Deferrable loads in an energy market: coordination under congestion constraints

Donatello Materassi, Saverio Bolognani, Mardavij Roozbehani, Munther A. Dahleh

Abstract— We consider a scenario where price-responsive energy consumers are allowed to optimize their individual utilities via mechanisms of load-shifting in a distribution network subject to capacity constraints. The uncoordinated selfish behavior of the consumers would lead, in general, to requests that could not be served by the distribution network because of such constraints. Thus, a centralized or hierarchically coordination mechanism is required. We derive algorithms and methods to determine in real-time the largest set of consumers' decisions that are compatible with the physical constraints of the network and capable of avoiding congestion phenomena in the future.

These methods are shown to be applicable to the design of coordination mechanisms with the aim of providing a large number of degrees of freedom to the users while guaranteeing the integrity of the system.

I. INTRODUCTION

The introduction of mechanisms of price-response in the power grid is expected to happen via the design, implementation, and deployment, of a large number of individual decision makers corresponding to different participants in the energy market, each one optimizing its own independent utility function [1], [2], [3], [4]. The behavior of the individual user necessarily depends on its specific utility of consumption, which, typically, is not just a static function of the delivered electric power. Indeed, in many noticeable cases – Electric vehicles (EV), thermostatic loads, heating, ventilation, and air conditioning (HVAC) systems, manifacturing processes, and residential loads – the user utility is described by quite complex individual models that might include the possibility of deferring the load, a penalty for postponing consumption and soft or hard deadlines.

Current power grids are managed under the assumptions that energy consumption is relatively predictable in aggregate and that user demand is substantially inelastic [5]. The introduction of a large number of price-responsive loads will challenge these assumptions and potentially affect how grids will need to be managed in many fundamental aspects.

The impact of uncoordinated response of flexible loads to a price signal has been analyzed, for example in [6], [7], [8]. In particular, [8] studies the effect of a widespread adoption of Electric Vehicles (EVs) on a large scale, while [7] considers the impact of EVs in terms of power losses at the level

of the distribution network. These studies suggest that, in order to guarantee the reliability of the grid, individual users cannot be allowed to participate freely in the retail energy market, and some level of supervision is needed. This is the main focus of this article, where we enlarge the problem perspective from the determination of optimal consumption policies of a single energy user, e.g. [9], to the scenario where multiple energy users need to be coordinated to meet the physical constraints imposed by the system.

In particular, we consider a specific class of congestion phenomena in the power network, consisting in overstepping capacity limits on the maximum aggregate power that can be delivered to a pool of users at the same time. Indeed, there are many practical scenarios where the operational constraints have this form - for example large fleet of electric vehicles connected to the same charging station, where each vehicle has to satisfy its energy demand (full battery charge) within some firm deadline. It is also reasonable to expect that the limit on the maximum power at the distribution substation would correspond to constraints of the same type.

This type of constraints have already been analyzed in [10], for example, where however no stochastic prices are considered. Heuristic scheduling strategies have been proposed in [11], in order to schedule deferrable loads in a energy market, for better exploitation of renewable sources and of reserve capacity. A decentralized protocol for day-ahead load scheduling has been proposed in [12], aiming at flat aggregate consumption profiles, while both [13] and [14] consider the issue of power losses minimization in their centralized load control. In all these examples, loads are dispatched by a (possibly distributed) scheduler, to which the user have to reveal information about their utility function. Our formulation, instead, intends to provide a framework where individual policies of the users in response to stochastic energy prices can be incorporated seamlessly, so that the aggregator has to guarantee only the satisfaction of operational constraints of the grid, while users are allowed to participate to the market according to their own strategy and interests, without even disclosing them.

We can use a simple example to describe the mechanism that make congestion to arise when different users are allowed to pursue their own individual utility with no supervision. Consider the problem of charging two identical batteries, $B^{(1)}$ and $B^{(2)}$. Assume that each battery requires a total of $E^{(1)} = E^{(2)} = 2$ units of energy by the time $T = 4$. At each time $k = 0, 1, 2, 3,$, the i-th battery can be charged by $u_k^{(i)} \geq 0$ units of energy with a maximum rate of $\overline{u}^{(i)} = 1$ per time unit, for $i = 1, 2$. In other terms, we have that

k	0	1	2	3
$B^{(1)}$	▩	▩		
$B^{(2)}$	▩	▩		
$u_k^{(1)} + u_k^{(2)}$	2	2	0	0

(a)

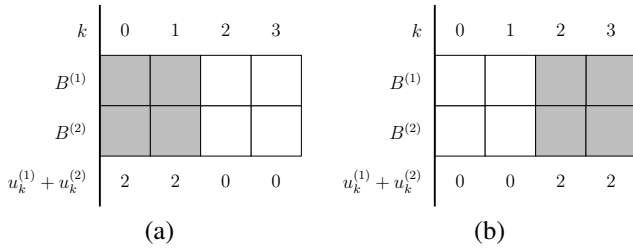| k	0	1	2	3
$B^{(1)}$			▩	▩
$B^{(2)}$			▩	▩
$u_k^{(1)} + u_k^{(2)}$	0	0	2	2

(b)

Fig. 1. Example of uncoordinated response of two batteries in a regime of dynamic-pricing. In (a) we see the response associated to *low* energy prices for $k = 0, 1$: both batteries decide to consume. In (b) we see the response to *low* prices for $k = 0, 1$: both batteries postpone consumption, activating too late when they cannot consume at the same time.

$0 \leq u_k^{(i)} \leq \overline{u}^{(i)}$ for $k = 0, 1, 2, 3$ and $i = 1, 2$. In addition, assume that the two batteries are subject to the capacity constraint $\overline{U}$ of their charging station: $u_k^{(1)} + u_k^{(2)} \leq \overline{U} = 1$ for $k = 0, 1, 2, 3$ and, being identical, are going to follow the same decision strategy - for example the optimal cost-minimizing strategy proposed in [9].

According to such a strategy, in the case of a "low" energy price during the time intervals $k = 0, 1$ (see Figure 1a) both users would decide to consume at their maximum rate, over-stepping the global constraint given by $u_k^{(1)} + u_k^{(2)} \leq \overline{U} = 1$. Via simple contention mechanisms, it might be possible to split the total rate $\overline{U}$ fairly between the two batteries, so that both users would have their demand satisfied by their deadlines. However, a more complex case of congestion happens in the case of "high" prices during the time intervals $k = 0, 1$ (see Figure 1b). In this case both batteries would postpone their consumption, expecting lower prices in the future. However, as soon as they both decide not to consume at some time k, they end up in a condition where their demands can not be satisfied any more within their deadlines.

Such congestion scenarios are not only possible, but also likely to appear, if the users react to the same price signal and therefore tend to have correlated behaviors.

Our results suggest that a possible architecture capable of avoiding congestion phenomena consists of an *aggregator* that has the duty of collecting bids from the individual batteries and clearing a auction. "Aggregators" that guarantee the correct behavior of the power grid have already been proposed (see [6] and [8] and references therein). However, the functions of these entities are often described in a qualitative way, without providing a formal characterization of what their behavior or their implementation should be.

Our main contribution is to precisely characterize the acceptable decisions from the the users that guarantee a feasible consumption and the satisfaction at the specified deadlines, without jeopardizing the schedulability of the charging problem in the future consumption horizon. In particular, we show that, the space of admissible consumption patterns can be algorithmically computed, and has the structure of a Partially Ordered set (POset). Such structure can be incorporated in the energy auction, in the form of a "constrained multi-object auctions" similar to those that have been studied in [15], [16], [17].

The paper has the following structure. We first provide a mathematical formulation of the problem (Section II), describe the solution in the simplified scenario in which all users have the same deadline (Section III), and then we extend the solution to the general case (Section IV). In Section V we illustrate one possible way to use the proposed methodology in the design of a constrained auction for the energy market.

II. PROBLEM FORMULATION

Each deferrable load (e.g. battery) is modeled as a *task*, defined as follows.[1]

Definition 1 (Task): A task is a 3-tuple $(E, T, \overline{u}) \in \mathbb{R}^3$ where $E > 0$ defines the demand, T is the deadline, i.e. the time left to satisfy such demand, and $\overline{u} > 0$ is the maximum rate at which the task can be served.

In line with many practical applications, and with minimal loss of generality, we introduce a discretization of time and of the parameters that define the tasks. We assume a unitary sampling time and introduce the following definition.

Definition 2 (Integer task): A task $(E, T, \overline{u})$ is *integer* if $T \in \mathbb{N}$ and if $\frac{E}{\overline{u}} \in \mathbb{N}$.

We can therefore introduce the following discrete time model to describe the dynamic process of task completion (or battery charging)

$$x_{k+1} = x_k + u_k$$
$$x_0 = -E$$

where $x_k \leq 0$ is the state of completion of the task (the state of charge), and $u_k \geq 0$ is the rate at which the task is served (in a piece-wise constant way) during the k-th interval, namely between time k and $k + 1$.

The constraint on the maximum rate, and the deadline for task completion, corresponds respectively to the inequalities

$$0 \leq u_k \leq \overline{u}, \quad \text{for } k = 0, \ldots, T - 1,$$

and the equality

$$x_T = 0.$$

We now consider the case of multiple users, whose behaviors are coupled by the fact that, at any time, the aggregate service rate (the sum of the rates at which different tasks are served) cannot exceed a given limit $\overline{U}$.

We therefore consider N tasks $(E^{(i)}, T^{(i)}, \overline{u}^{(i)})$, where $i = 1, ..., N$, and we consider the following problem.

Definition 3 (Task scheduling problem): We define the *task scheduling problem* as the problem of deciding the variables $u_k^{(i)}$ for $k = 0, ..., T := \max_{1 \leq i \leq N} T^{(i)}$ such that for any task i

$$\begin{aligned} x_{k+1}^{(i)} &= x_k^{(i)} + u_k^{(i)} \\ x_0^{(i)} &= -E^{(i)} \\ x_{T^{(i)}}^{(i)} &= 0 \end{aligned} \qquad (1)$$

---

[1]The generality of the terminology introduced in this section indicates how the same model can effectively describe other equivalent scenarios, e.g. real-time scheduling in multiprocessor machines [18].

where $x_k^{(i)} \leq 0$ is the state of task i at time k, and such that, at any time k,

$$0 \leq u_k^{(i)} \leq \overline{u}^{(i)} \quad \text{for } i = 1, ..., N$$
$$\sum_{i=1}^{N} u_k^{(i)} \leq \overline{U}. \qquad (2)$$

We refer to T as the horizon of the problem.

We introduce the following definitions.

Definition 4 (Action): We define the *action* $\mathbf{u}_k$ at time k as the vector in $\mathbb{R}^N$ obtained by stacking the rates $u_k^{(i)}$ of the different tasks.

Definition 5 (Feasible schedule): Given the tasks $(E^{(i)}, T^{(i)}, \overline{u}^{(i)})$ for $i = 1, ..., N$, and an aggregate limit $\overline{U}$, a schedule (i.e. a series of actions) $\{\mathbf{u}_k\}_{k=0}^{T-1}$ is *feasible* if it solves the task scheduling problem, i.e. it satisfies the constraints (1) and (2).

Definition 6 (Schedulable tasks): Given an aggregate limit $\overline{U}$, the tasks $(E^{(i)}, T^{(i)}, \overline{u}^{(i)})$ for $i = 1, ..., N$, are *schedulable* if a feasible schedule for the corresponding task scheduling problem exists.

As explained in the introduction, the goal of the analysis presented in this paper is to give necessary and sufficient conditions so that the actions of the users meet the constraints of the task scheduling problem, and do not compromise the possibility of generating a feasible schedule in the future, given the current state of completion of the tasks.

Our analysis is performed at time $k = 0$ and provides guarantees that, if the tasks are schedulable to begin with, then a certain action $\mathbf{u}_0$ leaves the tasks in a schedulable configuration over the horizon $[1, \max_{1 \leq i \leq N} T^{(i)}]$. Then, the same approach can be iteratively applied.

An effective characterization of the set of all (and only) the admissible actions allows to derive coordination strategies among the users, possibly in the form of an aggregator, as explained in the Introduction.

We therefore introduce the following formal definitions.

Definition 7 (Admissible action): Given the tasks $(E^{(i)}, T^{(i)}, \overline{u}^{(i)})$ for $i = 1, ..., N$, and an aggregate limit $\overline{U}$, an action

$$\mathbf{u}_0 = \begin{bmatrix} u_0^{(1)} \\ \vdots \\ u_0^{(N)} \end{bmatrix}$$

is *admissible* if:

- it satisfies

$$0 \leq u_0^{(i)} \leq \overline{u}^{(i)}, \; i = 1, \ldots, N, \quad \text{and} \quad \sum_{i=1}^{N} u_0^{(i)} \leq \overline{U},$$

- the tasks $(E^{(i)} - u_0^{(i)}, T^{(i)} - 1, \overline{u}^{(i)})$ are schedulable with the global constraint $\overline{U}$.

We define by $\mathcal{U}_0$ the set of all admissible actions.

We now make a technical assumption, with minimal loss of generality.

Assumption 8: The tasks $(E^{(i)}, T^{(i)}, \overline{u}^{(i)})$, $i = 1, ..., N$, are all integer tasks. Moreover, there exist a common $\overline{u}$, that

we denote as the *unit rate*, such that

$$\frac{E^{(i)}}{\overline{u}} \in \mathbb{N}, \quad \frac{\overline{u}^{(i)}}{\overline{u}} \in \mathbb{N}, \quad \frac{\overline{U}}{\overline{u}} \in \mathbb{N},$$

for all $i = 1, ..., N$.

Based on this, we introduce the following definition.

Definition 9 (Integer schedule): Given a task scheduling problem, a schedule is integer if all its elements are integer multiples of a unit rate $\overline{u}$.

In the following, we will focus only on *integer schedules*. This choice is reasonable in the common practice, where most of the time tasks can be served at some quantized rate of service (i.e. according to the number of processors assigned to the thread in a multiprocessor system [18], or according to some of the EV battery charging standards where different levels of charging speed are available).

It is worth remarking that, even if the problem of finding a solution to the scheduling problem (1)-(2) becomes easier if the integer constraints are relaxed (and reduces to the application of linear programming to load scheduling [19], [20]), a practical characterization of all the feasible actions remains a difficult problem. On the other hand, the following result shows that the restriction to integer schedules has no effect on the schedulability analysis of the task scheduling problem.

Theorem 10: Consider a set of N tasks $(E^{(i)}, T^{(i)}, \overline{u}^{(i)})$, with an aggregate bound $\overline{U}$, and let Assumption 8 hold. The tasks are schedulable if and only if an integer schedule exists.

Proof: See the appendix. ∎

The following proposition shows that if Assumption 8 holds, there is no loss of generality by assuming that all tasks have the same rate constraint $\overline{u}^{(i)} = 1$.

Proposition 11: Consider a task scheduling problem defined by the tasks $(E^{(i)}, T^{(i)}, \overline{u}^{(i)})$ for $i = 1, ..., N$, and let Assumption 8 hold. It is possible to define an equivalent task scheduling problem in the form

$$\chi_{k+1}^{(i,j)} = \chi_k^{(i,j)} + v_k^{(i,j)}$$
$$\chi_0^{(i,j)} = -\mathcal{E}^{(i,j)}$$
$$\chi_{\tau^{(i,j)}}^{(i,j)} = 0$$
$$0 \leq v_k^{(i,j)} \leq 1$$
$$\sum_{i=1}^{N} \sum_{j=1}^{\rho^{(i)}} v_k^{(i,j)} \leq \overline{V},$$

where

- the decision variables are $v_k^{(i,j)}$,
- $\rho^{(i)} = \overline{u}^{(i)}/\overline{u}$,
- $\overline{V} = \overline{U}/\overline{u}$,
- the tasks are defined as

$$\left(\mathcal{E}^{(i,j)} = \frac{E^{(i)}}{\overline{u}^{(i)}}, \tau^{(i,j)} = T^{(i)}, 1 \right)$$

for $i = 1, ..., N$ and $j = 1, ..., \rho^{(i)}$. Such task scheduling problem satisfies Assumption 8 with $\overline{u} = 1$. The decision

variables of the original problem can then be recovered via

$$u_k^{(i)} = \overline{u} \sum_{j=1}^{\rho^{(i)}} v_k^{(i,j)}. \tag{3}$$

Proof: The proof is by inspection. ∎

Based on this result, in the rest of the paper we will consider tasks with identical maximum rate equal to 1, unless otherwise specified. Therefore, the choice of considering only integer schedules corresponds, via (3), to having schedules with binary decision variables, i.e.

$$u_k^{(i)} \in \{0, 1\}, \quad \text{for } k =, ..., T - 1; \ i = 1, ..., N.$$

Definition 12 (Aggregate service rate): Given an action $\mathbf{u}_0$, we indicate the *aggregate service rate* as

$$\|\mathbf{u}_0\|_1 = \sum_{i=1}^{N} u_0^{(i)}. \tag{4}$$

We also borrow the notion of slack from the area of multi-processor scheduling [18].

Definition 13 (Slack): We define the *slack* of a task as

$$s^{(i)} := T^{(i)} - E^{(i)} \tag{5}$$

The slack represents the maximum idle time that the task can wait, before having to be served, in order to meet its deadline (remember that $\overline{u}^{(i)} = 1$). Based on this definition, we introduce a function of the action $\mathbf{u}_0$ that returns the slacks of the tasks that are served according to that action:

$$\mathbf{s} : \{0, 1\}^N \rightarrow \mathbb{N}^{\|\mathbf{u}_0\|_1}$$
$$\mathbf{s}(\mathbf{u}_0) = \left[s^{(i)}\right]_{i:u_0^{(i)}=1}. \tag{6}$$

As an example, in the case of an action serving tasks 2 and 3 among 5 tasks, we have that

$$\mathbf{s}\left(\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}\right) = \begin{bmatrix} s^{(2)} \\ s^{(3)} \end{bmatrix}.$$

It is possible to define a *partial ordering* relation between these vectors via the following definition.

Definition 14 (Slack partial ordering): Let $\mathbf{s}'$ and $\mathbf{s}''$ be two vectors of the dimensions d' and d'', respectively. We say that

$$\mathbf{s}' \preccurlyeq \mathbf{s}''$$

if $d' \geq d''$ and the sorted permutation of the d'' smallest elements of $\mathbf{s}'$ is element-wise smaller than the sorted permutation of $\mathbf{s}''$.

As an example, we have that $\begin{bmatrix} 1 \\ 2 \end{bmatrix} \preccurlyeq \begin{bmatrix} 3 \\ 5 \end{bmatrix}$, and also $\begin{bmatrix} 1 \\ 2 \\ 8 \end{bmatrix} \preccurlyeq \begin{bmatrix} 3 \\ 5 \end{bmatrix}$, while $\begin{bmatrix} 1 \\ 4 \end{bmatrix}$ and $\begin{bmatrix} 2 \\ 3 \end{bmatrix}$ are not comparable, and neither $\begin{bmatrix} 1 \\ 4 \\ 8 \end{bmatrix}$ and $\begin{bmatrix} 2 \\ 3 \end{bmatrix}$ are.

III. CHARACTERIZATION OF ADMISSIBLE POLICIES (HOMOGENEOUS DEADLINES)

In this section we consider the case where all the tasks have the same deadline, namely $T^{(i)} = T$, for $i = 1, ..., N$. The following result characterize the schedulability of a set of tasks, for generic maximum rates $\overline{u}^{(i)}$.

Lemma 15: Consider a set of N tasks $(E^{(i)}, T, \overline{u}^{(i)})$, for $i = 1, ..., N$, sharing the same deadline T, and let Assumption 8 hold. The tasks are schedulable if and only if the two following conditions are met

$$\sum_{i=1}^{N} \frac{E^{(i)}}{\overline{U}} \leq T, \qquad \max_{1 \leq i \leq N} \frac{E^{(i)}}{\overline{u}^{(i)}} \leq T. \tag{7}$$

Proof: See the Appendix. ∎

The characterization of schedulability proposed in Lemma 15 is the key tool that allows to give a complete characterization of the set $\mathcal{U}_0$ of admissible decision.

We can therefore state the main result for the case of homogeneous deadlines.

Theorem 16: Consider a set of N schedulable tasks $(E^{(i)}, T, 1)$, for $i = 1, ..., N$, sharing the same deadline T, and let Assumption 8 hold. Then the set of admissible integer decisions at time 0 is

$$\mathcal{U}_0 = \bigcup_{y=\max\{\underline{U}, z\}}^{\overline{U}} \{\mathbf{u}_0 : \|\mathbf{u}_0\|_1 = y \text{ and } \mathbf{s}(\mathbf{u}_0) \preccurlyeq \mathbf{0}_z\},$$

where z is the number of tasks with slack zero

$$z = |\{i : s^{(i)} = 0\}|,$$

the symbol $\mathbf{0}_z$ indicates a vector of size z in which all elements are 0, and $\underline{U}$ is defined as

$$\underline{U} = \sum_{i=1}^{N} E^{(i)} - (T - 1)\overline{U}. \tag{8}$$

Proof: See the Appendix. ∎

Notice that the characterization of the set of admissible decisions $\mathcal{U}_0$ can be translated in the two following necessary and sufficient conditions:

- the aggregate service rate $\|\mathbf{u}_0\|_1$ of all the tasks cannot be smaller than $\underline{U}$, as defined in (8);
- tasks with slack $s^{(i)}$ equal to zero must be served.

A. The two-battery problem revisited

Let us consider again the problem of charging two identical batteries that was presented in the introduction and let us analyze it in light of the results obtained in this section. We can formalize it as a task scheduling problem where $(E^{(1)}, T^{(1)}, \overline{u}^{(1)}) = (E^{(2)}, T^{(1)}, \overline{u}^{(1)}) = (2, 4, 1)$ with an aggregate limit $\overline{U} = 1$. From Equation (8), we can compute $\underline{U} = 1$, while from Equation (5) we can compute $s^{(1)} = s^{(2)} = 2$. As both the slacks are greater than zero, from Theorem 16 we know that any decision such that $\|\mathbf{u}_0\|_1 = 1$ is admissible, and thus

$$\mathcal{U}_0 := \{(1, 0)^T, (0, 1)^T\}.$$

In other words, at least one of the two batteries has to consume at $k = 0$, otherwise feasibility of the problem will be irreversibly lost at $k = 1$. We showed in the Introduction how, in the case of a high price at time $k = 0$, neither of the two batteries would consume if they are following a selfish optimal strategy for cost minimization. Our result suggests

that the existence of an Aggregator monitoring the situation and implementing, for example, a constrained multi-object auction to assign a minimal amount of energy when needed, is a viable solution to the congestion problem.

IV. CHARACTERIZATION OF ADMISSIBLE POLICIES (HETEROGENEOUS DEADLINES)

In this section, we study the set $\mathcal{U}_0$ of admissible decisions in the case of batteries with different deadlines.

We first present an algorithm which allows to test the schedulability of a given set of tasks.

Minimum Effort Algorithm

INPUT
- tasks: $(E^{(i)}, T^{(i)}, 1)$, $i = 1, ..., N$
- aggregate limit: $\overline{U}$

EXECUTE
1) INITIALIZE
    - $T := \max_{1 \le i \le N} T^{(i)}$
    - $u_k^{(i)} = 0$ for $0 \le k < T$, $1 \le i \le N$
2) FOR $k = T - 1 : -1 : 0$
    - Create **ActiveTasks**:= list of all tasks with deadline $T^{(i)} > k$ and demand $E^{(i)} > 0$
    - Compute $N_A := $ **#** of elements in **ActiveTasks**
    - Order **ActiveTasks** according to their *reverse* slack $r^{(i)} := k + 1 - E^{(i)}$
    - For the first $\min(N_A, \overline{U})$ tasks with least reverse slack $r^{(i)}$ in **ActiveTasks**, decrease $E^{(i)}$ by 1 and assign $u_k^{(i)} := 1$

    OUTPUT:
    - schedulability: if $E^{(i)} = 0$ for all $i = 1, ..., N$ return **TRUE** otherwise return **FALSE**
    - schedule: $u_k^{(i)} = 0$ for $0 \le k < T$, $1 \le i \le N$
    - effort: $\underline{U} := \sum_{i=1}^{N} u_0^{(i)}$

The algorithm is clearly polynomial in the number of tasks and in the length of the time horizon (the longest deadline). An example of application of the Minimum Effort Algorithm is presented in Figure 2 for the completion of 7 tasks with heterogeneous deadlines.

The following result shows how the proposed Minimum Effort Algorithm is effective in finding a feasible schedule, if such a schedule exists.

Theorem 17: Consider the tasks $(E^{(i)}, T^{(i)}, 1)$ for $i = 1, ..., N$, together with the aggregate constraint $\overline{U}$, and let Assumption 8 hold. The tasks are schedulable if and only if the Minimum Effort Algorithm returns **TRUE**. Also, if the tasks are schedulable the Minimum Effort Algorithm returns a feasible integer schedule.

Proof: See the Appendix. ∎

The scheduling generated by the algorithm exhibit one specific property that will be useful in the characterization of the set $\mathcal{U}_0$ of admissible decisions, and gives the name to the algorithm.

Lemma 18: Consider the tasks $(E^{(i)}, T^{(i)}, 1)$ for $i = 1, ..., N$, together with the aggregate constraint $\overline{U}$, and let Assumption 8 hold. Let $\underline{U}$ be the effort computed by the Minimum Effort Algorithm applied to such task scheduling problem. Then for any integer admissible decision $\mathbf{u}_0 \in \mathcal{U}_0$, we necessarily have that

$$\|\mathbf{u}_0\|_1 \ge \underline{U}.$$

Proof: See the Appendix. ∎

Lemma 18 states that the Minimum Effort Algorithm returns a lower bound $\underline{U}$ on the aggregate consumption that must happen at time 0, in order to maintain schedulability of the problem at the next time step. Also, the Minimum Effort Algorithm guarantees the existence of a feasible schedule with aggregate consumption exactly equal to $\underline{U}$ at time $k = 0$. However, an aggregate consumption larger or equal to $\underline{U}$ at time $k = 0$ is not a sufficient condition. Indeed, consider for example the task scheduling problem represented in Figure 2. The Minimum Effort Algorithm returned $\underline{U} = 3$, therefore via Lemma 18 we know that the aggregate consumption at time 0 must be at least 3, for any feasible scheduling. However, one can easily check that the decision of serving tasks $B^{(1)}$, $B^{(4)}$, and $B^{(7)}$ is not admissible: in fact, the system was scheduled to work at full aggregate rate $\overline{U}$ up to time $k = 4$, and therefore there is no way to serve task $B^{(7)}$ at time 0, as there is no other task that can be served at time 5 or later.

Conversely, it is also clear that the decision computed by the algorithm is not the only admissible one. In the same example, it is easy to check that the decision of serving tasks $B^{(1)}$, $B^{(3)}$, and $B^{(6)}$ is also admissible.

The following result clarifies what is the mathematical structure of the set $\mathcal{U}_0$ of admissible decisions, via the partial ordering relation introduced in Section II.

Theorem 19: Consider the tasks $(E^{(i)}, T^{(i)}, 1)$ for $i = 1, ..., N$, together with the aggregate constraint $\overline{U}$, and let Assumption 8 hold. Let $\mathbf{u}_0$ be an admissible decision. Then any decision $\mathbf{v}_0$ is also admissible if

$$\|\mathbf{v}_0\|_1 \le \overline{U} \quad \text{and} \quad \mathbf{s}(\mathbf{v}_0) \preccurlyeq \mathbf{s}(\mathbf{u}_0).$$

Proof: See the Appendix. ∎

In other words, Theorem 19 states that if a decision is admissible, then a decision smaller in slack, in the sense provided by the relation "$\preccurlyeq$", is also admissible.

The characterization given by Theorem 19 is exemplified in Figure 3. It represents the lattice of all possible choices of $\underline{U}$ tasks to be served among the set of 7 tasks already described in Figure 2. The label for each node represents the slacks associated with the tasks that are served. The Hasse diagram [21] highlights the POset structure defined by the relation "$\preccurlyeq$". In red (and on the right of the same figure) we have reported all the admissible decisions, illustrating the relation of partial ordering described by Theorem 19.

V. EXAMPLE: AN AUCTION WITH MINIMUM-EFFORT LEAST-SLACK CONSTRAINTS

The analysis in Section IV poses a lower bound on the aggregate service rate $\|\mathbf{u}_0\|_1$ of the admissible decisions, and shows that the set $\mathcal{U}_0$ has a poset structure. In the scenario that motivated this study, at every time the Aggregator has to

Battery	$B^{(1)}$	$B^{(2)}$	$B^{(3)}$	$B^{(4)}$	$B^{(5)}$	$B^{(6)}$	$B^{(7)}$
Demand $E^{(i)}$	3	2	4	3	1	5	1
Deadline $T^{(i)}$	3	3	5	5	5	8	8
Slack $s^{(i)}$	0	1	1	2	4	3	7

$$\overline{U} = 3$$

k	0	1	2	3	4	5	6	7
	$N^A=3$	$N^A=5$	$N^A=5$					
$B^{(1)}$	$E=1$ $r=0$	$E=2$ $r=0$	$E=3$ $r=0$					
$B^{(2)}$	$E=0$ $r=1$	$E=1$ $r=1$	$E=2$ $r=1$	$N^A=4$	$N^A=4$			
$B^{(3)}$	$E=0$ $r=1$	$E=1$ $r=1$	$E=2$ $r=1$	$E=3$ $r=1$	$E=4$ $r=1$			
$B^{(4)}$	$E=1$ $r=0$	$E=1$ $r=1$	$E=1$ $r=2$	$E=2$ $r=2$	$E=3$ $r=2$			
$B^{(5)}$			$E=0$ $r=3$	$E=1$ $r=3$	$E=1$ $r=4$	$N^A=1$	$N^A=1$	$N^A=2$
$B^{(6)}$	$E=1$ $r=0$	$E=1$ $r=1$	$E=1$ $r=2$	$E=1$ $r=3$	$E=2$ $r=3$	$E=3$ $r=3$	$E=4$ $r=3$	$E=5$ $r=3$
$B^{(7)}$							$E=0$ $r=7$	$E=1$ $r=7$
$\sum_{i=1}^{N} u_k^{(i)}$	3	3	3	3	3	1	1	2

Fig. 2. A set of 7 integer tasks with heterogeneous deadlines and the schedule computed by the Minimum Effort Algorithm. The aggregate bound is $\overline{U} = 3$. A gray box indicates that at a certain time k the task i is served ($u_k^{(i)} = 1$) while a white box indicates that the task is idle ($u_k^{(i)} = 0$). Each box also contains the numerical values of $E^{(i)}$ and of the reverse slack $r^{(i)}$ as computed by the Minimum Effort Algorithm during its execution. As all the demands are served ($E^{(i)} = 0$ for all tasks after the iteration at time $k = 0$), the problem is schedulable. The aggregate rate at time $k = 0$ represents the minimum effort ($\underline{U} = 3$).

Fig. 3. The Hasse diagram on the left represents the poset structure of all possible way of choosing $\underline{U}$ batteries that consume (at full rate) at time 0. The ordering relation describes the fact that if two decisions x, y satisfy $x \preccurlyeq y$ (and therefore a directed path goes from x to y in the diagram) then x is guaranteed to be an admissible decision if y is admissible. For this specific example, $[024]$ is the maximal element (the admissible decision with the largest slack), and all and only the decisions with smaller slack are admissible. The diagram on the right represents the subset of all admissible decisions. Clearly, as an admissible decision exists, the decision with the least slack is always admissible.

guarantee that the action $\mathbf{u}_0$ of the users belongs to $\mathcal{U}_0$. We propose here an example of one possible and very practical way to enforce such constraint, which we denote *Minimum-effort Least-slack* (MELS) policy, and which is based on the following result.

Corollary 20: Let $\underline{U}$ be the minimal aggregate consumption as returned by the Minimum Effort Algorithm. Let $i_1, \ldots, i_{\underline{U}}$ be the $\underline{U}$ tasks with the smallest slacks (resolving ties arbitrarily). Then any decision $\mathbf{u}_0$ in

$$\mathcal{U}_0^{\text{MELS}} = \left\{ \mathbf{u}_0 : \|\mathbf{u}_0\|_1 \leq \overline{U} \text{ and } u_0^{(i)} = 1 \text{ for } i = i_1, \ldots, i_{\underline{U}} \right\}$$

is admissible, being $\mathcal{U}_0^{\text{MELS}} \subseteq \mathcal{U}_0$.

Proof: See the Appendix. ∎

The proposed MELS policy consists in a regulated energy market described by the following steps.

1) In order to participate to the market, users declare their deadline and demand.
2) User are allowed to join only if their demand is feasible. Such test can be performed via the Minimum Effort Algorithm, and once a load is admitted, the aggregator guarantees that its demand will be served (in a form of commitment similar to the one in [22]).
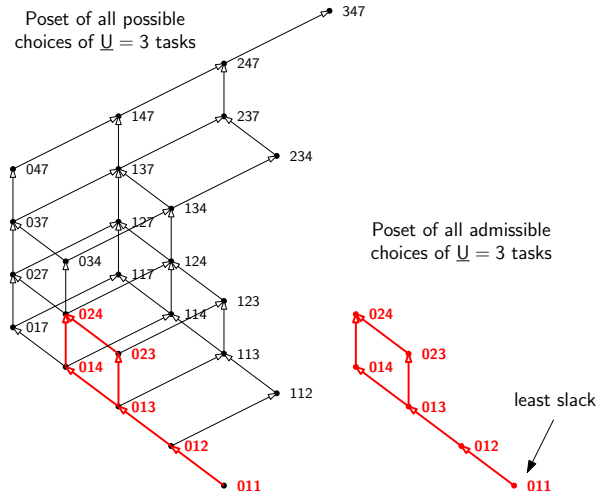3) At every time step, all users declare a *bid*, i.e. the price that they are willing to pay in order to consume at full rate, for the next time interval, according to their own selfish optimization policy (like for example the one derived in [9]).
4) Based on these date, an aggregator computes the lower bound $\underline{U}$ via the Minimum Effort Algorithm.
5) The aggregator clears a constrained multi-object auction with the following constraints:
   - the $\underline{U}$ loads with the least slack must be served;
   - no more than $\overline{U}$ loads are served, in total.

This example shows how a precise characterization of the set of admissible decisions allows the users to participate in the energy market with their specific, selfish, and possibly undisclosed, policy (i.e. via their bids), while ensuring that operational constraints and deadlines are satisfied.

VI. CONCLUSIONS

In an energy market with dynamic pricing, consumers can optimize their individual utilities adopting several mechanisms among which load-shifting is one of most effective ones. However, the response of a large number of uncoordinated price-responsive consumers might lead to local or global congestion of the power distribution system.

The goal of this paper is to suggest a possible methodology to enable a large number of price-responsive consumers to participate to the market without compromising the reliability of the network. This methodology is based on the computation of the set of admissible actions that are guaranteed to preserve the integrity of the system. It is shown that this set admits a precise mathematical characterization defined by a relation of partial ordering. This set of admissible actions can be used along with a constrained multi-object auction mechanism to establish a verified decision protocol providing

both the efficiency of a distributed dynamic pricing system and the reliability of a centralized approach.

REFERENCES

[1] A. Papavasiliou and S. Oren, "Supplying renewable energy to deferrable loads: Algorithms and economic analysis," in *Power and Energy Society General Meeting, 2010 IEEE*. IEEE, 2010, pp. 1–8.

[2] K. Afridi, A. M. Farid, J. M. Grochow, W. W. Hogan, H. D. Jacoby, J. L. Kirtley, H. G. Michaels, I. Perez-Arriaga, D. J. Perreault, N. L. Rose, and G. L. Wilson, *The Future of the Electric Grid*. MIT Press, 2011.

[3] D. Caves, K. Eakin, and A. Faruqui, "Mitigating price spikes in wholesale markets through market-based pricing in retail markets," *The Electricity Journal*, vol. 13, no. 3, pp. 13–23, 2000.

[4] D. Kirschen, "Demand-side view of electricity markets," *Power Systems, IEEE Transactions on*, vol. 18, no. 2, pp. 520–527, 2003.

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

[6] M. Caramanis and J. M. Foster, "Management of electric vehicle charging to mitigate renewable generation intermittency and distribution network congestion," in *Proceedings of the 48th IEEE Conference on Decision and Control*, Dec. 2009, pp. 4717–4722.

[7] K. Clement-Nyns, E. Haesen, and J. Driesen, "The impact of charging plug-in hybrid electric vehicles on a residential distribution grid," *IEEE Trans. on Power Systems*, vol. 25, no. 1, pp. 371–380, 2010.

[8] J. A. P. Lopes, F. J. Soares, and P. M. R. Almeida, "Integration of electric vehicles in the electric power system," *Proceedings of the IEEE*, vol. 99, no. 1, pp. 168–183, 2011.

[9] D. Materassi, S. Bolognani, M. Roozbehani, and M. Dahleh, "Deferrable loads in an energy market: optimal consumption policy," in *Proc. of the American Control Conference*, 2014.

[10] O. Sundström and C. Binding, "Flexible charging optimization for electric vehicles considering distribution grid constraints," *IEEE Transactions on Smart Grid*, vol. 3, no. 1, pp. 26–37, Mar. 2012.

[11] A. Subramanian, M. Garcia, A. Dominiguez-Garcia, D. Callaway, K. Poolla, and P. Varaiya, "Real-time scheduling of deferrable electric loads," in *Proc. of the American Control Conference*, 2012.

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

[13] 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. 2, no. 1, pp. 198–205, Mar. 2011.

[14] K. Clement, E. Haesen, and J. Driesen, "Coordinated charging of multiple plug-in hybrid electric vehicles in residential distribution grids," in *Proc. of the Power Systems Conference and Exposition*, 2009.

[15] M. Penn and M. Tennenholtz, "Constrained multi-object auctions and b-matching," *Inform. Process. Letters*, vol. 75, no. 1, pp. 29–34, 2000.

[16] M. Armstrong, "Optimal multi-object auctions," *The Review of Economic Studies*, vol. 67, no. 3, pp. 455–481, 2000.

[17] V. Krishna, *Auction theory*. Academic press, 2009.

[18] M. Pinedo, *Scheduling: theory, algorithms, and systems*. Springer, 2012.

[19] J. Chen, F. N. Lee, A. M. Breipohl, and R. Adapa, "Scheduling direct load control to minimize system opoperation cost," *IEEE Transactions on Power Systems*, vol. 10, no. 4, pp. 1994–2001, Nov. 1995.

[20] K.-H. Ng and G. B. Sheble, "Direct load control - a profit-based load management using linear programming," *IEEE Transactions on Power Systems*, vol. 13, no. 2, pp. 688–695, May 1998.

[21] R. Brèuggemann and G. P. Patil, *Ranking and Prioritization for Multi-indicator Systems: Introduction to Partial Order Applications*. Springer, 2011, vol. 5.

[22] S. Chen, L. Tong, and T. He, "Optimal deadline scheduling with commitment," in *Proc. of the 49h Annual Allerton Conf. on Communication Control and Computing*, 2011.

APPENDIX

PROOF OF THEOREM 10

If an integer schedule exists, then the task scheduling problem is trivially scheduling. The converse implication is a direct result of Theorem 17, which states that the proposed Minimum Effort Algorithmreturns an integer schedule, if the problem is schedulable.

PROOF OF THEOREM 15

The two conditions are obviously necessary. We will prove that the two conditions are sufficient by induction on the scheduling horizon T. When $T = 1$ the assertion is trivial. Now, assume that the two conditions guarantee schedulability for a set of tasks with common deadline $T-1$. Using Proposition 11, let us assume, without any loss of generality, that $\overline{u}^{(i)} = 1$, for $i = 1, ..., N$. Observe that the two conditions (7) are equivalent after the transformation of Proposition 11. Again with no loss of generality, assume that the tasks are already sorted in ascending order of demand $E^{(i)}$ and apply the consumption $u^{(i)} = 1$ for $i = 1, ..., \overline{U}$ and $u^{(i)} = 0$ for $i = \overline{U} + 1, ..., N$. After this consumption, we are left with the set of tasks $(E^{(i)} - u^{(i)}, T - 1, 1)$. We have that

$$\sum_{i=1}^{N} E^{(i)} - u^{(i)} \leq \overline{U}T - \sum_{i=1}^{N} u^{(i)} = \overline{U}(T - 1).$$

By contradiction suppose that

$$\max_{1 \leq i \leq N} \left(E^{(i)} - u^{(i)} \right) > T - 1.$$

This would necessary happen for a task $i > \overline{U}$, as for all tasks $i \leq \overline{U}$ we know that $u^{(i)} = 1$ and that $E^{(i)} \leq T$. Therefore it must be that for some $i > \overline{U}$ we have $E^{(i)} - 1 > T - 1$. Because the demands $E^{(i)}$ were ordered, this means that $\sum_{i=1}^{N} E^{(i)} > \overline{U}T$ leading to a contradiction. Thus, we must have

$$\max_{1 \leq i \leq N} \left(E^{(i)} - u^{(i)} \right) \leq T - 1.$$

The two conditions guaranteeing schedulability are met for the time horizon $T - 1$, implying that the problem is schedulable. ∎

PROOF OF THEOREM 16

All decisions in $\mathcal{U}_0$ are feasible beacuse for all of them $\|\mathbf{u}_0\|_1 \leq \overline{U}$. We have then to check that the problem at the next time step is schedulable. We use the characterization of Lemma 15. By hypothesis of schedulability, we have that $\sum_{i=1}^{N} E^{(i)} \leq T\overline{U}$ and that, for all tasks i, $E^{(i)} \leq T\overline{u}$. Therefore at time 1, as $\|\mathbf{u}_0\|_1$ is always larger than $\underline{U}$ as defined in (8), we have that $\sum_{i=1}^{N} E^{(i)} - \|\mathbf{u}_0\|_1 \leq (T-1)\overline{U}$. Moreover, as all the tasks with zero slack are served, we have that all slack are nonnegative at the next time step, and therefore, for all i, $E^{(i)} - u_0^{(i)} \leq (T - 1)\overline{u}$. ∎

PROOF OF THEOREM 17

We prove this by induction on the value of the largest deadline T. For $T = 1$ the assertion is trivial. Now, assume that the statement holds for T. Given a set of tasks with largest deadline $T + 1$, consider a generic feasible schedule $\{\{v_k^{(i)}\}_{k=0}^{T}\}_{i=1}^{N}$. Consider the pattern of consumption at the interval T, namely $\{v_T^{(i)}\}_{i=1}^{N}$. By contradiction assume that

the Minimum Effort Algorithm can not provide a feasible schedule and consider its pattern of consumption at at the interval T, namely $\{u_T^{(i)}\}_{i=1}^N$.

If $v_T^{(i)} \leq u_T^{(i)}$ for all $i = 1, ..., N$, the tasks $(E^{(i)} - u_T^{(i)}, \min\{T^{(i)}, T\})$ are obviously schedulable and applying the induction assumption we have that the Minimum Effort Algorithm provides an integer schedule. This leads to a contradiction.

Now assume that there exists some j for which $v_T^{(j)} > u_T^{(j)}$ leading necessarily to $u_T^{(j)} = 0$.

If $N = 1$, we would have that $j = 1$ and $T^{(1)} = T + 1$. Thus, the Minimum Effort Algorithm would have assigned $u_T^{(j)} = 1$ and this would be again a contradiction. For $N > 1$, $v_T^{(j)} > u_T^{(j)}$ implies that there is some $i \neq j$ for which $v_T^{(i)} < u_T^{(i)}$. If this were not the case we would have that

$$\sum_{l=1}^N v_T^{(l)} > \sum_{l=1}^N u_T^{(l)}$$

leading again to a contradiction: it is clear from the description of the Minimum Effort Algorithm that the global consumption at the last interval is maximized in the returned schedule.

So far we have proved that, if the Minimum Effort Algorithm can not provide feasibility for a set of tasks, then for every feasible scheduling (be it integer or not) $\{\{v_k^{(i)}\}_{k=0}^T\}_{i=1}^N$ there exist i and j such that $v_T^{(j)} > u_T^{(j)} = 0$ and $v_T^{(i)} < u_T^{(i)} = 1$. Thus, we can write that

$$\left[v_T^{(1)} \ ... \ v_T^{(i)} \ ... \ v_T^{(j)} \ ... \ v_T^{(N)} \right] = \left[1-\alpha^{(1)} \ ... \ 1-\alpha^{(i)} \ ... \ \alpha^{(j)} \ ... \ \alpha^{(N)} \right],$$

$$\left[u_T^{(1)} \ ... \ u_T^{(i)} \ ... \ u_T^{(j)} \ ... \ u_T^{(N)} \right] = \left[1 \ ... \ 1 \ ... \ 0 \ ... \ 0 \right],$$

where the vector $(\alpha^{(1)}, ..., \alpha^{(N)})$ has 1-norm that is different from zero.

The space of all feasible scheduling is a compact set. So, let us choose $\{\{v_k^{(i)}\}_{k=0}^T\}_{i=1}^N$ as the feasible scheduling minimizing the 1-norm of $(\alpha^{(1)}, ..., \alpha^{(N)})$. The minimal 1-norm of $(\alpha^{(1)}, ..., \alpha^{(N)})$ is larger than zero because of the compactness of the space of all feasible scheduling.

Since the Minimum Effort Algorithm allocates consumption giving priority to tasks with lower slack, we have that $s^{(i)} \leq s^{(j)}$ and, since $\overline{u}^{(i)} = \overline{u}^{(j)} = 1$, we also have that $E^{(i)} \geq E^{(j)}$.

Let $v_T^{(i)} = 1 - \alpha^{(i)}$, and $v_T^{(j)} = \alpha^{(j)}$ with $\min\{\alpha^{(i)}, \alpha^{(j)}\} > 0$. As a consequence, we have that the following relation holds

$$\left| \left\{ k < T \mid v_k^{(i)} > 0 \right\} \right| > E^{(i)} \geq E^{(j)} \geq \left| \left\{ k < T \mid v_k^{(j)} = 1 \right\} \right|.$$

Thus, there exists at least one interval $k < T$, where $v_k^{(i)} > 0$ and $v_k^{(j)} < 1$. Define $\alpha := \min\{\alpha^{(i)}, \alpha^{(j)}, v_k^{(i)}, 1 - v_k^{(j)}\}$ and define the following feasible consumption

$$\hat{v}_T^{(i)} = v_T^{(i)} + \alpha \qquad \hat{v}_T^{(j)} = v_T^{(j)} - \alpha$$
$$\hat{v}_k^{(i)} = v_k^{(i)} - \alpha \qquad \hat{v}_k^{(j)} = v_k^{(j)} + \alpha.$$

In particular, we have that

$$\left[\hat{v}_T^{(1)} \ ... \ \hat{v}_T^{(i)} \ ... \ \hat{v}_T^{(j)} \ ... \ \hat{v}_T^{(N)} \right]' =$$
$$\left[1-\alpha^{(1)} \ ... \ 1-\alpha^{(i)}+\alpha \ ... \ \alpha^{(j)}-\alpha \ ... \ \alpha^{(N)} \right]'$$

where the vector $(\alpha^{(1)}, ..., \alpha^{(i)} + \alpha, ..., \alpha^{(j)} - \alpha, ..., \alpha^{(N)})^T$ has 1-norm smaller than $(\alpha^{(1)}, ..., \alpha^{(i)}, ..., \alpha^{(j)}, ..., \alpha^{(N)})^T$ that was supposed to be minimal. Thus we have a contradiction.

PROOF OF LEMMA 18

Observe that the Minimum Effort Algorithm starts computing a feasible schedule iterating from the time $t = T - 1$, where $T = \max_{1 \leq i \leq N} T^{(i)}$, till $k = 0$. Thus, if the i-th task has deadline $T^{(i)} = 1$, it influences the iterations of the algorithm only when $t = 0$. Assume, by contradiction, that there is a feasible schedule such that $\|\mathbf{u}_0\|_1 \leq \underline{U} - 1$. Then, by adding to the original set of tasks also the tasks $(E^{(i)}, T^{(i)}, 1) = (1, 1, 1)$ for $i = N+1, ...N + \overline{U} - \underline{U} + 1$, we still have feasibility. However, by running the algorithm on this modified set of task, we execute the same operations till $t = 1$ and find that at $t = 0$ the problem is not feasible. This contradiction proves the assertion.

PROOF OF THEOREM 19

Assume that there is a feasible integer scheduling where $u_0^{(i_1)} = \ldots = u_0^{(i_M)} = 1$, Let j be a task such that $u_0^{(j)} = 0$ and $s^{(j)} \leq s^{(i_m)}$ for some $1 \leq m \leq M$. We will first prove that the decision where

$$v_0^{(i_1)} = \ldots = v_0^{(i_{m-1})} = v_0^{(i_{m+1})} = \ldots = v_0^{(i_M)} = 1$$
$$v_0^{(i_m)} = 0$$
$$v_0^{(j)} = 1$$
$$v_0^{(i)} = 0 \qquad \text{for } i \neq j, i_1, ..., i_M.$$

is admissible.

Without any loss of generality assume $m = M$. Let us consider first the case where $T^{(i_M)} \leq T^{(j)}$. If $T^{(i_M)} \leq T^{(j)}$, and since $s_0^{(j)} \leq s_0^{(i_M)}$, there is a time k such that $0 < k \leq T^{(i_M)}$ and $u_k^{(i_M)} = 0$, $u_k^{(j)} = 1$. Thus, it is possible to make the j-th task to be served in place of the i_M-th one at time 0 and vice-versa at time k without violating the constraints of the problem, as depicted in the following figure.
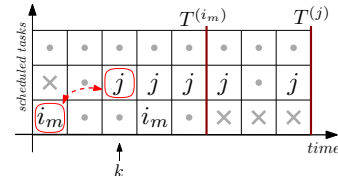
Now, let $T^{(i_M)} > T^{(j)}$. If there is a time $0 < k \leq T^{(i_M)}$ where $u_k^{(i_M)} = 0$ and $u_k^{(j)} = 1$, we can operate the same switch as before. Otherwise, since $u_0^{(j)} = 0$, there must be a time $0 < k' \leq T^{(j)}$ where both the j-th and the i_M-th batteries consume, namely $u_{k'}^{(i_M)} = 1$, $u_{k'}^{(j)} = 1$. Since $T^{(i_M)} > T^{(j)}$, there also must be a time $T^{(j)} < k'' \leq T^{(i_M)}$ where neither the j-th nor the i_M-th batteries consume. Now move the consumption of the i_M-th battery from k'

to k''. This is always possible: if at time k'' we have that $\sum_{i=1}^{N} u_{k''} < \overline{U}$ no constraint is violated, otherwise it is possible to find a battery $j' \neq j, i_M$ such that $u_{k''}^{(j')} = 1$, $u_{k'}^{(j')} = 0$ and anticipate its consumption at time k'. After this switch we have that at time k' the i_M-th battery does not consume while j-th does. Switching consumption beetween the two batteries at time 0 and k' is now possible without violating any constraint and the assertion is proved. This scenario is depicted in the following figure.
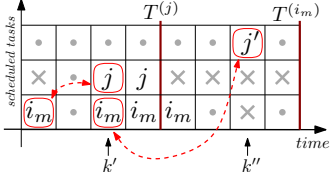
Iterated application of the same reasoning allow to prove that any decision $\mathbf{v}_0$ such that $\|\mathbf{v}_0\|_1 = \|\mathbf{u}_0\|_1$ and such that $\mathbf{s}(\mathbf{v}_0) \preccurlyeq \mathbf{s}(\mathbf{u}_0)$, is also admissibile (as it is the first decision in a feasible schedule). Finally, we conclude that this is also true in the case in which $\|\mathbf{v}_0\|_1 > \|\mathbf{u}_0\|_1$ (and therefore in all cases in which $\mathbf{s}(\mathbf{v}_0) \preccurlyeq \mathbf{s}(\mathbf{u}_0)$), because that simply means that other tasks have been served at time 0, together with the ones served by the feasible schedule, necessarily yielding another feasible schedule. $\blacksquare$

PROOF OF COROLLARY 20

Via Theorem 17 we know that a feasible decision such that $\sum_{i=1}^{(N)} u_0^{(i)} = \underline{U}$ exists. Because of the poset structure, we know that the decision $\mathbf{u}_0$ with

$$u_0^{(i)} = \begin{cases} 1 & \text{for } i = i_1, \dots, i_{\underline{U}} \\ 0 & \text{otherwise} \end{cases}$$

is admissible, as the tasks with the least slack are served. The result follows from the fact that any $\mathbf{v}_0$ in $\mathcal{U}_0^{\text{MELS}}$ satisfies $\mathbf{v}_0 \preccurlyeq \mathbf{u}_0$. $\blacksquare$