# Complexity and algorithms for min cost and max profit scheduling under time-of-use electricity tariffs

Michal Penn[1] and Tal Raviv[2]

[1]Technion, Haifa, Israel

[2]Tel Aviv University, Tel Aviv, Israel

October 2020

**Abstract**

Following a recent interest in sustainable scheduling under operational costs that vary over time, we study scheduling problems on a single machine under time-of-use (TOU) electricity tariffs. We consider two main variants of the problem: cost minimization and profit maximization. In the cost minimization problem, the set of jobs to be processed is given, and the goal is to schedule all jobs within a planning horizon so as to minimize the total cost, while in the profit maximization problem, one needs to select a set of jobs to be processed such that the total profit is maximized. The general cases of the cost minimization and profit maximization problems in which preemptions are forbidden are strongly NP-hard. In this paper, we show that some special cases with identical processing times can be solved by efficient algorithms**.** In addition, we consider several extensions of the problems, including release times, due dates, and variable energy consumption.

Keywords: Scheduling, Time-of-Use Tariffs, Single Machine, Maximum Profit, Minimum Cost

## 1. Introduction and literature review

Recently, many manufacturing and energy companies have shown an increased interest in sustainable scheduling. This interest has led to the implementation of variable pricing to manage the balance between supply and demand for electricity and to improve the reliability and efficiency of electrical power grids. For example, with time-of-use (TOU) tariffs, retail energy prices to customers vary hourly to reflect changes in wholesale energy prices. Such price structures are used to shift energy-intensive production jobs from peak hours to off-peak hours and to significantly reduce costs. Gahm et al. (2016) present a comprehensive survey on "energy-efficient scheduling" (EES); EES approaches are scheduling approaches that aim to improve energy efficiency. They further develop a research framework for EES scheduling and provide an empirical analysis of the reviewed literature and emphasize the benefits that can be achieved by EES in practice.

Some recent studies have taken a more theoretical approach, considering various scheduling environments. Among these is the work of Wan and Qi (2010), who study scheduling under operational costs that vary over time. They consider scheduling models on a single machine that minimize a linear combination of the total time slot costs and a traditional scheduling performance measure: total completion time, maximum lateness/tardiness, total weighted number of tardy jobs or total tardiness. They prove the intractability of the models under general parameters and provide polynomial-time algorithms for special cases with nonincreasing time slot costs. Zhong and Liu (2012) consider a single-machine scheduling problem in which processing of a job incurs a cost that depends on the time slots occupied by the job. Their objective is to minimize a linear combination of the makespan and the total time slot costs. They prove that the problem is strongly NP-hard and analyze a special case with nonincreasing time slot costs. Zhao et al. (2016) address the scheduling problem that aims to minimize the sum of the total weighted completion time and the total machine time slot cost. Focusing on the case of nonincreasing time slot cost with nonpreemptive jobs, they show that the problem can be solved in polynomial time when the time slot cost decreases with certain patterns and is NP-hard in the general decreasing-patterns case. Shrouf et al. (2014) propose a mathematical model to minimize energy consumption costs for single-machine production scheduling during production processes with "turning on" and "turning off" costs as well. The authors present a genetic algorithm heuristic for this problem.

Che et al (2016) investigate a single-machine scheduling problem under TOU tariffs to minimize the total electricity cost. A continuous-time MILP model is developed, and an efficient greedy insertion heuristic is proposed. A real-life case study from a Chinese company reveals that the total electricity cost can be reduced by about 30% using their algorithm.

Che et al (2017) address a single-machine scheduling problem with a power-down mechanism to minimize total energy consumption and maximize tardiness simultaneously. A MILP model based on position assignment has been developed to formulate the problem. To obtain the exact Pareto front of the problem, they propose a basic $\varepsilon-$ constraint method and develop a local search.

Fang et al. (2016) and Fang (2013) consider the problem of scheduling jobs on a single machine to minimize the total electricity cost of processing these jobs under TOU electricity tariffs. They study two variations of the problem: the uniform-speed case, in which all jobs have given processing times, and the speed-scalable case, in which the planner can control the processing speed and thus the processing time of each job. In the uniform-speed case, the energy consumption of each job is given and is not necessarily proportional to the processing time. Using the 3-partition problem, they prove that the nonpreemptive version of this problem is strongly NP-hard and inapproximable within a constant factor unless P=NP. On the other hand, for a special case of the problem, i.e., identical processing time, different power demands and the so-called pyramidal TOU electricity tariff function, the authors provide a polynomial-time algorithm.

Rubaiee et al. (2018) study a nonpreemptive scheduling problem on a single machine to minimize the total tardiness and total energy cost under TOU electricity tariffs. They formulate the problem as a mixed-integer multiobjective mathematical programming model and develop four multiobjective genetic algorithms to obtain a near-optimal Pareto front in a timely fashion.

The authors provide analysis and detailed experimental results evaluating the performance of the algorithms.

Aghelinejad et al (2019) address a nonpreemptive scheduling problem under TOU tariffs in which $n$ jobs are to be processed in a predefined order on a single machine. Each job has its own processing time and energy consumption. The machine may switch among three different states. The aim is to minimize the total energy consumption costs. The authors suggest a dynamic programming approach to model the problem. They use the Dijkstra algorithm to calculate a shortest path on a finite graph. Using this approach, the authors show that the uniform speed case of this problem is solvable in polynomial time. In addition, if the order of the jobs is not predefined, the authors provide a polynomial-time algorithm for constant, increasing, or decreasing energy costs. They use the 3-partition problem to show that the case with non-predefined job order and TOU tariffs is NP-hard.

Chen and Zhang (2019) consider the problem of scheduling jobs on a single machine to minimize the total electricity cost of processing these jobs under TOU electricity tariffs. They refer to this problem as scheduling with TOU costs (STOUC) and show that the STOUC problem is strongly NP-hard. They further study the STOUC problem with very restricted TOU costs, namely, the case in which the cost vector has two "valleys" (they call a period P a valley if its TOU cost is smaller than the TOU cost(s) of its neighboring period(s)) and show that this problem is NP-hard at least in the ordinary sense and is inapproximable within any constant factor. Under the very restricted condition of at most one valley, the authors show that the STOUC problem with either bounded lateness, bounded tardiness, or bounded flow-time is solvable in polynomial time.

Fang et al. (2016) and Chen and Zhang (2019) study scheduling on a single machine with uniform-speed processing time under TOU tariffs; these studies are probably the most relevant to our study. Since the energy cost minimization problem is NP-hard (Fang et al. (2016)), in the above studies, the authors develop polynomial (resp., pseudopolynomial) algorithms under very restricted cost (resp., TOU tariff) functions. We took a different approach by allowing any cost (resp., TOU tariff) function but considering uniform-speed and identical-processing-time jobs.

In this paper, we consider two variants of the single-machine scheduling problem with TOU tariffs: cost minimization and profit maximization. To the best of our knowledge, we are the first to study the more challenging profit maximization problem. More specifically, in the cost minimization problem, the set of jobs to be processed is given, and the goal is to schedule all jobs within a planning horizon in a way that minimizes the total cost. In the profit maximization problem, one needs to simultaneously select a subset of jobs to be processed and schedule them so that the total profit is maximized. We develop efficient algorithms for some variants of the problems with identical processing times as well as for preemptive versions of the problems. In particular, for the nonpreemptive case with identical processing times, we develop a strongly polynomial 3-step algorithm for both the cost-minimization and profit-maximization objectives. The first step of these algorithms is based on a greedy algorithm for the preemptive case, the second step uses dynamic programming that is applied to an instance of the problem with smaller dimension, and the third step merges the two solutions from the previous steps into a solution to the original problem. For the cost minimization problem, we further develop a polynomial-time algorithm for the preemptive case with release times and due dates. Our algorithm is based on a reduction to the Hitchcock transportation problem. We also present a

dynamic programming polynomial-time algorithm for the nonpreemptive identical-processing-time case with general release times or due dates. In addition, we present elegant and compact integer programming formulations that capture many variants of the nonpreemptive unit time problems, including time-varying electricity tariffs, release times, and due dates for both the minimum cost and the maximum profit objectives.

The rest of the paper is organized as follows: in Section 2, we introduce notations, define the problem variants formally and make some observations regarding the structure of an optimal solution. In Section 3, we present polynomial-time algorithms for the unit-time nonpreemptive cost-minimization problem. In Section 4, the profit-maximization problem is analyzed. The negative complexity result for the general case is presented as well as the polynomial-time algorithms for the preemptive and unit-time cases. In Section 5, we present a generalization of the cost minimization problems where release time and due dates are considered. In Section 6, we present integer programming formulations for an extension of problems in which the power consumption as well as other characteristics of each job may be different. A summary of the results obtained in this paper and some final thoughts and directions for future study are presented in Section 7.

# 2. Problem definitions and preliminaries

## 2.1 The general problem setting

Let $J$ be a set of $n$ jobs to be scheduled on a single machine during a given time horizon $[0, T]$. For each $j \in J$, job $j$ has processing time $p_j$ and revenue $\xi_j$. We consider the uniform-speed processing time where the production process consumes electrical energy at a constant rate and the electricity cost varies over time.

The time horizon $[0, T]$ is divided into $K$ electricity tariff intervals (ETIs), with $ETI_1 = [t_0, t_1]$ starting at time $t_0 = 0$ and $ETI_K = (t_{K-1}, t_K]$ ending at time $t_K = T$. The length of $ETI_k$ is $l_k = t_k - t_{k-1}$ with $\sum l_k = T$. The cost of the energy consumed by the machine in one unit of time during $ETI_k$ is denoted by $q_k$. Thus, the total energy cost of processing job $j$ in $ETI_k$ is $q_k p_j$. If the processing extends over several ETIs, the energy cost is calculated proportionally to the processing time during each ETI. We consider below two objective functions: minimum cost and maximum profit. We note that throughout the paper, we assume that the processing times of the jobs are positive integers and that all the other parameters are positive rational numbers. In addition, we assume that preemptions have no effect on the total processing times and processing costs of the jobs.

For ease of presentation, we use the three-field notation, $\alpha|\beta|\gamma$, introduced by Grahamet al. (1979) for classifying scheduling problems. The $\alpha$ field represents the machine environment, the $\beta$ field represents the processing characteristics and constraints, and the $\gamma$ field represents the objective function. For example, in $1|energy|cost$, $\alpha = 1$ stands for a single machine, $\beta = energy$ indicates that we consider scheduling under TOU electricity (energy) tariffs and $\gamma = cost$ stands for the minimum cost goal.

## 2.2 The cost minimization problem

In the above setting, the goal is to find a schedule for all $n$ jobs in the planning horizon $[0, T]$ such that the energy cost is minimized. The jobs' revenues are irrelevant and thus ignored. We call the cost minimization problem *identical* if all processing times are identical. In such cases, we assume that $p_j = 1$. Since we allow fractional values for the lengths of the ETIs, the unit time assumption is equivalent to the assumption that $p_j$ is constant ($p_j = p$). We consider four variants of the cost minimization problem. For general processing times, we distinguish between $1|energy|cost$, where preemptions of jobs are forbidden, and $1|energy, pmtn|cost$, where preemptions are allowed. In the identical case, we similarly differentiate between disallowing (resp., allowing) preemptions, i.e., $1|energy, p_j = 1|cost$ (resp., $1|energy, p_j = 1, pmtn|cost$).

Fang et al (2016) show that a more general problem than the $1|energy|cost$ problem is strongly NP-hard and not in APX using a reduction from the 3-partition problem. Chen and Zhang (2019) used similar arguments to prove the NP-hardness and inapproximability of the $1|energy|cost$ problem. In addition, Fang et al (2016) present a polynomial-time greedy algorithm that can be directly applied to $1|energy, pmtn|cost$.

## 2.3 The profit maximization problem

Given $n$ potential jobs with their processing times and revenues, the goal is to decide simultaneously upon the set of jobs to be produced and their schedule in order to maximize the profit during the planning horizon $[0, T]$. For a given schedule, the profit is defined as the total job revenues minus the total energy cost. We note that the additional degree of freedom resulting from the need to choose a subset of jobs to be processed leads to a more intricate problem, as indicated by our complexity analysis.

We consider several variants of the profit maximization problem. In particular, we distinguish between identical processing times (identical revenues), i.e., $p_j = 1$ ($\xi_j = 1$) for all jobs, and general processing times (revenues). Note that no generality is lost by assuming unit processing times (unit revenues) in the identical cases since the lengths of the ETIs (electricity tariffs) are rational numbers.

We call a profit maximization problem *identical* if all processing times and all revenues are equal, *p-semi-identical* if all processing times are identical ($p_j = 1$), and *ξ-semi-identical* if all revenues are equal ($\xi_j = 1$). The problem is *general* if it is neither *p-semi-identical* nor *ξ-semi-identical*.

Each profit maximization problem is further classified into the nonpreemptive case $1/energy/profit$ and the preemptive case $1|energy, pmtn|profit$. That is, we have eight variants of the profit maximization problem since we have three binary properties that distinguish among them.

## 2.4 The verge property

We present the notions of a busy period and the verge property below to describe a solution's characteristics.

**Definition 1**: *Busy period*. For a given solution of a single-machine scheduling problem a *busy period* is a maximal time interval in which jobs are processed continuously on the machine. We assume that switching between busy and idle periods does not incur any extra time or extra cost.

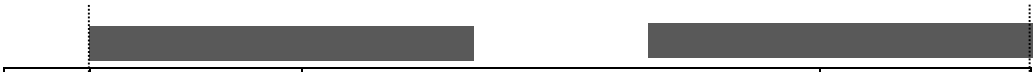
**Definition 2**: *The verge property*. We say that a feasible solution of the cost minimization or profit maximization problem satisfies the *verge property* if each busy period starts at the beginning of an ETI or ends at the end of an ETI. We note that the verge property plays an important role in the development of our algorithms.

**Proposition 1:** There exist optimal solutions for the cost minimization and for the profit maximization problems having the verge property.

This proposition, using different terminology, was proved independently by Chen and Zhang (2019) for the cost minimization version of the problem. The proof for the maximum profit version is very similar and thus the it is omitted.

The examples below demonstrate three instances of the $1|energy, p_j = 1, \xi_j = 1|profit$ problem in which the solutions satisfy the verge property. Consider a problem with four identical jobs three ETIs, $l_1 = 1.1, l_2 = 2.7, l_3 = 1.1$, $q_1 = q_3 = 0.9$ and $q_2$ as given below. Clearly, since $T = \sum l_k = 4.9$, four jobs can fit in the planning horizon but in optimal solution not necessarily all jobs are scheduled.

**Example 1**: If $q_2 = 1.01$, the optimal solution consists of two busy periods, each involving two jobs. The first starts at the beginning of $ETI_1$ and the second ends at the end of $ETI_3$, as illustrated in Figure 1.

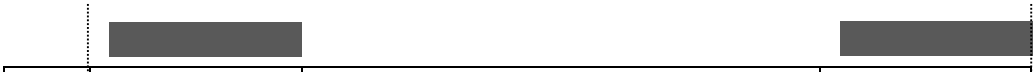


| | | | |
|---|---|---|---|
| $l_k$ | 1.1 | 2.7 | 1.1 |
| $q_k$ | 0.9 | 1.01 | 0.9 |
| $k$ | 1 | 2 | 3 |

**Figure 1**

Note that we would obtain the same solution for any $1.0111 \ldots > q_2 > 0.9$.

**Example 2**: For $q_2 > 1.0111\ldots$, any optimal solution consists of two busy periods, each involving one job. Due to the increase in the energy cost in $ETI_2$ it is not worth schedule more than two out of the four jobs. The first starts (resp., ends) at the beginning (resp., end) of $ETI_1$, and the second starts (resp. ends) at the beginning (resp., end) of $ETI_3$, as illustrated in Figure 2. There are infinitely many other optimal solutions that do not satisfy the verge property.



| | | | |
|---|---|---|---|
| $l_k$ | 1.1 | 2.7 | 1.1 |
| $q_k$ | 0.9 | 1.02 | 0.9 |
| $k$ | 1 | 2 | 3 |

**Figure 2**

**Example 3**: For the case of $q_2 < 0.9$, there are four optimal solutions that satisfy the verge property (and many others that do not). See Figure 3. At such low energy cost at $ETI_2$ it worth to utilize it completely and produce all the four jobs.

| | | | |
|---|---|---|---|
| $l_k$ | 1.1 | 2.7 | 1.1 |
| $q_k$ | 0.9 | 0.8 | 0.9 |
| $k$ | 1 | 2 | 3 |

**Figure 3**

Note that if $\boldsymbol{q_{k_1} < q_{k_2}}$ (resp., $\boldsymbol{q_{k_1} > q_{k_2}}$), a busy period that extends from $\boldsymbol{ETI_{k_1}}$ to $\boldsymbol{ETI_{k_2}}$ starts at the beginning of $\boldsymbol{ETI_{k_1}}$ (resp., ends at the end of $\boldsymbol{ETI_{k_2}}$).

Corollary 1 below is an immediate consequence of Proposition 1. It allows a solution of the identical-jobs variants in polynomial space to be described in terms of the number of ETIs.

**Corollary 1:** There exist optimal solutions for the cost minimization and profit maximization problems in which the number of busy periods is not greater than O($\boldsymbol{K}$), the number of ETIs.

## 2.5 Description of the input and output

There are several well-defined ways to describe the parameters of the problem at hand. Since the complexity of an algorithm is calculated relative to the size of the input, we assume that the most compact possible representation is used for each variant of the problem. In particular, in cases where the jobs are identical in all their characteristics, the size of the input is $O(K)$, the number of ETIs. Otherwise, it is $O(K + n)$, the number of ETIs plus the number of jobs. With these representations, an optimization algorithm for the identical jobs cases is strongly polynomial only if its running time is polynomial in $K$ and does not depend on $n$.

Similarly, the complexity of an optimization algorithm can be affected by the representation of the output. In the *1|energy|cost* and *1|energy|profit* problems, a solution of a single-machine scheduling problem consists of a list of the scheduled tasks and their respective starting times. That is, the output size is $O(n)$. However, in identical jobs cases, the solution can be described by a list of busy periods, which is of length $O(K)$. Therefore, with this output representation, it may be possible to develop a strongly polynomial-time algorithm for some identical jobs cases.

In Table 1, we list the variants of the problem in this study along with their input and output representation schemes and the size of these representations using O-notation.

**Table 1:** Input and output size and description for each variant of the problem

| Three-field notation | Input | Output |
|---|---|---|
| $1\|energy\|cost$ | $O(n+K)$: $p_j, q_k, l_k$ | Starting time of each job, $O(n)$ |
| $1\|energy, pmtn\|cost$ | | Starting time and length of each busy period, $O(K)$ |
| $1\|energy, p_j = 1\|cost$ | $O(K)$: $q_k, l_k, n$ | |
| $1\|energy, pmtn, p_j = 1\|cost$ | | |
| $1\|energy\|profit$ | $O(n+K)$: $p_j, \xi_j, q_k, l_k$ | List of processed jobs and their starting times, $O(n)$ |
| $1\|energy, p_j = 1\|profit$ | $O(n+K)$: $p_j, q_k, l_k$ | |
| $1\|energy, \xi_j = \|profit$ | $O(n+K)$: $\xi_j, q_k, l_k$ | |
| $1\|energy, pmtn/profit$ | $O(n+K)$: $p_j, \xi_j, q_k, l_k$ | List of processed jobs; starting time and length of each busy period, $O(n+K)$ |
| $1\|energy, pmtn, p_j = 1\|profit$ | $O(n+K)$: $\xi_j, q_k, l_k$ | |
| $1\|energy, pmtn, \xi_j = 1\|profit$ | $O(n+K)$: $p_j, q_k, l_k$ | |
| $1\|energy, p_j = 1, \xi_j = 1\|profit$ | $O(K)$: $q_k, l_k, n$ | Starting time and length of each busy period (the number of jobs is implied), $O(K)$ |
| $1\|energy,\ pmtn, p_j = 1, \xi_j = 1\|profit$ | | |

# 3. The cost minimization problem.

In this section, we present a polynomial-time 3-step algorithm for the nonpreemptive identical jobs case $1|energy, pmtn|cost$. The first step of this algorithm is based on a greedy algorithm for the preemptive case, the second step uses a dynamic program, and the third step merges the two solutions from the previous steps into a list of busy periods.

Recall that a solution for the $1|energy, pmtn|cost$ problem can be represented by a list of busy periods and the actual assignment of jobs or fractions of jobs to busy periods can then be carried out in any order. The following proposition is an adaptation of a result by Fang et al (2016).

**Proposition 2**: There exists an $O(K \log K + n)$ algorithm for the $1|energy, pmtn|cost$ problem.

**Proof:** Recall that the solution of this variant is a list of busy periods. The problem is solved by a greedy procedure that first calculates the total required processing time and then allocates processing times to the cheapest ETIs. The procedure requires sorting the ETIs in nondecreasing order of cost and then allocating the processing times of all the jobs to the ETIs one by one. The processing time allocated to the most expensive utilized ETI may be shorter than the length of the ETI. The amount of time needed to calculate the total processing time of all jobs and to allocate them to the ETIs is $O(n)$. The amount of time required to sort the ETIs by energy tariffs is $O(K \log K)$. Thus, the overall time complexity of the procedure is $O(K\ log\ K + n)$. ■

Fang et al (2016) present a very similar polynomial-time algorithm for a more general case in which the jobs have variable levels of energy consumption. The complexity of their algorithm is $O(K \log K\ + n \log n)$ because they need to sort the jobs by energy demand. Note that if the jobs are identical, i.e., $1|energy, pmtn, p_j = 1|cost$, the procedure works in exactly the same way,

but there is no need to calculate the sum of the processing times and to determine the start time of each job. In this case, the time complexity is $O(K \log K)$.

Next, we consider the identical jobs problem without preemptions $1|energy, p_j = 1|cost$. Observe that a lower bound on the value of the optimal solution is provided by the value of the optimal solution when preemptions are allowed. We use the following observation in the design of the algorithm for the $1|energy, p_j = 1|cost$ problem.

**Lemma 1**: Let $\boldsymbol{a_k}$ be the total amount of time allocated at $\boldsymbol{ETI_k}$ in an optimal solution of $\boldsymbol{1|energy, p_j = 1, pmtn|cost}$, as described in the proof of Proposition 2. Then, there exists an optimal solution for the nonpreemptive case ($\boldsymbol{1|energy, p_j = 1|cost}$) in which the utilization of each $\boldsymbol{ETI_k}$ is at least $\boldsymbol{a_k - 1}$.

***Proof***: We proceed by contradiction: consider a solution of $1|energy, p_j = 1|cost$ where the utilization of one or more ETIs is less than the corresponding $a_k - 1$, and let $k'$ be the index of the cheapest such ETI. Since the total processing times of the jobs are the same in both the preemptive and nonpreemptive versions, it must be the case that in the solution of the nonpreemptive problem, at least one job is scheduled entirely in an ETI (or several ETIs) that is at least as expensive as $ETI_{k'}$. However, since there is an idle time of more than one time unit in $ETI_{k'}$, it is possible to remove one job from the more expensive ETIs and schedule it in $ETI_{k'}$. Note that if the idle time in $ETI_{k'}$ is divided among several idle periods, the above modification may require rescheduling jobs within $ETI_{k'}$ but without affecting their energy cost. ■

We design a polynomial-time 3-step algorithm based on the insight obtained from Lemma 1. The idea is to schedule many of the unit-time jobs in ETIs according to the solution of the preemptive case. The remaining $O(K)$ jobs are scheduled in the remaining idle periods using a dynamic programing algorithm. Finally, the two solutions are combined. The procedure is described below:

**Step 1** (dimension reduction): Solve the $1|energy, pmtn, p_j = 1|cost$ problem. Let $a_k$ be the time allocated to $ETI_k$ in this solution. Tentatively schedule $\max(\lfloor a_k - 1 \rfloor, 0)$ jobs at each ETI in the solution of $1|energy, p_j = 1|cost$.

The remaining number of jobs, $n' = n - \sum_{k=1}^{K} \max(\lfloor a_k - 1 \rfloor, 0)$, cannot be too large; in particular, it is $O(K)$.

**Step 2** (dynamic programming): We create a streamlined version of the problem with $n'$ tasks and $K$ ETIs of length $l'_k = l_k - \max(\lfloor a_k - 1 \rfloor, 0)$, $T' = \sum_{k=1}^{K} l'_k$, and the same $q_k$s. This problem is solved by a dynamic programming algorithm described below:

Recall that by Proposition 1, there exists an optimal solution for the streamlined problem that satisfies the verge property, i.e., an optimal solution such that any busy period starts at the beginning or ends at the end of an ETI. For the identical-processing-time case, this solution implies that the other end of a busy period (the one that does not necessarily coincide with an ETI verge) must be at an integer time difference from such a verge. That is, the set of starting times of any (unit time) job is included in:

$$\Theta = \{\theta : \theta = t_k + i, k = 0, \dots, K, i = -n', \dots, n'\} \cap [0, T - 1]. \quad (1)$$

Clearly, the cardinality of $\Theta$ is $O(Kn')$. We define the state space of our dynamic program as $\{0, \dots, n'\} \times \Theta$, which represents the number of jobs processed so far and the current time within the set $\Theta$. We define the function $next(\theta)$ to return the next member of $\Theta$; that is,

$$next(\theta) = \min\{\theta' \in \Theta : \theta' > \theta\} \quad (2)$$

and $next(\theta) = T$ if $\theta$ is the greatest element in $\Theta$.

In addition, for each candidate starting point, we can calculate the total energy cost of processing a job starting at this point. If the next time unit falls entirely within a single ETI, this is the electricity tariff of the ETI. If it spans several ETIs, the cost is calculated based on the relative time in each ETI. The worst-case complexity of such a straightforward calculation is $O(K)$ for each of the $O(Kn')$ members of $\Theta$, i.e., $O(K^2 n')$. We denote the cost of processing a job starting at time $\theta$ by $\eta_\theta$.

Now, the decision at each point in $\Theta$ is whether to start processing a job or not. Our Bellman's equation is

$$f(\theta, j) = \min\{f(next(\theta), j), \eta_\theta + f(\theta + 1, j + 1)\}$$

for all $\theta \in \Theta : \theta < T$, and $j = 0, \dots, n' - 1$. The $f(\theta, j)$ function represents the minimal remaining cost at time $\theta$ when considering the $j^{\text{th}}$ job. The boundary conditions are given by

$$f(\theta, j) = \begin{cases} 0 & j = n' \wedge \theta < T \\ \infty & otherwise \end{cases}$$

**Step 3** (merging): Here, we create the busy periods for the original problem from the busy periods created in Steps 1 and 2. From each busy period of Step 2 that starts at $ETI_{k_1}$ and ends at $ETI_{k_2}$, we create a busy period in the original problem that consists of all the time allocated in Step 2 to this busy period and all the time allocated in Step 1 to ETIs $k_1, \dots, k_2 - 1$. In addition, if there is no other busy period in Step 2 that starts at $ETI_{k_2}$, all the time allocated to this ETI in Step 1 is also included in the busy period. Finally, any time that was allocated in Step 1 that is not included in the busy periods created by the above process are added as "stand-alone" busy periods to the solution. The complexity of this process is $O(K)$.

Below is a summary of our algorithm for the $1|energy, p_j = 1|cost$ problem, followed by an illustrative example (Example 4).

**Algorithm 1**: A 3-step procedure for the $1|energy, p_j = 1|cost$ problem

**Step 1** (dimension reduction): Solve the $1|energy, pmtn, p_j = 1|cost$ problem with the same input and allocate $\max(\lfloor a_k - 1 \rfloor, 0)$ unit-time jobs in each $ETI_k$, where $a_k$ is the total processing time allocated in the preemptive case to $ETI_k$.

**Step 2** (dynamic programming) : Create a streamlined version of the problem where the length of each ETI is $l_k - \max(\lfloor a_k - 1 \rfloor, 0)$ and the number of jobs to be processed is $n - \sum_{k=1}^{K} \max(\lfloor a_k - 1 \rfloor, 0)$. Solve this problem using the dynamic programming algorithm described above.

**Step 3** (merging): Combine the two solutions obtained in the previous steps into a list of busy periods that materializes their total utilization prescribed by the first two steps at each $ETI_k$.

**Example 4:** Consider an instance of the $1|energy, p_j = 1|cost$ problem with $n = 20$ jobs and $K = 5$ ETIs, where $q = (1,2,40,3,1)$ and $l = (8.8,1.1,2.1,2,10)$. When applying Algorithm 1, we first solve the problem with preemptions. The optimal solution in terms of the time allocated at each ETI is $a = (8.8,1.1,0,0.1,10)$. See Figure 4.

| | | | | | |
|---|---|---|---|---|---|
| $l_k$ | 8.8 | 1.1 | 2.1 | 2 | 10 |
| $q_k$ | 1 | 2 | 4 | 3 | 1 |

**Figure 4**: Example 4, initial preemptive solution

We round down this solution to create an initial allocation of jobs to the ETIs of (7,0,0,0,9), with a total energy cost of 16.

| | | | | | |
|---|---|---|---|---|---|
| $l_k$ | 8.8 | 1.1 | 2.1 | 2 | 10 |
| $q_k$ | 1 | 2 | 4 | 3 | 1 |

**Figure 5**: Example 4, Step 1 rounded down solution

For Step 2, we are left with $n' = 4$ jobs, ETI lengths $l = (1.8,1.1,2.1,2,1)$ and the same electricity tariffs. The optimal solution at this step (obtained by solving the above dynamic program) when represented as the time allocated at each ETI is (1.8,0.2,0,1,1), with a cost of 6.2.

| | | | | | |
|---|---|---|---|---|---|
| $l_k$ | 1.8 | 1.1 | 2.1 | 2 | 1 |
| $q_k$ | 1 | 2 | 4 | 3 | 1 |

**Figure 6**: Example 4, Step 2, solution of the reduced problem

Merging these two solutions, we obtain the solution (8.8, 0.2, 0, 1, 10), with a total cost of $16 + 6.2 = 22.2$. Nine jobs are processed from time 0 to time 9 and eleven from time 13 to time 24.

| | | | | | |
|---|---|---|---|---|---|
| $l_k$ | 8.8 | 1.1 | 2.1 | 2 | 10 |
| $q_k$ | 1 | 2 | 4 | 3 | 1 |

**Figure 7**: Example 4, optimal solution

Interestingly, in this optimal solution, the utilization of the second ETI is much lower than the utilization of the fourth one, although the electricity cost at the fourth is 50% higher. Clearly, this outcome occurs because adding additional jobs at time 9 will require the allocation of processing time at the third ETI, which is very costly.

**Theorem 1:** Algorithm 1 solves the $\mathbf{1}|\boldsymbol{energy}, \boldsymbol{p_j = 1}|\boldsymbol{cost}$ problem in $\boldsymbol{O(K^3)}$ time.

**Proof:** We first prove the feasibility of the obtained solution. That is, we show that the time allocated by Step 3 satisfies the following requirements:

1. for each ETI the time does not exceed the ETI length;
2. the total time allocated sums to $n$, the number of jobs;
3. the allocated time can be grouped into busy periods of integer length.

Let $\mathcal{S}(l'_1, \dots, l'_K) \subset \mathbb{R}^K_+$ be the set of vectors that represent feasible processing times allocated to each $ETI$ in the dimension-reduced problem that can be grouped into busy periods of integer length. Note that by definition, a busy period does not contain idle time. For example, the vector $(0.3, 0.7, 0, 2)$ is a member of $\mathcal{S}(0.3, 0.9, 0.9, 2)$, while the vector $(0.3, 0, 0.7, 2)$ is not; $(0.5, 0.25, 0.25)$ is not a member of $\mathcal{S}(0.5, 0.5, 0.25)$, but it is a member of $\mathcal{S}(0.5, 0.25, 0.5)$. Moreover, since the members of $\mathcal{S}(\boldsymbol{l'})$ represent feasible processing time allocation, then $\boldsymbol{y} \in \mathcal{S}(\boldsymbol{l'})$ implies $y_k \leq l'_k$ for all $k = 1, \dots, K$, which is Requirement 1 above.

Let us formulate the problem as the following mathematical program:

$$\min \sum_{k=1}^{K} q_k y_k \tag{3}$$

$$\sum_{k=1}^{K} y_k = n' \tag{4}$$

$$\mathbf{y} \in \mathcal{S}(l') \tag{5}$$

The objective function (3) is simply to minimize the total energy cost of all the allocated times. Constraint (4) stipulates that we allocate enough time to process all the jobs. Constraint (5) limits the allocation of times to ETIs in such a way that allows them to be arranged into busy periods of integer length.

Equivalently, the original problem can be formulated as

$$\min \sum_{k=1}^{K} q_k x_k \tag{6}$$

$$\sum_{k=1}^{K} x_k = n \tag{7}$$

$$\mathbf{x} \in \mathcal{S}(l) \tag{8}$$

Let $\boldsymbol{x'}$ be the vector of times allocated to each ETI at Step 1. That is,

$$x'_k = \max(0, \lfloor a_k \rfloor - 1) \quad \forall k.$$

Let $\boldsymbol{y}^*$ be an optimal solution of (3)-(5). The theorem can be restated as the claim that $\boldsymbol{x'} + \boldsymbol{y}^*$ is an optimal solution of (6)-(8). To see that $\boldsymbol{x'} + \boldsymbol{y}^*$ satisfies (7), note that $\sum_{k=1}^{K} x'_k = n - n'$ and thus $\sum_{k=1}^{K}(x'_k + y^*_k) = \sum_{k=1}^{K} x'_k + \sum_{k=1}^{K} y^*_k = (n - n') + n' = n$. By the construction of the input of the first two steps, we have $x'_k + y^*_k \leq l_k$ (Requirement 1) since $y^* \leq l'_k = l_k - x'_k$. It is only left to show that the requirement that the times allocated at the ETIs can be grouped into

integer-length busy periods (Requirement 3) is satisfied. Consider a busy period created in Step 3 that starts at $ETI_{k_1}$ and ends at $ETI_{k_2}$. This busy period was created by adding integer times allocated at Step 1 to the busy period (also of integer length) created at Step 2. Clearly, the sum of several integers is also an integer.

Next, we prove the optimality of the solution. Assume by contradiction that $x' + y^*$ is not optimal and let $x^*$ be an optimal solution of (6)-(8) that satisfies $x_k^* \geq x_k'$. Such an optimal solution exists by Lemma 1. Now, let $y' = x^* - x'$, so $y'$ is a feasible solution of (3)-(5). The value of the optimal solution of (6)-(8) can be expressed as

$$\sum_{k=1}^{K} q_k x_k^* = \sum_{k=1}^{K} q_k x_k' + \sum_{k=1}^{K} q_k y_k' \geq \sum_{k=1}^{K} q_k x_k' + \sum_{k=1}^{K} q_k y_k^* = \sum_{k=1}^{K} q_k (x_k' + y_k^*)$$

which contradicts the assumption that $x' + y^*$ is suboptimal. The first equality follows from the construction of $y'$ such that $x^* = y' + x'$. The inequality follows from the optimality of $y^*$ with respect to (3)-(5).

The complexity of Step 1 and Step 3 is $O(K)$. The overall complexity of the dynamic program of Step 2 is dictated by the number $n'^2 K$ of its states and by the effort of calculating the values of $\eta_\theta$, which is $O(K^2 n')$. Recall that $n' = O(K)$; thus, the complexity of our case is $O(K^3)$, which is polynomial with respect to the size of the input of the $1|energy, p_j = 1|cost$ problem. Since the dimension of the input of the original problem is $O(K)$, this is a polynomial-time algorithm. Therefore, the complexity of Algorithm 1 is dominated by the complexity of Step 2, which is $O(K^3)$. ■

Note that the dynamic program could be used to solve the original problem directly, but since the complexity of the procedure depends on the number of jobs that should be scheduled, it is not polynomial unless we reduce the number of jobs to $O(K)$, as we did during the first step.

Finally, if the lengths of the ETIs are all integers, preemption will not occur in the optimal schedule obtained from the greedy procedure since jobs have a unit size, and all events happen at integral times. Therefore, it is possible to use the same greedy procedure that solves the preemptive case to solve the non-preemptive one.

# 4. The profit maximization problem

We turn now to the maximum-profit version of the scheduling problem under TOU electricity tariffs. We begin with negative complexity results for $1|energy, \xi_j = 1|profit$ and $1|energy, pmtn|profit$. In Section 4.2, we present a pseudopolynomial algorithm for $1|energy, pmtn|profit$. In the remainder of Section 4, we continue to the more intricate cases in which preemptions are not allowed.

## 4.1 Negative complexity results

First, we show that the $\xi$-semi-identical problem, $1|energy, \xi_j = 1|profit$, is NP-hard in the strong sense and APX. Clearly, this result implies that the general problem, $1|energy|profit$, is also at least as hard.

**Proposition 3**: The $\xi$-semi-identical problem, $1|energy, \xi_j = 1|profit$, is NP-hard in the strong sense and not approximable with any constant unless $P = NP$.

**Proof**: The $1|energy|cost$ problem can be reduced to the $1|energy, \xi_j = 1|profit$ problem by multiplying all the energy costs $q_k$ by some constant $a$ such that $aq_k p_j < 1$ for all ETI $k$ and job $j$. For that, any $a < \frac{1}{\max_{j,k} p_j q_k}$ will work. With such energy tariffs, the cost of processing any job at any time is less than 1 and thus if in an optimal solution of the profit maximization problem all the jobs are scheduled this is also an optimal solution for the cost minimization problem. Otherwise, if in an optimal solution of the profit maximization problem some jobs are not scheduled the cost minimization problem admits no feasible solution. ■

Next, we show that the profit maximization problem with preemption ($1|energy, pmtn|profit$) is NP-hard in the weak sense.

**Proposition 4:** The $1|energy, pmtn|profit$ problem is NP-hard in the weak sense. Moreover, the problem remains NP-hard even if the revenue is proportional to the processing times; that is, $\xi_j = \alpha p_j$ for some $a > 0$ and for all jobs.

**Proof**: The proof is accomplished by a reduction of the NP-complete decision problem "subset sum" (Karp 1972). Recall that the subset sum problem is defined as follows: given a set of positive integers $a_1, a_2, \dots, a_n$ and an integer $b$, decide if there is a subset of this set whose sum equals $b$. We reduce this problem to the $1|energy, pmtn|profit$ problem by creating an instance with one "cheap" ETI of length $l_1 = b$ with $q_1 = 0$ and a set of jobs with $p_j = \xi_j = a_j$. Clearly, the optimal solution of the $1|energy, pmtn|profit$ problem equals $b$ if and only if the answer to the subset sum problem is true. ■

## 4.2 The profit maximization problem with preemption

The solution of the $1|energy, pmtn|profit$ problem is characterized merely by stating the jobs that are selected to be processed and the amount of time allocated at each ETI for processing. We first observe a straightforward property of the optimal solutions of the problem that greatly simplifies its solution.

**Lemma 2**: The $\mathbf{1|energy, pmtn|profit}$ problem admits an optimal solution in which all the ETIs excluding one, ETI $\widetilde{\boldsymbol{k}}$, are either fully utilized or not utilized at all. Moreover, the energy cost $\boldsymbol{q_{\widetilde{k}}}$ is not smaller than the energy cost $\boldsymbol{q_k}$ of any of the fully utilized ETIs and not greater than that of any of the nonutilized ETIs.

**Proof:** Assume by contradiction a solution that does not satisfy the conditions of the lemma. Such a solution can be improved by deallocating the processing time of a job from the most expensive utilized ETI and reallocating this processing time in some cheaper ETI. ■

Next, we present a pseudopolynomial-time dynamic programming algorithm for the problem. Recall that the processing times of the jobs are all integers, but the lengths of the ETIs are not necessarily integers.

In this section the ETIs are first indexed in nondecreasing order of their electricity cost, $q_k$. Note that under this ordering, no preemptions are required in an optimal solution and thus the solution can be characterized by the starting time of each job. Later, we revert to the original order of the ETIs and greedily schedule the jobs that are selected to be processed.

Let $\tilde{\rho}_{tj}$ be the profit of job $j$ (the revenue $\xi_j$ net of the electricity cost required to produce it) assuming that job $j$ started at time $t$ when the ETIs are indexed according to their electricity cost. The calculation of each constant $\tilde{\rho}_{tj}$ can be performed in $O(K)$ units of time, and thus, the overall complexity of calculating all of these values is $O(nTK)$.

We define $f(t,j)$ as the expected profit after considering $j$ jobs and utilizing the cheapest $t$ time units. The following Bellman equations define the dynamic program that solves the $1|energy, pmtn|profit$ problem:

$$f(t,j) = \max\{\tilde{\rho}_{tj} + f(t+p_j, j+1), f(t,j+1)\} \quad \forall t \in T, j \in J$$

$$f(t,n+1) = 0 \quad \forall t \in T$$

$$f(T+1,j) = 0 \quad \forall j \in J$$

Solving the dynamic program can be done in $O(nT)$ time, and thus, the time complexity is dominated by the preprocessing step, which is $O(nTK)$. This is clearly a pseudopolynomial running time since the input dimension is independent of $T$.

## 4.3 Identical and semi-identical cases with preemptions

Both the $1|energy, pmtn, p_j = 1|profit$ and the $1|energy, pmtn, \xi_j = 1|profit$ cases admit polynomial-time algorithms. A pseudocode that describes such a solution procedure for the $p$-semi-identical maximum profit problem is given in Algorithm 2.

**Algorithm 2:** Polynomial-time algorithm for $1|energy, pmtn, p_j = 1|profit$

Input:
The jobs in nonincreasing order of revenue, i.e., such that $\xi_1 \geq \xi_2 \geq \cdots \geq \xi_n$
The ETIs in a nondecreasing order of tariffs, i.e., such that $q_1 \leq q_2 \leq \cdots \leq q_K$

Let $j = 1, k = 1$
Let $A_k = 0$ for all $k = 1, \dots, K$ // *the processing time allocated to* $ETI_k$
While $k \leq K$ and $j \leq n$
// *calculate the electricity cost of assigning the next job*
$t \leftarrow p_j$, $a \leftarrow 0$, $k' \leftarrow k$
While $t \geq 0$ and $k' < K$
$a \leftarrow q_{k'} \cdot \min(t, l_{k'} - A_{k'})$
$t \leftarrow t - \min(t, l_{k'} - A_{k'})$
$k' \leftarrow k' + 1$

// *allocating the job processing time to the ETIs*
If $\xi_j > a$ and $t' = 0$
$t \leftarrow p_j$
For $k'' = k$ to $(k' - 1)$
$A_k \leftarrow A_k + \min(t, l_{k''} - A_{k''})$
$t \leftarrow t - \min(t, l_{k''} - A_{k''})$
$j \leftarrow j + 1$
$k \leftarrow k''$
Else
break // *exit the while loop if no profitable allocation remains*
Return:$\{1, \dots, j-1\}, A_1, \dots, A_k$

Algorithm 2 starts by sorting the jobs in nonincreasing order of revenue and sorting the ETIs in nondecreasing order of tariffs. For simplicity of notation, we assume that these lists are already ordered. The idea is to first schedule the most profitable jobs and to continue as long as it is still possible to add jobs while increasing the profit. The algorithm loops through the ETIs and jobs while maintaining the index of the next job to consider, $j$, and the index of the current $ETI$, $k$. For each job, the algorithm first calculates the electricity cost of allocating it to the cheapest remaining ETIs. If the cost is smaller than the profit, the processing time of the job is allocated to these ETIs. Next, the indexes of the current job and the ETI are updated. If the current job is already not profitable or cannot fit in the remaining time, no other jobs are considered. Recall that in Table 1, we specified that the solution of the semi-identical preemptive case is described by a list of busy periods and a list of the jobs that are being processed. Clearly, the busy periods can be easily constructed from the time allocated at each ETI, returned as $A_1, \dots, A_K$.

**Proposition 5:** Algorithm 2 solves the $1|energy, p_j = 1, pmtn|profit$ problem in $O(\mathrm{n} \cdot \log n + \mathrm{K} \cdot \log K)$ time.

**Proof**: We first prove that the algorithm solves the problem and then show its complexity. Let $S^{ALG}$ denote the solution delivered by the algorithm and proceed by contradiction: assume that

there is a better solution $S^*$ such that $S^*$ satisfies the conditions of Lemma 2 and has the minimal number of scheduled jobs among these solutions. Note that $S^{ALG}$ is the only solution that satisfies Lemma 2 with the same number of jobs (up to shifting busy time within the most expensive ETI, a modification that does not change its value). Thus, the number of scheduled unit time jobs in $S^*$ must be different from the number in $S^{ALG}$. If $S^*$ has more jobs than $S^{ALG}$, it means that it utilizes times that are not utilized in $S^{ALG}$ and schedules at least one job with a revenue that is not greater than the one with the smallest revenue scheduled by $S^{ALG}$. Now, since $S^*$ is optimal with a minimal number of jobs, it is not possible to remove one job from it without decreasing its value. This observation implies that the marginal electricity cost of its last and most expensive time unit interval (or intervals) is smaller than the revenue of the job with the smallest revenue that it schedules. However, this job and these marginal time intervals were available for the algorithm when it stopped. The fact that the job was not scheduled by ALG is a contradiction because it implies that the stopping condition did not hold. A similar argument can be used to show why it cannot be the case that the number of jobs in $S^*$ is smaller than the number of jobs in $S^{ALG}$, and thus the contradiction holds in this case as well.

To show its complexity of the algorithm, we note that the main loop is repeated at most $n + K$ times since at each iteration, at least one of the counters $j$ or $k$ is increased by at least one. The internal loop (for calculating the electricity cost of jobs that utilize several ETIs) is repeated at most $K$ times over all the iterations of the main loop. Thus, the complexity is dominated by the sorting of the list of jobs, which takes $O(n \cdot log\, n)$, or by the sorting of the list of ETIs, which takes $O(K \cdot log\, K)$. ■

The solution process for the $1|energy, pmtn, \xi_j = 1|profit$ problem is very similar, except that in this case, the jobs are ordered in nondecreasing order of $p_j$. We do not include the formal proof here for the sake of brevity. A sound heuristic for the general case of the $1/energy, pmtn/profit$ problem is to order the jobs in nonincreasing order of $\xi_j/p_j$. This procedure, of course, does not guarantee an optimal solution.

Recall that in the identical case, $1|energy, pmtn, p_j = 1, \xi_j = 1|profit$, the input and output dimensions are $O(K)$, and thus, applying Algorithm 2 for this special case does not result in a polynomial-time algorithm. A streamlined version of the solution procedure for the identical jobs case is presented as a pseudocode in Algorithm 3.

**Algorithm 3**: Polynomial algorithm for $1|energy, pmtn, p_j = 1, \xi_j = 1|profit$

Input: The ETIs in nondecreasing order of tariffs, i.e., such that $q_1 \leq q_2 \leq \cdots \leq q_K$

Let $N = 0, k = 1$
Let $A_k = 0$ for all $k = 1, \ldots, K$ // *the processing time allocated to* $ETI_k$

// *greedily allocate time to the cheapest ETI*
While $k \leq K$
    // *allocating time to ETIs*
    If $\lceil N \rceil \leq n$ and $q_k < 1$
        $A_k \leftarrow \min(n - \lceil N \rceil, l_k)$
        $k \leftarrow k + 1$
        $N \leftarrow N + A_k$
    else
        break // *exit the while loop*

// *check the electricity cost of the last job*
$t \leftarrow N - \lfloor N \rfloor, k' \leftarrow k - 1, a \leftarrow 0$ // *first: the cost of the already allocated fraction*
While $t > 0$
    $a \leftarrow a + q_{k'} \cdot \min(t, l_{k'})$
    $t \leftarrow t - \min(t, l_{k'})$
    $k' \leftarrow k' - 1$

$t \leftarrow \lceil N \rceil - N, k' \leftarrow k$ // *second: the cost of the remaining fraction*
While $t > 0$
    $a \leftarrow a + q_{k'} \cdot \min(t, l_{k'})$
    $t \leftarrow t - \min(t, l_{k'})$
    $k' \leftarrow k' + 1$

If $a < 1$ // *allocate the remaining processing time of the last job if it is profitable*
    $t \leftarrow \lceil N \rceil - N$
    While $t > 0$
        $A_k \leftarrow \min(t, l_k)$
        $t \leftarrow t - \min(t, l_k)$
        $k \leftarrow k + 1$
Else // *deallocate the processing time of the last job if it is not profitable*
    $t \leftarrow N - \lfloor N \rfloor$
    While $t > 0$
        $k \leftarrow k - 1$
        $A_k \leftarrow \min(t, l_k)$
        $t \leftarrow t - \min(t, l_k)$
Return $N, A_1, \ldots, A_K$

Algorithm 3 starts by allocating the times of the ETIs with electricity tariffs that are cheaper than the unit revenue of the unit time jobs. Clearly, this allocation may result in an

allocation of a noninteger total processing time. In such a case, we calculate the marginal cost of the last scheduled job, assuming it will be completed in more expensive ETIs. This cost consists of the cost of the already allocated time of the last job and the cost of processing its remaining time if allocated to the cheapest ETIs among the remaining ones. If this total cost is smaller than the unit revenue, the allocation of the remaining job is completed; otherwise, the fraction that was already allocated is deallocated.

**Proposition 6**: Algorithm 3 solves the $1|energy, pmtn, p_j = 1, \xi_j = 1|profit$ problem in $O(\mathrm{K} \cdot \log K)$ time.

The proof is very similar to that of Proposition 5.

## 4.4 The $p$-semi-identical case without preemptions

Interestingly, while the $\xi$-semi-identical case without preemption is proven to be strongly NP-hard, the $p$-semi-identical case admits a polynomial-time algorithm. The difference is due to the fact that identical processing times imply a relatively simple combinatorial structure with $O(Kn)$ optional points of time to start a job, while identical revenues have no effect on the combinatorial structure. There exists an optimal solution of $1/energy, p_j = 1/profit$ where all the starting times of jobs are values in $\Theta$, as defined below:

$$\Theta = \{\theta : \theta = t_k + i, k = 0, \dots, K, i = -n, \dots, n\} \cap [0, T-1]. \quad (9)$$

The function $next(\theta)$ is defined as in (2).

Note that since the size of the input of $1|energy, p_j = 1|profit$ is $O(K+n)$, an algorithm that is polynomial in $|\Theta|$ is a polynomial-time algorithm for the problem. Indeed, $|\Theta| = O(nK)$. Based on this observation, the following dynamic program solves the problem in $O(n^2K)$:

$$f(\theta, j) = \begin{cases} \max\{f(next(\theta), j), \xi_j - \eta_\theta + f(\theta+1, j+1)\} & \theta \le T-1 \\ 0 & otherwise \end{cases}$$

for all $\theta \in \Theta$, $j = 0, \dots, n-1$. The boundary conditions are

$$f(\theta, n) = 0 \text{ for all } \theta \in \Theta.$$

This dynamic program is similar to the one solved in Step 2 of Algorithm 1. The only differences are that we add $\xi_j - \eta_\theta$ instead of $\eta_\theta$ for the decision to include job $j$ in our plan and that the set $\Theta$ is defined with the original number of jobs $n$ rather than with $n'$.

When $n \gg K$, a more efficient algorithm can be crafted based on the idea of Algorithm 1 introduced for the minimum cost problem. Step 1 is now based on the optimal solution of the $1|energy, pmtn, p_j = 1|profit$ problem and Step 2 on the dynamic program described above for the reduced size problem. The two solutions are merged in Step 3 in the same way as in Algorithm 1. The complexity of this new algorithm is $O(K^3 + n \log n)$, where the $O(n \log n)$ arises from the need to sort the jobs by $\xi_j$ at Step 1.

## 4.5 The identical case without preemption

Recall that in the $1|energy, p_j = 1, \xi_j = 1|profit$ case, the input of the problem consists of a list of $K$ ETIs with their lengths and energy costs and an integer number $n$ of identical jobs. The optimal schedule (the output) should be returned implicitly as a list of busy periods with their starting times and lengths. Thus, the complexity of a polynomial-time algorithm for this problem must be polynomial in the number of ETIs $K$ but independent of the number of jobs $n$. Indeed, such an algorithm is presented below.

The length of each busy period in the solution should be an integer, and the total lengths of the busy periods should not exceed $n$. However, in contrast to the minimum cost problem, here, producing fewer than $n$ jobs can be an optimal decision if it is profitable to do so. Hence, in this problem, the scheduler has to decide the number of jobs to schedule and their starting times.

Note that given the optimal number of jobs that should be processed, the optimal schedules of the maximum profit problem and of the minimum cost problem with the same number of jobs are identical. Hence, an optimal solution where the starting times of all jobs are in the set $\Theta$, as defined in (1), exists. Our proposed solution method follows Algorithm 1 for the identical minimum cost problem. We start by solving the preemptive version of the identical maximum profit problem ($1|energy, p_j = 1, \xi_j = 1, pmtn|profit$). Let $a_k$ denote the total processing time of $ETI_k$ in this solution. Next, we tentatively schedule $\max(\lfloor a_k \rfloor - 1, 0)$ jobs to each ETI. The remaining $O(K)$ jobs are scheduled in a modified version of the problem where the number of jobs that have already been scheduled is deduced from the lengths of the ETI$s$. The schedule is carried out using the dynamic program presented below:

$$f(\theta, j) = \begin{cases} \max\{f(next(\theta), j), 1 - \eta_\theta + f(\theta + 1, j + 1)\} & \theta \in \Theta \\ 0 & otherwise \end{cases}$$

for all $j = 0, \dots, n-1, \theta \in \Theta$. The function $f(\theta, j)$ represents the maximal profit that can be obtained by scheduling a subset of the set of jobs $\{j, \dots, n\}$ in the interval $[\theta, T]$. The boundary conditions are

$$f(\theta, n) = 0, \qquad \text{for all } \theta \in \Theta$$

This dynamic program differs from the one presented for solving the $1|energy, p_j = 1|cost$ problem mainly in the boundary conditions that allow reaching the end of the planning horizon without scheduling all $n$ jobs.

The optimal solutions of the first and second steps are then merged into a single solution, as in the minimum cost problem. The overall complexity is again $O(K^3)$.

We will now discuss a simple solution method for a special case of the problem where the number of jobs, $n$, is large enough so that the value of an optimal solution cannot be improved by increasing the number of available jobs. The fact that the number of jobs is large enough may be known a priori. For example, if the total number of available jobs is greater than the length of the planning horizon or if the solution of the preemptive case uses less than $n$ jobs. We denote this case by $1|energy, p_j = 1, \xi_j = 1, n = \infty|profit$.

With a large enough number of jobs, the second step can be solved by a simpler dynamic program with a smaller state space. This space consists of only the elements of $|\Theta|$ because we do not need to keep track of the number of jobs already scheduled. The Bellman equation for this problem is

$$f(\theta) = \begin{cases} \max\{f(next(\theta), 1 - \eta_\theta + f(\theta+1)\} & \theta \leq T-1 \\ 0 & otherwise \end{cases}$$

for all $\theta \in \Theta$. The boundary conditions are included in the equation. The time complexity of this procedure is $O(K^2)$, and it dominates all of Algorithm 1.

# 5. Cost minimization with release times and due dates

## 5.1 The cost minimization problem with preemption

The problem $1|energy, \text{pmtn}, r_j, d_j|cost$ can be cast as an instance of the Hitchcock transportation problem. We define the set $E$ of epoch ends as the union of all the release times, due dates and ends of ETIs:

$$E = \{t_k : k = 0, \dots, K) \cup \{r_j : j = 1, \dots, n\} \cup \{d_j : j = 1, \dots, n\}$$

The set $E$ is sorted in increasing order, and we let $e_i$ denote the $i^{th}$ element, where $i = 0$ refers to the first element $e_0 = 0$. Next, we define a set of intervals

$$I(E) = \{(e_{i-1}, e_i) : i = 1, \dots, |E| - 1\}$$

Note that the electricity tariff is constant during each interval in $I(E)$. Let $q_i'$ denote the electricity tariff during the $i^{th}$ interval.

Using this notation, we define a transportation problem with a source for each interval in $I(E)$ and a sink for each job. The supply of each interval $(e_{i-1}, e_i)$ is its length $e_i - e_{i-1}$. The demand of each job $j$ is $p_j$. The transportation cost between each interval $i$ and job $j$ is $q_i'$ if $(e_{i-1}, e_i) \subseteq (r_j, d_j)$ and infinite otherwise. The solution of the transportation problem prescribes the time allocated to each job during each interval.

The state of the art in solving the Hitchcock transportation problem is the algorithm given by Brenner (2008). He presents an algorithm that solves an instance with $s$ sources and $t$ sinks in $O(st^2(\log s + t \log t))$ time. In our case, there are $O(n + K)$ sources and $n$ sinks. That is, our problem can be solved in $O((n^4 + n^3K)\log n + (n^3 + n^2K)\log(n + K))$ time. In particular, if the number of jobs, $n$, is not much smaller than the number of ETIs, $K$, the complexity is dominated by $O(n^4 \log n)$.

## 5.2 The nonpreemptive min-cost problem with unit processing times

In this section, we present a polynomial-time algorithm for the $1|energy, p_j = 1, d_j|cost$ problem. The same idea is also applicable to $1|energy, p_j = 1, r_j|cost$. Our algorithm is based on the following two simple observations.

**Observation 1**: There exists an optimal solution of the $1|energy, p_j = 1, d_j|cost$ problem where the jobs are sorted in nondecreasing order of their due dates (EDD). This observation can be shown by a simple swapping argument.

The following observation is valid under the assumption that the jobs are indexed by EDD order.

**Observation 2**: The starting time of job $j$ in a solution of the $1|energy, p_j = 1, d_j|cost$ problem is not greater than

$$\omega_j = \min_{j' \in \{j,\dots,n\}} \{d_{j'} - (j' - j + 1)\}. \quad (10)$$

Indeed, under the EDD rule, for any pair of jobs $j < j'$, job $j$ must start at least $j' - j$ units of time before job $j'$ starts and job $j'$ must start at least one unit of time before its due date.

The values of $\omega_j$ for all $j = 1, \dots, n$ can be calculated in linear time using the following recursive formula:

$$\omega_j = \min(d_j, \omega_{j+1}) - 1 \quad \forall j = 1, \dots, n-1$$

and

$$\omega_n = d_n - 1$$

Based on the definition of $\Theta$ in (9), we define

$$H = \Theta \cup \{\omega_j + h : j = 1, \dots, n, h = 0, \dots, n - j\}$$

$$H_j = \{\theta \in H : j - 1 \le \theta \le \omega_j\}$$

**Proposition 7**: There exists an optimal solution where the starting time of each job $j$ is in the set $H_j$.

Proof: Assume by contradiction that there is no such optimal solution and consider an optimal solution where the first job that starts at a time not in $H_j$ has a maximal index, among all the optimal solutions that satisfy the EDD property. We use $\tau_j$ to denote the starting time of job $j$ in a solution. Let $j^*$ be the first job in this solution that does not satisfy $\tau_{j^*} \in H_j$. Next, let $h$ denote the position of job $j^*$ in its busy period in this solution, where $h = 0$ refers to the first job in the busy period. If $h > 0$, then job $j^* - 1$ starts at time $(\tau_{j^*} - 1) \in H$, and in particular, $(\tau_{j^*} - 1) \in H_{j^*-1}$. Now, we consider two cases:

1) $(\tau_{j^*} - 1) \in \Theta$,

2) $(\tau_{j^*} - 1) \in \{\omega_j + h : j = 1, \dots, n, h = 0, \dots, n - j\}$.

In case 1, it follows from the definition of $\Theta$ that $\tau_{j^*} \in \Theta$ and thus $\tau_{j^*} \in H$, which is a contradiction.

In case 2, since $\tau_{j^*-1} \leq \omega_{j^*-1}$, we know that for some $j' \leq j^* - 1$, $\tau_{j^*-1} = \omega_{j'} + (j^* - 1 - j')$ and therefore $\tau_{j^*} = \omega_{j'} + (j^* - j')$. Since it must be the case that $j^* - j' \leq n - j'$, we conclude that $\tau_{j^*} \in \{\omega_j + h: j = 1, \dots, n, h = 0, \dots, n - j\}$, which is again a contradiction.

If, on the other hand, $h = 0$, meaning that $j^*$ is the first job in its busy period, we consider two other cases. Either

1) The electricity tariff of the first ETI of the busy period is not greater than the tariff at the last ETI of the busy period, or

2) The electricity tariff of the first ETI of the busy period is greater than the tariff at the last ETI of the busy period.

In case 1, the entire busy period may be shifted backward until it meets either another busy period or the beginning of the ETI. In either case, the busy period will satisfy the verge property, and all the jobs in it will start at times in $\Theta$. In case 2, it would be desirable to shift the busy period forward, but since this is an optimal solution, such a shift must be impossible because one of the jobs in the busy period ends at its due date. However, if this is the case, this job and all the jobs $j$ that precede it in the busy period start at $\omega_j$, which is also in $H$.

Finally, since $\tau_{j^*} \in H$ and $j - 1 \leq \tau_{j^*} \leq \omega_j$, we conclude that $\tau_{j^*} \in H_j$. ■

We define a function

$$next(\theta) = \min\{\theta' \in \mathrm{H}: \theta' > \theta\}$$

and the function

$$succ(\theta, j) = min\{\theta' \in \mathrm{H}_{j+1}: \theta' \geq \theta + 1\}.$$

Recall that the constant $\eta_\theta$ is the energy cost of the job that starts at time $\theta$. The optimal solution can be obtained from the following Bellman equation defined for each state $\theta \in \Theta$ and $j \in \{1 \dots, n\}$:

$$f(\theta, j) = \begin{cases} \min\{f(next(\theta), j), \eta_\theta + f(succ(\theta, j), j + 1,)\} & \theta \in \mathrm{H_j}: \mathrm{next}(\theta) \in \Theta_\mathrm{j} \\ \eta_\theta + f(\, succ(\theta, j), j + 1) & \theta = \omega_j \end{cases}. \quad (11)$$

The function $f(\theta, j)$ represents the minimal cost of scheduling jobs $\{j, \dots, n\}$ in the time interval $[\theta, T]$. The optimal solution is obtained by evaluating the program starting from $f(1,0)$. The size of the state space of (11) is $O(n^2 K)$, and the calculation of the value of each state can be done in constant time. The preprocessing step of calculating $\eta_\theta$ for each $\theta \in \Theta \cup \{\omega_j: j \in 1, \dots, n\}$ is $O(K)$. However, a more cautious implementation allows the calculation of all $\eta_\theta$ to be done in $O(|\Theta \cup \{\omega_j: j \in 1, \dots, n\}|) = O(nK)$ time, though the straightforward procedure requires $O(nK^2)$ time. Therefore, the Bellman equations (11) can be solved in $O(n^2 K)$ time.

A similar approach could be applied to the $1|energy, r_j|cost$ problem with the jobs sorted in nondecreasing order of their release time and a lower bound for the starting time calculated as

$$\delta_j = \max(r_j, \delta_{j-1}) - 1 \quad \forall j = 2, \dots, n$$

and

$$\delta_1 = r_1$$

However, the problem $1|energy, r_j, d_j|cost$ is left for future consideration. We note that the solution approach presented in this section does not work because we do not know of any order that jobs follow in an optimal solution.

# 6. The variable-energy-consumption generalization

Fang et al. (2016) study a generalization of the $1|energy|cost$ problem in which power consumption varies among jobs and denote it as Problem U. We let $\pi_j$ denote the power consumption of job $j$. The cost of processing job $j$, assuming it is processed entirely during $ETI_k$, is thus $\pi_j \cdot p_j \cdot q_k$ (the power consumption times the duration times the electricity tariff). If the processing of a job spans over several ETIs, its cost is calculated proportionally. Using the 3-field notation, we refer to this problem as $1|energy, var\text{-}power|cost$. This problem is proven by Fang et al (2016) to be strongly NP-hard and not approximable. Note that the $1|energy|cost$ problem is clearly obtained as a special case of the $1|energy, var\text{-}power|cost$ problem with $\pi_j = 1$. Recall that the complexity results of the $1|energy, var\text{-}power|cost$ are clearly implied by the fact that the special case of identical power consumption is also NP-hard and not in APX (Theorem 1).

Fang et al (2016) discuss the $1|energy, var\text{-}power, p_j = p|cost$ problem (using our notation) where $p$ and the ETI lengths are integers. We consider the equivalent problem $1|energy, var\text{-}power, p_j = 1|cost$ with rational ETI lengths. For the special case of a particular electricity tariff structure known as a pyramid structure, they develop a polynomial-time greedy algorithm. However, the complexity status of the $1|energy, var\text{-}power, p_j = 1|cost$ problem for a general tariff structure is left open.

The method for solving the $1|energy, pmtn, r_j, d_j|cost$ problem using the transportation problem as described in Section 5 can easily be generalized to solve the $1|energy, var-power, pmtn, r_j, d_j|cost$ problem. This method requires setting the transportation cost between each interval $(e_{i-1}, e_i)$ and job $j$ to $\pi_j q'_i$.

In the rest of this section, we derive an interesting structural property of optimal solutions of the $1|energy, var\text{-}power|cost$ problem and present a compact integer programming formulation for the identical-processing-time case that is based on this property.

It turns out that the verge property does not hold for the $1|energy, var\text{-}power|cost$ problem even in the identical-processing-time case. For a counterexample, consider an instance of the problem with $n = 3, k = 5, p_j = 1, \pi = (4,8,8), q = (100,1,10,1,100), l = (1.25,0.75,1,0.75,1.25)$. It is easy to see that it is optimal to schedule the three jobs in order (2,1,3) at one busy period starting at time 1 with a cost of 452. However, this solution does not satisfy the verge property, as seen in Figure . No other schedule of the three jobs can yield equal or smaller cost (except for the one obtained by swapping jobs 2 and 3 within the same busy period). For example, shifting the busy period forward to start at time 1.25 (to the beginning of

$ETI_2$) results in a schedule with a cost of 461. Clearly, this example could be transformed to one with integer length ETIs and identical-length jobs by multiplying all times by 4.

| | | $\pi_2 = 8$ | $\pi_1 = 4$ | $\pi_3 = 8$ | |
|---|---|---|---|---|---|
| $l_k$ | 1.25 | 0.75 | 1 | 0.75 | 1.25 |
| $q_k$ | 100 | 1 | 10 | 1 | 100 |
| $ETI_k$ | 1 | 2 | 3 | 4 | 5 |

**Figure 8:** Example of an instance of the $\mathbf{1}|\boldsymbol{energy}, \boldsymbol{var\text{-}power}, \boldsymbol{p_j = 1}|\boldsymbol{cost}$ problem that does not satisfy the verge property.

Fortunately, although the verge property does not hold, we can prove that a similar useful property is satisfied.

**Definition 3**: the *boundary property*. A solution of the $1|energy, var\text{-}power|cost$ problem is said to satisfy the *boundary property* if, in each busy period, the start time or end time of at least one job coincides with a boundary of an ETI.

The solution presented in Figure satisfies this property because the start time of job 1 coincides with the start time of $ETI_3$.

**Proposition 8:** There exists an optimal solution of the $1|energy, var\text{-}power|cost$ problem that satisfies the boundary property.

**Proof:** Consider an optimal solution with a minimal number of busy periods that violate the boundary property. Let us examine the first (relative to the starting times of the busy periods) such violating busy period. Note that this busy period can be shifted backward or forward without increasing either the total electricity consumption or its cost until either the busy period hits another busy period or the boundary property is satisfied. In the first case, the number of busy periods is reduced by one, contradicting the minimality of the number of busy periods. In the second case, the number of busy periods that violate the boundary conditions is reduced by one. We can continue with this process until either there are no violating busy periods, or the number of busy periods is reduced by one, contradicting our assumption. Hence, there exists an optimal solution of the $1|energy, var\text{-}power|cost$ problem in which all busy periods satisfy the boundary property. ■

Recall that the identical-processing-time case ($p_j = p$) is equivalent to the unit-time case ($p_j = 1$) when the lengths of the ETIs are not necessarily integers. An implication of Proposition 8 for the solution of the $1|energy, var\text{-}power, p_j = 1|cost$ problem is that there exists an optimal solution of the problem in which the starting time of any job differs by an integer number of time units from an ETI boundary. That is, the cardinality of the set of candidate starting times is $O(nK)$. In particular, this set can be defined as

$$\Theta = \{t: t = t_k + i, k = 0, \dots, K, i = -n, \dots, n\} \cap [0, T-1]. \quad (12)$$

Note that $\Theta$ is defined in a similar way to the set with the same notation introduced in Step 2 of Algorithm 1 as presented in (1), only here $n'$ is replaced by the total number of jobs, $n$. Unfortunately, the boundary property does not lead to a polynomial-time dynamic program

algorithm for the $1|energy, var\text{-}power, p_j = 1|cost$ problem similar to the one applicable to the $1|energy, p_j = 1|cost$ problem since a state space that consists of all the pairs $\Theta \times \{0, \dots, n\}$ is not enough to encode the information for an optimal local decision. For this purpose, one needs to know which jobs remain to be processed.

Next, we formulate the $1|energy, var\text{-}power, p_j = 1|cost$ problem as a linear integer program. For each $\theta \in \Theta$, let $\eta_\theta$ be the electricity cost per power unit for a time unit that starts at time $\theta$. In addition, for each $\theta$, let $\mathcal{B}_\theta = \Theta \cap [\theta, \theta + 1)$. That is, $\mathcal{B}_\theta$ consists of the time $\theta$ and all the other members of $\Theta$ that are within one unit from $\theta$. We define a set of $O(Kn^2)$ binary decision variables $x_{\theta,j}$ that equal "1" if job $j$ is scheduled to start at time $\theta$ and "0" otherwise:

$$\min \sum_{\theta \in \Theta} \sum_{j \in J} \eta_\theta \pi_j x_{\theta,j} \tag{13}$$

$$\sum_{\theta \in \Theta} x_{\theta,j} = 1 \quad \forall j \in J \tag{14}$$

$$\sum_{\theta \in \mathcal{B}_\theta} \sum_{j \in J} x_{\theta,j} \leq 1 \quad \forall \theta \in \Theta \tag{15}$$

$$x_{\theta,j} \in \{0,1\} \quad \forall \theta \in \Theta, j \in J$$

In the objective function(13), the total cost of the schedule is minimized. Constraint (14) stipulates that each job is scheduled exactly once at a candidate starting time in $\Theta$. Constraint (15) assures that no two jobs are scheduled to be processed at an overlapping time.

A profit maximization version of this problem can be formulated and is denoted by $1|\text{energy}, var\text{-}power, p_j = 1|profit$, where the schedulers select which jobs to process and according to what schedule so to maximize their net profit. The problem exhibits the same boundary property as the cost minimization problem; thus, in the unit time case, the set of possible starting times of jobs is the set $\Theta$ defined above. It is possible to formulate the problem as a linear integer program as follows:

$$\max \sum_{\theta \in \Theta} \sum_{j \in J} (\xi_j - \eta_\theta) \pi_j x_{\theta,j} \tag{16}$$

$$\sum_{\theta \in \Theta} x_{\theta,j} \leq 1 \quad \forall j \in J \tag{17}$$

$$\sum_{\theta \in \mathcal{B}_\theta} \sum_{j \in J} x_{\theta,j} \leq 1 \quad \forall \theta \in \Theta \tag{18}$$

$$x_{\theta,j} \in \{0,1\} \quad \forall \theta \in \Theta, j \in J$$

The objective function (16) maximizes the revenue net of the electricity cost of the scheduled jobs. Constraint (17) is similar to (14) except that here, in the profit maximization problem, a job is scheduled at most once rather than exactly once. Constraint (14) works exactly as in the cost minimization model to eliminate the possibility of scheduling jobs at overlapping times.

It should be noted that formulations (13)-(15) and (16)-(18) can be easily extended to include release time and due date constraints as well as lateness or tardiness components in the objective function.

# 7. Summary and conclusions

Recently, we have observed growing interest and awareness in better ways to operate and manage systems to improve their energy efficiency. We follow this direction and study operational systems that aim to save energy by taking into account time-of-use (TOU) electricity tariffs. In particular, we study cost minimization and profit maximization scheduling problems on a single machine in such environments.

Tables 2 and 3 summarize our results, presenting the complexity status of all four variants of the cost minimization problem and all eight variants of the profit maximization problem. For the release time and due date extensions, we also report our findings regarding the cost minimization problem.

**Table 2:** Complexity status for the minimum cost problem

| Problem | Complexity |
|---|---|
| $1\|energy\|cost$ | Strongly NP-hard and not in APX. A reduction from 3-partition by Fang et al (2016). |
| $1\|energy, pmtn\|cost$ | Greedy, select the cheapest ETIs, $O(K \log K + n)$. Fang et al (2016) presented an $O(K \log K + n \log n)$ algorithm for a more general case. |
| $1\|energy, p_i = 1, pmtn\|cost$ | Greedy, select the cheapest ETIs, $O(K \log K)$ |
| $1\|energy, p_i = 1\|cost$ | $O(K^3)$, a 3-step algorithm based on the of solution of the preemptive case, dynamic programming and merging the two solutions. |
| $1\|energy, pmtn, r_j, d_j\|cost$ | $O((n^4 + n^3K) \log n + (n^3 + n^2K) \log(n + K))$ Formulated as a transportation problem |
| $1\|energy, p_j = 1, d_j\|cost$<br>$1\|energy, p_j = 1, r_j\|cost$ | $O(n^2K)$ using dynamic programming |

**Table 3:** Complexity status for the maximum profit problem

| Problem | Complexity and solution method |
|---|---|
| $1\|energy\|profit$ | Strongly NP-hard (reduction from bin packing decision problem as in the minimum cost problem) and in APX (reduction from the set partitioning). |
| $1\|energy, pmtn\|profit$ | Weakly NP-hard (reduction from knapsack), pseudopolynomial-time algorithm, $O(nTK)$. |
| $1\|energy, \xi_j = 1\|profit$ | Strongly NP-hard (reduction from bin packing decision problem as in the minimum cost problem). |
| $1\|energy, p_j = 1\|profit$ | $O(n^2K)$ dynamic programming or $O(K^3 + n \log n)$, 3-step algorithm. |
| $1\|energy, p_j = 1, \xi_j = 1\|profit$ | $O(K^3)$, 3-step algorithm. If there are more than $T$ jobs ($n \geq T$) then the problem can be solved in $O(K^2)$. |

| | |
|---|---|
| $1\|energy, pmtn, \xi_j = 1\|profit$ | $O(K \log K + n \log n)$, greedy algorithm |
| $1\|energy, pmtn, p_j = 1\|profit$ | $O(K \log K + n \log n)$, greedy algorithm. |
| $1\|energy, pmtn, p_j = 1, \xi_j = 1\|profit$ | $O(K \log K)$, greedy algorithm. |

The solution methods that we used for the cost minimization problems with release times and due dates cannot be easily adapted for the profit maximization variations; the development of such methods is left for future research. Another interesting question that is left open by this study is the approximability of the $1|energy, pmtn|profit$ problem. We show that this problem is weakly NP-hard.

For a special case of the $1|energy, var - power, p_i = 1|cost$ problem where the TOU tariffs satisfy the pyramidal structure, Fang at el. (2016) presented a greedy algorithm but left the case of general TOU tariffs open. Based on the boundary property of optimal solutions (Proposition 8), we formulated a compact linear integer program for the $1|energy, var - power, p_j = 1|cost$ and $1|energy, var - power, p_j = 1\ |profit$ problems. These formulations are easily extendable to include release times and due dates. It is still a challenge to find the complexity status of these unit time problems.

**Acknowledgment**: the authors wish to express their thanks to the anonymous reviewers for their constructive remarks.

# Bibliography

Aghelinejad, M., Ouazene, Y., and Yalaoui, A. (2019). Complexity analysis of energy-efficient single machine scheduling problems. Operations Research Perspectives, 6, 100105.

Brenner U, 2008, A faster polynomial algorithm for the unbalanced Hitchcock transportation problem. Operations Research Letters, Vol 36, 408-413.

Che, A., Wu, X., Peng, J., and Yan, P., 2017, Energy-efficient bi-objective single-machine scheduling with power-down mechanism. Computers & Operations Research, 85, 172-183.

Che, A., Zeng, Y., and Lyu, K., 2016. An efficient greedy insertion heuristic for energy-conscious single machine scheduling problem under time-of-use electricity tariffs. Journal of Cleaner Production, 129, 565-577.

Chen, B., and Zhang, X., 2019, Scheduling with time-of-use costs. European Journal of Operational Research, 274(3), 900-908.

Fang K., 2013, Algorithmic and mathematical programming approaches to scheduling problems with energy-based objectives, Ph.D. dissertation, Purdue University, ProQuest Dissertations Publishing, 2013. 3613120.

Fang, K., Uhan, N.A., Zhao, F. and Sutherland, J.W., 2016. Scheduling on a single machine under time-of-use electricity tariffs. Annals of Operations Research, 238(1-2), 199-227.

Gahm C., Denz F., Dirr M. and Tuma A., 2016, Energy-efficient scheduling in manufacturing companies: A review and research framework, European Journal of Operational Research, 248(3).

Garey M. R. and Johnson D. S., 1978, "Strong" NP-Completeness Results: Motivation, Examples, and Implications Journal of the ACM, 25, 499-508.

Garey, M.R. and Johnson, D.S., 1979, Computers and Intractability: A Guide to the Theory of NP-Completeness. pp. 96–105. A Series of Books in the Mathematical Sciences. San Francisco, California: W. H. Freeman and Co.

Graham, R.L., Lawler, E.L., Lenstra, J.K., and Rinnooy Kan, A.H.G., 1979, Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey, Annals of Discrete Mathematics, 3, 287-326.

Karp R., 1972, Reducibility among Combinatorial Problems, in R. E. Miller and J. W. Thatcher (eds). Complexity of Computer Computations, Plenum Press, NY, 85-103.

Rubaiee, S., Cinar, S., and Yildirim, M. B., 2018, An energy-aware multiobjective optimization framework to minimize total tardiness and energy cost on a single-machine nonpreemptive scheduling. IEEE Transactions on Engineering Management (Early access).

Shrouf F., J. Ordieres-Meré, A. García-Sánchez and M. Ortega-Mier, 2014, Optimizing the production scheduling of a single machine to minimize total energy consumption costs. Journal of Cleaner Production Vol. 67, 197-207.

Wan G. and X. Qi, 2010, Scheduling with Variable Time Slot Costs, Naval Research Logistics, 57, 159–171.

Zhao Y., X. Qi, and Minming L. 2016. On scheduling with non-increasing time slot cost to minimize total weighted completion time, Journal of Scheduling 19,759–767.

Zhong, W. and Liu, X. 2012. A single machine scheduling problem with time slot costs. Recent advances in computer science and information engineering. Heidelberg: Springer, 678-681.