Water-filling Exact Solutions for Load Balancing of Smart Power Grid Systems

Peter He, Mushu Li, Lian Zhao, Bala Venkatesh and Hongwei Li

***Abstract*—For the demand side management (DSM), the elastic power loads can be scheduled to achieve load balancing and to minimize the fluctuation of the overall load. In a process of power supply, the inelastic power loads can be regarded as a group of parameters. Then the demand response (DR) can be utilized to realize the optimal allocation of the elastic power loads. Importantly, the power load balancing can reduce the cost from power generations since no cost is spent on the requirement of frequency control. This paper focuses on the optimal allocation of the elastic load for load balancing. A load balancing problem, with the node and the group power upper bound constraints, is investigated in this paper. Water-filling algorithm is proposed for exactly and efficiently computing the optimal solutions to the power load balancing optimization problem with lower degree polynomial computational complexity. The proposed algorithm can be applied to solve the target optimization problems with large-scale due to utilization of non-derivative water-filling method. To the best of the authors' knowledge, there is no existing algorithm reported in the open literature that can compute the exact solution to the target problem.**

Keywords

Demand side management (DSM), energy demand management, power load allocation, optimal load allocation, optimization theory and methods, water-filling algorithm.

I. Introduction

Electricity has been playing a critical important role in our daily life. Current electricity grid was designed over half century ago with rather simple one-way electricity flow requirement at that time. This grid system has been showing some deficiencies nowadays. With the advances of information technology and more complicated electricity demand, the concept of "Smart Grid" has been evolved as an enhancement of the conventional power grid [1]. The smart grid is capable of managing the electricity system more efficiently with the application of advanced information technologies.

Unlike other commodities, electricity cannot be stored by most users when it is generated before the mass commercialization of electrical storage devices. Electricity demand/load can vary significantly over time, as reported, for example, in [1], [2]. This results in electricity generation to vary significantly if no strategy is taken. In a traditional grid, the electricity generation tries to match the peak demand, or the average demand with extra standby generators. Both approaches have been shown to be not only expensive but also impractical [1]. The wide spread distribution of the total demand requires the electricity suppliers to invest more capital for installing and running those standby generators which are utilized to reduce failure rate; and as a result, the consumers' bills reflected on the expensive cost to run those extra standby generators. Therefore, smoothing or flatting power generation, to reduce the dependency on those standby generators, is one of the most important objectives for smart grid. In addition, it is seen that to have the generators' stable and smooth working condition is important, under the load constraints for nodes, groups, epochs, and load supply capability as well [1].

Demand side management (DSM) or energy demand management [3], aims to match electricity supplies with demand. Strategies for load balancing to flatten the overall demand/load have attracted significant research interests in the context of smart grid, for example, load balancing by regulating PHEV (Plug-in Hybrid Electric Vehicles, a typical kind of load with large amount of electricity requirement) charging [4]-[6]; load balancing by adjusting time-depend on electricity pricing or applying game approach [4]- [14]. Generally, the electric load is categorized as an elastic load and an inelastic load. Elastic loads, such as home appliances like washers and dryers, can be scheduled to switch on at appropriate time to achieve overall load balancing. Inelastic loads, such as lighting requirements, cannot be scheduled, and are considered to be a group of parameters of the system. The general concept is to shift elastic loads from high-demand peak times to low-demand off-peak times. To schedule the elastic load, the prediction of the inelastic loads has been developed in [15] and [16].

In this paper, we focus on using a well-known optimization tool: water-filling algorithm to achieve load balancing. Water-filling has been widely used for power allocation in communication systems to maximize system throughput or sum date rate, for example [17], [18] and the references cited in. In recent years, water-filling has also been applied in a smart grid system to solve load balancing problems [2], [19], [20], [21], [22].

In the early work [2], the water-filling concept was applied to schedule the general elastic loads from an analogue approach. The defined problem and solution from this approach was investigated without a solid mathematical model of the load balancing problem. In [19], load management problem was modelled as a typically simplest form of the water-filling problem, *i.e.*, the load to be nonnegative 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 [20], on top of the typical load balancing water-filling problem, delay cost was added as another consideration factor; in [21], the work was focused on reducing the fluctuation of the power load by controlling all the "soft" or elastic loads drawn by HVAC (Heating, Ventilating, and Air Conditioning) systems with

Peter He, Mushu Li, Lian Zhao and Bala Venkatesh are with Dept. of Electrical and Computer Engineering, Ryerson University, Ontario, Canada. E-mail: phe, mushu1.li, lzhao, bala@ee.ryerson.ca.

Hongwei Li is with School of Computer Science and Engineering, University of Electronic Science and Technology of China. E-mail: hongweili@uestc.edu.cn.

comfortability considerations; in [22], the authors formulated an overall load fluctuation minimization problem and applied water-filling principle to solve the target problem.

In these previous works, the problem was solved by the conventional Lagrangian approach to obtain the solution. Generally speaking, there is no closed form solution to a general water-filling problem by this approach; and normally the procedure results in an approximate solution by numerical methods. In contrast to the previous works, we modelled the problem in a more general form and exploited our proposed GWF (geometric water-filling) [18] approach to provide a closed-form, exact valued and strictly proven optimal solution, and a finite amount of computation with a low degree polynomial computational complexity.

In this paper, we consider the case that the users (or nodes) are grouped into different groups. Each user has an individual peak power constraint and each group has a group peak power constraint due to different restrictions of the power loads used by the users and the groups. Adding these restrictions or bounds has its practical significance. It enables transmission lines and transformers not to exceed the corresponding delivery capacity. Some or all of the peak load constraints can be relaxed by assigning values large enough to the corresponding load upper bounds. Thus, our presented problem and proposed algorithm are more general than previous models. While adding this set of constraints significantly escalate the complexity level to solve the optimization problem. Due to introduction of these constraints, the "water level" is no longer a unique level. The conventional water level searching approach shows its limits in solving this generalized problem. Furthermore, the proposed algorithm uses non-derivative methods, so that the implementation of the proposed algorithm invokes neither the derivative nor the gradient operations. This has avoided computing the inverse of Hessian matrices via existing numerical optimization methods. This feature is especially suitable for solving large-scale load balancing problems.

II. Problem Statement

For convenience and without loss of generality, the process is assumed to be a discrete time process. Table I is a list of the variables and abbreviations used in our analysis, with appropriate SI units (International System of Units) defined where needed.

It has been assumed that the power loads for users are divided into two categories: elastic and inelastic power loads. As mentioned above, for DSM, it means the modification of user demand for energy through various methods [7] to encourage users to use less energy during peak hours, or to move the time of energy use to off-peak times such as night time and weekends [3]. *As an assumption in this paper, the inelastic loads can be predicted as parameters, like [2], [21], and [22].*

Let K be the projected time window length with the time unit of epoch. Assume that there are J users, each of which utilizes the elastic load and inelastic load with the index j. According to the priority and the structure of power systems

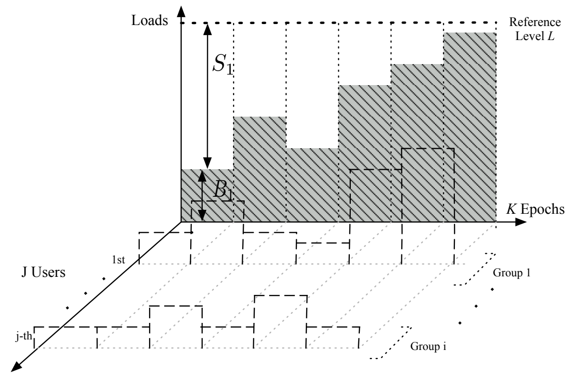


Fig. 1. System Model: 3-dimensional power allocation problem

to distribute the power loads, we can categorize the users into I groups. For example, the power supply and its infrastructure of a residential community can be a different group of users from an industrial center. Let $\{\chi_i\}_{i=1}^{I}$ be a partition of the set of $\{j\}_{j=1}^{J}$, where χ_i is the set of indexes of all the members or users for group i for $1 \le i \le I$. Cardinality of the set, χ_i, is denoted by $|\chi_i|$. Thus, we may list all the members of χ_i by $\{i_1, \cdots, i_{|\chi_i|}\}$ to emphasize the index of the ith group.

The inelastic loads are denoted by a matrix $\mathbf{A}$, where the (k, j)th element, denoted by $A_{k,j}$, represents the observed or predicted inelastic load of the jth user at the beginning time of the kth epoch, as illustrated in the K-J plane in Fig. 1. Similarly, the elastic loads are denoted by another matrix $\mathbf{R}$. The elements of the matrices, $A_{k,j}$ and $R_{k,j}$, represent the power consumption of inelastic and elastic loads of the jth user in the kth time slot respectively. $A_{k,j}$ can be predicted from power utility, which is shown as the wide-dashed lines in Fig. 1. The shadow areas, B_k, in Fig. 1 are the projection of the inelastic load for all the users at the kth time slot. L is an optimization variable as reference level of overall power consumption, where water-filling concept is applied.

If user j does not utilize the inelastic load at epoch k, $A_{k,j}$ can take the value of zero. $R_{k,j} = 0$ is similarly understood. In this paper, J can be selected large enough, $J \gg 0$. This assumption accounts for the proposed algorithm being capable of balancing the load in large-scale systems, especially for metropolises. We can form the following optimization problem for minimizing power load fluctuation,

$$\begin{array}{ll} \min_{\{\mathbf{R},L\}} & \sum_{k=1}^{K} \left[\sum_{i=1}^{I} \sum_{j\in\chi_i} (A_{k,j} + R_{k,j}) - L \right]^2 \\ \text{subject to:} & 0 \le R_{k,j} \le P_{k,j},\ j = 1, \ldots, J, \forall k; \\ & \sum_{j\in\chi_i} R_{k,j} \le PG_{k,i},\ \forall i, k; \\ & \sum_{j=1}^{J} R_{k,j} \le PU_k, \forall k; \\ & \sum_{k=1}^{K} \sum_{j=1}^{J} R_{k,j} = P_T; \\ & L \ge 0, \end{array} \tag{1}$$

where $P_{k,j}$ is non-negative and it stands for the jth user elastic load upper bound at the kth epoch; $PG_{k,i}$ is non-negative and it stands for the load upper bound for group i at epoch k; PU_k is non-negative and it stands for the elastic load upper bound at epoch k; and P_T, which is the sum of all the loads, is non-

TABLE I
LIST OF VARIABLES AND ABBREVIATIONS

Group	Variable	Meaning
Index group	k	index of epoch, or index of the time slot, for $k = 1, \ldots, K$
	i	index of group, for $i = 1, \ldots, I$
	j	index of user (or node), for $j = 1, \ldots, J$
	χ_i	the ith group
Load group	$\mathbf{A}$	inelastic load matrix. The (k, j)th element, $A_{k,j}$, denotes inelastic load of user j at epoch k
	$\mathbf{B}$	inelastic load vector, obtained by sums of $\mathbf{A}$ over rows. The ith element, B_k, denotes the total inelastic load at epoch $k, 1 \leq k \leq K$
	$\mathbf{R}$	elastic load matrix. The (k, j)th element, $R_{k,j}$, denotes the allocated elastic load of user j at epoch k, $1 \leq k \leq K, 1 \leq j \leq J$
	$\mathbf{S}$	elastic load vector. The kth element, S_k, denotes the allocated total elastic load at epoch $k, 1 \leq k \leq K$
Constraint group	$\mathbf{P}$	individual load upper bound (or peak) matrix. The (k, j)th element, $P_{k,j}$, denotes the load upper bound for the jth user in the kth epoch
	$\mathbf{PG}$	group load upper bound) matrix. The (k, i)th element, $PG_{k,i}$, denotes the group load upper bound for the ith group in the kth epoch $(1 \leq i \leq I)$
	$\mathbf{PU}$	load upper bound vector for epochs. The kth element, PU_k, denotes the load upper bound at the kth epoch
	P_T	total load budget or supply capability for entire elastic power loads over all the epochs, *i.e.*, a sum of loads
	$P_2(n)$	middle variable, total elastic load allocated above step n in water tank illustration
	L	reference horizon, reference level
	$\mathcal{L}$	Lagrange function
Abbreviations	DSM	Demand side management
	PBLA1	Preparation for Basic load allocation 1 (BLA1)
	BLA1	Basic load allocation 1, to be generalized into BLA2
	BLA2	generalized Basic load allocation for ELPA
	ELPA	Elastic load power allocation for target problem (1)
	WF	water-filling, often means conventional water-filling
	GWF	Geometric water-filling, an exact and efficient algorithm [18]
	the k^* search	a method to search the step of water level in water-filling
	KKT conditions	a group of optimality conditions, named after W. Karush, H. Kuhn and A. Tucker

negative and it stands for the total elastic load budget for all the epochs and all the J nodes. Under the given inelastic loads, we can obtain the total load budget from P_T. Plainly, (1) has the node and the group power upper bound constraints, expressed in the first line and the second line of the constraint part. The optimization variable L is a reference level. Since this L does not depend on k, it just reflects a "flatten" requirement. Problem in (1) can be regarded as minimizing the sum of squared errors over all the epochs. The error is the difference between the total loads at an epoch and the reference level L. Obviously, load balancing can be achieved by minimizing this objective function. It is seen that suitable physical units may be chosen for all the quantities in Problem (1).

III. Water-filling: Basic Load Allocation Algorithms

In this section, we introduce two Basic Load Allocation algorithms (BLA1 and BLA2) using the water-filling principle. These two algorithms provide mappings and serve as functional blocks for solving the target problem.

A. Basic Load Allocation Water-filling Problem 1: BLA1

1) Preparation of BLA1: Assume there are K epochs. The inelastic load sequence is represented by the monotonically increasing sequence $\{B_k, 1 \leq k \leq K\}$ (the indexes can be arbitrarily renumbered to satisfy this condition) which can be illustrated by the stairs inside a water tank in Fig. 2. The total available elastic load is P_T. Find the elastic load allocation $\{S_k\}$ to balance the overall loads, through solving the following optimization problem:

$$\begin{array}{lll} \min_{\{\{S_k\},L\}} & \sum_{k=1}^{K}(S_k + B_k - L)^2 & \\ \text{subject to:} & 0 \leq S_k, \ \forall k; & \\ & \sum_{k=1}^{K} S_k = P_T; & (2) \\ & 0 \leq L. & \end{array}$$

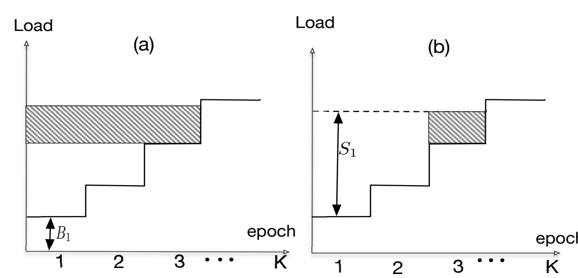


Fig. 2. Preparation of Basic Load Water-filling Problem 1: PBLA1

This is a load allocation problem with the sum load constraint. Similar problems that maximize the sum rate, have been extensively discussed in [18]. Looking for the optimal solution, water-filling, as an approach, can be imagined into pouring P_T units of water into the tank. We may use a similar approach to the geometric water-filling in [18], to solve (2).

We introduce four variables for solving (2). The first variable, $P_2(k)$, is illustrated by the water volume (load) above the kth step, as shown in the shadowed area in Fig. 2(a) when

$k = 3$, where the surface denotes the water-level, and can formally be defined as

$$P_2(k) = \left\{ P_T - \left[\sum_{l=1}^{K-1} (B_k - B_l) \right] \right\}^+, \quad k = 1, 2, \ldots, K. \tag{3}$$

Note that a reminder of the definition of a special case for the summation is: $\sum_{i=m}^{n} g_i = 0$, as $m > n$, where $\{g_i\}$ is a general sequence of numbers. Then the maximum index k^* of keeping $P_2(k)$ positive is given by

$$k^* = \max \left\{ k \middle| P_2(k) > 0, \ 1 \le k \le K \right\}, \tag{4}$$

where k^* can be understood as the highest step under water. The allocated load for this index is

$$S_{k^*} = \frac{1}{k^*} P_2(k^*). \tag{5}$$

S_{k^*} is illustrated by the shadowed area in Fig. 2(b) when $k^* = 3$. Then the loads of all the epochs are

$$S_k = \begin{cases} S_{k^*} + (B_{k^*} - B_k), & 1 \le k \le k^* \\ 0, & k^* < k \le K. \end{cases} \tag{6}$$

Solution (6) is obtained by observing Fig. 2(b). When the step is below the water level ($1 \le k \le k^*$), the allocated soft load is the sum of both the shadow box (S_{k^*}) and the difference, between the step height and the k^*th step. When the step is above the water level, simply there is no soft load being allocated.

Therefore, for the given $(\mathbf{B}, P_T)$, the allocation of the soft load, $\{S_k\}$, can be obtained from (6). This relationship can formally be written as a mapping or algorithm:

$$(\mathbf{S}, L) = \text{PBLA1}(\mathbf{B}, P_T) \tag{7}$$

where boldface $\mathbf{S}$ and $\mathbf{B}$ are the corresponding elastic load allocation vector and inelastic load vector. Also, $L = \frac{1}{K}(\sum_{k=1}^{K} B_k + P_T)$. $\mathbf{S}$ can be regarded as the first part of the output of PBLA1, denoted as $\mathbf{S} = \text{PBLA1}(\mathbf{B}, P_T)|_I$. Note that without confusion, the problem and this mapping take the same name for convenience. This mapping can provide an optimal solution to the problem (2).

The algorithm: PBLA1 is formally stated as below.

Algorithm 1 Preparation for the Basic Load Allocation Algorithm 1 (PBLA1)

Input: $\mathbf{B}$, P_T
Output: $\mathbf{S}$, L
1: Solve the problem (3) - (6) to obtain $\{S_k\}_{k=1}^{K}$ and L via $\{B_k\}_{k=1}^{K}$ and P_T.

The guaranteed optimality of PBLA1 is stated by the following proposition.

Proposition 3.1: Output of the mapping PBLA1 is an optimal solution to (2).

Proof. Since (2) has a non-empty bounded sub-level set, the set of optimal solutions is non-empty. For any optimal solution to (2), this optimal solution and its optimal dual variables satisfy the Karush-Kuhn-Tucker (KKT) conditions (refer to [23]) of (2). For the optimization variable L and the constraint of $L \ge 0$ in (2), the related KKT conditions, stemming from the Lagrange function of (2), are:

$$\begin{cases} 2\sum_{k=1}^{K}(B_k + S_k) - 2KL + \psi = 0; \\ L \ge 0; \psi \ge 0; \psi \cdot L = 0, \end{cases} \tag{8}$$

where the Lagrange function of (2) is

$$\begin{aligned} &\mathcal{L}(\{S_k\}, L; \{\mu_k\}, \psi, \lambda) = \textstyle\sum_{k=1}^{K}(B_k + S_k - L)^2 - \\ &\textstyle\sum_{k=1}^{K} \mu_k S_k - \lambda(P_T - \sum_{k=1}^{K} S_k) - \psi \cdot L, \end{aligned} \tag{9}$$

where the dual variables $\{\mu_k\}$, ψ and λ mentioned above correspond to the constraints of $\{S_k \ge 0\}$, $L \ge 0$ and $\sum_{k=1}^{K} S_k = P_T$ respectively.

It is seen that $\psi = 0$. This point can be proven by an indirect approach. Let us assume to the contrary that $\psi > 0$. Then the last equality $\psi \cdot L = 0$ in the KKT conditions implies $L = 0$. Thus, the first equality in the KKT conditions above is simplified by $2\sum_{k=1}(B_k + S_k) + \psi = 0$. Due to non-negativeness of $B_k + S_k$ and $\psi > 0$, the contradiction of $2\sum_{k=1}(B_k + S_k) + \psi = 0$ and $2\sum_{k=1}(B_k + S_k) + \psi > 0$ holding at the same time, is obtained. Therefore, $\psi = 0$. As a result, from the first equality in the KKT conditions, it can equivalently be transformed into the

$$L \;= \tfrac{1}{K}\textstyle\sum_{k=1}^{K}(B_k + S_k) = \tfrac{1}{K}(\sum_{k=1}^{K} B_k + P_T). \tag{10}$$

To find other optimal dual variables, first, construct $\lambda = 2[L - (S_{k^*} + B_{k^*})]$. It has been obtained that $\psi = 0$, *i.e.*, ψ is equal to zero. Then, let the dual variable $\mu_k = 0$ for $1 \le k \le k^*$; while $\mu_k = 2[(S_k + B_k) - (S_{k^*} + B_{k^*})] \ge 0$ for $k^* < k \le K$.

The constructed dual variables, and the solution by PBLA1, including (6), are observed. Indeed, they are the optimal dual variables and the optimal solution to problem (2), since they satisfy the KKT conditions of the convex optimization problem (2). Note that the constraint qualification of problem (2) holds.

Therefore, Proposition 3.1 is proved. □

The water level of (2) is $S_{k^*} + B_{k^*}$ due to $S_{k^*} + B_{k^*} = S_i + B_i$, for $1 \le i \le k^*$. Note that, although the problem in [18] and (2) have some similarities, there is an important difference between them. The water level, a key concept, in [18] is the **reciprocal** of the optimal dual variable λ that corresponds to the total sum power constraints; while the water level for (2) is an **affine function** in the optimal dual variable λ. This difference will be further explained in Numerical Examples section.

2) Statement and Optimality of BLA1: The Basic Load Allocation water-filling Problem 1 (BLA1) can be expressed as:

$$\begin{array}{ll} \min_{\{\{S_k\}, L\}} & \sum_{k \in \{1,\ldots,K\}} (S_k + B_k - L)^2 \\ \text{subject to:} & 0 \le S_k \le PU_k, \ \forall k \in \{1, \ldots, K\}; \\ & \sum_{k \in \{1,\ldots,K\}} S_k = P_T; \\ & 0 \le L. \end{array} \tag{11}$$

Compared to PBLA1, individual upper power constraints, PU_k, are added in the first constraint line in (11). This difference is illustrated in Fig. 3. In Fig. 3(a), it is assumed that PU_k is infinity and the load vector $\mathbf{B}$ is sorted as a monotonically increasing sequence, in the same way as Fig. 2. Fig. 3(b) indicates the system with load upper bound constrained by PU_k , as shown in the height of the shadow areas. S_k can not be greater than the corresponding PU_k.

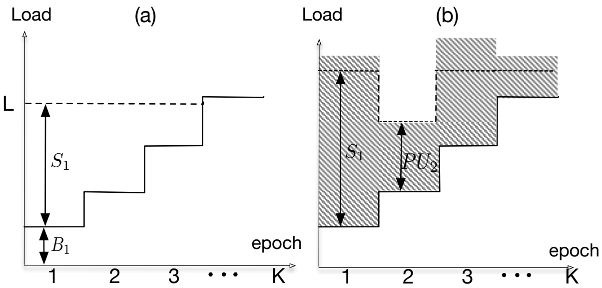


Fig. 3. Illustration of Basic Load Allocation (BLA)

To solve problem (11), Basic Load Allocation 1 (BLA1) is stated as follows, which is also called Algorithm 2. Four concepts are generalized to solve problem BLA1, and they are stated as follows under the given E:

$$P_2(k) = \left\{ P_T - \left[\sum_{l=1}^{|E|-1} (B_{t_k} - B_{t_l}) \right] \right\}^+, \quad k = 1, \ldots, |E|; \tag{12}$$

the maximum index k^* of keeping $P_2(k)$ positive is given by

$$k^* = \max \left\{ k \middle| P_2(k) > 0, \ \ 1 \le k \le |E| \right\}; \tag{13}$$

the (power) load level for this index is

$$S_{i_{k^*}} = \frac{1}{k^*} P_2(k^*); \tag{14}$$

and then the generalized (power) load levels of all the epochs are given by

$$S_{t_k} = \begin{cases} S_{t_{k^*}} + (B_{t_{k^*}} - B_{t_k}), & 1 \le k \le k^* \\ 0, & k^* < k \le |E|. \end{cases} \tag{15}$$

Note that k^* in (13) can use the k^* discussed above. For the given E and its restriction on $\mathbf{B}$ and P_T, the $\{S_k\}_{k \in E}$ in (15) can be obtained. The restriction is still labelled as $\mathbf{B}$, without loss of generality. Thus, this relationship can also formally be written as a mapping:

$$(\mathbf{S}, L) = \text{BLA1}(\mathbf{B}, \mathbf{PU}, P_T). \tag{16}$$

The algorithm description of BLA1 is stated below.

For Algorithm BLA1, we have:

Proposition 3.2: Output of Algorithm BLA1 is an optimal solution to (11).

The proof of Proposition 3.2 is similar to that of Proposition 3.1.

B. Basic Load Allocation Water-filling Problem 2: BLA2

The second basic problem is described below:

$$\begin{array}{ll} \min_{\{\{S_k\}, L\}} & \sum_{k \in E} (S_k + B_k - L)^2 \\ \text{subject to:} & 0 \le S_k \le PU_k, \ \forall k \in E; \\ & \sum_{k \in E} S_k = P_T; \\ & 0 \le L. \end{array} \tag{17}$$

Algorithm 2 Basic Load Allocation Algorithm 1 (BLA1)

Input: $\mathbf{B}$, $\mathbf{PU}$, P_T, $\mathbf{E} = \{1, \ldots, K\}$
Output: $\mathbf{S}$, L

while $E \neq \emptyset$ **do**
  Compute (12)-(15) to obtain $\{S_k\}_{k \in E}$ and L via $\{B_k\}_{k \in E}$ and P_T.
  $\Lambda \leftarrow \{k \mid S_k > PU_k, \ k \in E\}$.
  if $\Lambda \neq \emptyset$ **then**
    if $k \in \Lambda$ **then**
      $S_k = PU_k$.
    end if
    $E \leftarrow E \setminus \Lambda$, $P_T = P_T - \sum_{k \in \Lambda} PU_k$.
  else
    Output $\{S_k\}$ as $k \in E$.
  end if
end while

Compared with BLA1, BLA2 only generalizes the initial index set of E from $\{1, \ldots, K\}$ of BLA1 *to* the more general increasing sequence which is taken from the set of natural numbers. BLA2 does not have an essential change over BLA1. The motivation to introduce BLA2 is to make the statement of the following proposed algorithm to be more concise.

Thus, the second mapping, from $(\mathbf{B}, \mathbf{PU}, P_T)$ to $(\mathbf{S}, L)$, is obtained. It can be read as:

$$(\mathbf{S}, L) = \text{BLA2}(\mathbf{B}, \mathbf{PU}, P_T) \tag{18}$$

where the boldface $\mathbf{S}, \mathbf{B}$, and $\mathbf{PU}$ are the allocated elastic load vector, the inelastic power vector, and the group load peak or upper bound vector, respectively, with $L = \frac{1}{K}(\sum_{k=1}^{K} B_k + P_T)$.

We have the following proposition for the mapping or algorithm of BLA2.

Proposition 3.3: Output of Algorithm BLA2 is an optimal solution to (17).

Similarly to proving optimality of the output for PBLA1, we can also prove that the output of BLA2 is an optimal solution to (17). Hence, its proof is omitted for brevity.

IV. Proposed ELPA to Solve the Target Problem

A. Statement of Algorithm: ELPA

In this section, we discuss the proposed Elastic Power Allocation algorithm (ELPA), to solve the target problem (1).

In order to have a feasible solution to problem (1), assume $\sum_{k=1}^{K} \sum_{j=1}^{J} P_{k,j} \ge P_T$, $\sum_{k=1}^{I} PU_k \ge P_T$ and $\sum_{k=1}^{K} \sum_{i=1}^{I} PG_{k,i} \ge P_T$. If this assumption does not hold, we can assign $P_T = \min\{\sum_{k=1}^{K} \sum_{j=1}^{J} P_{k,j}, \sum_{k=1}^{K} \sum_{i=1}^{I} PG_{k,i}, \sum_{k=1}^{K} PU_k\}$, to make the feasible set non-empty. In addition, it is seen that we may assume that $\sum_{j \in \chi_i} P_{k,j} \ge PG_{k,i}, \forall k, i$, or else $PG_{k,i}$ is assigned by $\sum_{j \in \chi_i} P_{k,j}$. Similarly, $\sum_{i=1}^{I} PG_{k,i} \ge PU_k$ can be assumed without loss of generality. Then, it is seen that the solution to problem (1) exists, although the feasible set is unbounded. The reason is stated as follows: since the zero vector is a feasible point, the feasible set of (1) is not empty. Further, it is also seen that there exists a non-empty bounded sub-level set, which includes the zero vector. Since the intersection of the feasible set and the sub-level set is a compact set, and the object function of problem (1) is continuous, then an optimal solution to problem (1) exists. If $P_T = 0$, the unique optimal solution is the zero vector; Else, $P_T > 0$ implies that problem (1) has an optimal solution.

The proposed ELPA is a synthesis of previous BLA1 and BLA2, to directly compute the optimal solutions to the target problem. Using BLA1 and BLA2, ELPA can be described as below.

Algorithm ELPA:

Input: the inelastic loads $\mathbf{A}$, the individual load upper bounds $\mathbf{P}$, the group load upper bounds $\mathbf{PG}$, the epoch load upper bounds $\mathbf{PU}$, the index set $E = \{1, 2, \ldots, K\}$, the group partition $\{\chi\}_{i=1}^{I}$, and the sum load constraint which is equal to P_T.

1) Form vector $\mathbf{B}$ by computing the sum of matrix $\mathbf{A}$ over its rows. Then call PBLA1 to solve load allocation vector $\mathbf{S}$, by

$$(\mathbf{S}, L) = \text{PBLA1}(\mathbf{B}, P_T)$$

where $L = (\sum_{k,j} A_{k,j} + P_T)/K$.

2) The set Λ is defined by the set $\{k | S_k > PU_k, 1 \leq k \leq K\}$. If Λ is the empty set, solve the elastic load matrix, $\mathbf{R}$, by calling the mapping BLA2 twice. For any $k \in E$, the first time calling BLA2 completes the allocation for the total elastic loads for the groups,

$$\{\overline{R}_{k,i}\}_{i=1}^{I} = \text{BLA2}(\{\sum_{j \in \chi_i} A_{k,j}\}_{i=1}^{I}, \{PG_{k,i}\}_{i=1}^{I}, S_k)|_I,$$

where $\overline{R}_{k,i}$ denotes the total elastic loads for the ith group at the kth epoch. The second time calling BLA2 allocates the elastic load for the specific users within each group,

$$\{R_{k,j}\}_{j \in \chi_i} = \text{BLA2}\left(\{A_{k,j}\}_{j \in \chi_i}, \{P_{k,j}\}_{j \in \chi_i}, \overline{R}_{k,i}\right)|_I,$$

where $R_{k,j}$ is the allocated elastic load for the jth user at the kth epoch. Step 2) solves the following problem:

$$\begin{cases} \sum_{n=1}^{N} R_{k,n} = S_k, \forall k; \\ \sum_{j \in \chi_i} R_{k,j} \leq PG_{k,i}, \forall k, i; \\ 0 \leq R_{k,n} \leq P_{k,n}, \end{cases} \tag{19}$$

and then output the solution $\mathbf{R}$; else go to 3).

3) For any $k \in \Lambda$, the total group elastic load is allocated with its corresponding upper bound. The first calling of BLA2 completes elastic load for the ith group at the kth epoch as

$$\{\overline{R}_{k,i}\}_{i=1}^{I} = \text{BLA2}(\{\sum_{j \in \chi_i} A_{k,j}\}_{i=1}^{I}, \{PG_{k,i}\}_{i=1}^{I}, PU_k)|_I;$$

and then for any i,

$$\{R_{k,j}\}_{j \in \chi_i} = \text{BLA2}\left(\{A_{k,j}\}_{j \in \chi_i}, \{P_{k,j}\}_{j \in \chi_i}, \overline{R}_{k,i}\right)|_I,$$

to solve the following problem:

$$\begin{cases} \sum_{j=1}^{J} R_{k,j} = PU_k; \\ \sum_{j \in \chi_i} R_{k,j} \leq PG_{k,i}, \forall i; \\ 0 \leq R_{k,j} \leq P_{k,j}. \end{cases} \tag{20}$$

4) $E \leftarrow E \setminus \Lambda$, $P_T \leftarrow [P_T - \sum_{k \in \Lambda} PU_k]$. Then return to 1) of ELPA.

As a note, in 4) of Algorithm ELPA, procedures are carried out, to shrink the index set of $\{1, 2, \ldots, K\}$. In particular, given E, through the first statement of 2) of Algorithm ELPA, the set of Λ is obtained. If Λ is empty, then output $\{R_{k,j}\}$, which were just obtained through solving (19) in 2) of Algorithm ELPA. Else, for any $k \in \Lambda$, compute the solution to (20). Following, the index set E is replaced with $E \setminus \Lambda$ in 4) of Algorithm ELPA. Through these procedures, the index set has been shrunk.

Remark 4.1. Algorithm ELPA is a dynamic load distribution process. The state of this process is the difference between the epoch load upper bound sum sequence, $\{PU_k\}_{k \in E}$, and current group sum load distribution sequence, $\{S_k\}_{k \in E}$, obtained by PBLA1. The control of this process is to use PBLA1 based on the state mentioned above. A new state for next time stage forms and an optimal dynamic load distribution process, ELPA, with the state feedback is formed. Since the finite set E is getting smaller and smaller until the set Λ is empty, ELPA iterates out at most K loops to compute the optimal solution.

Remark 4.2. For solving (19) and (20), besides the proposed method, there are many other methods from the linear optimization theory to solve them. For example, we may compute a set of basic feasible solutions, *i.e.*, vertices of the feasible region that is a convex polyhedron. Other solutions can be obtained by a convex combination. Similarly, we may deal with the case of.

Even with the existence of multiple optimal solutions, an optimization problem is said to be solvable iff one optimal solution can be computed [23, p. 128]. In Numerical Examples section, we will discuss the existence of multiple optimal solutions through examples, and illustrate graphically how to obtain a completed optimal solution set.

Remark 4.3. Given a set of parameters, it is possible that some of the soft load cannot be accommodated. If a household's soft demand cannot be fully satisfied, this signals user side to increase its corresponding load bound. For example, with newly added EV charging function in this specific household, this household might need to update its electricity service size from 100 Amps to 200 Amps to accommodate this increased load demand. From Utility company point of view, load increasing for this specific household flags that the total soft load (P_T) needs to be increased correspondingly. This procedure reflects a kind of mechanism to balance the electricity demand and supply.

B. Illustration of ELPA

Fig. 4 is used to illustrate the steps of the proposed ELPA algorithm. As shown in Fig. 4 (a), the shadowed 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 corresponding time slots. The elastic load vector $\mathbf{S}$ and the reference level L can be determined by calling PBLA1 in Step 1) of ELPA.

Fig. 4 (b) illustrates the allocation of the total elastic load S_1 at the first epoch among the groups. In the first statement

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TSG.2016.2590147

of Step 2) or Step 3) in ELPA, S_k, obtained from Step 1) in ELPA, is treated as the new elastic power budget for all the groups at the kth epoch. Therefore, we call BLA2 algorithm, which distributes elastic load power S_k to the groups 1 to I. **PG**, the group elastic power upper bound vector, will determine the elastic load group constraints in this step.

Fig. 4 (c) illustrates the allocation of the elastic load among all the users in any specify group using the total group load obtained from Fig. 4 (b), through the second time calling of BLA2 in Step 2) or Step 3) in ELPA. Through these steps, all the users' elastic loads $R_{k,j}, \forall j \in \chi_i$ can be solved optimally.

By this strategy, the elastic power is optimally distributed from the first time slot to the Kth slot; and the peak power constraints from the users, the groups, and any time slot for elastic loads are satisfied. Since PBLA1 and BLA2 are based on water-filling, this allocation process is fast.

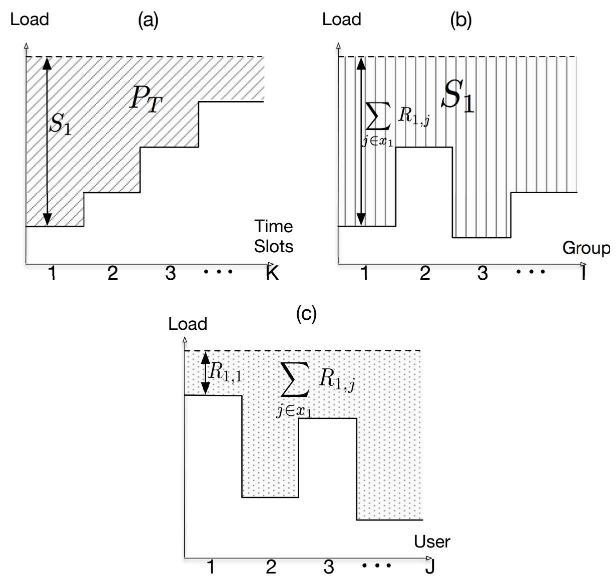


Fig. 4. (a) Elastic load allocation among epochs or the time slots. (b) Elastic load allocation among groups. (c) Elastic load allocation among users in a group.

C. Optimality of ELPA

The optimality of the proposed ELPA is discussed below.

Proposition 4.1: Output of Algorithm ELPA is an optimal solution to problem (1) by finite amounts of computation.

Proof. From the definition of Algorithm ELPA, at most K loops are needed to empty the final set Λ. It implies that solutions to (19) and those to (20) form the solution to problem (1). The reason is as follows.

Let $\{R_{k,j}\}$ be the solution to (20) and (19) (corresponding to the regular order of procedure in the algorithm) upto the final loop. It is seen that the final set Λ must be empty due to the assumptions of the upper bounds for the loads or the sums of the loads. Thus, for convenience, the set of $\{1, 2, \ldots, K\}\backslash E$ is denoted by E_c, where the set E is the one obtained by the final loop in Algorithm ELPA.

Following the existence of the optimal solution, there exists an optimal solution to the optimization problem (1) for minimizing the overall load fluctuation. That is to say, the set of optimal solutions to (1) is not empty. Thus, we may take any optimal solution, denoted by $\{R_{k,j}\}$, to (1). This optimal solution and its optimal dual variables satisfy the KKT conditions to (1). For the optimization variable L and the constraint of $L \geq 0$ for (1), the related KKT conditions, stemming from the Lagrange function of (1), are:

$$\begin{cases} 2\sum_{k=1}^{K}\sum_{j=1}^{J}(A_{k,j}+R_{k,j}) - 2KL + \psi = 0; \\ L \geq 0; \psi \geq 0; \psi \cdot L = 0, \end{cases} \tag{21}$$

where the Lagrange function of (1) is

$$\begin{aligned} & \mathcal{L}(\{R_{k,j}\}, L; \{\underline{\mu}_{k,n}, \overline{\mu}_{k,j}\}, \{\lambda_{k,i}\}, \{\phi_k\}, \psi, \lambda) \\ = \; & \textstyle\sum_{k=1}^{K} \left[\sum_{j=1}^{J}(a_{k,j}+R_{k,j}) - L\right]^2 \\ - \; & \textstyle\sum_{k=1}^{K}\sum_{j=1}^{J} \underline{\mu}_{k,j} R_{k,j} \\ - \; & \textstyle\sum_{k=1}^{K}\sum_{j=1}^{J} \overline{\mu}_{k,j}(P_{k,j} - R_{k,j}) \\ - \; & \textstyle\sum_{k=1}^{K}\sum_{i=1}^{I} \lambda_{k,i}[PG_{k,i} - \sum_{j\in\chi_i} R_{k,j}] \\ - \; & \textstyle\sum_{k=1}^{K} \phi_k[PU_k - \sum_{j=1}^{J} R_{k,j}] \\ - \; & \lambda(P_T - \sum_{k=1}^{K}\sum_{j=1}^{J} R_{k,j}) - \psi \cdot L. \end{aligned} \tag{22}$$

It is seen that $\psi = 0$. This point can be acquired by the following reason. Let us assume to the contrary that $\psi > 0$. Then the last equality $\psi \cdot L = 0$ in the KKT conditions mentioned above implies $L = 0$. Thus, the first equality in the KKT conditions is simplified by $2\sum_{k=1}\sum_{j=1}^{J}(A_{k,j}+R_{k,j})+\psi = 0$. Due to non-negativeness of $A_{k,j}, R_{k,j}$ and $\psi > 0$, the contradiction of $2\sum_{k=1}\sum_{j=1}^{J}(A_{k,j}+R_{k,j})+\psi = 0$ and $2\sum_{k=1}\sum_{j=1}^{J}(A_{k,j}+R_{k,j})+\psi > 0$ holding at the same time, is obtained. Therefore, $\psi = 0$. As a result, from the first equality in the KKT conditions, it can equivalently be transformed into the

$$\begin{aligned} L &= \tfrac{1}{K}\textstyle\sum_{k=1}^{K}\sum_{j=1}^{J}(A_{k,j}+R_{k,j}) \\ &= \tfrac{1}{K}(\textstyle\sum_{k=1}^{K}\sum_{j=1}^{J}A_{k,j}+P_T). \end{aligned} \tag{23}$$

That is to say, the optimal variable L can be determined just by the parameters of problem (1).

To find other optimal dual variables, first, construct $\lambda = 2[L - (S_{t_{k^*}} + \sum_{j=1}^{J} A_{t_{k^*},j})]$. As a reminder, it has been obtained that $\psi = 0$, *i.e.*, ψ is equal to zero. Then, for $k \in E$ and $1 \leq k \leq t_{k^*}$, let the corresponding remaining dual variables, with the epoch subscript k, be zeros. For $k \in E$ and $t_{k^*} < k$, let $\underline{\mu}_{k,j} = \{\lambda + 2[\sum_{l=1}^{K}\sum_{j=1}^{J}(A_{l,j}+R_{l,j}) - L]\}, \forall j$, and the corresponding other remaining dual variables, with the subscript of the epoch: k, be zeros. It is seen that $\underline{\mu}_{k,j} \geq 0, \forall j$. While $k \notin E$, let $\phi_k = -\{\lambda + 2[\sum_{j=1}^{J}(A_{l,j}+R_{l,j}) - L]\}$, and it is seen that this ϕ_k is non-negative, and the other corresponding remaining dual variables, with the subscript of the epoch: k, are assigned into zeros.

The constructed dual variables, and the solution by ELPA, including (19) and (20), are observed. Indeed, they are the optimal dual variables and the optimal solution to problem (1), respectively, since they satisfy the KKT conditions of problem (1).

Since problem (1) is a differentiable convex optimization problem with linear constraints, not only are the KKT conditions mentioned above sufficient, but they are also necessary for optimality. Note that the constraint qualification of problem (1) holds. Proposition 4.1 is hence proved, concisely and strictly. □

Remark 4.4. The proposed algorithm has great robustness feature. Two reasons can account for this point. First, the proposed algorithm ELPA doesn't need any initial point. With many existing algorithms, different initial points would have different effects, even with all the parameters or their changes being kept the same. Second, (21) leads to $L > 0$, due to $P_T > 0$ and $\frac{1}{K}\sum_{k=1}^{K}\sum_{j=1}^{J} A_{k,j}$ being non-negative, whether or not $A_{k,j}$ is assumed to be the two cases: a precise realization, or a prediction with the prediction error. Then $\psi = 0$ in (21). Therefore, the difference between the two cases of equation (21) leads to $\sum_{k,j}[\delta(A_{k,j}) + \delta(R_{k,j})] = 0$. It implies the error sum of the solution approaching to zero, as the estimation or prediction errors for non-elastic loads approach to zero. That is to say, the optimal objective function is also continuous in the parameters. At the same time, the solution is continuous in the parameters of inelastic loads. This point results from the set of expressions in the statement of PBLA1. According to the continuity of a function by the (ϵ, δ) approach (refer to [23, A.3.2]), for permissiveness ϵ of the optimal value or the solution varying, there exists the domain of the prediction errors with the radius δ, such that only when the vector consisting of the KJ errors falls into this domain then permissiveness ϵ of the solution or the optimal value varying is met. Thus, from the continuity, a relationship between the ϵ and the δ exists.

V. Numerical Examples and Computational Complexity

In this section, numerical examples and computational complexity analysis are presented to illustrate the steps of the proposed algorithm and the efficiency of the algorithm.

A. Numerical Example

Example 1. We consider load balancing for 4 users in 2 epochs ($J = 4, K = 2$). Assume that the inelastic load matrix is given as:

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \end{pmatrix},$$

as shown in Fig. 5(a). Each of the elastic load upper bounds is: for any user at any epoch it is 3; users 1 and 2 form a group, users 3 and 4 form another group, with the group upper bounds to be 5 for both epochs; the upper bound for each epoch is 8; and the total allocation load is 16. Then the problem can be formulated as

$$\begin{array}{ll} \min_{\{\mathbf{R}_{2\times 4}, L\}} & \sum_{k=1}^{2}[\sum_{j=1}^{4}(A_{k,j} + R_{k,j}) - L]^2 \\ \text{subject to:} & 0 \le R_{k,j} \le 3,\ \forall k, j; \\ & R_{k,1} + R_{k,2} \le 5,\ \forall k; \\ & R_{k,3} + R_{k,4} \le 5,\ \forall k; \\ & \sum_{j=1}^{4} R_{k,j} \le 8,\ \forall k; \\ & \sum_{k=1}^{2}\sum_{j=1}^{4} R_{k,j} = 16,\ \forall k; \\ & L \ge 0. \end{array} \tag{24}$$

The above target problem can be solved by using Algorithm ELPA as illustrated in Fig. 5. In Step 1) of the first loop, we can obtain the elastic load matrix and the reference level L as

$$\mathbf{R}^{(1)} = \begin{pmatrix} 4 & 3 & 2 & 1 \\ 3 & 2 & 1 & 0 \end{pmatrix} \quad \text{and} \quad L = \frac{1}{2}(24 + 16) = 20,$$

as shown in Fig. 5(b), where the superscript specifies the number of computation loops. After the operation of Step 2) and Step 3), the result is updated as

$$\mathbf{R}^{(1)} = \begin{pmatrix} 3 & 2 & 2 & 1 \\ 3 & 2 & 1 & 0 \end{pmatrix},$$

as shown in Fig. 5(c).

At the end of the first loop, the set Δ is not empty, the second loop is conducted and the result is given as

$$\mathbf{R}^{(2)} = \begin{pmatrix} 3 & 2 & 2 & 1 \\ 3 & 2 & 2 & 1 \end{pmatrix},$$

as shown in Fig. 5(d), which is our final output solution.

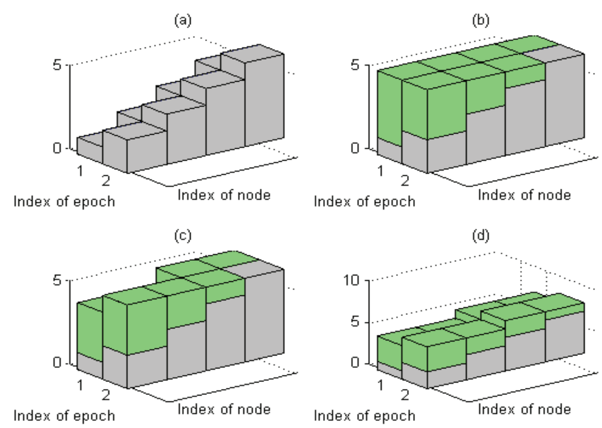


Fig. 5. Node load distribution of ELPA for Example 1

The target problem focuses on the load balancing across the epochs. Fig. 6 depicts the sum load vs. epochs, where the solid horizontal line denotes the reference level $L = 20$. The total load vector is $\mathbf{S} + \mathbf{B} = [18, 22]^T$, where $[\cdot]^T$ denotes matrix transpose. The objective function value (sum squared error) is 8. Here we can see the difference of the reference level L and water level. Here the reference level L is a constant level no matter how many water levels there might be.

Next we discuss the existence of multiple optimal solutions for Example 1 through Fig. 7. The constraints in (24), with (19) and (20), determine the line segment between point $X = (3, 5)$ and point $Y = (5, 3)$ in the (U, V) plane as shown in Fig. 7(a), where the first coordinate U stands for the first group two node sum elastic load; while the second one for the second group sum load. Arbitrarily choosing an operation point $Z = (3.5, 4.5)$ on the line XY, the point Z determines two straight lines of $R_{11} + R_{12} = 3.5$ (lower straight line); and $R_{13} + R_{14} = 4.5$ (higher straight line) in the (X, Y) plane in Fig. 7(b). This denotes that in the first epoch, sum elastic load for group 1 is allocated with 3.5, and second group is allocated with 4.5.

Further consider the first constraint in (24), *i.e.*, the bound of the elastic load for any user at any epoch is 3, we plot one vertical dashed line and one horizontal dashed line in Fig. 7(b) to meet this constraint. The lower-left region determined by this pair of dashed lines is referred as "feasible region". Then any pair of the allocation for R_{11} and R_{12} can be selected on the lower solid line, and R_{13} and R_{14} by the upper solid line within the "feasible region" or it swaps. The shadowed strip illustrates the region of the straight line path when the point Z shifts between X and Y. Similarly, we can also obtain those at epoch 2.

Fig. 8 illustrates the specified optimal solution presented in Figs. 5 and 6. In Fig. 8(a), the operation point Z falls down at the end of the line XY. Correspondingly, the two lines in Fig. 8(b) determining the group sum load are located at the end positions of the shadowed strip. The solution, as a point, Z is on the top line, and specified as $G1 = (3, 2)$ for $R_{11} = 3$ and $R_{12} = 2$ for the first group. Similarly, the solution $G2 = (2, 1)$ specifies $R_{13} = 2$ and $R_{14} = 1$ for the second group.

Therefore, the proposed algorithm not only rapidly computes an optimal solution, but also offers an alternative path to obtain all the optimal solutions to the target Problem in (1). Even though from optimization theory, an optimization problem is solvable iff one optimal solution can be computed [23, p. 128], the proposed solution by (19) and (20) is providing a path for computing all the optimal solutions beyond providing one optimal solution.

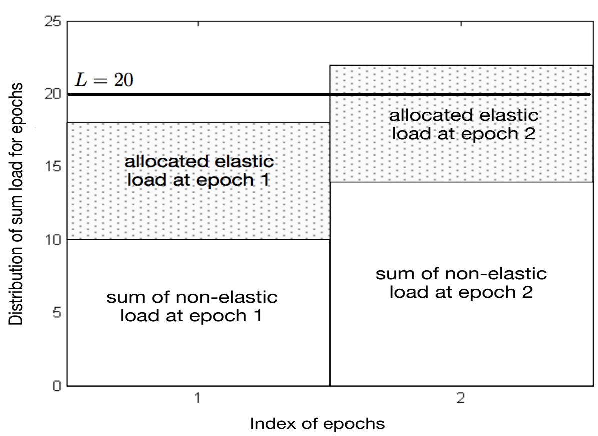


Fig. 6. Epoch load distribution of ELPA for Example 1

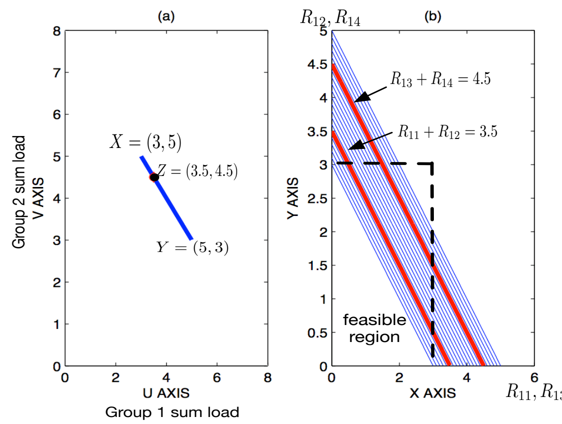


Fig. 7. Illustration of the existence of the multiple optimal solutions of ELPA for Example 1

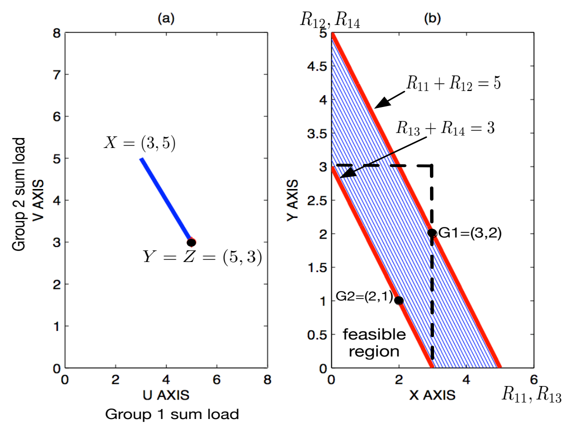


Fig. 8. Illustration of the optimal solutions in Figs. 5 and 6 for Example 1

In the following of this subsection, we simulate a multiple user multiple group system. The simulation model for a house is based on the parameters listed in Tables 1 and 2 of Section VII in [2]. We consider a part of power system with $I = 4$ groups or communities, and each group has 100 users. We divide the scheduling interval as 30-minutes. Once an elastic load is scheduled, it will continue operation until the load finishes its work. In other words, all loads are used continuously without disruption. We predict one day power demand and all the demand of the inelastic loads is satisfied in a day.

Fig. 9(a) shows the elastic load of the four groups in the first five hours in a day with no group upper bound restriction. The allocation of the four groups is identically distributed. Fig.9(b) plots the total inelastic (dotted curve) and elastic (dash curve) loads, and the total loads (solid curve). The overall load is flattened through scheduling the elastic load to fill the valleys of the inelastic load.

Figs. 9(c) and (d) show the power allocation when the group elastic loads demand constraints are limited. The elastic power constraints **PG** in group 1 to 4 are set probabilistically among time slots by normal distribution in this simulation, with mean of {900kWh, 800kWh, 700kWh, 600kWh} respectively. We see the elastic load in each group and time slot is bounded by corresponding power constraint. The elastic power is adjusted simultaneously among groups to reach the overall reference level value in group domain.

It is ensured that our approach provides a stable and exact solution for the elastic power allocation problem.

B. Complexity Analysis

For the proposed target problem (1), PBLA1 has been developed into BLA1, then BLA2, and finally into ELPA to compute an optimal solution. Since ELPA uses PBLA1 once

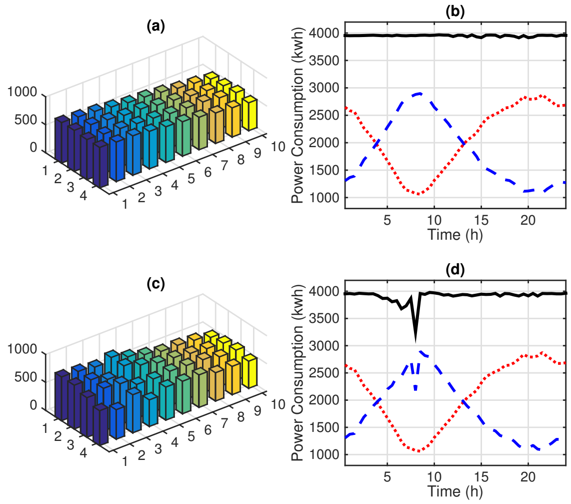


Fig. 9. (a) Elastic load power consumption in group view, where PG is infinity in five hours, 30 minutes in a time slot; (b) Power consumption of inelastic, elastic and total loads, where PG is infinity in a day; (c) Elastic load power consumption in group view, where PG is bounded by randomized matrix in the system, 30 minutes in a time slot; (d) Power consumption of inelastic, elastic and total loads, where PG is bounded by randomized matrix in the system.

and BLA2 $K \times I$ times in one loop, further considering that the algorithms BLA1-2 are similar, and both are based on PBLA1, we first focus on the discussion of the complexity of PBLA1, then that of BLA1-2, and finally that of ELPA.

The proposed PBLA1 algorithm occupies less computational resource. It is seen that PBLA1 needs at most K loops to search k^* through the k^* search method. Thus it needs 4 arithmetic operations and 2 logical operations to complete each loop. The worst-case computational complexity of the proposed solution is $6K + 2$ fundamental arithmetical and logical operations under the $K + 2$ memory units to store $\{B_i\}, W_s$, and PM. It is written into $O(K)$.

For the BLA1, it needs at most K loops to compute the optimal solution. The required number of operations is, at worst, $\sum_{i=1}^{K}(6i+2) = 3K^2 + 5K$ fundamental arithmetical and logical operations. The worst case complexity of BLA1 is $O(K^2)$. Similarly, that of BLA2 is also $O(K^2)$, under the E being a subsequence of $\{1, \ldots, K\}$.

For the ELPA, it needs at most K loops to compute the optimal solution. Further, each loop of ELPA uses PBLA1 once, and BLA2 $K \times I$ times. Considering that the numbers of the groups and the users, respectively, with $I \leq J$, computational complexity of the corresponding PBLA1 and BLA2, and sorting all the entries of $\mathbf{A}$ needing $O(KJ)\log(KJ)$, therefore, the required number of operations, at worst, is $O(K)+O(KI^2)+O(KIJ^2)+O(KJ)\log(KJ)$ fundamental arithmetical and logical operations.

Therefore, Algorithm ELPA uses its computational complexity is not beyond $O((KJ)^3)$. This point plays a key role in the proposed algorithm being able to compute the exact solution to the proposed problem, especially suitable for the proposed problems which are large-scale.

VI. Conclusion

In this paper, we proposed a load balancing problem and its optimal solutions through the allocation of the elastic load. The target problem is subject to the peak and group load upper bound constraints. The proposed algorithm decomposes the three-dimensional allocation problem into two-dimensional problems. We proved that the load balancing problem aiming to minimize the load fluctuation has the optimal solution in the form of water-filling.

The proposed target problem is a quadratic optimization problem with constraints. To the best of the authors' knowledge, no algorithms have been reported in the open literature to exactly compute the optimal solution to this class of the problems. The proposed algorithm computes the target problem with low computational complexity. Numerical examples are provided to illustrate the steps that can obtain the optimal solutions by the proposed algorithm. We also analyzed the path to determine the family of all the optimal solutions.

References

[1] X. Fang, S. Misra, G. L. Xue, and D. J. Yang, "Smart grid - the new and improved power grid: a survey," *IEEE Communications surveys & toturials*, vol. 14, no. 4, pp. 944–980, 2012.
[2] M. Shinwari, A. Youssef, and W. Hamouda, "A water-filling based scheduling algorithm for the smart grid," *IEEE Transactions on Smart Grid*, vol. 3, pp. 710–719, 2012.
[3] Government of Western Australia Office of Energy, "Demand management," 2010.
[4] S. Bahrami and M. Parniani, "Game theoretic based charging strategy for plug-in hybrid electric vehicles," *IEEE Trans. Smart Grid*, , no. 5, pp. 2368–2375, Sept. 2014.
[5] Y. F. He, B. Venkatesh, and L. Guan, "Optimal scheduling for charging and discharging of electric vehicles," *IEEE Transactions on Smart Grid*, vol. 3, no. 3, pp. 1095–1105, Sept 2012.
[6] E. Sortomme, M. M. Hindi, S. D. J. MacPherson, and S. S. Venkata, "Coordinated charging of plug-in hybrid electric vehicles to minimize distribution system losses," *IEEE Transactions on Smart Grid*, vol. 6, no. 2, pp. 726–736, 2015.
[7] W. Y. Chiu, H. Sun, and H.V. Poor, "Energy imbalance management using a robust pricing scheme," *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 896–904, 2013.
[8] T. T. Kimand and H.V. Poor, "Scheduling power consumption with price uncertainty," *IEEE Transactions on Smart Grid*, vol. 2, pp. 519–527, 2011.
[9] A.-H. Mohsenian-Rad and A. Leon-Garcia, "Optimal residential load control with price prediction in real-time electricity pricing environments," *IEEE Transactions on Smart Grid*, vol. 1, pp. 120–133, 2010.
[10] C. P. Rodriguez and G. J. Anders, "Energy price forecasting in the ontario competitive power system market," *IEEE Transactions on Power Systems*, vol. 19, pp. 366–374, 2004.
[11] F. L. Alvarado, "Is system control entirely by price feasible?," *in Proc. 36th annual hawaii international conference on system sciences*, pp. 6–9, 2003.
[12] R. L. Deng, Z. Y. Yang, J. M. Chen, N. R. Asr, and M. Y. Chow, "Residential energy consumption scheduling: A coupled-constraint game approach," *IEEE Trans. Smart Grid*, , no. 3, pp. 1340–1350, May 2014.
[13] L. Qian, Y. Zhang, J. Huang, and Y. Wu, "Demand response management via real-time electricity price control in smart grids," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 7, pp. 1268–1280, July 2013.
[14] P. Yi, X. Dong, A. Iwayemi, C. Zhou, and S. Li, "Real-time opportunistic scheduling for residential demand response," *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 227–234, 2013.
[15] F. Bonanno, G. Capizzi, and G. Tina, "Long-term energy performance forecasting of integrated generation systems by recurrent neural networks," *in Proc. 36th annual hawaii international conference on system sciences*, pp. 673–678, 2009.
[16] J. Wu, X. Liu, and J. Qian, "Wind speed and power forecasting based on rbf neural network," *in Proc. Computer Application and System Modeling (ICCASM '10), Shanxi, China*, vol. 5, pp. 298–301, 2010.

[17] D. Palomar, "Practical algorithms for a family of waterfilling solutions," *IEEE Trans. Signal Processing*, vol. 53, pp. 686–695, 2005.

[18] 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 Communications*, vol. 12, no. 7, pp. 3637–3647, 2013.

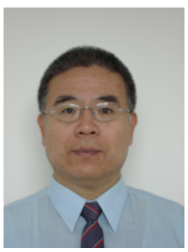
[19] D. Li, S. K. Jayaweera, O. Lavrova, and R. Jordan, "Load management for price-based demand response scheduling - a block scheduling model," *International Conference on Renewable Energy and Power Quality (ICREPQ'11), Canary Islands, Spain*, pp. 1–6, 2011.

[20] D. Li and S. K. Jayaweera, "Uncertainty modeling and price-based demand response scheduling in smart grid," *IEEE System Journal*, pp. 1–12, 2014.

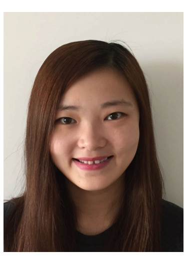
[21] K. Zhou and L. Cai, "A dynamic water-filling method for real-time HVAC load control based on model predictive control," *IEEE Transactions on Power Systems*, vol. 30, pp. 1405–1414, 2015.

[22] Y. Mou, H. Xing, Z. Lin, and M. Fu, "Decentralized optimal demand-side management for PHEV charging in a smart grid," *IEEE Transactions on Smart Grid*, vol. 6, no. 2, pp. 726–736, 2015.

[23] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, 2004.

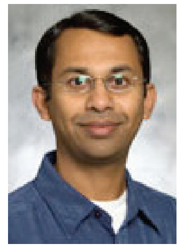


Peter He (S'15-M'15) received the M.A.Sc. degree in electrical engineering from McMaster University, Hamilton, ON, Canada, in 2009. He received the Ph.D. degree in electrical engineering from Ryerson University, Toronto, Canada, in 2015. At present, his post-doc research interests are the union of large-scale optimization, information theory, communications, and smart power grid. During his Ph.D. studies, he has been continuously awarded Graduate Research Excellence Award in the Department of Electrical and Computer Engineering at Ryerson University and OGS (Ontario Graduate Scholarship) awards. He was awarded NSERC (Natural Sciences and Engineering Research Council) Postdoctoral Fellowship in 2016. He is a recipient of the Best Paper Award of the 2013 International Conference on Wireless Communications and Signal Processing (WCSP).

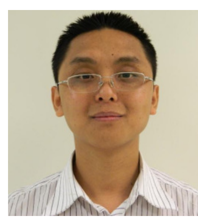


Mushu Li obtained her Bachelor Degree from University of Ontario Institute of Technology (UOIT) in 2015. She is currently pursuing her master degree at Ryerson University. She has been awarded OGS (Ontario Graduate Scholarship) in 2015. Her research interests are load balancing for smart grid, EV charging scheduling, and system optimizations.

Lian Zhao (S'99-M'03-SM'06) received her PhD degree from the Department of Electrical and Computer Engineering (ELCE) from University of Waterloo, Canada, in 2002. She joined the Electrical and Computer Engineering Department, Ryerson University, Toronto, Canada, as an Assistant Professor in 2003 and a Professor in 2014. Her research interests are in the areas of wireless communications, radio resource management, power control, cognitive radio and cooperative communications, and smart grid. She has been an Editor for IEEE Transaction on Vehicular Technology since 2013 and awarded TOP 15 Editor in 2015. She has been a member of NSERC (Natural Sciences and Engineering Research Council) Evaluation Group for ELCE since 2015, Associate Chair for Graduate Studies at ELCE department from 2013 to 2015.



Bala Venkatesh (M'95SM'08) received the Ph.D. degree in Electrical Engineering from Anna University, Chennai, India, in 2000. He is an Academic Director of the Centre for Urban Energy and a Professor with the Department of Electrical and Computer Engineering at Ryerson University, Toronto, Canada. His research interests are power system analysis, economics, and optimization.



Hongwei Li (M'12) is currently the head and an Associate Professor with the School of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC). He received his PhD degree from UESTC, Chengdu, China in 2008. He has worked as a post-doctoral fellow in the Department of ELCE at University of Waterloo, Waterloo, Canada, for one year until October 2012. His research interests include network security, applied cryptography, and trusted computing. He serves as the Associate Editor of Peer-to-Peer Networking and Applications, the Guest Editor for Peer to-Peer Networking and Applications Special Issue on Security and Privacy of P2P Networks in Emerging Smart City.