Coordinating Flexible Loads via Optimization in the Majorization Order

Yanfang Mo, Wei Chen, and Li Qiu

Abstract— **A key feature of the smart grid is the integration of a large group of flexible loads which, depending on their respective natures, are deferrable, and/or interruptible. To fully exploit such load flexibilities, the system operator attempts to make decisions towards an efficient coordination of flexible loads so as to achieve specifically aimed energy consumption patterns. One desirable consumption pattern, motivated by reducing generation costs and customer expenditures, is to make the total load profile as smooth as possible. To this end, we model the load coordination as an optimal zero-one matrix completion problem. In particular, we propose an optimization problem in the majorization order. Although such problem seems combinatorially hard at first sight, due to its nice structure, we show that it can be solved with low complexity. We firstly discuss the existence and uniqueness of the optimal solution. Then, a sequential algorithm is proposed to solve the optimization problem efficiently, even in the case of a large population of loads. Moreover, we address the connection between our work and the valley-filling behavior presented by a substantial number of works in the literature.**

I. Introduction

Nowadays, on the supply side, more and more electric energy is generated from renewable resources, e.g., solar and wind energy [1]. Also, on the demand side, more and more flexible loads are integrated to the grid. Depending on their respective natures, some loads are deferrable, and/or interruptible, e.g., pool pumps and electric vehicles [2]–[5]. This raises worldwide interest in how to improve the performance of the smart grid by fully exploiting such load flexibilities [6]–[9].

In general, loads in a region can be categorized into two separate groups. The first group takes up the majority of the total power consumption, over which the system operator has little or no control. These are termed as base loads, and can be estimated under most circumstances. The second group comprises the flexible loads mentioned in the last paragraph. The system operator attempts to make decisions towards an efficient coordination of flexible loads so as to achieve aimed energy consumption patterns.

A natural question arises – what kind of consumption pattern is the most desirable? As a starting point, consider the total load profile, i.e., the sum of the aggregated flexible load profile and the base load profile. One desirable consumption pattern is to make the total load profile as smooth as possible. The underlying rationale is as follows. Firstly, augmenting the peak in demand could require additional generation capacity and ramping capacity [7]. In addition, a more fluctuant total load profile may result in more power losses, voltage deviations, and emission costs [10]. Last but not least, the market generally penalizes peak times with higher prices, which increases customer expenditures.

In this paper, we consider a group of flexible loads which can only be charged at a certain fixed rate and its multiples. The practicability of such an assumption is supported by [11]. Thus, as explained in later sections, the load coordination can be mathematically modeled as an optimal zero-one matrix completion problem. In order to evaluate the smoothness of the total load profile, we introduce the majorization relation, which is well-known as a measure of the fluctuation of a sequence of numbers. Given the base load, the system operator coordinates the charging processes of these discrete flexible loads such that the total load profile is as small as possible in the majorization order.

At first sight, it seems combinatorially hard to solve such an optimization problem in the majorization order. However, due to its nice structure, we show that such a problem can be solved with low complexity. Firstly, we analyze the existence and uniqueness of the optimal solution, which implies that the majorization relation is suitable for evaluating the smoothness of the total load profile. Next, an efficient algorithm is proposed to find the optimal solution. This algorithm is sequential in terms of flexible loads, and thus friendly to the augmentation of the number of loads. In detail, when a new group of flexible loads arrive, the system operator does not need to rearrange the coordinated charging profiles of existing loads. Furthermore, we address the connection between our work and the valley-filling behavior. A number of works in the literature aim at finding a valley-filling charging profile. See, for instance, [7], [10], and [12]–[15]. Although a valley-filling charging profile may not necessarily exist in our model, our results are still consistent with the so-called valley-filling behavior in the smart grid.

The paper is structured as follows. In Section II, we introduce some preliminary knowledge and notation. The model and main problems are elaborated in Section III. In Section IV, we present the analysis of the corresponding total load smoothing problem, consisting of the properties of optimal solutions, the solution algorithm, and the connection to the valley-filling behavior. After that, some illustrative numerical examples are presented in Section V. Finally, we articulate the conclusion and some future work in Section VI.

This work was partially supported by the Research Grants Council of the Hong Kong Special Administrative Region, China, under the Theme-Based Research Scheme T23-701/14-N and the Hong Kong PhD Fellowship.

Y. Mo, W. Chen, and L. Qiu are with the Department of Electronic and Computer Engineering, the Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China ymoaa@connect.ust.hk, wchenust@gmail.com, eeqiu@ust.hk

For fluency and the page limitation, most proofs are omitted and available from the authors.

II. Notation and Preliminaries

We use a bold italic letter to denote a vector. A set is denoted by a calligraphic capital letter, except that we use $\mathbb{R}$ and $\mathbb{N}$ to refer to real numbers and non-negative integers respectively. Let $\boldsymbol{e}_i$ denote the vector $\begin{bmatrix}0 & \dots & 0 & 1 & 0 & \dots & 0\end{bmatrix}'$, where the only 1 is located at the ith position. The length of $\boldsymbol{e}_i$ can be inferred from the context. The symbol $\boldsymbol{0}_T$ denotes a vector of length T that has only zero elements. Let $\lfloor x \rfloor$ denote the largest integer not exceeding $x \in \mathbb{R}$. The notation $\mathbb{1}(\cdot)$ is the indicator function, mapping an assertion to $\{0, 1\}$.

A. Preliminary on Majorization

The notation regarding majorization varies in the existing literature, while we stick to the notation from the famous monograph [16]. Consider a vector $\boldsymbol{x} \in \mathbb{R}^n$ and rearrange the elements of $\boldsymbol{x}$ in non-increasing order to obtain a new vector $\boldsymbol{x}^{\downarrow}$. Vectors $\boldsymbol{x}$ and $\boldsymbol{x}^{\downarrow}$ are respectively denoted by

$$\begin{aligned} \boldsymbol{x} &= \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}', \\ \boldsymbol{x}^{\downarrow} &= \begin{bmatrix} x_{[1]} & x_{[2]} & \cdots & x_{[n]} \end{bmatrix}', \end{aligned}$$

where $x_{[1]} \geq x_{[2]} \geq \cdots \geq x_{[n]}$.

Definition 1 ([16]): For two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, we write $\boldsymbol{x} \prec \boldsymbol{y}$, saying that $\boldsymbol{x}$ is majorized by $\boldsymbol{y}$, if

$$\begin{cases} \sum_{i=1}^{k} x_{[i]} \leq \sum_{i=1}^{k} y_{[i]}, & k = 1, 2, \ldots, n-1, \\ \sum_{i=1}^{n} x_{[i]} = \sum_{i=1}^{n} y_{[i]}. \end{cases}$$

Note that it suffices to restrict $\mathbb{R}^n$ to $\mathbb{N}^n$ in our application. This restriction will be made clear in following sections. We say that $\boldsymbol{x} \in \mathbb{N}^n$ is equivalent to $\boldsymbol{y} \in \mathbb{N}^n$ if $\boldsymbol{x}^{\downarrow} = \boldsymbol{y}^{\downarrow}$, denoted by $\boldsymbol{x} \sim \boldsymbol{y}$. As we can see, the majorization relation is reflexive and transitive over $\mathbb{N}^n$, since

$$\begin{cases} \boldsymbol{x} \prec \boldsymbol{x}, & \text{(reflexivity)} \\ \boldsymbol{x} \prec \boldsymbol{y} \text{ and } \boldsymbol{y} \prec \boldsymbol{z} \Rightarrow \boldsymbol{x} \prec \boldsymbol{z}. & \text{(transitivity)} \end{cases}$$

Although $\boldsymbol{x} \prec \boldsymbol{y}$ and $\boldsymbol{y} \prec \boldsymbol{x}$ do not imply that $\boldsymbol{x} = \boldsymbol{y}$, they do imply that $\boldsymbol{x} \sim \boldsymbol{y}$, i.e., $\boldsymbol{x}$ is a rearrangement of $\boldsymbol{y}$. As analyzed above, the majorization order is merely a preorder in $\mathbb{N}^n$, since it is reflexive and transitive but not antisymmetric.

B. Partition Set – Partially Ordered by Majorization

Concepts in this subsection are mainly adopted from references [16] and [17], where readers can find more details.

A partition $\boldsymbol{x}$ of a positive integer τ $(\leq n)$ is a vector only consisting of non-negative integers such that

$$\boldsymbol{x} \in \mathbb{N}^n, \ \boldsymbol{x} = \boldsymbol{x}^{\downarrow}, \text{ and } \sum_{i=1}^{n} x_i = \tau.$$

Denote the set of all the partitions of τ by $\mathcal{P}_\tau$. Restricted to the partition set $\mathcal{P}_\tau$, the majorization is then a partial ordering (reflexivity, transitivity and antisymmetry). Thus, we say that the set $\mathcal{P}_\tau$ is partially ordered by the majorization relation. Actually, the majorization order over $\mathcal{P}_\tau$ is also well-known as the dominance order, as stated by James in [18].

Before proceeding, it is important to distinguish "a minimal element" and "the least element", especially when we tackle a partially ordered set (poset). Given a poset $(\mathcal{R}, \prec)$, an element $\boldsymbol{x} \in \mathcal{R}$ is minimal if there exists no other element $\boldsymbol{y} \in \mathcal{R}$ such that $\boldsymbol{y} \prec \boldsymbol{x}$, while the least element $\boldsymbol{o}$ in $(\mathcal{R}, \prec)$ is the unique element of $\mathcal{R}$ such that $\boldsymbol{o} \prec \boldsymbol{y}$, for every $\boldsymbol{y} \in \mathcal{R}$. In a similar way, we can define a maximal element and the greatest element. For $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{R}$, define $\inf\{\boldsymbol{x}, \boldsymbol{y}\}$ as the greatest element of $\{\boldsymbol{z} | \boldsymbol{z} \in \mathcal{R}, \boldsymbol{z} \prec \boldsymbol{x}, \ \& \ \boldsymbol{z} \prec \boldsymbol{y}\}$, while define $\sup\{\boldsymbol{x}, \boldsymbol{y}\}$ as the least element of $\{\boldsymbol{z} | \boldsymbol{z} \in \mathcal{R}, \boldsymbol{x} \prec \boldsymbol{z}, \ \& \ \boldsymbol{y} \prec \boldsymbol{z}\}$. Note that the least/greatest element may not necessarily exist in a poset.

Interestingly, the minimal element and the least element coincide in the partition poset $(\mathcal{P}_\tau, \prec)$. Firstly, the poset $(\mathcal{P}_\tau, \prec)$ is a lattice since $\inf\{\boldsymbol{x}, \boldsymbol{y}\}$ and $\sup\{\boldsymbol{x}, \boldsymbol{y}\}$ both exist in $\mathcal{P}_\tau$, for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{P}_\tau$. Secondly, as a lattice with finite elements, $(\mathcal{P}_\tau, \prec)$ has the least/greatest element, which is also the unique minimal/maximal element of $\mathcal{P}_\tau$. However, subsets of $(\mathcal{P}_\tau, \prec)$ may no longer possess such nice properties and should be treated with caution. Specifically, there may be more than one minimal elements and thus the least element does not exist.

C. Majorization in Zero-one Matrix Completion

Given $\boldsymbol{x} \in \mathbb{N}^m$ and the integer n no less than the largest element of $\boldsymbol{x}$, define the conjugate vector of $\boldsymbol{x}$ by $\boldsymbol{x}^* \in \mathbb{N}^n$, where

$$x_t^* = \sum_{i=1}^{n} \mathbb{1}(x_i \geq t).$$

The following theorem is widely known as the analytic condition for the existence of a zero-one matrix with given row sum and column sum vectors, as shown in [6], [16], [19], and [20]. Furthermore, it is of particular theoretical interest in characterizing the feasible aggregated charging profiles, which will be elaborated in the following section.

Theorem 2.1 (Gale-Ryser): There exists an $m \times n$ zero-one matrix with the row sum vector $\boldsymbol{x} \in \mathbb{N}^m$ and the column sum vector $\boldsymbol{y} \in \mathbb{N}^n$ if and only if $\boldsymbol{y} \prec \boldsymbol{x}^*$.

III. Problem Formulation

The operational period is evenly divided into a sequence of time slots, indexed by $t \in \mathcal{T} = \{1, 2, \ldots, T\}$. Let d_t denote the base load at time slot t. Then, the base load profile is given by a non-negative integral vector

$$\boldsymbol{d} = \begin{bmatrix} d_1 & d_2 & \cdots & d_T \end{bmatrix}'.$$

The flexible loads are indexed by $\mathcal{N} = \{1, 2, \ldots, N\}$. The requirement of a flexible load n is specified by two parameters, the charging demand r_n and maximum charging rate $\bar{r}_n$. Let $r_{n,t}$ denote the charging rate of the load n at time slot t. Due to the assumption of feasible loads, both $r_{n,t}$ and r_n are integral. Thus, a feasible charging profile for the load n is a charging profile $\boldsymbol{r}_n = \begin{bmatrix} r_{n,1} & r_{n,2} & \dots & r_{n,T} \end{bmatrix}'$, which satisfies the following constraints:

$$\sum_{t=1}^{T} r_{n,t} = r_n,$$
$$0 \le r_{n,t} \le \bar{r}_n, r_{n,t} \in \mathbb{N}, t \in \mathcal{T}.$$

Denote the aggregated charging profile $\boldsymbol{h}$ by

$$\boldsymbol{h} = \begin{bmatrix} h_1 & h_2 & \dots & h_T \end{bmatrix}',$$

where

$$h_t = \sum_{n=1}^{N} r_{n,t}, t \in \mathcal{T}.$$

Naturally, an aggregated charging profile is feasible if it is the summation of N feasible charging profiles corresponding to the N loads. Let $\mathcal{H}$ denote the set of all the feasible aggregated charging profiles. The total load profile is the summation of the base load profile and the aggregated charging profile, denoted by $\boldsymbol{h}+\boldsymbol{d}$. Let τ denote the amount of the total load; i.e.,

$$\tau = \sum_{t=1}^{T} (h_t + d_t).$$

Considering the base load profile $\boldsymbol{d}$ and the charging requirements $(r_n, \bar{r}_n)$ of the N loads, the coordinator schedules the feasible charging profiles such that the total load profile is as smooth as possible. Mathematically, the total load profile smoothing problem (TLPS) is formulated as follows:

$$\begin{aligned} \min_{\prec} \quad & \boldsymbol{h}+\boldsymbol{d} \\ \text{subject to} \quad & \boldsymbol{h} \in \mathcal{H}. \end{aligned} \tag{1}$$

As explained later in Remark 1, without loss of generality, we can assume that $\bar{r}_n = 1, \forall n \in \mathcal{N}$. As a result, all the demand requirements can be specified by the demand profile

$$\boldsymbol{r} = \begin{bmatrix} r_1 & r_2 & \dots & r_N \end{bmatrix}'.$$

Furthermore, the TLPS problem (1) can be reformulated as follows:

$$\begin{aligned} \min_{\prec} \quad & \boldsymbol{h}+\boldsymbol{d} \\ \text{subject to} \quad & \boldsymbol{h} = \sum_{n=1}^{N} \boldsymbol{r}_n, \quad r_n = \sum_{t=1}^{T} r_{n,t}, \\ & r_{n,t} \in \{0,1\}, \quad n \in \mathcal{N}, \quad t \in \mathcal{T}. \end{aligned} \tag{2}$$

Alert readers may find that the above problem is nothing but an optimal zero-one matrix completion problem. Specifically, given the row sum vector $\boldsymbol{r}$, complete a zero-one matrix such that the sum of the column sum vector $\boldsymbol{h}$ and the base load profile $\boldsymbol{d}$ is as small as possible under the majorization order. As a result, along with the Gale-Ryser theorem, we can substitute $\boldsymbol{h} \prec \boldsymbol{r}^*$ for the constraints in (2). Finally, the TLPS problem (1) leads to the following optimization problem in the majorization order:

$$\begin{aligned} \min_{\prec} \quad & \boldsymbol{h}+\boldsymbol{d} \\ \text{subject to} \quad & \boldsymbol{h} \prec \boldsymbol{r}^*. \end{aligned} \tag{3}$$

Now, we can further see why it is reasonable to coordinate flexible loads via optimization in the majorization order. It is not only because the smoothness of the objective variable can be evaluated by the majorization order, but also because the constraint of the TLPS problem can be characterized by an inequality in the majorization order. As a whole, the optimization problem (3) is of essentially theoretical interest to the TLPS problem (1).

Remark 1: A consumer n specified by $(r_n, \bar{r}_n)$ can be theoretically regarded as the combination of $\bar{r}_n$ sub loads. Specifically, the first δ of the sub loads are all specified by $(\lfloor r_n/\bar{r}_n \rfloor + 1, 1)$ and the remaining $(\bar{r}_n - \delta)$ sub loads are all specified by $(\lfloor r_n/\bar{r}_n \rfloor, 1)$, where $r_n = \lfloor r_n/\bar{r}_n \rfloor * \bar{r}_n + \delta$. The equivalence is explicit and will be illustrated by numerical examples in Section V.

IV. Analysis of TLPS Problem

The section consists of three parts. Firstly, we will show that the optimal solutions to the TLPS problem have a nice property and thus the majorization order is suitable for evaluating the smoothness of the total load profile. Then, an algorithm is put forward to find the optimum of the TLPS problem. Finally, we connect our results to the valley filling, which has been widely studied in existing literature.

A. Existence of Solution

Before proceeding, we define

$$\mathcal{S} = \{(\boldsymbol{h}+\boldsymbol{d})^{\downarrow} \mid \boldsymbol{h} \in \mathcal{H}\}.$$

For simplicity, the total load profile $\boldsymbol{h}+\boldsymbol{d}$ will also be denoted by $\boldsymbol{s}$. In the following, we will exploit the poset $(\mathcal{S}, \prec)$. Obviously, by properly adding some possible zeros behind the vectors, we can write $\mathcal{S} \subseteq \mathcal{P}_\tau$ for convenience. Just like $(\mathcal{P}_\tau, \prec)$, the poset $(\mathcal{S}, \prec)$ also has a nice structure, as shown in the following theorem.

Theorem 4.1: The least element exists in $(\mathcal{S}, \prec)$.

Generally, we can see that there are no minimals in $\{(\boldsymbol{h}+\boldsymbol{d}) \mid \boldsymbol{h} \in \mathcal{H}\}$, so, a fortiori, neither is the least element. However, the above theorem says that a unique minimal (the least element) exists in $\mathcal{S} = \{(\boldsymbol{h}+\boldsymbol{d})^{\downarrow} \mid \boldsymbol{h} \in \mathcal{H}\}$. This least element corresponds to all the optimal total load profiles with respect to the demands $\boldsymbol{r}$ and the base load $\boldsymbol{d}$. That is to say, all such total load profiles possess the same elements and thus certainly share the same fluctuation level. Strictly speaking, we should rewrite the objective function in the TLPS problem as $\min_{\prec,\mathcal{S}}(\boldsymbol{h}+\boldsymbol{d})$ to make the optimization problem better defined. Without confusion, we can omit $\mathcal{S}$ in the context for simplicity. As a whole, the essence of the TLPS problem is to find a feasible total load profile $\boldsymbol{s}$ such that $\boldsymbol{s}^{\downarrow}$ is the least element of $\mathcal{S}$.

B. Solution Algorithm

As an optimal zero-one matrix completion problem, the TLPS problem is combinatorially hard at first sight. A natural way to solve a combinatorial optimization problem is to search the optimum in the feasible set. Although the superset $\mathcal{P}_\tau$ of $\mathcal{S}$ has a nice structure for searching, such brute-force search is not desirable. Besides, we still need to take extra efforts to find the charging profile of each load after attaining the least element. Thus, we pursue a more efficient algorithm, which can help us find a smoothest total load profile, together with corresponding charging profiles.

Concerning the zero-one matrix completion problem with given row and column sum vectors, the well-known Ryser's algorithm works efficiently. Interested readers can find details in [20]–[22]. As for the TLPS problem, which is aimed at completing a zero-one matrix with a given row sum vector such that the column sum vector is the desired one, we propose an algorithm as efficient as the Ryser's algorithm. Shown below as Algorithm 1, the proposed algorithm is iterative and has at most N iterations. The complexity of each iteration is uniformly $\mathcal{O}(T)$, and thus, the complexity of the total algorithm is $\mathcal{O}(T \cdot N)$. The complexity of Ryser's algorithm is also $\mathcal{O}(T \cdot N)$, to find a zero-one matrix with the column sum vector $\boldsymbol{h}$ and row sum vector $\boldsymbol{r}$.

As shown in this paper, the designed algorithm solves the problem in a sequential way. Thus, one of our concerns is whether the output total load profile $\bar{\boldsymbol{s}}$ will be changed if the elements in the vector $\boldsymbol{r}$ are rearranged. The following proposition gives the answer.

Proposition 1: Rearranging elements of the input $\boldsymbol{r}$, the output total load profile $\bar{\boldsymbol{s}}$ remains equivalent.

The above invariance property plays a vital role in verifying the optimality of Algorithm 1. The optimality means that the least element is given by $\bar{\boldsymbol{s}}^{\downarrow}$, where $\bar{\boldsymbol{s}}$ is the total load profile generated by Algorithm 1.

Another interesting observation is that if a new group of loads arrive after we have coordinated N loads by Algorithm 1, we can still achieve the optimum without changing the charging profiles of existing loads. This observation implies that our algorithm can still work when the number of flexible loads N is extremely large. When the storage space of a processor is limited, the flexible loads can be coordinated in batches and we can still achieve the optimum in the presence of all the flexible loads.

On the other hand, in our current setting, the system operator has full control over the charging processes of flexible loads. Based on Algorithm 1, the following scenario can heighten the customer engagement without reducing the optimal performance in principle. Denote the price function by $f(\cdot)$, which is a strictly increasing function. The original price profile is given by

$$\boldsymbol{p} = \begin{bmatrix} p_1 & p_2 & \dots & p_T \end{bmatrix}',$$

with $p_t = f(d_t), t \in \mathcal{T}$. The flexible loads are served by the first-come first-served rule. Once a new load specified by $(r_n, 1)$ arrives, the system operator shows it the current price profile $\boldsymbol{p}$. Then, the load chooses a feasible charging profile $\boldsymbol{r}_n$ such that the cost $\sum_{t=1}^{T} p_t r_{n,t}$ is minimum, and reports it back. The system operator receives the information, successively updates the total load profile $\boldsymbol{s}$ and the price profile $\boldsymbol{p}$ by $s_t = s_t + r_{n,t}$ and $p_t = f(s_t)$, $t \in \mathcal{T}$, and then waits for another load. By Proposition 1, for the same group of flexible loads and the same base load, the above process will generate an equivalent total load profile to that from Algorithm 1.

Algorithm 1 Total Load Smoothing Algorithm

Input: The base load profile $\boldsymbol{d}$ and the demand profile $\boldsymbol{r}$, where the lengths of $\boldsymbol{d}$ and $\boldsymbol{r}$ are respectively T and N.
Output: The total load profile $\bar{\boldsymbol{s}}$ and a group of feasible charging profiles $\boldsymbol{r}_n, n \in \mathcal{N}$.
1: Initialization: $i = N; \boldsymbol{r}_n = \boldsymbol{0}_T, n \in \mathcal{N}; \bar{\boldsymbol{s}} = \boldsymbol{d}$;
2: Identify position indices $i_1, i_2, \dots, i_{r_i}$ corresponding to the r_i smallest elements of $\bar{\boldsymbol{s}}$. In case of ties, randomly pick them such that only r_i positions are identified. Update the charging profile $\boldsymbol{r}_i$ by adding one to elements in positions indexed by the aforementioned indices;
3: Update the total supply profile: $\bar{\boldsymbol{s}} = \bar{\boldsymbol{s}} + \boldsymbol{r}_i; i = i - 1$;
4: If i achieves zero, then the algorithm terminates with outputs; Otherwise, go to step 2.

C. Valley-filling Behavior

In this section, we intend to relate the TLPS problem to an interesting concept in the smart grid – valley filling.

A valley-filling aggregated charging profile $\boldsymbol{h}^{vf}$ is defined as follows [7]:

$$h_t^{vf} = \max\{0, v - d_t\}, \tag{4}$$

where v is a constant and $\boldsymbol{h}^{vf} = \begin{bmatrix} h_1^{vf} & h_2^{vf} & \dots & h_T^{vf} \end{bmatrix}'$.

In some models, a valley-filling profile does not necessarily exist in general, as shown in [14], and neither does the model in this paper. Firstly, there may not exist an integer v such that $\tau = \sum_{t=1}^{T} h_t^{vf}$. If such a v exists, it can be found by Algorithm 2. Otherwise, Algorithm 2 will go to an endless loop. Secondly, even if such an integral v exists, the resulted $\boldsymbol{h}^{vf}$ may not belong to $\mathcal{H}$; i.e., $\boldsymbol{h}^{vf} \not\prec \boldsymbol{r}^*$.

If Algorithm 1 generates a valley-filling aggregated charging profile $\boldsymbol{h}$, then it is exactly the one obtained by some v and formula (4). Moreover, the corresponding v can be obtained by Algorithm 2. Thus, our concern is whether the aggregated charging profile $\boldsymbol{h}$ resulted by Algorithm 1 is valley-filling, if we can actually achieve the valley-filling behavior by coordinating the given flexible loads appropriately. The answer is positive. When a valley-filling aggregated charging profile $\boldsymbol{h}^{vf}$ does exist, the resulted total load profile $(\boldsymbol{h}^{vf} + \boldsymbol{d})$ also belongs to the feasible set $\{(\boldsymbol{h} + \boldsymbol{d}) \mid \boldsymbol{h} \in \mathcal{H}\}$.

Algorithm 2 Finding Integral Constant v

Input: The base load profile $\boldsymbol{d}$ and the total demand τ.
Output: Integral constant v.
1: Initialization: $\boldsymbol{h}^{vf} = \boldsymbol{0}_T, v = 0, v^1 = 0, v^2 = \lfloor \tau / T \rfloor + \max\{d_t, t \in \mathcal{T}\} + 1$;
2: $v = \lfloor (v^1 + v^2)/2 \rfloor$; Update $\boldsymbol{h}^{vf}$ by v and formula (4); $\tau^1 = \sum_{t=1}^{T} h_t^{vf}$;
3: If $\tau^1 > \tau$, then let $v_2 = v$ and go to step 2; Otherwise, if $\tau^1 < \tau$, then let $v_1 = v$ and go to step 2; Otherwise, if $\tau^1 = \tau$, then the algorithm terminates with v.

In the following, we will show that $(\boldsymbol{h}^{vf} + \boldsymbol{d})$ is the only element in $\{(\boldsymbol{h} + \boldsymbol{d}) \mid \boldsymbol{h} \in \mathcal{H}\}$, which corresponds to the least element of $\{(\boldsymbol{h} + \boldsymbol{d})^{\downarrow} \mid \boldsymbol{h} \in \mathcal{H}\}$. In other words, $\boldsymbol{h}^{vf}$ can be obtained by Algorithm 1; i.e.,

$$\boldsymbol{h}^{vf} = \sum_{i=1}^{N} \boldsymbol{r}_i,$$

where $\boldsymbol{r}_i, i \in \mathcal{N}$ are the outputs of Algorithm 1.

Partition the elements of $(\boldsymbol{h}^{vf} + \boldsymbol{d})$ into two parts. The elements of the first part consist of those indexed by t with $(d_t + h_t^{vf}) > v$, while the remaining elements of $(\boldsymbol{h}^{vf} + \boldsymbol{d})$ are all v. It is obvious that we cannot obtain a different vector $\hat{\boldsymbol{s}}$ by rearranging the elements of $(\boldsymbol{h}^{vf} + \boldsymbol{d})$ such that $(\hat{\boldsymbol{s}} - \boldsymbol{d})$ is elementwise non-negative. We claim that $(\boldsymbol{h}^{vf} + \boldsymbol{d})^{\downarrow}$ is indeed the least element of $\mathcal{S}$. Since the total load profile $\bar{\boldsymbol{s}}$ generated by Algorithm 1 also corresponds to the least element of $\{(\boldsymbol{h} + \boldsymbol{d})^{\downarrow} \mid \boldsymbol{h} \in \mathcal{H}\}$, i.e., $\bar{\boldsymbol{s}}^{\downarrow} = (\boldsymbol{h}^{vf} + \boldsymbol{d})^{\downarrow}$, we have $\bar{\boldsymbol{s}} = (\boldsymbol{h}^{vf} + \boldsymbol{d})$.

The above analysis concludes that Algorithm 1 can check whether a valley-filling profile exists. A valley-filling profile exists if and only if the profile generated by Algorithm 1 is valley-filling. An alternative way to check the existence of a valley-filling profile is as follows. Firstly, the integral constant v can be calculated by Algorithm 2 and then we obtain $\boldsymbol{h}^{vf}$ by the formula (4). According to the Gale-Ryser theorem, we say that there exists a valley-filling profile if and only if $\boldsymbol{h}^{vf} \prec \boldsymbol{r}^*$.

V. Numerical Examples

In this section, we will present some numerical examples to interpret partial concepts and algorithms discussed in previous sections.

Example 1 illustrates the decomposition of loads with no single maximum charging rates, as shown in Remark 1.

Example 1: Let $T = 4$. Consider a flexible load n, whose requirement is specified by the charging demand $r_n = 7$ and the maximum charging rate $\bar{r}_n = 3$. Two feasible charging profiles are

$$\boldsymbol{r}_{n1} = \begin{bmatrix} 3 & 3 & 1 & 0 \end{bmatrix}' \text{ and } \boldsymbol{r}_{n2} = \begin{bmatrix} 1 & 1 & 2 & 3 \end{bmatrix}'.$$

Since $r_n = 2 * \bar{r}_n + 1$, this load can be regarded as the combination of three flexible sub loads, whose maximum charging rates are all 1s. Herein, the first sub load has the charging demand 3, while the other two sub loads require 2 units each.

As a result, the feasible charging profile $\boldsymbol{r}_{n1}$ can only be decomposed into the three zero-one charging profiles:

$$\begin{bmatrix} 1 & 1 & 1 & 0 \end{bmatrix}', \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}', \text{ and } \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}'.$$

On the other hand, the feasible charging profile $\boldsymbol{r}_{n2}$ can be recovered by several possible triples of zero-one charging profiles, e.g.,

$$\begin{bmatrix} 1 & 0 & 1 & 1 \end{bmatrix}', \begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix}', \text{ and } \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix}';$$

$$\begin{bmatrix} 1 & 1 & 0 & 1 \end{bmatrix}', \begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix}', \text{ and } \begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix}'.$$

However, the following triple is not consistent with the equivalent decomposition of load n:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}', \begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix}', \text{ and } \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}',$$

though the combination of each triple is still a feasible charging profile for the load n.

The next example recovers the process of Algorithm 1 and verifies Proposition 1 numerically.

Example 2: The base load profile and the demand profile are respectively given by

$$\begin{bmatrix} 7 & 1 & 2 & 5 & 2 \end{bmatrix}' \text{ and } \begin{bmatrix} 3 & 2 & 1 & 4 \end{bmatrix}'.$$

Assume all the loads take 1 as the maximum charging rate. The process of Algorithm 1 is recovered in Table I. On the other hand, if the input is the reversal of the original demand profile, the corresponding process is presented in Table II.

As shown in the Table I and II, the output total load profiles are equivalent, for the two different processes.

3	*	*	*	*	*
2	*	*	*	*	*
1	*	*	*	*	*
4	*	*	*	*	*
$\bar{\boldsymbol{s}}$	7	1	2	5	2
$\boldsymbol{d}$	7	1	2	5	2

$\Longrightarrow$

3	*	*	*	*	*
2	*	*	*	*	*
1	*	*	*	*	*
$\bar{\boldsymbol{s}}$	7	2	3	6	3
4	0	1	1	1	1
$\boldsymbol{d}$	7	1	2	5	2

$\Longrightarrow$

3	*	*	*	*	*
2	*	*	*	*	*
$\bar{\boldsymbol{s}}$	7	3	3	6	3
1	0	1	0	0	0
4	0	1	1	1	1
$\boldsymbol{d}$	7	1	2	5	2

$\Rightarrow$

3	*	*	*	*	*
$\bar{\boldsymbol{s}}$	7	3	4	6	4
2	0	0	1	0	1
1	0	1	0	0	0
4	0	1	1	1	1
$\boldsymbol{d}$	7	1	2	5	2

$\Longrightarrow$

$\bar{\boldsymbol{s}}$	7	4	5	6	5
3	0	1	1	0	1
2	0	0	1	0	1
1	0	1	0	0	0
4	0	1	1	1	1
$\boldsymbol{d}$	7	1	2	5	2

TABLE I

4	*	*	*	*	*
1	*	*	*	*	*
2	*	*	*	*	*
3	*	*	*	*	*
$\bar{\boldsymbol{s}}$	7	1	2	5	2
$\boldsymbol{d}$	7	1	2	5	2

$\Longrightarrow$

4	*	*	*	*	*
1	*	*	*	*	*
2	*	*	*	*	*
$\bar{\boldsymbol{s}}$	7	2	3	5	3
3	0	1	1	0	1
$\boldsymbol{d}$	7	1	2	5	2

$\Longrightarrow$

4	*	*	*	*	*
1	*	*	*	*	*
$\bar{\boldsymbol{s}}$	7	3	4	5	3
2	0	1	1	0	0
3	0	1	1	0	1
$\boldsymbol{d}$	7	1	2	5	2

$\Rightarrow$

4	*	*	*	*	*
$\bar{\boldsymbol{s}}$	7	4	4	5	3
1	0	1	0	0	0
2	0	1	1	0	0
3	0	1	1	0	1
$\boldsymbol{d}$	7	1	2	5	2

$\Longrightarrow$

$\bar{\boldsymbol{s}}$	7	5	5	6	4
4	0	1	1	1	1
1	0	1	0	0	0
2	0	1	1	0	0
3	0	1	1	0	1
$\boldsymbol{d}$	7	1	2	5	2

TABLE II

Example 3 presents the connection between our results and the valley-filling behavior. In general, a valley-filling profile is not feasible. As long as it is feasible, it can be achieved by Algorithm 1.

Example 3: The base load profile and the demand profile are those given in Example 2, and the conjugate of the demand supply is $\begin{bmatrix} 4 & 3 & 2 & 1 \end{bmatrix}'$.

If a valley-filling aggregated charging profile exists, it should be $\begin{bmatrix} 0 & 4 & 3 & 0 & 3 \end{bmatrix}'$, where $h_t = \max\{0, v - d_t\}$

and $v = 5$. However, such an aggregated charging profile is not feasible, since

$$\begin{bmatrix}0 & 4 & 3 & 0 & 3\end{bmatrix}' \not\preceq \begin{bmatrix}4 & 3 & 2 & 1 & 0\end{bmatrix}'.$$

If the demand profile is changed to $\begin{bmatrix}2 & 2 & 3 & 3\end{bmatrix}'$, the possible valley-filling aggregated profile remains the same. Moreover, the majorization condition is satisfied. The constructing process is given in Table III.

$$\begin{array}{c|ccccc} \hline \mathbf{2} & * & * & * & * & * \\ \mathbf{2} & * & * & * & * & * \\ \mathbf{3} & * & * & * & * & * \\ \mathbf{3} & * & * & * & * & * \\ \hline \bar{s} & 7 & 1 & 2 & 5 & 2 \\ \hline \hline d & 7 & 1 & 2 & 5 & 2 \\ \hline \end{array} \Longrightarrow \begin{array}{c|ccccc} \hline \mathbf{2} & * & * & * & * & * \\ \mathbf{2} & * & * & * & * & * \\ \mathbf{3} & * & * & * & * & * \\ \hline \bar{s} & 7 & 2 & 3 & 5 & 3 \\ \hline \hline \mathbf{3} & 0 & 1 & 1 & 0 & 1 \\ d & 7 & 1 & 2 & 5 & 2 \\ \hline \end{array} \Longrightarrow \begin{array}{c|ccccc} \hline \mathbf{2} & * & * & * & * & * \\ \mathbf{2} & * & * & * & * & * \\ \hline \bar{s} & 7 & 3 & 4 & 5 & 4 \\ \hline \hline \mathbf{3} & 0 & 1 & 1 & 0 & 1 \\ \mathbf{3} & 0 & 1 & 1 & 0 & 1 \\ d & 7 & 1 & 2 & 5 & 2 \\ \hline \end{array}$$

$$\Rightarrow \begin{array}{c|ccccc} \hline \mathbf{4} & * & * & * & * & * \\ \hline \bar{s} & 7 & 4 & 5 & 5 & 4 \\ \hline \hline \mathbf{2} & 0 & 1 & 1 & 0 & 0 \\ \mathbf{3} & 0 & 1 & 1 & 0 & 1 \\ \mathbf{3} & 0 & 1 & 1 & 0 & 1 \\ d & 7 & 1 & 2 & 5 & 2 \\ \hline \end{array} \Longrightarrow \begin{array}{c|ccccc} \hline \hline \bar{s} & 7 & 5 & 5 & 5 & 5 \\ \hline \hline \mathbf{2} & 0 & 1 & 0 & 0 & 1 \\ \mathbf{2} & 0 & 1 & 1 & 0 & 0 \\ \mathbf{3} & 0 & 1 & 1 & 0 & 1 \\ \mathbf{3} & 0 & 1 & 1 & 0 & 1 \\ d & 7 & 1 & 2 & 5 & 2 \\ \hline \end{array}$$

TABLE III

VI. Conclusion and Future Work

In this paper, a total load profile smoothing problem (1) is considered. Given the base load profile, we intend to smooth the total load profile in terms of majorization by way of coordination of a group of flexible loads. Such problem can be reformulated as an optimal matrix completion problem (2) and further an optimization problem in the majorization order (3). We show that, for such a minimization problem, the least element exists in the corresponding partially ordered set. An efficient algorithm can help us find the optimal total load profile together with the corresponding charging profile of each flexible load. As a sequential algorithm, it can still work when the number of flexible loads is extremely large. Moreover, this algorithm can help us check whether a valley-filling profile exists. If it does, the outputs of the algorithm will present the valley-filling behavior.

In the future, we plan to solve the TLPS problem in the presence of heterogeneous flexible loads. Concretely speaking, different loads may have different arrival times and different deadlines. Under such constraints, a simple solution algorithm will no longer work in general. On the other hand, the majorization relation gives us a hint to treat the total loads of several time slots as a whole. As often observed in real life, the price of electricity may be determined by not only the total load of the current time slot but also the total loads of previous and future time slots. We intend to find the theoretical implications by way of majorization theory.

References

[1] M. L. Tuballa and M. L. Abundo, "A review of the development of smart grid technologies," *Renew. Sustain. Energy Rev.*, vol. 59, pp. 710–725, 2016.

[2] S. Majumdar, D. Chattopadhyay, and J. Parikh, "Interruptible load management using optimal power flow analysis," *IEEE Trans. Power Syst.*, vol. 11, no. 2, pp. 715–720, 1996.

[3] A. Subramanian, M. Garcia, A. Dominguez-Garcia, D. Callaway, K. Poolla, and P. Varaiya, "Real-time scheduling of deferrable electric loads," in *Proc. Am. Control Conf. (ACC)*, pp. 3643–3650, 2012.

[4] W. Chen, L. Qiu, and P. Varaiya, "Duration-deadline jointly differentiated energy services," in *54th IEEE Conf. Dec. Control (CDC)*, pp. 7220–7225, 2015.

[5] Y. Mo, W. Chen, and L. Qiu, "Duration-differentiated energy services with peer-to-peer charging," in *55th IEEE Conf. Dec. Control (CDC)*, pp. 7514–7519, 2016.

[6] Y. Mo, W. Chen, and L. Qiu, "Differentiated energy services: Multiple arrival times and multiple deadlines," in *20th IFAC World Congress*, pp. 209–214, 2017.

[7] A. Chakrabortty and M. D. Ilić, *Control and Optimization Methods for Electric Smart Grids*. Springer, 2012.

[8] P. Palensky and D. Dietrich, "Demand side management: Demand response, intelligent energy systems, and smart loads," *IEEE Trans. Ind. Informat.*, vol. 7, no. 3, pp. 381–388, 2011.

[9] P. P. Varaiya, F. F. Wu, and J. W. Bialek, "Smart operation of smart grid: Risk-limiting dispatch," *Proc. IEEE*, vol. 99, no. 1, pp. 40–57, 2011.

[10] G. Binetti, A. Davoudi, D. Naso, B. Turchiano, and F. L. Lewis, "Scalable real-time electric vehicles charging with discrete charging rates," *IEEE Trans. Smart Grid*, vol. 6, no. 5, pp. 2211–2220, 2015.

[11] M. Yilmaz and P. T. Krein, "Review of battery charger topologies, charging power levels, and infrastructure for plug-in electric and hybrid vehicles," *IEEE Trans. Power Electron.*, vol. 28, no. 5, pp. 2151–2169, 2013.

[12] L. Gan, U. Topcu, and S. H. Low, "Optimal decentralized protocol for electric vehicle charging," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 940–951, 2013.

[13] Z. Ma, D. S. Callaway, and I. A. Hiskens, "Decentralized charging control of large populations of plug-in electric vehicles," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 1, pp. 67–78, 2013.

[14] W.-J. Ma, V. Gupta, and U. Topcu, "On distributed charging control of electric vehicles with power network capacity constraints," in *Proc. Am. Control Conf. (ACC)*, pp. 4306–4311, 2014.

[15] N. Chen, C. W. Tan, and T. Q. Quek, "Electric vehicle charging in smart grid: Optimality and valley-filling algorithms," *IEEE J. Sel. Top. Signal Process.*, vol. 8, no. 6, pp. 1073–1083, 2014.

[16] A. W. Marshall, I. Olkin, and B. C. Arnold, *Inequalities: Theory of Majorization and Its Applications*. Springer Series in Statistics, 2 ed., 2011.

[17] G. Szász, *Introduction to Lattice Theory*. Academic Press, 1964.

[18] G. D. James, *The Representation Theory of the Symmetric Groups*, vol. 682. Springer, 2006.

[19] D. Gale, "A theorem on flows in networks," *Pacific J. Math.*, vol. 7, no. 2, pp. 1073–1082, 1957.

[20] H. J. Ryser, "Combinatorial properties of matrices of zeros and ones," *Canad. J. Math.*, vol. 9, pp. 371–377, 1957.

[21] R. A. Brualdi and H. J. Ryser, *Combinatorial Matrix Theory*, vol. 39. Cambridge University Press, 1991.

[22] R. A. Brualdi, *Combinatorial Matrix Classes*, vol. 13. Cambridge University Press, 2006.