09%
13	1h	16895931	0.01%	69.28	16950110	0.32%
14	1h	20148929	0.03%	7.91	20356607	1.03%
15	1h	17369489	0.03%	67.13	17470305	0.58%
16	1h	19508290	0.04%	11.58	19583330	0.38%
17	1h	19692032	0.03%	12.42	19748998	0.28%
18	1h	19603663	0.03%	11.49	19704270	0.51%
19	1h	20119361	0.03%	10.90	20282731	0.81%
20	1h	19721585	0.04%	12.34	19793775	0.36%

The proposed H-ADMM can be executed parallel by using Matlab Parallel Computing Toolbox with 4 workers, and the calculating time and speedup are listed in Table IV. In this table, column "p_{spu}" represents the parallel speedup; column "p_{pe}" represents the parallel efficiency. The results of 1, 2, and 4 workers show that as the number of workers increases, the calculating time reduces dramatically, but parallel efficiency decreases either. For the communication between too many processors and low efficient allocation taking too much time, the parallel efficiency is a decreasing function of CPU number.

TABLE IV
RESULTS OF H-ADMM IN PARALLEL ENVIRONMENT

No.	1 worker			2 workers			4 workers		
	C_{time}(s)	p_{spu}	p_{pe}	C_{time}(s)	p_{spu}	p_{pe}	C_{time}(s)	p_{spu}	p_{pe}
1	6.77	1	100.0%	4.30	1.57	36.5%	3.54	1.91	47.7%
2	4.66	1	100.0%	3.24	1.43	30.4%	2.25	2.07	51.7%
3	9.45	1	100.0%	6.03	1.56	36.2%	4.19	2.26	55.7%
4	8.57	1	100.0%	5.08	1.68	40.7%	3.58	2.39	58.2%
5	86.58	1	100.0%	51.90	1.67	40.0%	42.01	2.06	51.5%
6	44.19	1	100.0%	27.72	1.59	37.3%	18.9	2.34	57.2%
7	9.79	1	100.0%	5.87	1.66	40.0%	4.24	2.31	56.7%
8	64.88	1	100.0%	39.26	1.65	39.5%	28.71	2.26	55.7%
9	91.55	1	100.0%	54.29	1.68	40.7%	46.83	1.95	48.8%
10	36.04	1	100.0%	22.13	1.62	38.6%	15.77	2.29	56.2%
11	18.43	1	100.0%	11.18	1.64	39.3%	7.37	2.50	60.0%
12	37.18	1	100.0%	19.55	1.90	47.4%	16.09	2.31	56.7%
13	170. 67	1	100.0%	98.98	1.72	42.0%	69.28	2.46	59.4%
14	18.36	1	100.0%	11.35	1.62	38.2%	7.91	2.32	56.9%
15	151.21	1	100.0%	90.27	1.67	40.3%	67.13	2.25	55.6%
16	28.61	1	100.0%	21.20	1.35	25.9%	11.58	2.47	59.5%
17	29.08	1	100.0%	17.60	1.65	39.4%	12.42	2.34	57.3%
18	28.53	1	100.0%	17.45	1.63	38.8%	11.49	2.48	59.7%
19	26.82	1	100.0%	16.15	1.66	39.8%	10.90	2.46	59.4%
20	29.68	1	100.0%	18.24	1.63	38.5%	12.34	2.41	58.4%

V. CONCLUSION

The need for the coordination of distributed energy resources in the future electric power system provides motivation to move from the current highly centralized control of resources towards a more distributed control structure. In this paper, a novel hierarchical method for fully distributed solving UC problem was proposed under the framework of ADMM. The separable objective function and physical constraints for each unit can be decoupled in outer ADMM. And we decouple the coupling system constraints by using inner consensus ADMM to solve the Lagrangian dual problem of the outer ADMM z-update step. The proposed method can be carried out in master-slave distributed and parallel schema and can protect privacy for independent power producer. Simulation results show that the proposed method can obtain high-quality solutions for D-UC problems and is suit for large-scale electricity market application.

REFERENCES

[1] B. Saravanan, S. Das, S. Sikri, and D. P. Kothari, "A solution to the unit commitment problem - a review," *Frontiers in Energy*, vol. 7, no. 2, pp. 223-236, Jun. 2013.

[2] K. Hossein, A. Jamshid, K. Benyamin, and S. Pierluigi, "Optimal battery sizing in microgrids using probabilistic unit commitment," *IEEE Trans. Ind. Informat.*, vol. 12, no. 2, pp. 834-843, Apr. 2016.

[3] M. Carrión, and J. M. Arroyo, "A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem," *IEEE Trans. Power Syst.*, vol. 21, no. 3, pp. 1371–1378, Aug. 2006.

[4] M. Shahidehpour, H. Yamin, Z. Li, Market Operations in Electric Power Systems: Forecasting, Scheduling, and Risk Management. *New York: Wiley-IEEE Press*, 2002.

[5] K. Zhang, Y. Mao, S. Leng, S. Maharjan, Y. Zhang, A. Vinel and M. Jonsson, "Incentive-driven energy trading in the smart grid", *IEEE Access*, vol.4, pp.1243-1257, Apr. 2016.

[6] Y.M. Wang, L. Wu, S.X. Wang. A Fully-Decentralized Consensus-Based ADMM Approach for DC-OPF With Demand Response[J]. *IEEE Trans. on Smart Grid*, 2016, PP(99):1-11.

[7] Hug G, Kar S, Wu C. Consensus + Innovations Approach for Distributed Multiagent Coordination in a Microgrid[J]. *IEEE Trans. on Smart Grid*, 2015, 6(4):1893-1903.

[8] Nutkani I U, Loh P C, Wang P, et al. Decentralized Economic Dispatch Scheme With Online Power Reserve for Microgrids[J]. *IEEE Trans. on Smart Grid*, 2016, 8(1):139-148.

[9] Tang Y, Dvijotham K, Low S. Real-time Optimal Power Flow[J]. *IEEE Trans. on Smart Grid*, 2017, PP(99):1-1.

[10] Dall'Anese E, Zhu H, Giannakis G B. Distributed Optimal Power Flow for Smart Microgrids[J]. *IEEE Trans. on Smart Grid*, 2013, 4(3):1464-1475.

[11] R. Yu, W. Zhong, S. Xie, C. Yuen, S. Gjessing and Y. Zhang, "Balancing power demand through EV mobility in vehicle-to-grid mobile energy networks," *IEEE Transactions on Industrial Informatics*, vol.12, no.1, pp.79-90, Feb.2016.

[12] B. Chai, J. Chen, Z. Yang and Y. Zhang, "Demand response management with multiple utility companies: A Two-Level Game Approach", *IEEE Transactions on Smart Grid*, vol.5, no.2, pp.722-731, March 2014.

[13] H. Liu, H. Ning, Y. Zhang, Q. Xiong and L. Yang, "Role-dependent privacy preservation for secure V2G networks in the smart grid", *IEEE Trans. on Information Forensics & Security*, vol.9, no.2, pp.208-220, Feb. 2014.

[14] Zhang Z, Rahbari-Asr N, Chow M Y. Asynchronous distributed cooperative energy management through gossip-based incremental cost consensus algorithm[C]// North American Power Symposium. IEEE, 2013:1-6.

[15] Tsai S C, Tseng Y H, Chang T H. Communication-Efficient Distributed Demand Response: A Randomized ADMM Approach[J]. *IEEE Trans. on Smart Grid*, 2017, 8(3):1085-1095.

[16] Y. Liu, C. Yuen, R. Yu, Y. Zhang and S. Xie, "Queuing-based energy consumption management for heterogeneous residential demands in smart grid", *IEEE Transaction on Smart Grid*, vol.7, no.3, pp.1650-1659, May 2016.

[17] S. Maharjan, Q. Zhu, Y. Zhang, S. Gjessing and T. Basar, "Demand response management in the smart grid in a large population regime", *IEEE Transaction on Smart Grid*, vol.7, no.1, pp.189-199, Jan. 2016.

[18] D. He, S. Chan, Y. Zhang, and M. Guizani, "An enhanced public key infrastructure to secure smart grid wireless communication networks", IEEE Network Magazine, vol. 28, no.1, pp.10-16, Jan/Feb. 2014.

[19] H. Liu, H. Ning, Y. Zhang and L. Yang, "Aggregated-proofs based privacy-preserving authentication for V2G networks in smart grid", *IEEE Transactions on Smart Grid*, vol.3, no.4, pp.1722-1733, 2012.

[20] R. Yu, X. Huang, J. Kang, J. Ding, S. Maharjan, S. Gjessing, and Y. Zhang, "Cooperative resource management in cloud-enabled vehicular networks", *IEEE Transactions on Industrial Electronics*, vol.62, no.12, pp. 7938-7951, Dec.2015.

[21] R. Yu, J. Ding, S. Maharjan, S. Gjessing, Y. Zhang, and D. Tsang, "Decentralized and optimal resource cooperation in geo-distributed mobile cloud computing", accepted by IEEE Transactions on Emerging Topics in Computing.
[22] C.H. Lo, N. Ansari, "Decentralized controls and communications for autonomous distribution networks in smart grid," *IEEE Trans. Smart Grid*, vol. 4, no. 1, pp. 66-77, Mar. 2013.
[23] J. Xu, R.D. Christie, "Decentralized unit commitment in competitive energy markets," *In: The next generation of electric power unit commitment models*, Springer, pp. 293-313, 2001.
[24] C.K. Simoglou, P.N. Biskas, A.G. Bakirtzis, "Optimal self-scheduling of a thermal producer in short-term electricity markets by MILP," *IEEE Trans. Power Syst.*, vol. 25, no. 4, pp. 1965-1977, Dec. 2010.
[25] A. L. Motto, F.D. Galiana FD, "Equilibrium of auction markets with unit commitment: the need for augmented pricing," *IEEE Trans. Power Syst.*, vol. 17, no. 3, pp. 798-805, 2002.
[26] P. Correia, "Decentralised unit commitment in a market structure: problem formulation and solution advancement," *IEE Proc-Gener, Trans. Distrib.*, vol. 153, no. 1, pp. 121-126, 2006.
[27] R. Sioshansi, S.S. Oren, R. ONeill, "The cost of anarchy in self-commitment based electricity markets," In: Competitive electricity markets: design, implementation and performance, *Elsevier*, pp. 245-266, 2008.
[28] R. Fischer, P. Serra, "Energy prices in the presence of plant indivisibilities," *Energy Econ.*, vol. 25, no. 4, pp. 302-314, 2003.
[29] R. Sioshansi, R. O'Neill, S.S. Oren, "Economic consequences of alternative solution methods for centralized unit commitment in day-ahead electricity markets," *IEEE Trans. Power Syst.*, vol. 23, no. 2, pp. 344-352, 2008.
[30] W. Gatterbauer, "Economic efficiency of decentralized unit commitment from a generator perspective," In: Engineering electricity services of the future, Kluwer Academic Publishers (Springer Verlag), 2010.
[31] T. Nagata, M. Ohno, H. Sasaki, and H. Fujita, "A multi-agent approach to unit commitment," Electr. Eng. Jpn., vol. 141, no. 2, pp. 41-47, Nov. 2002.
[32] B. Ramachandran, S. K. Srivastava, C. S. Edrington, and D. A. Cartes, "An intelligent auction scheme for smart grid market using a hybrid immune algorithm," *IEEE Trans. Ind. Electron.*, vol. 58, no. 10, pp. 4603-4612, Oct. 2011.
[33] A. Pantoja and N. Quijano, "A population dynamics approach for the dispatch of distributed generators," *IEEE Trans. Ind. Electron.*, vol. 58, no. 10, pp. 4559-4567, Oct. 2011.
[34] H.S.V.S. Kumar-Nunna, S. Doolla, "Multiagent-based distributedenergy-resource management for intelligent microgrids," *IEEE Trans. Ind. Electron.*, vol. 60, no. 4, pp. 1678-1687, Apr. 2013.
[35] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. and Trends Mach. Learning*, vol. 3, no. 1, pp. 1-122, 2010.
[36] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers & Mathematics with Applicaions*, vol. 2, no. 1, pp. 17–40, 1976.
[37] A. Frangioni, and C. Gentile, "A computational comparison of reformulations of the perspective relaxation: SOCP vs. cutting planes," *Operations Research Letters*, vol. 37, no. 3, pp. 206-210, May 2009.
[38] O. Günlük, and J. Linderoth, "Perspective reformulations of mixed integer nonlinear programs with indicator variables," *Math. Program.*, vol. 124, pp. 183-205, 2010.
[39] A. Frangioni, C. Gentile, and F. Lacalandra, "Tighter approximated MILP formulations for unit commitment problems," *IEEE Trans. Power Syst.*, vol. 24, no. 1, pp. 105–113, Jan. 2009.
[40] A. Frangion, C. Gentile, and F. Lacalandra, "Sequential Lagrangian-MILP approaches for unit commitment problems," *Int. J. Elect. Power Energy Syst.*, vol. 33, no. 3, pp. 585-593, Mar. 2011.
[41] J. Ostrowski, M.F. Anjos, and A. Vannelli, "Tight mixed integer linear programming formulations for the unit commitment problem", *IEEE Trans. Power Syst.*, vol. 27, no. 1, pp. 39-46, Feb. 2012.
[42] D. Rajan, and S. Takrit, "Minimum up/down polytopes of the unit commitment problem with start-up costs," *IBM research report*, 2005.
[43] G. Morales-Espana, J. M. Latorre, and A. Ramos, "Tight and compact MILP formulation for the thermal unit commitment problem," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4897-4908, Mar. 2013.
[44] L.F. Yang, C. Zhang, J.B. Jian, K. Meng, Y. Xu, Z.Y. Dong, "A novel projected two-binary-variable formulation for unit commitment in power systems," *Appl. Energy*, DOI: 10.1016/j.apenergy.2016.11.096.
[45] R. Chartrand, "Nonconvex splitting for regularized low-rank + sparse decomposition," *IEEE Trans. on Signal Processing*, vol. 60, no. 11, pp. 5810-5819, 2012.
[46] B. S. He, M. Tao, and X. M. Yuan, "Alternating direction method with Gaussian back substitution for separable convex programming," *SIAM J. Optim.*, vol. 22, no. 2, pp. 313-340, Apr. 2012
[47] C. H. Chen, B. S. He, Y. Y. Ye, and X. M. Yuan, "The direct extension of ADMM for multiblock convex minimization problems is not necessarily convergent," *Math. Program.*, Ser. A, vol. 155, pp. 57-79, 2016.
[48] M. Fukushima, "Application of the alternating direction method of multipliers to separable convex programming problems", *Computational Optimization and Applications*, vol. 1992, no. 1, pp. 93-111, 1992.
[49] M.S. Bazaraa, H.D. Sherali, C. M. Shetty, Nonlinear programming: theory and algorithms (third edition), Wiley-Interscience, 2006.
[50] Takapoui R, Moehle N, Boyd S, et al. A Simple Effective Heuristic for Embedded Mixed-Integer Quadratic Programming[C]// American Control Conference. IEEE, 2015:5619-5625.
[51] T. Senjyu, K. Shimabukuro, K.Uezato, and T. Funabashi, "A fast technique for unit commitment problem by extended priority list," *IEEE Trans. Power Syst.*, vol. 18, no. 2, pp. 882-888, May 2003.