

Dynamic Load Balancing Applying Water-Filling Approach in Smart Grid Systems

Mushu Li, Peter He, *Member, IEEE*, and Lian Zhao, *Senior Member, IEEE*

Abstract—To enhance the reliability of the power grid, further processing of the power demand to achieve load balancing is regarded as a critical step in the context of smart grids with Internet of Things technology. In this paper, dynamic offline and online scheduling algorithms are proposed to minimize the power fluctuations by applying a geometric water-filling approach. For the offline approach, full information in the power demand is available, possibly by predicting from the power utilities. We present an exact approach in order to allocate the elastic loads based on the inelastic load's information considering the group- and node-power upper constraints. For the online approach, the reference level is computed dynamically using historical demand data to minimize the fluctuation in the grid, and the elastic loads can only be scheduled in the future time slots. Two dynamic algorithms are investigated to achieve load balancing in the power grid without influencing user experience by real-time reference level adjustment. Facilitated by the proposed methodologies, the power utilities can significantly reduce the cost of improving the power capacity, and the consumers are able to enjoy more stable electrical power.

Index Terms—Internet of Things (IoT), load management, load modeling, optimization, resource management, smart grids, upper bound.

I. INTRODUCTION

THE SMART grid has been considered as one of the most important applications of Internet of Things (IoT) technologies in recently years [1]. It requires the integration of load control, information communication, and cloud data computing via coordinating among smart meters, smart infrastructures and smart facilities, in which IoT technologies provide a framework for communications. Via IoT technologies, the smart grid enables improving energy efficiency by energy monitoring, energy modeling, practical changes evaluation, and strategy adjustments [2].

It is well-known that power demand is time-varying in a day. In a typical summer weekday in Ontario, the power consumption difference between peak hours and nonpeak hours

can be as much as 65% [3]. Therefore, two problems would happen: when the utility can not provide enough power to supply the peak demand, some loads have to be shut down and it will cause power shortage. On the other hand, to keep the stability of the power grid system, the power plant has to pay more generators and reach the considerable capacity to supply the power in peak hours. The problem brings the mass cost to the power utility and reduces the efficiency of the power grid.

Demand side management (DSM) is a mechanism to increase the power stability and efficiency by scheduling elastic loads on demand side considering the user's utility and electricity cost. Users' power consumption is able to be shaped to produce a desired pattern by DSM [4]. Less power fluctuation in the power system indicates improved efficiency and safety operation of the power grid. Therefore, flattening the power consumption to achieve load balancing via DSM attracted lots of research attention in recent years. To achieve this objective, in [5]–[9], charging of plug-in hybrid electrical vehicles (PHEVs) was scheduled to satisfy DSM by minimizing the energy cost depending on a floating electricity price; [10]–[15] achieved DSM to maximize user's payoff by the mechanism of real-time pricing approach; [16]–[18] implemented the real-time demand side control of heating, ventilation, and air conditioning (HVAC) by energy prediction, while the users' comfort level was not influenced. Recent contribution [1] proposed centralized and decentralized real-time energy distribution strategies to maximize users' utility, minimize cost, and smooth the grid load by applying IoT technologies.

Water-filling (WF) algorithm is a well-known important tool for optimal radio resource management in communication systems to maximize channel capacity when power is constrained [19]. In recent years, WF approach has been used to achieve load balancing in the context of the smart grid [3], [18], [20]–[22]. Early work [3] applied WF concept to schedule elastic loads with the assumption that all load information in a day was predicted. The overall power consumption is flattened through appropriate scheduling with a probability distribution model. In [20], load management problem was modeled as a typically simplest form of the WF problem, i.e., the load to be non-negative and subject to sum load restriction. It is important to note that although these early investigations were preliminary, they paved the ways for later research works. For more later works, in [21], on top of the typical load balancing WF problem, delay cost was added as another consideration factor. In [18] and [22],

Manuscript received November 1, 2016; accepted December 28, 2016. Date of publication January 4, 2017; date of current version February 8, 2017. This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) under Grant RGPIN-2014-03777 and in part by a Dean's Research Fund from the Faculty of Engineering and Architecture Science of Ryerson University.

M. Li and L. Zhao are with the Department of Electrical and Computer Engineering, Ryerson University, Toronto, ON M5B 2K3, Canada (e-mail: mushu1.li@ryerson.ca; l5zhao@ryerson.ca).

P. He is with the Department of Electrical and Computer Engineering, University of Waterloo, Toronto, ON N2L 3G1, Canada (e-mail: z85he@uwaterloo.ca).

Digital Object Identifier 10.1109/JIOT.2016.2647625

TABLE I
LIST OF VARIABLES AND ABBREVIATIONS

Group	Variable	Meaning	
Index Group	k	index of time slots, for $k = 1, \dots, K$.	
	i	index of group, for $i = 1, \dots, I$.	
	j	index of user (or node), for $j = 1, \dots, J$.	
	χ_i	the i th group.	
	n	index of time slots in the reference window, for $n \in [k - N + 1: k]$.	
Load Group	n_1	index of time slots in past part of the reference window, for $n_1 \in N_1$, where $N_1 = [k + m - N + 1: k]$.	
	n_2	index of time slots in future part of the reference window, for $n_2 \in N_2$, where $N_2 = [k + 1: k + m]$.	
	A	inelastic load matrix. The (k, j) th element, $A_{k,j}$, denotes inelastic load of user j at time k .	
	B	inelastic load vector, obtained by sums of A over rows. The k th element, B_k , denotes the total inelastic load at time k .	
	R	elastic load matrix. The (k, j) th element, $R_{k,j}$, denotes inelastic load of user j at time k .	
	S	elastic load vector. The k th element, S_k , denotes the total elastic load at time k .	
	\hat{A}	predicted inelastic load matrix. The (k, j) th element, $\hat{A}_{k,j}$, denotes predicted inelastic load of user j at time k .	
	Constraint Group	P	individual load upper bound (or peak) matrix. The (k, j) th element, $P_{k,j}$, denotes load upper bound for user j at time k .
		PG	group load upper bound matrix. The (k, i) th element, $PG_{k,i}$, denotes load upper bound for the i th group at time k .
		PU	load upper bound vector for time slots. The k th element, PU_k , denotes load upper bound at time k .
P_T		total load budget or supply capability for entire elastic power loads over all the time slots, <i>i.e.</i> , a sum of loads.	
P_S		elastic load demand vector. The k th element, $P_{S,k}$, denotes the elastic load demand at time k .	
Reference Level Group	L	reference horizon, reference level for offline scheduling.	
	L	reference level vector for online scheduling. The k th element, L_k , denotes reference level at time k .	
Abbreviations	WF	Water-filling	
	BLA	Basic load allocation.	
	DSM	Demand side management.	
	ELPA	Elastic load power allocation.	
	OELPA	Online elastic load power allocation.	
	EOELPA	Computation efficient online elastic load power allocation.	

WF algorithm was used to schedule the demand of HVACs and PHEVs, respectively. In the above works, WF concept provides an efficient approach to balance the overall load in order to minimize power fluctuation. In our recent work [23], we modeled the problem in a more general form and exploited our proposed geometric WF [19] approach to provide a closed-form, exact valued and strictly proven optimal solution with a low degree polynomial computational complexity.

The main contribution of this paper is to dynamically implement the optimal elastic load scheduling to achieve load balancing in IoT environment. Based on WF algorithm, we propose dynamic approaches to schedule the elastic load both offline and online to flatten the overall power consumption considering the peak power constraints. First, we inherit the basic concept and the problem formulation for computing the general exact solution for load balancing applying WF approach from our previous work [23], where the offline approach with full given load information was investigated. A smart grid operator can generate a constant reference level and the solution from the one-time computation. In this paper, we implement the new online load balancing approaches on top of the offline approach using the load prediction model from [3], considering the group and node power upper constraints for elastic loads. Comparing with our previous work [23], the main differences are summarized below.

- 1) Beyond the offline solution that we already proposed in [23], we modified the problem formulation and methodology to adopt the dynamic change in the real-time power grid system. The online approach is implemented to balance the loads in real-time without future load request information. It provides a robust

algorithm independent of the power profile prediction. Therefore, the online algorithm can allocate the power loads without knowledge of the future information or in the scenario of inaccurate prediction, which the offline solution cannot achieve.

- 2) The demand of inelastic and elastic loads changes randomly in the real time, and the elastic loads can not be scheduled in the elapsed time slots. We strategically adjust the overall load reference level in real-time to be compatible with real power demand variation, which is computed from the demand history monitored by the smart meters. With our strategy, the load can be balanced in real time without losing user satisfaction.
- 3) The load allocation is no longer in one direction from the smart grid operator to the end users. In each allocation cycle, the utility allocates the power to the microgrids and users. Meanwhile, the utility adjusts the reference level for next cycle from the real demand pattern, which requires coordinating among the smart infrastructures, smart meters, and smart facilities in IoT environment. The power allocation operation turns to be more stable due to the roundway communication and feedback in the network.
- 4) We also extend a computation efficient online algorithm with lower computational complexity in real-time scheduling, which is expected to reduce the communication and computation burden in IoT networks within the admissible variance of overall power fluctuation.

In the remainder of this paper, the system model of the load balancing problem via WF approach is presented

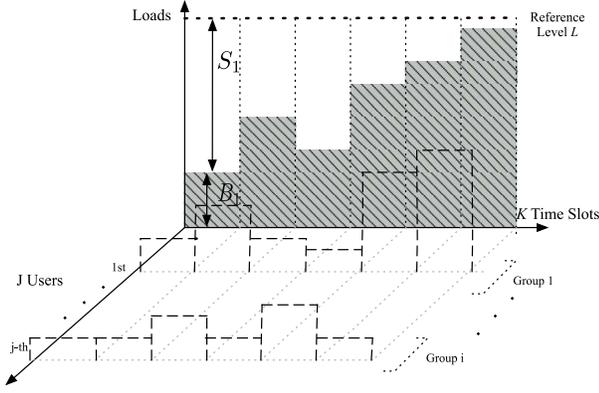


Fig. 1. System model: 3-D power allocation problem.

in Section II. Basic elastic load allocation approach is introduced in Section III. In Section IV, we propose the algorithms to allocate the elastic loads for offline and online elastic load allocation problems, respectively. Simulation results and performance evaluation are presented in Section V. The conclusion and future works are summarized in Section VI.

II. SYSTEM MODEL

A. Problem Statement

Similar as [3] and [18], depending on the flexibility of the power requests in the system, we assume that there are two categories of the loads: 1) elastic loads and 2) inelastic loads. Inelastic loads have strict time constraints so that the demand is not schedulable. The demand of the elastic loads can be scheduled, such as charging for PHEVs. In our offline approach, power requirements of the inelastic loads are predicted as parameters. To reduce the fluctuation of the overall power load in the system, the elastic load demand is shifted from the peak hours to the nonpeak hours in order to flatten the power profile. Table I is a list of the variables and abbreviations used in our analysis. In our work [23], the system is modeled as a 3-D problem regarding time, users, and load as shown in Fig. 1.

The index k denotes the time slot from 1 to K . There are J users in the system, indexed by j . All the users are grouped into I groups, where i is group index, and χ_i being a set of users in group i , where $i \in [1, I]$. The inelastic and elastic loads are denoted by matrix A and matrix R , respectively. $A_{k,j}$ and $R_{k,j}$ represent the power consumption of inelastic and elastic loads of the j th user in the k th time slot, respectively. $A_{k,j}$ can be predicted from the smart grid operator, which is shown as the wide dashed lines in Fig. 1. The shadow areas, B_k are the projection of the inelastic load for all the users at the k th time slot. L is an optimization variable as the reference level of the overall power consumption, where the WF concept is applied.

The objective of load balancing is to determine an optimal reference value L , then schedule the elastic load power R to minimize the difference between L and the overall load from

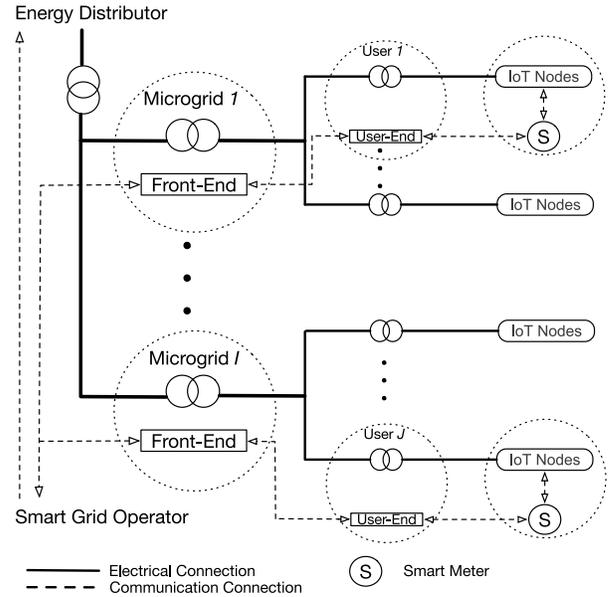


Fig. 2. Schematic of the network of the distribution grid.

time slots 1 to K . It can be written as

$$\begin{aligned}
 & \min_{\{R, L\}} \sum_{k=1}^K \left[\sum_{i=1}^I \sum_{j \in \chi_i} (A_{k,j} + R_{k,j}) - L \right]^2 \\
 & \text{subject to } 0 \leq R_{k,j} \leq P_{k,j}, j = 1, \dots, J, \forall k \\
 & \sum_{j \in \chi_i} R_{k,j} \leq PG_{k,i}, \forall i, k \\
 & \sum_{j=1}^J R_{k,j} \leq PU_k, \forall k \\
 & \sum_{k=1}^K \sum_{j=1}^J R_{k,j} = P_T \\
 & L \geq 0
 \end{aligned} \tag{1}$$

where $P_{k,j}$ is the upper bound of the elastic load of user j in time k . $PG_{k,i}$ and PU_k are the upper bounds of the elastic loads for the i th group at the k th time slot and for all users at the k th time slot, respectively. These constraints offer an approach to control the system directly in power grid when some particular issues occur, such as fault detection. To ensure the system feasibility, all the peak power values are not less than 0. P_T is the total load budget for the whole system's elastic loads over all time slots.

B. Network Structure

We consider the power distribution model in an IoT framework, where the smart grid operator conducts load balancing for the overall system. Fig. 2 shows the power distribution and communication network model. There were

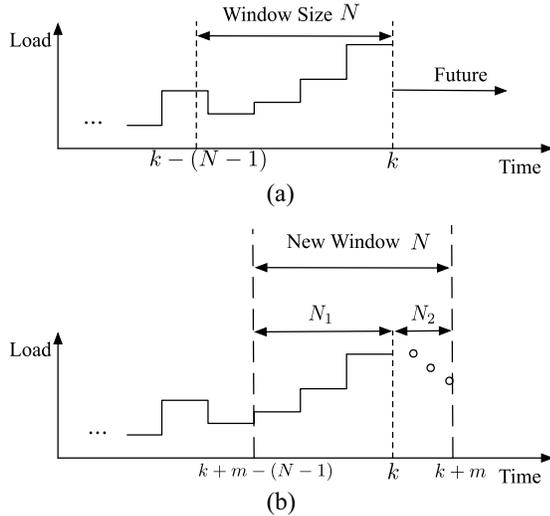


Fig. 3. Moving window model. (a) Window with size N before shifting. (b) New window with size N after shifting.

three levels in the network: 1) energy distributor; 2) microgrids (groups); and 3) users. In each of the levels, there was a server to receiving the load requests from the lower level and protect the transformers with upper power constraints. At time k , first, the smart grid operator generates a reference level of the overall power distribution to balance the loads in real-time, which is computed by predicting the power requests from the smart meters. It also controls the overall power flow in real time to be not greater than PU_k in order to protect the transformer of the whole system. Furthermore, the front-ends of I microgrids collaborate and balance the power in group level depending on their group power constraint PG . As the same as at the group level, the user level distributes the group power constrained by individual power constraint P . Then the smart meter of each user schedules the infrastructures and receives the load requests by Internet connection. The real loads are regarded as IoT nodes.

C. Moving Window

In this paper, we discuss both offline and online solutions for the load balancing problem. For offline operation, it is assumed that all the load demand is known. The reference level L is a constant, and it is computed once. For online operation, the available load information is the load demand from the past time slots. The reference level L is a vector which is different by time slots. We shall use a moving window to solve reference level L_k for next time slot scheduling. Fig. 3(a) shows the model of moving window approach with window size N . When the algorithm runs to time k , we apply a reference level L_k determined by the parameters from the time slots $k - (N - 1)$ to k in order to schedule the loads for the $(k + 1)$ th time slot. With a smaller window size, the reference level is shaped more likely to the unscheduled power profile. In contrast, with a larger window size, the variation of the reference level is slow. Furthermore, we propose a computation reduced online approach. We predict inelastic load information for the m th time slot in the future, as shown in Fig. 3(b). Then we shift the starting point of the

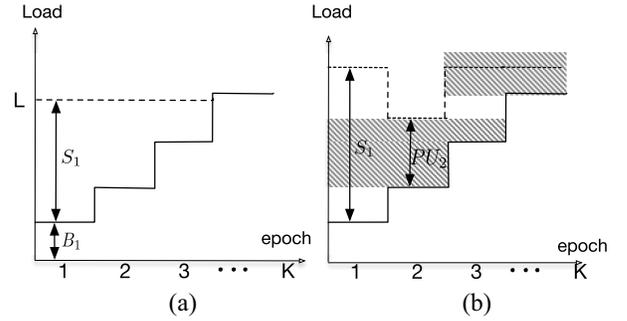


Fig. 4. Illustration of BLA problem. (a) BLA problem with infinite PU_k . (b) BLA problem with finite PU_k .

window by m slots, so that the window size is kept at N but we analyze the data which includes both the past and future information.

III. BASIC LOAD ALLOCATION MODEL

To simplify the problem (1), this 3-D problem is transformed to a 2-D problem by projecting the overall load for all the users. Let S_k and B_k represent the elastic and inelastic load vectors from all the users in the k th time slot, respectively. They can be obtained by the summations of the matrix \mathbf{A} and \mathbf{R} over their columns

$$\sum_{i=1}^I \sum_{j \in \chi_i} R_{k,j} = S_k; \quad \sum_{i=1}^I \sum_{j \in \chi_i} A_{k,j} = B_k. \quad (2)$$

Then the objective function in (1) can be simplified by the basic load allocation (BLA) model

$$\begin{aligned} \min_{\{S_k, L\}} & \sum_{k=1}^K (S_k + B_k - L)^2 \\ \text{subject to} & 0 \leq S_k \leq PU_k, \forall k \\ & \sum_{k=1}^K S_k = P_T \\ & L \geq 0 \end{aligned} \quad (3)$$

where the vector \mathbf{B} is computed from the inelastic load matrix A in (2). A and P_T are given by predicting from the utility. The details of the solution are presented in [23] using geometric WF, as illustrated in Fig. 4.

In Fig. 4(a), we assume that PU_k is infinite and sort the load vector \mathbf{B} as a monotonically increasing sequence. The original sequence can be recovered easily after the solution of the problem is obtained. $P_2(k)$ is defined as the allocated total power (elastic load or water volume) above the k th stair. k^* is defined as the maximum index of the stair keeping $P_2(k)$ non-negative, i.e., the highest step under water. The optimal solution can be obtained through the steps

$$P_2(k) = \left\{ P_T - \left[\sum_{l=1}^{k-1} (B_k - B_l) \right] \right\}^+, \quad k = 1, \dots, K \quad (4)$$

where B_l denotes the total elastic load power at time slot l . Index l indicates any time slots less than k . $\{\cdot\}^+$ denotes zero if the value inside the bracket is negative. Then

$$k^* = \max\{k | P_2(k) > 0, 1 \leq k \leq K\}. \quad (5)$$

Algorithm 1 BLA WF Problem Algorithm**Input:** \mathbf{B} , \mathbf{PU} , P_T , $E = \{1, \dots, K\}$ **Output:** \mathbf{S} , L

```

1: while  $E \neq \emptyset$  do
2:   Solve equations (4) - (8) to obtain  $\{\mathbf{S}\}_{k \in E}$  and  $L$  via  $\{\mathbf{B}\}_{k \in E}$  and  $P_T$ .
3:    $\Delta \leftarrow \{k \mid S_k > PU_k, k \in E\}$ .
4:   if  $\Delta \neq \emptyset$  then
5:     if  $k \in \Delta$  then
6:        $S_k = PU_k$ .
7:     end if
8:      $E \leftarrow E \setminus \Delta, P_T = P_T - \sum_{k \in \Delta} PU_k$ .
9:   else
10:    Set  $\{S_k\}$  when  $k \in E$ .
11:     $E = \emptyset$ .
12:   end if
13: end while

```

From Fig. 4(a), the allocated elastic load in the k^* time slot is

$$S_{k^*} = \frac{1}{k^*} P_2(k^*). \quad (6)$$

Then the solution of the elastic load power in the k th slot is

$$S_k = \begin{cases} S_{k^*} + (B_{k^*} - B_k), & 1 \leq k \leq k^* \\ 0, & k^* \leq k \leq K \end{cases} \quad (7)$$

and the reference level is

$$L = \frac{1}{k^*} \sum_{k=1}^{k^*} (B_k + S_k) = \frac{1}{k^*} \left(\sum_{k=1}^{k^*} B_k + P_T \right). \quad (8)$$

In this paper, we assume that there are sufficient amount of elastic load requests, and therefore, there are always some elastic loads available for scheduling, i.e., $k^* = K$. Furthermore, Fig. 4(b) indicates the system with the upper load constraints, when PU_k is a finite number, as shown in the height of the shadow areas. S_k cannot be greater than the corresponding PU_k . To solve this problem, BLA was proposed in [23], described by the following Algorithm 1.

Therefore, the 2-D allocation problem can be written as a mapping function of the inelastic load power vector, the upper power constraint vector, and the total elastic power budget

$$(\mathbf{S}, L) = \text{BLA}(\mathbf{B}, \mathbf{PU}, P_T). \quad (9)$$

The optimality of BLA mapping is proven in [23].

IV. DYNAMIC LOAD BALANCING WITH WF APPROACH

In this section, we first evaluate the solution of the 3-D offline load allocation problem in (1) and implement it to allocate elastic load, which is solved in [23]. Then, a dynamic online approach is presented in the second section when the future demand information is not known. At the third section, we propose a computation efficient online approach to reduce the computation.

Algorithm 2 Offline ELPA**Input:** \mathbf{A} , \mathbf{PU} , \mathbf{PG} , \mathbf{P} , P_T , $j \in \{1, \dots, J\}$, $i \in \{1, \dots, I\}$, $k \in \{1, \dots, K\}$ **Output:** \mathbf{R} , reference level L

```

1:  $\mathbf{B} = \sum_{i=1}^I \sum_{j \in \chi_i} A_{k,j}$ . The vector  $\mathbf{B}$  works as parameter vector and it is not changed in this algorithm.
2:  $(\mathbf{S}, L) = \text{BLA}(\mathbf{B}, \mathbf{PU}, P_T)$ . Steps of BLA are in Algorithm 1.
3:  $\{R_i\}_{i=1}^I = \text{BLA}(\{\sum_{j \in \chi_i} A_{k,j}\}_{i=1}^I, \{PG_{k,i}\}_{i=1}^I, S_K)$ .
4:  $\{R_{k,j}\}_{j \in \chi_i} = \text{BLA}(\{A_{k,j}\}_{j \in \chi_i}, \{P_{k,j}\}_{j \in \chi_i}, R_i)$ .
5: Move to next time period,  $k^{(2)} = \{1, \dots, K^{(2)}\}$ , and back to step 1.

```

A. Offline Approach

Similar to the outline in [3], we summarize the steps of our offline algorithm as follows.

- 1) *Demand Forecasting*: The smart grid operator forecasts all of the demand from previous energy behavior via smart meters to predict the elastic load budget P_T and the matrix of inelastic load \mathbf{A} for users from 1 to J in time vector \mathbf{K} .
- 2) *Reference Level Computation*: After the load information is available, the reference level L for the first time slot to the K th time slot can be solved by using the WF algorithm in (8).
- 3) *Elastic Load Allocation for the System*: Once the reference level is computed, the elastic load power consumption, \mathbf{R} , for the whole system can be solved by elastic load allocation algorithm. In this step, L may be adjusted, depending on the influence of the constraint \mathbf{PU} .
- 4) *Elastic Load Filling*: The elastic loads are settled by the elastic power \mathbf{R} which is allocated in the last step. In this paper, we assume that all loads operate continuously until the tasks being completed, the same treatment as in [3].

After the inelastic and elastic load demand is predicted for the whole system, matrix \mathbf{A} and the elastic load budget P_T are known. In order to have a feasible solution for problem (1), we assume that $\sum_{k=1}^K \sum_{j=1}^J P_{k,j} \geq P_T$, $\sum_{k=1}^K PU_k \geq P_T$, and $\sum_{k=1}^K \sum_{i=1}^I PG_{k,i} \geq P_T$. If the assumption is not hold, then we choose $P_T = \min\{\sum_{k=1}^K \sum_{j=1}^J P_{k,j}, \sum_{k=1}^K PU_k, \sum_{k=1}^K \sum_{i=1}^I PG_{k,i}\}$, to make sure that the power allocated does not violate the constraints. In the same idea, $\sum_{j \in \chi_i} P_{k,j} \geq PG_{k,i}$, otherwise, $PG_{k,i} = \sum_{j \in \chi_i} P_{k,j}$, and $\sum_{i=1}^I PG_{k,i} \geq PU_k$, otherwise $PU_k = \sum_{i=1}^I PG_{k,i}$. Also, in order to make the problem solution exist, P_T should be greater than zero, so that the matrix \mathbf{R} is nonempty. Elastic power allocation can be obtained by Algorithm 2 when above information is predicted.

In Algorithm 2, we call the BLA algorithm three times in order to solve the elastic load matrix, \mathbf{R} . Fig. 5 illustrates Algorithm 2 graphically. First, the total elastic load vector \mathbf{S} and the reference level L can be determined by calling BLA algorithm in line 2, where the constraint for the elastic load,

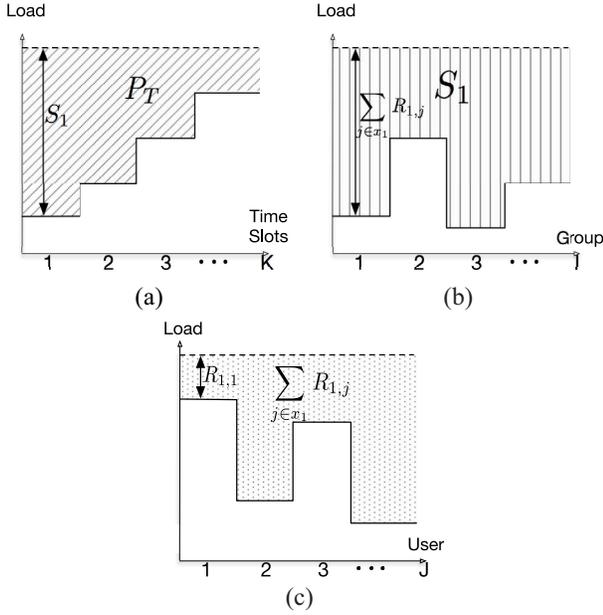


Fig. 5. Elastic load allocation among (a) time slots, (b) groups, and (c) users in a group.

PU, is applied. Fig. 5(a) visualizes this procedure: the shaded area is the elastic power budget P_T , where $\sum_{k=1}^K S_k = P_T$. The surface denotes the reference level. Stairs represent the inelastic load at the corresponding time slots. S_k can be obtained by applying (4)–(7). In line 3, S_k is treated as the new elastic power budget for group $\{1, \dots, I\}$ at time k . So we call BLA algorithm again, which distributes elastic load power S_k to groups 1 to I . PG, upper group elastic power bound, is the new constraint in this step. Fig. 5(b) indicates this concept. In line 4, the individual elastic power matrix $R_{k,j}$ is allocated by calling BLA again considering the constraint P , shown in Fig. 5(c). By this strategy, the elastic power is maximally flattened from the first time slot to the K th slot; and the peak power constraints from the users, groups, and time slots for elastic loads are satisfied.

B. Online Approach

The discussed offline approach requires all of the load information being predicted by the smart grid operator. However, normally, the present and future load information are unknown when the smart grid operator supplies the power to the system. Therefore, we are not able to use a static variable L since the demand is uncertain and time-varying. In this section, we consider the reference level L to be a time-varying variable. Then we adjust problem (1) to the following objective function:

$$\begin{aligned} & \min_{\{\mathbf{R}, \mathbf{L}\}} \sum_{k \in N} \left[\sum_{i=1}^I \sum_{j \in \chi_i} (A_{k,j} + R_{k,j}) - L_k \right]^2 \\ & \text{subject to } 0 \leq R_{k,j} \leq P_{k,j}, j = 1, \dots, J, \forall k \\ & \sum_{j \in \chi_i} R_{k,j} \leq PG_{k,i}, \forall i, k \end{aligned}$$

Algorithm 3 Reference Level Adjustment

Input: Inelastic and elastic load consumption $A_{n,j}$ and $R_{n,j}$ where $n \in [k-N+1, k]$, L_k , and Acc_{k-1} .

Output: L_{k+1} .

- 1: $Acc_k = Acc_{k-1} + P_{S_k} - \sum_{j=1}^J R_{k,j}$.
- 2: $\Delta = [L_k - (\sum_{j=1}^J R_{k,j} + \sum_{j=1}^J A_{k,j})]^2$.
- 3: **if** $\Delta \geq \epsilon$ **or** $Acc_k \geq \xi$ **then**
- 4: $P_T = \sum_{n=k-N+1}^k \sum_{j=1}^J R_{n,j} + Acc_k$.
- 5: $B = \sum_{n=k-N+1}^k \sum_{j=1}^J A_{n,j}$.
- 6: $L_{k+1} = \frac{1}{N} (P_T + B)$.
- 7: **else**
- 8: $L_{k+1} = L_k$.
- 9: **end if**

$$\begin{aligned} & \sum_{j=1}^J R_{k,j} \leq PU_k, \forall k \\ & 0 \leq \sum_{k=1}^K \sum_{j=1}^J R_{k,j} \leq \sum_{k=1}^K P_{S_k} \\ & L_k \geq 0 \end{aligned} \quad (10)$$

where P_{S_k} is the elastic load demand in the k th time slot. The objective function aims to minimize the difference between the overall power load and the reference level to achieve load balancing. The load information inside the moving window is used to predict the reference level for next time slot as described in Algorithm 3.

Algorithm 4 updates the reference level dynamically. Accumulator Acc is the elastic loads which were not scheduled; Δ represents the squared difference between the reference level and the actual total power load; N is window size and n is time index in the window. Inside the window, P_T is the overall elastic load demand, and B is inelastic load demand. Lines 3–9 learn the load behavior in the tuning window when the reference level cannot provide enough accuracy for allocation or the certain amount of elastic load demand cannot be satisfied. The parameters, ϵ and ξ , are the tolerance values determined by the smart grid operator. The reference level will be adjusted frequently with smaller ϵ and ξ , which will lead to a more smooth result. In contrast, the computation of the reference level will be less with larger tolerance, while the performance will be influenced. The window size N is also a factor to affect the performance. With a larger window size N , the reference level L varies more slowly. Otherwise, if the window size is small, L changes more quickly, and then the elastic demand can be satisfied rapidly, but with more overshoot in the overall load.

Overall, OELPA algorithm is shown in Algorithm 4. OELPA is developed by applying Algorithms 1 and 3 to solve the online elastic load allocation problem in (10). The idea is the same as offline elastic load power allocation (offline ELPA) but the reference level is dynamically adjusted. The elastic loads assigned in time k is based on the inelastic load in slot k and L_k which is predicted in time slot $(k-1)$. After the allocation of the elastic load by the reference level in real time, the reference level for the next time slot will be triggered by

Algorithm 4 OELPA

Input: Inelastic load consumption $A_{k,j}$, PU_k , $PG_{k,i}$ and $P_{k,j}$ at time k , where user $j = 1, \dots, J$, group $i = 1, \dots, I$, and window size N .

Output: Elastic load allocation at time k , $R_{k,j}$.

- 1: $S_k = L_k - \sum_{j=1}^J A_{k,j}$, where L_k is obtained from the previous loop.
- 2: **if** $S_k \geq PU_K$ **then**
- 3: $S_k = PU_K$.
- 4: **end if**
- 5: $\{R_i\}_{i=1}^I = \text{BLA}(\{\sum_{j \in \chi_i} A_{k,j}\}_{i=1}^I, \{PG_{k,i}\}_{i=1}^I, S_k)$.
- 6: $\{R_{k,j}\}_{j \in \chi_i} = \text{BLA}(\{A_{k,j}\}_{j \in \chi_i}, \{P_{k,j}\}_{j \in \chi_i}, R_i)$.
- 7: Run reference level adjustment algorithm.
- 8: Move to next time slot $k+1$. Back to step 1.

Algorithm 3, if the conditions are met. OELPA algorithm provides an online dynamic solution when the future inelastic load information is unknown. Reference level in next epoch is modified by inheriting from previous demand information in each time slot, such that the updated reference level enables to adopt the power fluctuation uncertainty in the future. Namely, when future inelastic load power is unknown or predicted unreliably, OELPA does elastic load allocation in each time slots by real-time scheduling. However, ELPA is not suitable in the scenario, since the fixed reference level cannot respond to the uncertain power fluctuations.

C. Computation Efficient Online Approach

In OELPA, we predict the reference level L_k by previous load behavior. In Algorithm 3, if the condition is met, the reference level for next time slot L_{k+1} will be computed to satisfy the demand in the power system. However, to get the optimal solution, ϵ and ξ will be modified as small as possible to reduce the power fluctuation without influencing the users' comfort level. Also, the inelastic load information \mathbf{A} in next time slot is unknown. Therefore, \mathbf{L} will be adjusted frequently because of the fluctuation of $A_{k,j}$. In another word, the computation of OELPA will be a burden. Considering to reduce the computation effort and ensure the quality of service, we propose a computation efficient online approach by predicting future load information in Algorithm 5 (EOELPA).

In enhanced OELPA (EOELPA), first, we fit cubic spline to the inelastic load in the tuning window, from $A_{k-N+1,j}$ to $A_{k,j}$, and predict the inelastic load information $\hat{A}_{k+1,j}$ to $\hat{A}_{k+m,j}$ by extrapolation m points, with \hat{A} being the predicted value for future time slots at time k . The parameter, m , is greater than zero, and it is selected by the smart grid operator. With a smaller m , the prediction accuracy will increase and computation will increase correspondingly. Otherwise, with a larger m , the accuracy and computation will descend. Then we move the time window forward by m slots. The indices, n_1 and n_2 , are time index for $N_1 = [k+m-N+1 : k]$, which is the historical part in the tuning window, and $N_2 = [k+1, k+m]$, which is future part in the tuning window, respectively. Then we predict the reference level periodically by Algorithm 6 instead of adjusting the reference level in Algorithm 3.

Algorithm 5 Computation Reduced Online Elastic Load Allocation Algorithm (EOELPA)

Input: Inelastic load consumption \mathbf{A} in past window $[k-N+1, k]$, \mathbf{PU} , \mathbf{PG} and \mathbf{P} at future m slots, for all users $j = 1, \dots, J$, groups $i = 1, \dots, I$.

Output: Elastic load allocation \mathbf{R} for future m time slots.

- 1: Fit inelastic load consumption $A_{n,j}$ to cubic spline, where $n \in [k-N+1, k]$.
- 2: Predict inelastic load consumption $\hat{A}_{n_2,j}$ by previous spline, where $n_2 \in [k+1, k+m]$.
- 3: Window moves from $[k-N+1 : k]$ to $[k+m-N+1 : k+m]$, where N_1 is the past part in the window, N_2 is the future part in the window. $N_1 = [k+m-N+1 : k]$, $N_2 = [k+1 : k+m]$.
- 4: Compute L_{N_2} at time k by Algorithm 6.
- 5: $S_{N_2} = L_{N_2} - \sum_{j=1}^J \hat{A}_{N_2,j}$.
- 6: **if** $S_{N_2} \geq PU_{N_2}$ **then**
- 7: $S_{N_2} = PU_{N_2}$.
- 8: **end if**
- 9: $\{R_i\}_{i=1}^I = \text{BLA}(\{\sum_{j \in \chi_i} \hat{A}_{n_2,j}\}_{i=1}^I, \{PG_{n_2,i}\}_{i=1}^I, S_{N_2})$.
- 10: $\{R_{n_2,j}\}_{j \in \chi_i} = \text{BLA}(\{A_{n_2,j}\}_{j \in \chi_i}, \{P_{n_2,j}\}_{j \in \chi_i}, R_i)$.
- 11: Back to step 1 after m time slots are elapsed.

Algorithm 6 Reference Level Prediction Algorithm

Input: Inelastic and elastic load consumption $A_{n_1,j}$ and $R_{n_2,j}$ where $n_1 \in N_1$, predicted inelastic load consumption $\hat{A}_{n_2,j}$ where $n_2 \in N_2$, the elastic load demand P_S in past m slot and Acc_{k-m} .

Output: L_{N_2} .

- 1: $Acc_k = Acc_{k-m} + \sum_{n=k-m+1}^k P_S n - \sum_{n=k-m+1}^k \sum_{j=1}^J R_{n,j}$.
- 2: $P_T = \sum_{n_1 \in N_1} \sum_{j=1}^J R_{n_1,j} + Acc_k$.
- 3: $L_{N_2} = \frac{1}{N} (P_T + \sum_{n_1 \in N_1} \sum_{j=1}^J A_{n_1,j} + \sum_{n_2 \in N_2} \sum_{j=1}^J \hat{A}_{n_2,j})$.

In Algorithm 6, the reference level L in future period N_2 is predicted by WF approach. The idea is the same as Algorithm 3 but we predict a reference level for future m time slots, while the reference level is only predicted for next one slot in Algorithm 3. After the inelastic loads and reference level are determined, S_{N_2} can be obtained, which is the overall elastic power for the system in future m time slots. Then we can schedule the elastic load for future m slots by calling twice BLA algorithm, the same as OELPA does. The algorithm is revisited when m time slots are elapsed, then $k^{(2)} = k^{(1)} + m$.

Compared with the online algorithm OELPA, the computation of the reference level can be reduced by $(1/m)$ in EOELPA because the reference level for m slots is predicted in one run. Furthermore, since the elastic loads are scheduled only depending on the predict information, the elastic load allocation does not have to always stay online in each time slot. The algorithm allocates the elastic load for m slots at one time, so the real-time information needs to be communicated once in every m periods, instead of keeping online computation in every time slot. Considering the computing efficiency, EOELPA solves the problem when communication

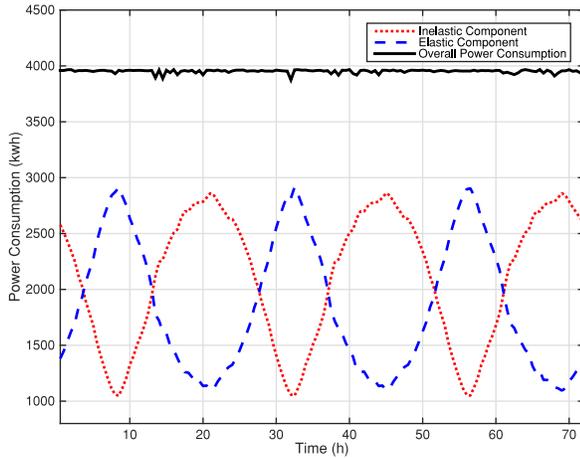


Fig. 6. Power consumption of inelastic, elastic, and total loads scheduled by offline ELPA for three days.

resources are limited for the large-scale load balancing, though the overall fluctuation is higher than the result from OELPA because of the uncertainties of the prediction.

V. SIMULATION RESULTS

In this section, we evaluate the performance of the proposed offline and online ELPA (OELPA) algorithms. The simulation model for a house is based on the parameters listed in [3, Sec. VII, Tables 1 and 2]. We consider a power system with $I = 4$ groups or microgrids, and in each group there are 100 users. We divide the scheduling interval as 30 min. Once an elastic load is scheduled, it will continue operation until the load finishes its work. Namely, all loads are used continuously without disruption. We predict one-day power demand, and all the demand of the elastic loads is satisfied in a day. First, we simulate the offline ELPA algorithm and compare its performance with unscheduled power consumption profile and the algorithm proposed in [3], referred as “SDWF.” Then we evaluate the performance when the constraints of **PU** and **PG** are applied. We also analyze the performance of OELPA approach and computation EOELPA algorithm. Finally, we compare the results of those two algorithms with the unscheduled case.

A. Offline ELPA Performance Evaluation

As shown in Algorithm 2, we flatten the power consumption around a reference level L , which is determined by the inelastic load demand and elastic load budget P_T . Inelastic load demand, which cannot be shifted, serves as a parameter. Fig. 6 shows the performance of scheduling the elastic power demand in three days by a single round using the proposed Algorithm 2, where the dashed curve and the dotted curve represent the allocated elastic load and fixed inelastic load, respectively. The overall load, denoted by the solid curve, is flattened through scheduling the elastic load to fill the valleys of the inelastic load. In the simulated range, the mean of the overall load power is 3952, with a standard deviation of 14.9, and peak to average ratio to be 1.004, which is very close to the unity (ideal case). We compare our results

TABLE II
SIMULATION RESULTS FOR OFFLINE ELPA,
SDWF, AND UNSCHEDULED CASE

	ELPA	SDWF	Unscheduled
Standard Deviation	14.9	268.4	1650.0
Peak-to-mean Ratio	1.004	1.105	1.681

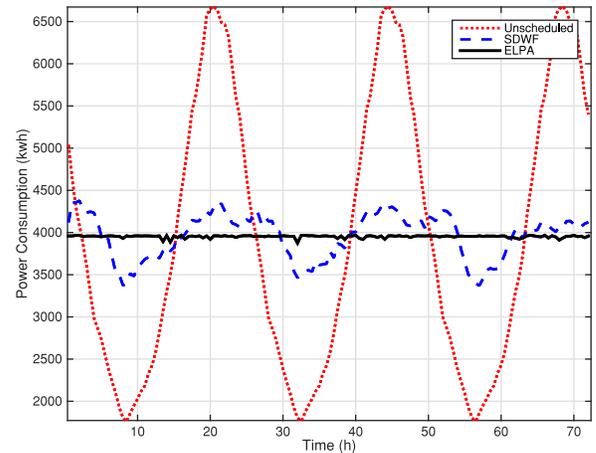


Fig. 7. Free-run (unscheduled) total power consumption and the power consumption achieved by offline ELPA and SDWF for three consecutive days.

with those of SDWF [3] and the unscheduled case with the same set of parameters. The result is shown in Table II and Fig. 7. Fig. 7 depicts the overall load for these three schemes: 1) proposed ELPA (solid curve); 2) SDWF (dashed curve); and 3) unscheduled operation (dotted curve). Improvement can be observed clearly by applying scheduling algorithms. As shown in Table II, both the overall load fluctuation and the peak-to-mean ratio of ELPA are significantly reduced over SDWF; while SDWF has also achieved great improvement over the unscheduled operation. It is ensured that our approach provides a more stable and an accurate solution for the elastic power allocation problem when all load information is available.

Furthermore, we apply the constraints of upper elastic power bound in time slot **PU** and upper elastic power bound in a group **PG** into the system. In Fig. 8, we set the **PU** in all time slots as 2000 kWh, where elastic power in all time slots cannot exceed that limit. We see the maximum of elastic load power is in 2000 kWh, and the reference level is also adjusted to a higher level to make sure all the user demand is satisfied in a day. In addition, we implement the upper elastic power bound in a group into the simulation. Fig. 9 shows the influence of group elastic power constraint **PG** in the algorithm. Fig. 9(a) shows elastic load distribution for four groups in 5 h period when **PG** is infinity. We see the elastic power is distributed by WF in group domain among four groups. The corresponding overall power profile is in Fig. 9(b). Fig. 9(c) and (d) shows the power allocation when **PG** in group 4 is set as 600 kWh in a day. We see the elastic load power for group 4 is upper bounded at 600 kWh. The elastic power is adjusted correspondingly in other groups to reach the overall reference level value in group domain.

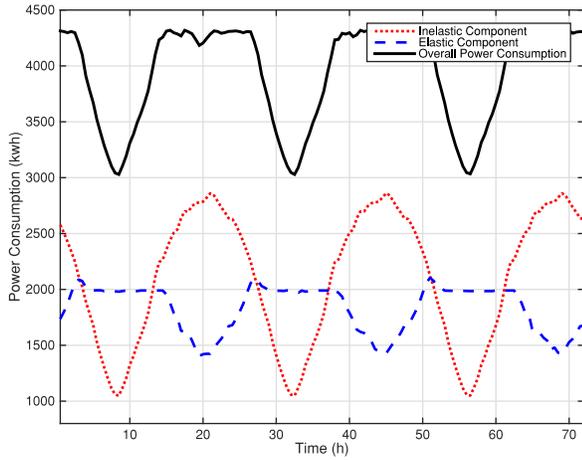


Fig. 8. Power consumption of inelastic, elastic, and total loads, where PU is 2000 kWh.

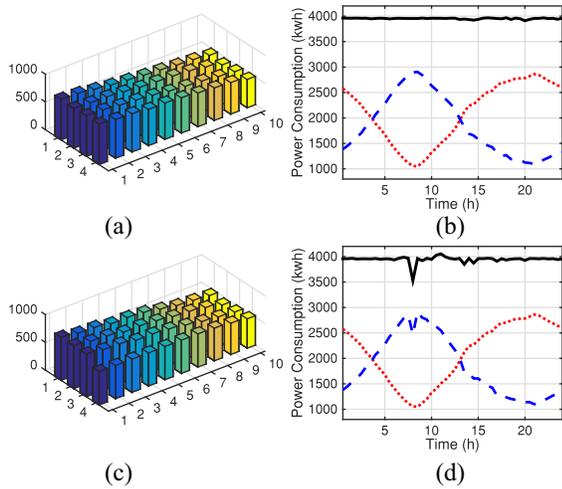


Fig. 9. (a) Elastic load power consumption in group view, where PG is infinity in 5 h, 30 min in a time slot. (b) Power consumption of inelastic, elastic, and total loads, where PG is infinity in a day. Legend is the same as Fig. 6. (c) Elastic load power consumption in group view, where group 4 PG is bounded in 600 kWh in 5 h, 30 min in a time slot. (d) Power consumption of inelastic, elastic, and total loads, where group 4 PG is bounded in 600 kWh in a day. Legend is the same as Fig. 6.

B. Online Approach Performance Evaluation

We first compare the overall power consumption scheduled by the proposed OELPA with unscheduled power profile. For the OELPA algorithm, we set the window size N to be 24 h. ϵ is set as 30 kWh and ξ is set as 100 kWh. When the power system runs to the k th time slot, the information beyond the k th time slot is unknown in power demand. Utilities offer the power in reference level, which is adjusted dynamically by Algorithm 3. Fig. 10 illustrates the power profile by OELPA (solid curves) and unscheduled operation (dotted curves) for five consecutive days. In Fig. 10(a), we see the reference level (dashed-dotted curve) keeps in less fluctuation (standard deviation = 57.09) when the total load budget is identical in these five days. The real power load is close to the reference level. Compared with the unscheduled result, OELPA achieved peak-cut and valley-filling (standard deviation from 1603 to 154).

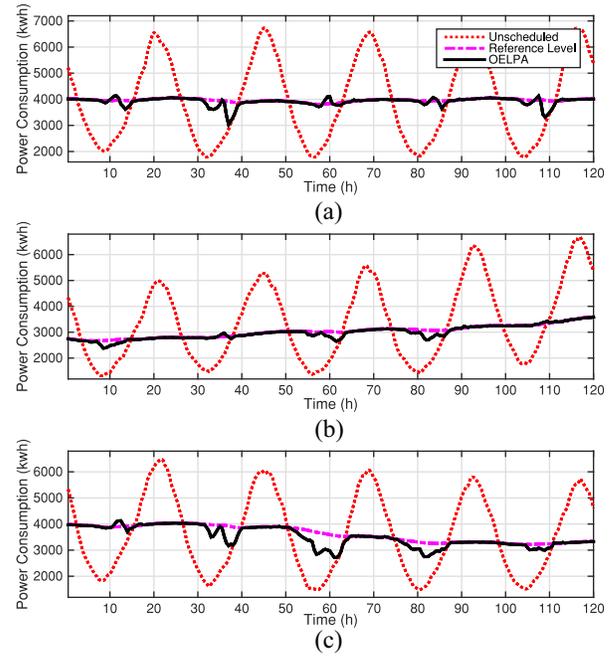


Fig. 10. Overall power consumption scheduled by OELPA and unscheduled power consumption for five days. (a) With same load budget. (b) Load budget increasing. (c) Load budget descending.

Then we apply the same demand adjustment methodology in [3] to show different behavior in five consecutive days. Fig. 10(b) is the result when inelastic load power consumption is raising by increasing the probability to start the load. Correspondingly, the reference level is increasing (from 2742 to 3572 kWh) when the overall power budget is increasing. 81.25% of the scheduled overall power consumption is close to the reference level, where tolerance is applied (100 kWh). The fluctuation is decreased from 1503 to 258 in the standard deviation. In Fig. 10(c), inelastic load power consumption attenuates with decreasing the probability to start the load. The reference level is adjusted from 3971 to 3023 kWh since the distributed power (reference level) is greater than the power consumed. The standard deviation is reduced from 1522 to 379. We see although the power demand is varied, the OELPA still achieves load balancing to accommodate the real-time demand.

We also evaluate the performance of computation efficient OELPA (EOELPA) algorithm. Fig. 11 illustrates the zoomed-in overall load of EOELPA and OELPA in five consecutive days, with the same system setting as in Fig. 10. The predicted reference level of EOELPA algorithm is also shown in Fig. 11. In EOELPA, the inelastic load consumption for next three time slots ($m = 3$) is predicted by cubic extrapolation in MATLAB. The system allocates elastic loads for those three slots offline. Then the system backs to online and the inelastic loads are predicted again after three time slots being elapsed. Comparing these two algorithms, EOELPA shows more ripples than OELPA results since the gap exists between the predicted inelastic load information and practical inelastic power consumption. However, the trend of those two algorithms' results is consistent, and the fluctuation is still much smaller than that of unscheduled operation.

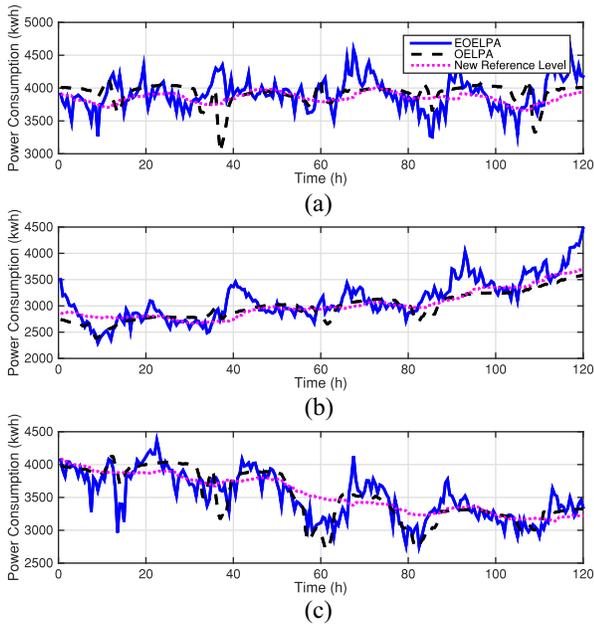


Fig. 11. Power consumption of inelastic, elastic, and total loads scheduled by EOELPA and OELPA for five days. (a) With same load budget. (b) Load budget increasing. (c) Load budget descending.

TABLE III
STANDARD DEVIATION OF OVERALL POWER CONSUMPTION
IN FIVE CONSECUTIVE DAYS (FIG. 11)

Loads Demand	OELPA	EOELPA	Unscheduled
Identical (a)	154	241	1603
Ascending (b)	258	395	1503
Descending (c)	379	351	1522

Table III shows the standard deviation of the overall load in Fig. 11 under three cases (OELPA, EOELPA, and unscheduled operation). The results of EOELPA is slightly floating on the results in OELPA because EOELPA allocates the loads according to prediction. Nevertheless, the results in EOELPA are around (1/4) of the standard deviation of the unscheduled case. Note that the reference level computation by EOELPA is reduced to (1/3) of computation by OELPA, so EOELPA is also a considerable solution to achieve load balancing when the system computation is limited.

Fig. 12 depicts the impact of the factor m selection in the EOELPA algorithm. With increased m values, the computing efficiency of the load balancing is improved but the power fluctuation raises correspondingly because of the prediction accuracy decreasing. Although the result of EOELPA has higher variance than the result of OELPA, especially in higher m values, the overall power fluctuation of EOELPA is still significantly reduced than the unscheduled case in the range of simulation as shown in Fig. 12. Overall, OELPA is more appropriate when the system emphasizes on the better performance with minimizing the power fluctuation in real time. Otherwise, if the communication resource is limited with large-scale computation in the system, EOELPA is more suitable for the solution with appropriate tolerance on the power fluctuation.

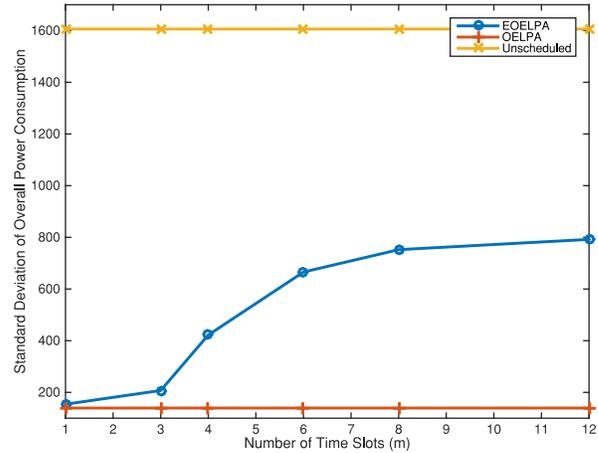


Fig. 12. Standard deviation versus parameter m for unscheduled, OELPA, and EOELPA schemes.

VI. CONCLUSION

In this paper, we presented three algorithms to solve the load balancing problem in the smart grid. First, we reviewed the load balancing model and its optimal offline solution investigated in our earlier work. Then we extend the solution to its online algorithm to compute and distribute the elastic load among the time slots and individuals based on the WF concept and load prediction. Furthermore, we proposed computation efficient online algorithm making use of prediction and extrapolation. Simulation results are presented to show that both online algorithms significantly reduce the load fluctuation. We also extended this paper by enabling different levels of constraints to play roles in the algorithms in order to improve the controllability in the smart grid. For future works, first, we need to consider the tolerance of the elastic load flexibility. Second, the research needs to be extended to get the precise prediction for reference levels in the online algorithm. The influence of the system when error prediction occurred is another scenario in our further study.

REFERENCES

- [1] Y. Wang, S. Mao, and R. M. Nelms, "Distributed online algorithm for optimal real-time energy distribution in the smart grid," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 70–80, Feb. 2014.
- [2] J. Pan *et al.*, "An Internet of Things framework for smart energy in buildings: Designs, prototype, and experiments," *IEEE Internet Things J.*, vol. 2, no. 6, pp. 527–537, Dec. 2015.
- [3] M. Shinwari, A. Youssef, and W. Hamouda, "A water-filling based scheduling algorithm for the smart grid," *IEEE Trans. Smart Grid*, vol. 3, no. 2, pp. 710–719, Jun. 2012.
- [4] T. Logenthiran, D. Srinivasan, and T. Z. Shun, "Demand side management in smart grid using heuristic optimization," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1244–1252, Sep. 2012.
- [5] Z. Fan, "A distributed demand response algorithm and its application to PHEV charging in smart grids," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1280–1290, Sep. 2012.
- [6] Z. Tan, P. Yang, and A. Nehorai, "An optimal and distributed demand response strategy with electric vehicles in the smart grid," *IEEE Trans. Smart Grid*, vol. 5, no. 2, pp. 861–869, Mar. 2014.
- [7] X. Wang and Q. Liang, "Energy management strategy for plug-in hybrid electric vehicles via bidirectional vehicle-to-grid," *IEEE Syst. J.*, to be published.
- [8] S. Bahrami and M. Parniani, "Game theoretic based charging strategy for plug-in hybrid electric vehicles," *IEEE Trans. Smart Grid*, vol. 5, no. 5, pp. 2368–2375, Sep. 2014.

- [9] H. K. Nguyen and J. B. Song, "Optimal charging and discharging for multiple PHEVs with demand side management in vehicle-to-building," *J. Commun. Netw.*, vol. 14, no. 6, pp. 662–671, Dec. 2012.
- [10] Q. Hu and F. Li, "Hardware design of smart home energy management system with dynamic price response," *IEEE Trans. Smart Grid*, vol. 4, no. 4, pp. 1878–1887, Dec. 2013.
- [11] L. P. Qian, Y. J. A. Zhang, J. Huang, and Y. Wu, "Demand response management via real-time electricity price control in smart grids," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 7, pp. 1268–1280, Jul. 2013.
- [12] Z. Chen, L. Wu, and Y. Fu, "Real-time price-based demand response management for residential appliances via stochastic optimization and robust optimization," *IEEE Trans. Smart Grid*, vol. 3, no. 4, pp. 1822–1831, Dec. 2012.
- [13] S. Althaher, P. Mancarella, and J. Mutale, "Automated demand response from home energy management system under dynamic pricing and power and comfort constraints," *IEEE Trans. Smart Grid*, vol. 6, no. 4, pp. 1874–1883, Jul. 2015.
- [14] P. Yang, G. Tang, and A. Nehorai, "A game-theoretic approach for optimal time-of-use electricity pricing," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 884–892, May 2013.
- [15] Y. M. Ding, S. H. Hong, and X. H. Li, "A demand response energy management scheme for industrial facilities in smart grid," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2257–2269, Nov. 2014.
- [16] S. Li and D. Zhang, "Developing smart and real-time demand response mechanism for residential energy consumers," in *Proc. Clemson Univ. Power Syst. Conf. (PSC)*, Clemson, SC, USA, 2014, pp. 1–5.
- [17] J. H. Yoon, R. Baldick, and A. Novoselac, "Dynamic demand response controller based on real-time retail price for residential buildings," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 121–129, Jan. 2014.
- [18] K. Zhou and L. Cai, "A dynamic water-filling method for real-time HVAC load control based on model predictive control," *IEEE Trans. Power Syst.*, vol. 30, no. 3, pp. 1405–1414, May 2015.
- [19] P. He, L. Zhao, S. Zhou, and Z. Niu, "Water-filling: A geometric approach and its application to solve generalized radio resource allocation problems," *IEEE Trans. Wireless Commun.*, vol. 12, no. 7, pp. 3637–3647, Jul. 2013.
- [20] D. Li, S. K. Jayaweera, O. Lavrova, and R. Jordan, "Load management for price-based demand response scheduling—A block scheduling model," in *Proc. Int. Conf. Renew. Energy Power Qual. (ICREPO)*, Canary Islands, Spain, 2011, pp. 1–6.
- [21] D. Li and S. K. Jayaweera, "Uncertainty modeling and price-based demand response scheduling in smart grid," *IEEE Syst. J.*, to be published.
- [22] Y. Mou, H. Xing, Z. Lin, and M. Fu, "Decentralized optimal demand-side management for PHEV charging in a smart grid," *IEEE Trans. Smart Grid*, vol. 6, no. 2, pp. 726–736, Mar. 2015.
- [23] P. He, M. S. Li, L. Zhao, B. Venkatesh, and H. W. Li, "Water-filling exact solutions for load balancing of smart power grid systems," *IEEE Trans. Smart Grid*, to be published.



Mushu Li received the B.S. degree from the University of Ontario Institute of Technology, Oshawa, ON, Canada, in 2015. She is currently pursuing the M.A.Sci. degree at Ryerson University, Toronto, ON, Canada.

Her current research interests include load balancing for smart grid, EV charging scheduling, and system optimizations.

Ms. Li was a recipient of the Ontario Graduate Scholarship in 2015 and 2016, respectively, and the Aditya Jha Diversity Award from Ryerson University

to recognize her academic achievement in 2016.



Peter He (S'14–M'15) received the M.A.Sci. degree in electrical engineering from McMaster University, Hamilton, ON, Canada, in 2009, and the Ph.D. degree in electrical engineering from Ryerson University, Toronto, ON, Canada, in 2015.

Since then, he has been a Post-Doctoral Research Fellow with the Broadband Communications Research Group Laboratory, University of Waterloo, Toronto, ON, Canada, and Ryerson University. His current research interests include union of large-scale optimization, information theory, communications, and smart power grid.

Dr. He was a recipient of the Graduate Research Excellence Award from the Department of Electrical and Computer Engineering, Ryerson University, the Ontario Graduate Scholarship awards, the Natural Sciences and Engineering Research Council Post-Doctoral Fellowship in 2016, and the Best Paper Award of the 2013 International Conference on Wireless Communications and Signal Processing.



Lian Zhao (S'99–M'03–SM'06) received the Ph.D. degree from the University of Waterloo, Toronto, ON, Canada, in 2002.

She joined the Department of Electrical and Computer Engineering, Ryerson University, Toronto, in 2003, where she became a Professor in 2014. Her current research interests include wireless communications, radio resource management, power control, cognitive radio and cooperative communications, and optimization for complicated systems.

Dr. Zhao is a Licensed Professional Engineer in the Province of Ontario. She was a recipient of the Best Land Transportation Paper Award from the IEEE Vehicular Technology Society in 2016, the Top 15 Editor in 2015 for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the Best Paper Award from the 2013 International Conference on Wireless Communications and Signal Processing and Best Student Paper Award (with her student) from Chinacom in 2011, the Ryerson Faculty Merit Award in 2005 and 2007, the Canada Foundation for Innovation New Opportunity Research Award in 2005, and the Early Tenure and promotion to Associate Professor in 2006. She has been an Editor of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY since 2013, the Workshop Co-Chair for IEEE/CIC ICC 2015, the Local Arrangement Co-Chair for IEEE Infocom 2014, and the Co-Chair for IEEE Global Communications Conference 2013 Communication Theory Symposium. She served as a Committee Member for Natural Science and Engineering Research Council of Canada Discovery Grants Evaluation Group for Electrical and Computer Engineering from 2015 to 2018 and an Associate Chair with the Department of Electrical and Computer Engineering, Ryerson University from 2013 to 2015.