RAIRO-Oper. Res. 50 (2016) 715–732
DOI: 10.1051/ro/2015063

RAIRO Operations Research
www.rairo-ro.org

BI-OBJECTIVE OPTIMIZATION OF SINGLE-MACHINE BATCH SCHEDULING UNDER TIME-OF-USE ELECTRICITY PRICES

Junheng Cheng[1,2], Feng Chu[2,3], Chengbin Chu[4] and Weili Xia[4]

Abstract. Time-of-use (TOU) electricity pricing has been a common practice to enhance the peak load regulation capability of power grid. Meanwhile, it provides a good opportunity for industries to reduce energy costs, especially for energy-intensive ones, where batch scheduling is often involved. Majority of batch scheduling problems have been proved to be NP-hard, even for most single-machine environments. Optimizing batch scheduling under TOU policy in these industries will be of great significance. Single-machine batch scheduling is an important basis for more complicated shop scheduling problems. This paper investigates a bi-objective single-machine batch scheduling problem under TOU policy: the first objective is to minimize the makespan and the second is to minimize the total electricity costs. The considered problem is first formulated as a bi-objective mix-integer linear programming (MILP) model and is demonstrated to be NP-hard. Subsequently, the MILP is simplified by analyzing properties and search space for a Pareto optimal solution is greatly reduced. Then, an exact ε-constraint method is adapted to obtain its Pareto front, which is accelerated due to these properties. Finally, a preferable solution is recommended for decision makers *via* a fuzzy-logic-based approach. Computational results on randomly generated instances show the effectiveness of the proposed approach.

Mathematics Subject Classification. 90B35, 90B50, 90C11.

Received May 20, 2015. Accepted December 2, 2015.

1. Introduction

Electricity, as one of the most widely used energies, plays a very important role in modern industries. For example, its consumption accounts for approximately 30% of the total industry energy consumption in APEC area [1]. The rapid and ongoing growth of electricity demand has become a bottleneck for sustainable economic development. Therefore, improving electric power efficiency and saving electricity have been important issues for many countries. It usually can be achieved through three strategies: the first one is called structure energy-saving, which is to reduce the share of high energy consumption industries; the second one is called

Keywords. Batch scheduling, TOU pricing policy, bi-objective optimization, makespan, electricity cost.

[1] School of Management, Northwestern Polytechnical University, Xi'an, P.R. China.

[2] Laboratoire IBISC, University of Evry-Val d'Essonne, Evry, France. feng.chu@ibisc.univ-evry.fr

[3] School of Transportation Engineering, Hefei University of Technology, Hefei, P.R. China.

[4] Laboratoire Genie Industriel, CentraleSupélec, Université Paris-Saclay, Grande Voie des Vignes, 92290 Chatenay-Malabry, France.

Article published by EDP Sciences

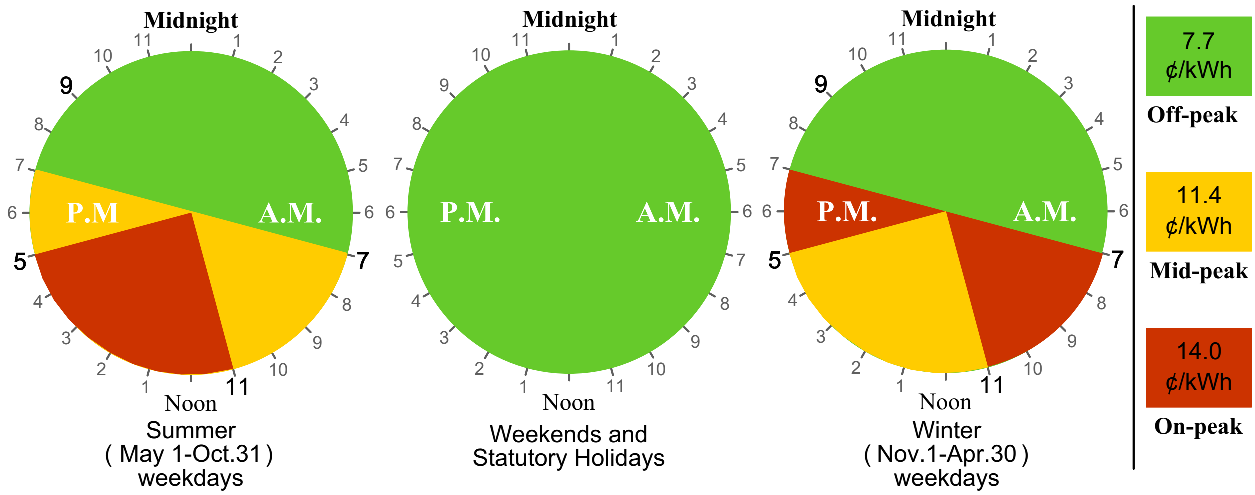

FIGURE 1. An example of Time-of-Use electricity prices (Source: Ontario Energy Board).

technical energy-saving, which is to improve the techniques of production devices and decrease their energy consumption; the third one is called management energy-saving, which is to reduce energy consumption by management approaches. Compared with the first two strategies, the third one is the fastest, most inexpensive and flexible one.

As a practical management energy-saving strategy, Demand Response (DR) strategy has been implemented in many countries such as the United States, Canada, France and China [5]. Time-of-use (TOU) electricity pricing policy (see Fig. 1) is one of the most important and popular DR strategies [29]. It aims to improve the peak load regulation ability and promote the balance of electricity demand between on-peak and off-peak periods by price control, so that the electricity demand in on-peak periods can be met without constructing more costly backup facilities [29]. Under TOU policy, manufacturing industries, especially power-intensive ones, are motivated to improve their competitiveness by reducing energy costs of production. For example, the total electricity costs can be saved by processing high energy consumption operations in off-peak periods. Certainly, considering electricity costs reduction together with traditional optimization criteria in production management will be of huge significance.

Scheduling is an important part of production management and optimal scheduling scheme can significantly improve production efficiency and reduce production costs. As an important branch of production scheduling, batch scheduling has been extensively applied to high technology and modern industries such as semiconductor manufacturing [13, 32], steel manufacturing [24], and aircraft industry [28]. Most of them are power-intensive industries and the electricity cost accounts for 10∼50% of the final product costs [9]. Batch scheduling involves batch-processing machines that can process multiple jobs simultaneously. Different from classical scheduling problems, batch scheduling needs to allocate jobs into batches, and then schedule the formed batches. It is known that majority of batch scheduling problems are NP-hard, even for most single-machine environments. Single-machine batch scheduling problem is an important foundation for studying more complex batch scheduling problems and the makespan is a basic production optimization criterion. Interested readers can refer to [11, 17, 20, 26, 27, 31] for batch scheduling optimization.

In recent years, optimizing production scheduling to save electricity costs under TOU policy has attracted the attention of many researchers. Shrouf *et al.* [22] studied a single machine production scheduling problem under TOU policy, to minimize the total electricity costs under traditional work shifts. A genetic algorithm (GA) was proposed to obtain near-optimal solutions and 13 instances with up to 60 jobs were tested. Fang *et al.* [7]

also focused on scheduling jobs on uniform-speed and speed-scalable single machine under TOU policy with the same optimization criterion. Heuristics were developed for the problems. Moon *et al*. [18] examined a parallel machine scheduling problem to minimize the weighted sum of the makespan and time-dependent electricity costs. A hybrid GA was suggested to yield near-optimal solutions for 24 instances with up to 60 jobs under two scenarios. Wang *et al*. [29] addressed scheduling with power-down strategy in manufacturing systems considering TOU policy to minimize the total electricity costs. A particle swarm-optimization-based meta-heuristic was proposed to solve the problem and 26 instances with up to three machines were tested. Luo *et al*. [15] developed a novel ant colony based meta-heuristic algorithm to solve a bi-objective hybrid flow shop scheduling problem with minimizing both makespan and total electricity costs under TOU policy. And the method was tested on 18 instances with up to 50 jobs. Later, Zhang *et al*. [33] formulated a time-indexed integer programming model for a bi-objective flow shop scheduling problem under TOU policy, in which the objectives were to minimize total electricity costs and the carbon emissions, while simultaneously ensuring the production throughput. In their paper, several electricity costs were simply selected for the electricity cost optimization, and the carbon emissions was optimized with integer programming for each selected electricity cost to obtain a set of Pareto optimal solutions. Only one instance involving one product and eight machines was tested. It can be found that the above studies on production scheduling under TOU policy mainly focused on classical scheduling problems, such as single machine scheduling, parallel machine scheduling, and flow shop scheduling. Almost all these studies proposed meta-heuristic algorithms to obtain near-optimal solutions and relatively few instances were tested.

To the best of our knowledge, our preliminary work [3] is the first study on batch scheduling under TOU policy, in which the makespan and the total electricity costs are simultaneously considered. In [3], a nonlinear bi-objective model was formulated. The ε-constraint method was used to solve the bi-objective model. The corresponding single-objective models considered in the ε-constraint method were linearized *via* theoretical properties, and 75 instances with up to 40 jobs were tested. However, the complexity of the considered problem was not demonstrated and the proofs of the properties were not given in [3].

This paper is a natural extension of our previous work on the following aspects: 1) the complexity of the considered problem is demonstrated; 2) the formulated bi-objective model is transformed into an equivalent simplified one *via* new analytical properties and search space is greatly reduced; 3) an exact ε-constraint method is adapted to generate the Pareto front for the problem, which is accelerated by new proposed properties; 4) a fuzzy-logic-based method is employed to recommend a preferable Pareto optimal solution for decision makers; and 5) 475 instances with up to 650 jobs (65 batches) are generated to evaluate the effectiveness and efficiency of the proposed method and parameter sensitivity analysis is conducted.

The remainder of this paper is organized as follows. Section 2 describes the problem, formulates it mathematically, analyzes its computational complexity, and proposes a simplified mix-integer linear program (MILP) based on analytical properties. In Section 3, an exact ε-constraint method is adapted for generating the Pareto front of the considered problem. Subsequently, a fuzzy-logic-based approach is employed to recommend a preferable solution for decision makers. Section 4 reports the computational results. Section 5 concludes our work and discusses future directions.

2. Problem formulation and complexity demonstration

2.1. Problem description and formulation

A bi-objective single-machine batch scheduling problem under TOU policy can be represented as $TOU, 1|B|C_{\max}, EC$ by using the three-field notation of Graham [8]. The problem can be described as follows:

A given set of n jobs $J = \{1, 2, \ldots, n\}$ is to be processed on a single batch processing machine within a scheduling horizon I of m periods, $I = \{1, 2, \ldots, m\}$. Each job $j, \forall j \in J$, is nonresumable and has a processing time p_j. Any p_j is less than the length of any period i, denoted by T_i; *i.e.*, $T_i \gg p_j, \forall j \in J, \forall i \in I$. Without loss of generality, we assume that the jobs are numbered in nonincreasing order of the processing times; *i.e.*,

$$p_1 \geq p_2 \geq \cdots \geq p_n.$$

The jobs can be regrouped to k (to be optimized) batches and each batch can contain at most C jobs. Therefore, we must have $\lceil n/C \rceil \leq k \leq n$ [12]. The processing time of a batch is determined by the longest processing time of the jobs in the batch.

In the paper, the periods can be defined by electricity prices or work shifts. However, with the job nonresumable assumption in multiple periods, the processing of a batch should be completed before the end of a period or it must wait the beginning of another period. In the case where the periods are defined by electricity prices, the above assumption may lead that a batch waits a new electricity price period to be processed. This is not appropriate and appreciated in manufacturing industry, while the case that each job has to be completed in a single work shift exists in some production environments, for example, French Atomic Energy Industry, CEA. Therefore, a work shift is regarded as a period and its average unit electricity cost is implemented here. One work day is often composed of two or three work shifts according to the types of products, so-called two-shift and three-shift, respectively. The average unit electricity cost $E_i, \forall i \in I$ for each work shift can be calculated according to the tariffs information in Figure 1. For example, for a three-shift in a work day: 8h–16h, 16h–0h, 0h–8h, the corresponding unit electricity costs are as follows, $E_{8\mathrm{h}-16\mathrm{h}} = (11.4*3+14.0*5)/8(¢/\mathrm{kWh})*1(\mathrm{kWh/h}) = 13.0250$ ¢/h, $E_{16\mathrm{h}-0\mathrm{h}} = 9.4125$ ¢/h, and $E_{0\mathrm{h}-8\mathrm{h}} = 8.1625$ ¢/h. This problem can be easily extended to single-machine batch scheduling problem with unavailability periods.

Before formulating the problem, the parameters and decision variables are summarized as follows:

Parameters:

J: set of all jobs, *i.e.*, $J = \{1, 2, \ldots, n\}$;
C: capacity of a batch;
B: set of batches, *i.e.*, $B = \{1, 2, \ldots, k\}$;
p_j: processing time of job j, $\forall j \in J$;
I: set of time periods on the planning horizon, *i.e.*, $I = \{1, 2, \ldots, m\}$;
t_i: ending time of period i, $\forall i \in I$;
T_i: duration of period i, $\forall i \in I$, in which $T_i = t_i - t_{i-1}$;
E_i: unit electricity cost of period i, $\forall i \in I$.

Decision Variables:

k: number of the batches;
x_{jbi}: $x_{jbi} = 1$, if job j is assigned to batch b and processed in period i; otherwise $x_{jbi} = 0$; $\forall j \in J$, $\forall b \in B$, $\forall i \in I$;
y_{bi}: $y_{bi} = 1$, if batch b is assigned to period i; otherwise $y_{bi} = 0$; $\forall b \in B$, $\forall i \in I$;
z_i: $z_i = 1$, if at least one batch is assigned to period i; otherwise $z_i = 0$; $\forall i \in I$;
P_{bi}: $P_{bi} = P_b = \max\{p_j \mid j \in b\}$, if batch b is processed in period i, where P_b is the processing time of the longest job in the batch, and otherwise $P_{bi} = 0$; $\forall b \in B$, $\forall i \in I$.

In the paper, the number of batches k is initially considered as its upper bound n. A batch is opened if there is at least one job allocated to the batch. On the contrary, a batch is closed without any job and its corresponding processing time equals to 0, *i.e.*, $P_b = 0$. Now the considered problem can be formulated as the following bi-objective MILP model $\mathcal{P}$ [3].

$$\mathcal{P}: \ \min f_1 = C_{\max} \tag{2.1}$$

$$\min f_2 = EC = \sum_{i=1}^{m} E_i \sum_{b=1}^{n} P_{bi} \tag{2.2}$$

$$s.t. \ \sum_{i=1}^{m}\sum_{b=1}^{n} x_{jbi} = 1, \forall j \in J \tag{2.3}$$

$$\sum_{i=1}^{m} y_{bi} = 1, \forall b \in B \tag{2.4}$$

$$\sum_{j=1}^{n} x_{jbi} \leq C y_{bi}, \forall b \in B, \forall i \in I \tag{2.5}$$

$$x_{jbi} p_j \leq P_{bi}, \forall j \in J, \forall b \in B, \forall i \in I \tag{2.6}$$

$$\sum_{b=1}^{n} P_{bi} \leq T_i z_i, \forall i \in I \tag{2.7}$$

$$t_{i-1} z_i + \sum_{b=1}^{n} P_{bi} \leq C_{\max}, \forall i \in I \tag{2.8}$$

$$x_{jbi}, y_{bi}, z_i \in \{0,1\}, \forall j \in J, \forall b \in B, \forall i \in I \tag{2.9}$$

$$P_{bi} \geq 0, C_{\max} \geq 0 \tag{2.10}$$

Objective (2.1) is to minimize the makespan $C_{\max}$, which is the completion time of the last batch. Objective (2.2) is to minimize the total electricity costs EC on the horizon I. Constraint (2.3) ensures that job j, $\forall j \in J$, is assigned to only one batch and one period. Constraint (2.4) guarantees that each batch $b, \forall b \in B$, is processed in only one period. Constraint (2.5) aims to restrict the number of jobs assigned to any batch should not exceed the batch capacity C, and any job $j, \forall j \in J$, cannot be assigned to period i if its corresponding batch is not processed in this period. Constraint (2.6) determines the batch processing time. Constraint (2.7) ensures that the total processing time of batches in period $i, \forall i \in I$, should not exceed its duration, and $z_i = 1$ if there is at least one batch assigned to period i. Constraint (2.8) defines the makespan $C_{\max}$. Constraint (2.9) and (2.10) enforce the restrictions on decision variables.

It is worth pointing out that this study mainly focuses on $TOU, 1|B|C_{\max}, EC$ with multiple periods, *i.e.*, $m \geq 2$. Because the case with single period reduces to a classic single-machine batch scheduling problem with release time $r_j = 0$ to minimize the makespan, *i.e.*, $1|B|C_{\max}$ [23], which can be solved in polynomial time.

2.2. Complexity and properties of $TOU, 1|B|C_{\max}, EC$

As mentioned above, the number of batch k equals to the number of jobs n in model $\mathcal{P}$. we can observe that the solution search space of the model is very large, since it is already time consuming even k is set as its lower bound $\lceil n/C \rceil$ in [3]. This subsection is devoted to reducing the search space *via* problem property analysis. We show that the formation of batches can be solved independent of the scheduling of batches, with two objectives we consider in this paper.

In the remainder, let $w(W)$ and $P(W)$ denote the least indexed (hence the longest) job and the processing time of the batch made of set of jobs W; *i.e.*

$$w(W) = \min(W), \quad P(W) = \max_{j \in W} p_j = p_{w(W)}$$

A solution of the problem is uniquely defined by $(k, \{W_b, 1 \le b \le k\}, \{\tau_b, 1 \le b \le k\})$, where k is the number of batches, W_b and τ_b are the set of jobs involved in the batch b and the period the batch is processed, respectively.

We consider in particular those solutions where the batches are formed with a so-called LPT-based method. In this method, any job j with $(b-1)C < j \le bC$ and $1 \le b \le \lceil n/C \rceil - 1$ is put into batch b, and the remaining jobs, to batch $\lceil n/C \rceil$. Let k^* and W_b^* denote the number of batches obtained with this method and the set of jobs contained in the batch b ($1 \le b \le k^*$), respectively. We have

$$\begin{aligned} k^* &= \lceil n/C \rceil \\ W_b^* &= \{(b-1)C+1, (b-1)C+2, \dots, bC\}, \quad b = 1, 2, \dots, k^*-1 \\ W_{k^*}^* &= \{(k^*-1)C+1, (k^*-1)C+2, \dots, n\} \end{aligned}$$

By the construction, we have

$$w(W_b^*) = (b-1)C+1 \tag{2.11}$$
$$P(W_b^*) = p_{(b-1)C+1} \tag{2.12}$$

The following theorem shows that we only need to consider such solutions in order to find the Pareto front.

Theorem 2.1. *Any solution in which the batches are different from those formed with the LPT-based method is* (*at least weakly*) *dominated.*

Proof. Consider a feasible solution $(\hat{k}, \{\hat{W}_b, 1 \le b \le \hat{k}\}, \{\hat{\tau}_b, 1 \le b \le \hat{k}\})$ in which the batches are different from those formed with the LPT-based method. Obviously, we must have

$$\hat{k} \ge \lceil n/C \rceil = k^* \tag{2.13}$$

In other words, there are at least as many batches as those formed with the LPT-base method.

Without loss of generality, we renumber the batches in an increasing order of $w(\hat{W}_b)$'s; *i.e.*,

$$w(\hat{W}_1) < w(\hat{W}_2) < \cdots < w(\hat{W}_{\hat{k}})$$

As a consequence, we have

$$w(\hat{W}_b) = \min\{j \in \hat{W}_\beta | b \le \beta \le \hat{k}\} \tag{2.14}$$

Considering the fact that

$$\min(S) \le n - |S| + 1 \quad \forall S \subseteq \{1, 2, \dots, n\},$$

(2.14) implies, for any b such that $1 \le b \le \hat{k}$,

$$\begin{aligned} w(\hat{W}_b) &\le n - \sum_{\beta=b}^{\hat{k}} |\hat{W}_\beta| + 1 \\ &= \sum_{\beta=1}^{b-1} |\hat{W}_\beta| + 1 \\ &\le \sum_{\beta=1}^{b-1} C + 1 \\ &= (b-1)C + 1 \end{aligned}$$

which implies that

$$P(\hat{W}_b) = p_{w(\hat{W}_b)} \ge p_{(b-1)C+1} = P(W_b^*), \quad 1 \le b \le k^* \tag{2.15}$$

In other words, the processing times of the batches are at least as long as those formed with the LPT-based method.

Construct a new solution by removing batches $k^*+1, \ldots, \hat{k}$, if any, and replacing each batch $\hat{W}_b$ $(1 \leq b \leq k^*)$ by the corresponding one formed with the LPT-based method (*i.e.*, batch W_b^*), without changing starting time. In other words, consider solution $(k^*, \{W_b^*, 1 \leq b \leq k^*\}, \{\hat{\tau}_b, 1 \leq b \leq k^*\})$. Relation (2.15) implies that this new solution is also feasible. Furthermore, due to the fact that some batches are removed and the processing times of the remaining batches are reduced, neither the electrical consumption costs nor the makespan is increased, which means that the initial solution is (at least weakly) dominated by the new one. $\square$

As a consequence, by considering batches formed with the LPT-method as new jobs, the problem is transformed into a classical production scheduling problem without batching machine. Due to the fact that each batch (new job) should be entirely executed in one period, these new jobs are non-preemptive. There is an (infinitely short) unavailability period between two successive periods. The latter problem has been proved to be NP-hard in the strong sense, even when the single objective is to minimize the makespan. Hence, we have the following theorem.

Theorem 2.2. *The $TOU, 1|B|C_{\max}, EC$ is strongly NP-hard.*

2.3. An equivalent simplified model

According to Theorem 2.1, we can focus on scheduling problems of the batches. Therefore, the decision variables can be restricted to y_{bi}' and z_i's. And $P_b = p_{(b-1)C+1}$ with the LPT-based method. The initial model can be simplified into the following model $\mathcal{P}'$.

$$\mathcal{P}': \quad \min f_1 = C_{\max}$$

$$\min f_2 = EC = \sum_{i=1}^{m} \sum_{b=1}^{k^*} E_i P_b y_{bi} \tag{2.16}$$

$$\text{s.t.} \sum_{i=1}^{m} y_{bi} = 1, \forall b \in B^* \tag{2.17}$$

$$\sum_{b=1}^{k^*} P_b y_{bi} \leq T_i z_i, \forall i \in I \tag{2.18}$$

$$t_{i-1} z_i + \sum_{b=1}^{k^*} P_b y_{bi} \leq C_{\max}, \forall i \in I \tag{2.19}$$

$$y_{bi}, z_i \in \{0, 1\} \tag{2.20}$$

where $B^* = \{1, 2, \ldots, k^*\}$ denotes the set of batches formed with the LPT-based method. Constraint (2.17) ensures that a formed batch $b, \forall b \in B^*$ is allocated to exactly one period. Constraint (2.18) guarantees that the total processing time of the batches in period i does not exceed its duration and $z_i = 1$ if there is at least one batch allocated to period i, $\forall i \in I$. Constraint (2.19) restricts the makespan. Constraint (2.20) specifies binary restrictions on the variables. Since part of variables and constraints are removed, the search space for Pareto optimal solutions of the initial problem is significantly reduced.

3. Solution method

In this section, we first present the principles of multi-objective optimization and the ε-constraint method. Then, an exact ε-constraint method is adapted to find the Pareto front for $TOU, 1|B|C_{\max}, EC$. Finally, a fuzzy-logic-based approach is employed to recommend a preferable solution for decision makers.

3.1. Multi-objective optimization and the principle of ε-constraint method

In general, a multi-objective optimization problem (MOOP) can be represented as follows:

$$\min\{f_1(x), f_2(x), \ldots, f_m(x)\}, \text{s.t. } x \in \chi$$

where m objectives have to be optimized simultaneously, and χ represents the $feasible\ solution\ space$. Generally, due to the conflicting nature of the objectives, there exists no optimal solution for all the objectives but a set of Pareto optimal solutions. x^* is called a Pareto optimal solution if and only if no $x \in \chi$ exists such that $f_i(x) \leq f_i(x^*)$ for $i \in \{1, 2, \ldots, m\}$ with at least one inequality being strict [25]. The objective vector of a Pareto optimal solution x^*, *i.e.*, $\{f_1(x^*), f_2(x^*), \ldots, f_m(x^*)\}$ is called a non-dominated point. All the non-dominated points constitute the Pareto front. Two particular points, namely, Ideal and Nadir points, define the lower and upper limits of objective values of the Pareto front, respectively. For a bi-objective optimization problem (BOOP), the two points can be defined as follows [2]:

Definition 3.1. The Vector (f_1^I, f_2^I) with $f_1^I = \min\{f_1(x), x \in \chi\}$, and $f_2^I = \min\{f_2(x),\ x \in \chi\}$, denotes the Ideal point; and the Vector (f_1^N, f_2^N) with $f_1^N = \min\{f_1(x) : f_2(x) = f_2^I, x \in \chi\}$, and $f_2^N = \min\{f_2(x) : f_1(x) = f_1^I, x \in \chi\}$, denotes the Nadir point.

Based on the above definition, we have

Definition 3.2. The Vector (f_1^I, f_2^N) and (f_1^N, f_2^I) are two extreme points on the Pareto front.

The *weighted-sum method* and the *ε-constraint method* are the two most popular methods for exactly solving MOOPs. Compared with the former, the latter can avoid several drawbacks: 1) the former may be time-consuming due to large number of redundant runs caused by inappropriate weights; 2) a part of non-dominated points cannot be obtained when the Pareto curve is non-convex [4]; and 3) the weighted-sum method is not appropriate to the case where the linear combination is not suitable for integrating different objectives into a single one. Moreover, since the ε-constraint method was introduced by Haimes *et al*. [10], it has been successfully used to solve many BOOPs, *e.g.*, [2, 3, 16, 19, 30, 34]. These successful applications motivate us to apply it to solve our problem.

The ε-constraint method aims to optimize a single preferred objective function while formulating the other objectives as constraints, called ε-constraints. For the bi-objective case, the ε-constrained problem can be illustrated as follows if the first objective is considered as the preferred one:

$$\mathcal{P}(\varepsilon) : \min f_1(x), \text{s.t. } f_2(x) \leq \varepsilon, x \in \chi$$

where $\varepsilon \in [f_2^I, f_2^N]$. A widely used way for determining the values of ε is to uniformly divide the interval of ε into a number of subintervals and take each subinterval's upper limit as the value of ε. Such an ε-constraint method is referred to as *equidistant ε-constraint method*. Due to its simplicity of implementation, it has been used in our previous work [3] and many other papers, *e.g.*, [14, 21, 34]. However, this method cannot guarantee that all the obtained solutions are non-dominated and the Pareto front is found. In this paper, an exact ε-constraint method introduced by Bérubé *et al*. [2] is adapted to generate the Pareto front, which solves a sequence of ε-constrained problems based on a gradual reduction of the values of ε's. Its framework is shown as follows.

Algorithm 1. Exact ε-constraint method for the Pareto front of BOOPs.

1. Compute $f^I = (f_1^I, f_2^I)$ and $f^N = (f_1^N, f_2^N)$.
2. Set $\mathcal{F}' = \{(f_1^I, f_2^N)\}$, $\varepsilon_l = f_2^N - \triangle$ ($\triangle$ is set as the minimum unit value of f_2 [30]).
3. While $\varepsilon_l \geq f_2^I$, do:
   3.1. Solve problem $\mathcal{P}(\varepsilon_l)$ to optimality, obtain an optimal solution x^* and add $f(x^*)$ to $\mathcal{F}'$.
   3.2. Reset $\varepsilon_l = f_2(x^*) - \triangle$.
4. Remove dominated points from $\mathcal{F}'$ if existing and obtain the Pareto front $\mathcal{F}$.

3.2. Exact ε-constraint method for $TOU, 1|B|C_{\max}, EC$

In this subsection, an exact ε-constraint method is adapted to obtain the Pareto front for $TOU, 1|B|C_{\max}, EC$. Since this study mainly focuses on minimizing electricity costs under TOU policy, the total electricity cost is considered as the preferred objective. With the ε-constraint method, $TOU, 1|B|C_{\max}, EC$ can be transformed into the following ε-constrained problem:

$$\mathcal{P}_E(\varepsilon): \qquad \min f_2 = EC = \sum_{i=1}^{m} E_i \sum_{b=1}^{k^*} P_b y_{bi}$$

s.t. constraints (2.17), (2.18), (2.20), and

$$t_{i-1} z_i + \sum_{b=1}^{k^*} P_b y_{bi} \le \varepsilon, \quad \forall i \in I, \tag{3.1}$$

where ε denotes a given upper bound of $C_{\max}$.

It is easy to see that $\mathcal{P}_E(\varepsilon)$ is strongly NP-hard. In fact, $\mathcal{P}_E(\varepsilon)$ involves a constraint requiring that the makespan be not larger than ε. It is therefore equivalent to minimizing the makespan from a computational complexity point of view. Since the latter problem is known to be strongly NP-hard, as explained above Theorem 2.2, $\mathcal{P}_E(\varepsilon)$ is also strongly NP-hard.

From now on, we adapt the exact ε-constraint method to yield the exact Pareto front for the considered problem. To be specific, we first determine the lower and upper limits of the Pareto front by the Ideal and Nadir points; then we define the minimum unit value of $C_{\max}$.

3.2.1. Computation of Ideal and Nadir points

According to the framework of the exact ε-constraint method, we first need to compute the Ideal and Nadir points by exactly solving the following four single-objective optimization problems by Definitions 3.1.

$$\mathcal{P}^I_{C_{\max}}: \qquad C^I_{\max} = \min f_1 = C_{\max}$$

s.t. constraints (2.17)-(2.20)

$$\mathcal{P}^I_{EC}: \qquad EC^I = \min f_2 = EC = \sum_{i=1}^{m} E_i \sum_{b=1}^{k^*} P_b y_{bi}$$

s.t. constraints (2.17), (2.18) and (2.20).

The problem $\mathcal{P}^N_{C_{\max}}$ is formed by adding to $P^I_{C_{\max}}$ constraint (3.2) that fixes the optimal objective value of the total electricity costs.

$$\mathcal{P}^N_{C_{\max}}: \qquad C^N_{\max} = \min f_1 = C_{\max}$$

s.t. constraints (2.17)–(2.20), and

$$\sum_{i=1}^{m} E_i \sum_{b=1}^{k^*} P_b y_{bi} = EC^I. \tag{3.2}$$

The problem $\mathcal{P}^N_{EC}$ is formed by adding to P^I_{EC} constraint (3.3) that fixes the optimal objective value of $C_{\max}$.

$$\mathcal{P}^N_{EC}: \qquad EC^N = \min f_2 = EC = \sum_{i=1}^{m} E_i \sum_{b=1}^{k^*} P_b y_{bi}$$

s.t. constraints (2.17), (2.18), (2.20), and

$$t_{i-1} z_i + \sum_{b=1}^{k^*} P_b y_{bi} = C^I_{\max}, \forall i \in I. \tag{3.3}$$

By Definition 3.2, $(C^I_{\max}, EC^N)$ and $(C^N_{\max}, EC^I)$ are extreme points on the Pareto front of $TOU, 1|B|C_{\max}, EC$.

3.2.2. Definition of parameter $\triangle$

In this study, parameter $\triangle$ should be set as the minimum unit value of $C_{\max}$ according to its definition in Algorithm 1. The objective function $C_{\max}$ has the following equivalent form: $C_{\max}=\max_{i\in I}\{t_{i-1}z_i+\sum_{b=1}^{k^*}P_b y_{bi}\}$. It is not difficult to find that the minimum unit value of $C_{\max}$ is the minimal unit value of t_{i-1} and P_b, $\forall i \in I$ and $\forall b \in B^*$. Hence, parameter $\triangle$ is set as the minimal unit value of t_{i-1} and P_b.

3.3. Selection of the most preferable Pareto optimal solution

Among all the obtained nondominated points, a decision maker may desire to select a preferable one. In this subsection, the *fuzzy-logic-based* approach [6] is employed to recommend a preferable solution, since it can not only take into account the preferences of the decision maker, but also indicate the optimality degree each obtained nondominated point for each objective.

With the fuzzy-logic-based approach, the *membership function* $\delta_i(f_i^s)$, which represents the optimality degree of the sth Pareto optimal solution for the ith objective function, is presented as follows [6]:

$$\delta_i(f_i^s)=\begin{cases}1, & \text{if } f_i^s \le f_i^I \\ \frac{f_i^N-f_i^s}{f_i^N-f_i^I}, & \text{if } f_i^I < f_i^s < f_i^N,\ 1\le i\le m, 1\le s\le S \\ 0, & \text{if } f_i^s \ge f_i^N\end{cases} \tag{3.4}$$

where f_i^I and f_i^N represent the lower and upper limits of the ith objective function, respectively, and f_i^s expresses the value of the ith objective of the sth Pareto solution. S denotes the total number of Pareto solutions.

On the basis of the membership functions $\delta_i(f_i^s)$, the *membership degree* δ^s of the sth solution can be calculated as follows [6]:

$$\delta^s=\frac{\sum_{i=1}^m \omega_i\delta_i(f_i^s)}{\sum_{i=1}^m \omega_i} \tag{3.5}$$

where ω_i denotes the weight of objective i. It can be determined according to the preferences about the objectives of the decision maker. The most preferable solution is the one giving the maximum value of δ^s.

3.4. Overall algorithm for $TOU, 1|B|C_{\max}, EC$

To sum up, the algorithm to obtain the Pareto front and recommend the most preferable solution for single-machine batch scheduling under TOU policy is illustrated as follows.

Algorithm 2. Exact ε-constraint and fuzzy-logic combined method for $TOU, 1|B|C_{\max}, EC$.

1. Transform model $\mathcal{P}'$ into $\mathcal{P}_E(\varepsilon)$;
2. Obtain $f^I=(C_{\max}^I, EC^I)$ and $f^N=(C_{\max}^N, EC^N)$ by exactly solving $\mathcal{P}_{C_{\max}}^I$, $\mathcal{P}_{EC}^I$ and $\mathcal{P}_{C_{\max}}^N$, $\mathcal{P}_{EC}^N$;
3. Set $\mathcal{F}'=\{(C_{\max}^N, EC^I)\}$, $\varepsilon_l=C_{\max}^N-\triangle$($\triangle$ is set to the minimal unit value of t_{i-1} and P_b);
4. While $\varepsilon_l \ge C_{\max}^I$, do:
   4.1. Solve problem $\mathcal{P}_E(\varepsilon)$ exactly, obtain the optimal solution and its corresponding objective vector $(C_{\max}(\varepsilon), EC(\varepsilon))$;
   4.2. Reset $\varepsilon_l=C_{\max}(\varepsilon)-\triangle$;
5. Remove dominated points from $\mathcal{F}'$ if any and obtain the Pareto front $\mathcal{F}$;
6. Calculate the membership function $\delta_i(f_i^s)$ and membership degree δ^s;
7. Recommend a preferable solution.

TABLE 1. Computational results for the instances in [3].

Set	n	m	F'	F	CT'	CT	CT/CT'
1	20		3.0	3.0	2.442	0.246	0.101
2	25		6.2	6.2	6.123	0.347	0.057
3	30	3	4.6	4.6	9.900	0.328	0.033
4	35		7.0	7.0	22.583	0.658	0.029
5	40		8.8	9.0	194.547	0.797	0.004
6	20		3.0	3.0	4.827	0.278	0.058
7	25		6.2	6.2	7.530	0.449	0.060
8	30	4	11.2	11.2	9.860	0.747	0.076
9	35		16.2	16.2	30.387	1.320	0.043
10	40		17.2	18.2	536.743	1.526	0.003
11	20		8.6	8.6	5.333	0.446	0.084
12	25		14.8	14.8	8.373	0.983	0.117
13	30	5	15.8	16.0	10.193	1.161	0.114
14	35		18.0	24.2	30.857	1.912	0.062
15	40		16.4	26.4	675.913	2.172	0.003
Average			10.467	11.640	103.707	0.891	0.009

4. Computational results

In this section, the performance of the proposed algorithm is evaluated by 475 randomly generated instances (95 sets × 5 instances) in which 15 sets were proposed by [3]. The proposed algorithm is implemented in Visual C++ embedded with commercial optimization software CPLEX 12.4. All the single-objective optimization problems, *i.e.*, $\mathcal{P}^I_{C_{\max}}$, $\mathcal{P}^I_{EC}$, $\mathcal{P}^N_{C_{\max}}$, $\mathcal{P}^N_{EC}$ and $\mathcal{P}_E(\varepsilon)$, are solved by CPLEX in default setting and CPLEX is allowed to run until the optimal solution is output. All the numerical experiments are performed on a Personal Computer with 1.7 GHz processor and 4.0 GB RAM in windows seven environment. The computational time of each instance is limited to 18 000 s.

4.1. Comparison with the work [3]

In order to evaluate the performance of the proposed algorithm, we first test the instances in [3] and compare the computational results with that reported in [3] in terms of solution quality (number of nondominated points) and computational time. Let F and F' represent the number of nondominated points found by the proposed exact ε-constraint method and that by the equidistant ε-constraint method proposed in [3]. Besides, let CT and CT' denote the computational time by the proposed method and that by the method proposed in [3], respectively. Table 1 presents the results.

It can be observed from Table 1 that the average number of non-dominated points obtained by the equidistant ε-constraint method, *i.e.*, F', is less than that obtained by the proposed exact ε-constraint method, *i.e.*, F. It means that the equidistant ε-constraint method fails to obtain the Pareto front. This demonstrates that the proposed method in this paper is more effective than that in [3] in terms of solution quality.

In terms of computational time, we can see that CT' varies from 2.44 to 675.91 s, while CT varies from 0.25 to 2.17 s. The proposed method spends much less time than the method proposed in [3]. This indicates that the proposed algorithm is much more efficient than that in [3] in terms of computational time. This may be because new derived properties in this paper significantly reduce the search space for Pareto optimal solutions.

In summary, the proposed method in this paper dramatically outperforms the method proposed in [3] in terms of both solution quality and computational time.

TABLE 2. Computational results for the instances with $E_i = \{30, 15, 5\}$, $r_m \in [0.6, 1.0]$, $C = 10$, $p_j \in (100, 200]$.

Set	n	m	F	CT(s)	$\overline{CT}$	R_{EC}(%)	$IN_{C_{\max}}$(%)
16	50	2.0	12.0	1.122	0.094	19.79	21.88
17	60	2.6	16.0	2.114	0.147	21.03	15.02
18	70	3.0	34.8	4.942	0.142	36.55	28.85
19	80	3.0	43.4	7.197	0.166	22.67	16.81
20	90	3.8	45.2	8.569	0.186	41.62	28.51
21	100	4.0	61.0	10.127	0.166	32.51	20.09
22	110	4.2	89.6	14.730	0.163	24.89	16.10
23	120	4.6	76.2	15.242	0.198	22.91	14.78
24	130	5.0	120.2	27.363	0.228	26.02	15.26
25	140	5.2	148.2	43.454	0.318	21.80	15.68
26	150	6.0	211.4	80.817	0.387	29.48	23.90
27	160	6.6	246.6	215.775	0.899	32.19	25.11
28	170	6.6	253.0	1461.130	6.318	21.47	16.52
29	180	7.0	320.0	3169.854	9.799	30.79	22.73
30	190	7.6	304.2	6996.100	23.836	29.39	22.04
31	200	7.8	353.4	16390.878	45.039	22.07	16.81
32	210	8.4	–	–	–	–	–
Average			144.988	1777.831	5.473	26.78	19.71

4.2. Larger-size instances

To further evaluate the effectiveness of the proposed algorithm, we test our algorithm by 80 new larger-size problems sets. The instances are randomly generated in a similar way with [3]. The processing times $p_j, \forall i \in J$, are randomly and uniformly generated from (100, 200] and (50, 100] respectively. Batch capacity C is set as 10 and 5, respectively. The duration of each period $T_i, \forall i \in I$, is taken as 480, which corresponds to one period of three work shifts in each day, *i.e.*, 8*60 min. The number of periods m is set as $r_m \max_{j \in J}\{p_j \times k^*/T_i\}$, where r_m is a given number, which is randomly generated from the interval [0.6, 1.0] in the default case to avoid overmuch idle periods in scheduling horizon and to make it closer to reality. Three different average unit electricity cost in a work day are set as 30, 15 and 5, respectively. Considering that the both objectives have importance in industrial production, the weights in fuzzy-logic-based method are set to be equal, *i.e.*, $\omega_1 = \omega_2 = 0.5$, to select the most preferable solution. In addition, to more comprehensively evaluate the proposed algorithm, the sensitivity analysis for input parameter r_m as well as $E_i, \forall i \in I$ is conducted.

Let $\overline{CT}$ and $(C^S_{\max}, EC^S)$ denote the average computational time of each nondominated point, *i.e.*, $\overline{CT} = CT/F$, and the objective vector of the selected preferable solution calculated by fuzzy-logic-based approach, respectively. To evaluate the electricity cost reducing effectiveness of the selected solution, we will compare its objective value vector with the vector $(C^I_{\max}, EC^N)$, where $C^I_{\max}$ can be regarded as the optimal makespan without constraint of electricity costs and EC^N is the corresponding cost. Thus, we calculate the reduced electricity costs R_{EC} as $(EC^N - EC^S)/EC^N$. Similarly, the increased makespan is calculated as $(C^S_{\max} - C^I_{\max})/C^I_{\max}$, denoted by $IN_{C_{\max}}$. With these notations, the computational results are reported in Tables 2–6 and Figures 2 and 3. Note that each value in these tables is the average value of five instances.

Table 2 reports the computational results for instances with n varying from 50 to 200 and average m varying from 2.0 to 7.8 in default case. From the table, we can see that the proposed exact ε-constraint method can find the Pareto front for all the instances with up to 200 jobs, *i.e.*, 20 batches, within 16 391 s with the average time being 1777.831 s. This work aims to obtain the Pareto front, *i.e.*, all the non-dominated points, with exact ε-constraint method. The computational time mainly depends on two factors: the actual number of non-dominated points of the instance, *i.e.*, F, and average computational time for solving each ε-constraint

TABLE 3. Computational results for the instances with $E_i = \{30, 15, 5\}$, $r_m \in [0.6, 1.0]$, $C = 5$, $p_j \in (100, 200]$.

Set	n	m	F	CT(s)	$\overline{CT}$	R_{EC}(%)	$IN_{C_{\max}}$(%)
33	50	4.0	74.4	11.616	0.156	35.51	21.52
34	60	5.6	136.8	24.231	0.181	35.17	31.28
35	70	6.0	277.8	63.971	0.229	38.74	35.34
36	80	6.2	239.8	226.190	1.003	28.47	22.00
37	90	6.8	276.6	2488.884	8.638	25.94	18.70
38	100	7.8	436.4	16954.372	38.851	20.46	17.09
39	110	8.2	–	–	–	–	–
Average			240.300	3294.877	8.176	30.71	24.32

TABLE 4. Computational results for the instances with $E_i = \{30, 15, 5\}$, $r_m \in [0.6, 1.0]$, $C = 10$, $p_j \in (50, 100]$.

Set	n	m	F	CT(s)	$\overline{CT}$	R_{EC}(%)	$IN_{C_{\max}}$(%)
40	50	1.2	6.8	0.494	0.062	13.33	23.02
41	100	2.2	229.4	21.576	0.092	26.82	30.16
42	150	3.0	280.6	43.826	0.156	30.96	24.09
43	200	4.0	370.8	207.734	0.557	39.67	25.32
44	250	4.2	101.8	205.057	0.676	9.14	5.92
45	300	5.0	104.2	55.265	0.543	7.81	4.51
46	350	7.0	617.0	505.571	0.813	34.46	26.55
47	400	7.4	502.2	1283.992	2.006	23.91	17.42
48	450	8.6	660.8	1424.866	2.496	25.20	21.13
49	500	9.4	554.4	2357.996	3.377	22.55	18.21
50	550	9.8	386.8	4066.906	6.750	15.72	12.65
51	600	11.4	674.4	11323.212	15.511	20.62	20.85
52	650	12.0	659.8	16353.410	25.084	17.90	15.81
53	700	13.2	–	–	–	–	–
Average			396.231	2911.531	4.471	22.16	18.89

problem, *i.e.*, $\overline{CT}$. It can be seen from Table 2 that CT rapidly increases with the problem size since both factors increase. Moreover, it is not difficult to find that the increase of CT is mainly caused by $\overline{CT}$ due to the NP-hard nature of the problem. Take sets 16 and 31 for example, $\overline{CT}$ increases 479.14 (45.039/0.094) times while the number of F only increases 29.45 (353.4/12.0) times. We note that because of the problem complexity, the proposed algorithm cannot obtain the Pareto front within 18000s when the number of jobs increases to 210.

In addition, we can see that R_{EC} ranges from 19.79% to 41.62% with its average value being 26.78%. In other words, the total electricity costs under the TOU policy can reduced from 19.79% up to 41.62% with appropriate scheduling. This shows that appropriate scheduling under TOU policy can offer great benefits to reduce energy costs for industrial users. $IN_{C_{\max}}$ varies from 10.09% to 28.85% with its average value being 19.71%, which means that the total electricity costs and the makespan are two conflicting objectives. However, note that R_{EC} is greater than $IN_{C_{\max}}$ for almost all the problem sets, which indicates that industrial users can benefit more than loss with the selected preferable solution.

Now we set the batch capacity as 5 and other parameters in Table 2 remain unchanged. Note that the optimal number of batches $k^* = \lceil n/C \rceil$ has increased due to the reduction of batch capacity C, which leads that it is necessary to increase the number of periods m such that all the jobs can be completed in the scheduling horizon. Thus, m is regenerated according to $r_m \max_{j \in J}\{p_j \times k^*/T_i\}$. The computational results are presented

TABLE 5. Comparison for sensitivity analysis of r_m $m = r_m \max_{j \in J}\{p_j \times k^*/T_i\}$.

Set	r_m	n	m	F	CT(s)	$\overline{CT}$	R_{EC}(%)	$IN_{C_{\max}}$(%)
54		90	3.6	45.0	8.321	0.183	31.03	21.73
55		100	4.0	65.0	12.942	0.200	33.50	20.79
56		110	4.2	91.0	19.332	0.210	23.81	15.44
57	[0.6, 1.0]	120	4.8	116.4	22.567	0.196	31.49	21.63
58		130	5.0	119.0	29.596	0.249	27.39	16.49
59		140	5.4	166.2	62.309	0.367	24.74	19.28
60		150	5.8	193.6	102.540	0.527	25.71	20.54
Average				113.743	36.801	0.276	28.24	19.41
61		90	4.0	64.4	9.578	0.149	49.55	34.36
62		100	4.8	106.8	17.086	0.161	32.89	28.87
63		110	5.0	134.0	21.403	0.160	35.27	28.05
64	[1.1,1.5]	120	5.8	197.0	34.427	0.175	46.17	46.50
65		130	7.0	220.6	54.315	0.252	35.40	33.26
66		140	7.4	248.4	67.485	0.273	26.07	22.69
67		150	8.4	322.6	104.918	0.321	16.05	11.93
Average				184.829	44.173	0.213	34.49	29.38
68		90	5.0	85.4	12.966	0.152	25.83	31.17
69		100	5.2	119.4	19.031	0.159	32.89	37.00
70		110	6.0	140.8	24.971	0.179	30.72	26.98
71	[1.6,2.0]	120	6.8	165.2	24.532	0.151	13.17	12.90
72		130	7.8	222.8	52.482	0.239	37.18	32.02
73		140	8.6	345.4	79.176	0.235	30.51	25.65
74		150	9.4	413.6	145.823	0.355	37.01	32.65
Average				213.229	51.283	0.210	29.62	28.34

in Table 3. By comparing the two Tables, it can be observed that for the instances with the same number of jobs, CT increases when the batch capacity reduces. This is because the number of batches increases. Take sets 16 and 33 for example, both sets have 50 jobs while their computational time are 1.122s and 11.616s, respectively. Moreover, we can find from Tables 2 and 3 that computational time of most instances with same number of batches are almost the same, such as sets 21 and 33, sets 27 and 36, *etc.*

The number of batches per work shift is relatively few due to the relatively large job processing time. To test the performance of the proposed algorithm for instances with more batches per work shift, we have additionally conducted the experiments on the instances with $p_j \in (50, 100]$. The computational results are reported in Table 4. It can be seen that the proposed algorithm is able to obtain the Pareto fronts for instances with up to 650 jobs, *i.e.*, 65 batches, and 12.0 periods within 16 354 s with average time being 2911.531 s. Due to the complexity of the problem, the proposed method is not able to obtain the Pareto front for the instances with 700 jobs within 18000 s. We can also find that CT rapidly increases with the number of jobs. For example, CT for set 40 is 0.494 s while for set 52 is 16 353.410 s. Similar to Table 2, the increase of CT is mainly caused by $\overline{CT}$ since it increases more times than that of F. For example, by comparing sets 52 and 40, we can see that $\overline{CT}$ increases 404.58 (25.084/0.062) times while F only increases 97.03 (659.8/6.8) times. Moreover, by comparing Table 4 with Table 2, it can be observed that for the instances with the same number of jobs, $\overline{CT}$ in Table 4 is less than that in Table 2. This may be because the instances in Table 2 have more periods. Besides, we can find that CT in Table 2 increases faster than that in Table 4. Take sets 16 and 31 in Table 2 and sets 40 and 50 in Table 4 for example, for sets 16 and 31 CT increases 14051.93 (15766.264/1.122) times when the number of jobs and periods increase from 50 to 200 and 2.0 to 7.8, respectively, while for sets 40 and 50 CT increases 8232.60 (4066.906/0.494) times when the number of jobs and periods increase from 50 to 550 and 1.2 to 9.8, respectively. This indicates that the proposed algorithm is more efficient for the instances with less job processing time.

TABLE 6. Comparison for sensitivity analysis of E_i

Set	E_i	n	m	F	CT(s)	$\overline{CT}$	R_{EC}(%)	$IN_{C_{\max}}$(%)
75		90	3.0	32.2	4.274	0.153	5.00	4.35
76		100	4.0	86.0	12.858	0.150	33.30	27.38
77		110	4.0	57.0	9.752	0.168	6.20	4.69
78	{20, 12, 5}	120	5.0	92.8	16.382	0.180	22.64	19.36
79		130	5.0	109.8	25.377	0.230	13.55	11.55
80		140	5.0	99.4	43.517	0.360	12.10	9.21
81		150	6.0	148.4	64.410	0.409	18.28	16.33
Average				89.371	25.224	0.236	15.87	13.27
82		90	3.0	21.0	2.705	0.129	3.82	2.58
83		100	4.0	68.4	11.928	0.175	35.21	21.58
84		110	4.0	64.0	11.353	0.178	17.62	10.35
85	{30, 15, 5}	120	5.0	113.0	23.687	0.210	37.85	25.62
86		130	5.0	124.4	29.671	0.239	28.09	17.25
87		140	5.0	123.0	36.473	0.296	17.77	10.59
88		150	6.0	217.2	92.663	0.441	29.06	23.40
Average				104.429	29.783	0.238	24.20	15.91
89		90	3.0	21.0	2.824	0.135	4.29	2.58
90		100	4.0	68.6	11.148	0.163	40.02	21.58
91		110	4.0	64.4	10.477	0.166	20.47	10.47
92	{50, 25, 5}	120	5.0	114.0	20.486	0.280	42.03	25.62
93		130	5.0	124.2	27.214	0.219	31.53	17.25
94		140	5.0	124.5	35.885	0.287	20.51	10.85
95		150	6.0	218.0	92.543	0.439	31.89	23.40
Average				104.957	28.654	0.227	27.25	15.96

To further evaluate the performance of the combined approach, sensitive analysis experiments for input parameters m and E_i are conducted. Since the performance of the instances with $p_j \in (100, 200]$ (Tab. 2) is worse than that with $p_j \in (50, 100]$ (Tab. 4), to better test the stability of the proposed algorithm, we conduct the sensitive analysis experiments based on the former job processing time generation scheme.

Table 5 reports the computational results for three scenarios regrading number of periods m. m is defined as $r_m \max_{j\in J}\{p_j \times k^*/T_i\}$. Parameter r_m is generated from [0.6, 1.0], which is regarded as the baseline. Other two cases r_m are generated from $[1.1, 1.5]$ and $[1.6, 2.0]$, respectively, and the other parameters remain unchanged.

From Table 5, we can see that for each type of r_m, the computational time CT and number of nondominated points F increase with n and m. Moreover, we can see in Figure 2 that the changing trends of CT and F more obviously for larger r_m. More precisely, we can observe that given the number of jobs n, the computational time increases with the number of periods m. Take sets 67 and 74 as an example, both sets have 150 jobs but different periods, while CT of set 74 is greater than that of set 67, which is mainly because that F increases from 322.6 to 413.6. This shows that the increasing of m adds the complexity of the problem, since more periods result in more nondominated points. On the other hand, we can also find that given m, CT increases with n. Take sets 63 and 58 for example, both of them have five periods, but CT are 21.403 and 29.596 s, respectively. This implies that the complexity of the problem increases with the number of jobs. Besides, by comparing R_{EC} and $IN_{C_{\max}}$ under three different scenarios, we can find that they slightly increases when r_m increases. This shows that more periods may result in more electricity costs reduction, but may incur a longer makespan.

Table 6 presents the results of sensitivity analysis of unit electricity cost E_i, which are set as {20, 12, 5}, {30, 15, 5} and {50, 25, 5}, respectively. The computational time of the three scenarios range between 4.274 and 64.410 s, 2.705 and 92.663 s, 2.824 and 92.543 s, respectively. Moreover, it can be found in Figure 3 that the changing trends of CT for the three scenarios are almost the same. Furthermore, the average computational time

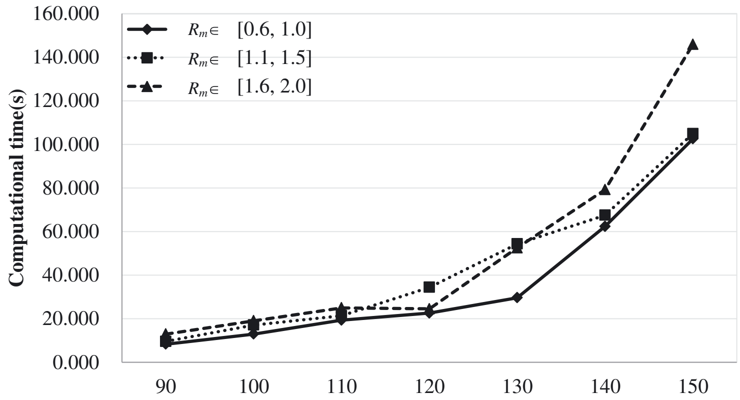

Figure 2. Computational time for sensitivity analysis of m. $m = r_m \max_{j \in J}\{p_j \times k^*/T_i\}$.

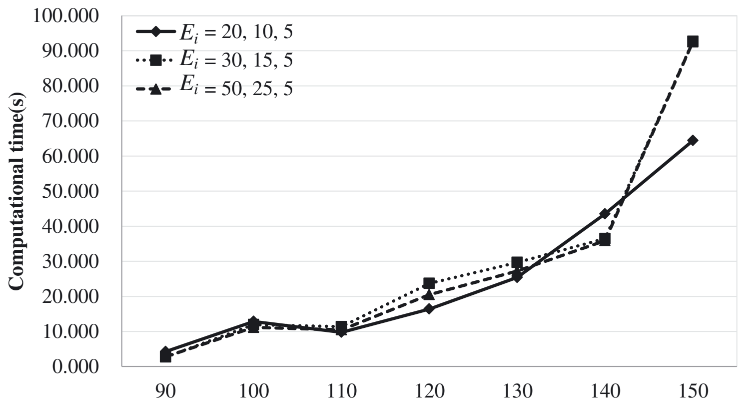

Figure 3. Computational time for sensitivity analysis of E_i.

of the proposed algorithm for all scenarios are 25.224 s, 29.783 s and 28.654 s, respectively, which are almost the same. These results indicate that our proposed algorithm is insensitive to the changes of unit electricity cost.

Besides, by comparing R_{EC} of the three scenarios, we can see that given the number of jobs and periods, in general the greater difference of unit electricity cost among different periods, the more electricity costs can be reduced. Take sets 81 and 95 for example, both problem sets have 150 jobs and six periods, set 81 reduces 18.28% electricity costs while the electricity costs in the latter set is reduced up to 31.89%. This demonstrates that industrial users may benefit more from the TOU policy when the differences of electricity prices among different periods are bigger.

5. Conclusion

In this paper, we have investigated a bi-objective single-machine batch scheduling problem under TOU electricity prices. The two objectives are to minimize the makespan and to minimize the electricity costs. New properties are firstly analyzed for the problem, which greatly reduce its search space for a Pareto optimal solution. With such properties, the problem is demonstrated to be NP-hard and a simplified MILP is proposed. Then, we propose an exact ε-constraint method to obtain its Pareto front and a fuzzy-logic-based approach is employed to help decision makers derive a most preferable solution. Computational results show that the proposed approach can efficiently solve instances with up to 650 jobs within reasonable time and obtain the Pareto front.

In future research, on one hand we may resort to efficient problem-specific heuristics to generate well-distributed nondominated points for larger-size problems within a shorter time. We will also extend our mathematical model to the case where the assumption of a job must be completed in single work shift is relaxed. On the other hand, we may extend our study from the following aspects: 1) from the perspective of energy saving, power off and speed scaling strategies can be introduced to reduce energy consumption and costs. For example, when the processing is completed in a period, the machine may be idle for a certain amount of time until the next processing begins, then the machine can be turned off with power off strategy to decrease the total energy consumption; 2) from the perspective of machine environment, this study can be extended from single machine to parallel and flow shop machines; and 3) from the perspective of job characteristics, we may extend this work to those considering non-identical job sizes and non-identical job release times. Finding effective and efficient approaches for these potential problems requires more work.

Acknowledgements. This work was partially supported by the National Natural Science Foundation of China, under Grants Nos. 71571061 and 71431003, the Cai Yuanpei Program between the French Ministries of Foreign and European Affairs and Higher Education and Research and the Chinese Ministry of Education, under Grant No. 27927VE.

References

[1] A. Pacific, Energy Research Centre APERC. Chapter 6: Apec energy demand and supply outlook, 5th edition. Asia-Pacific Economic Cooperation (2013) 70.

[2] J.-F. Bérubé, M. Gendreau and J.-Y. Potvin, An exact-constraint method for bi-objective combinatorial optimization problems: Application to the traveling salesman problem with profits. *Eur. J. Oper. Res.* **194** (2009) 39–50.

[3] Junheng Cheng, Feng Chu, Weili Xia, Jianxun Ding and Xiang Ling. Bi-objective optimization for single-machine batch scheduling considering energy cost. In *2014 International Conference on Control, Decision and Information Technologies (CoDIT)*. IEEE (2014) 236–241.

[4] I. Das and J.E. Dennis, A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. *Struct. Optim.* **14** (1997) 63–69.

[5] M.P. Lee, L. Medearis, P. Sporborg, M. Tita, D. Wight, A. Wilkerson, F. Kreger, V. Richardson, W. Gifford, C. Elsner, D. Kathan and R. Aldina, Assessment of demand response and advanced metering: Staff report 2012. Federal Energy Regulatory Commission, Washington (2012).

[6] M. Esmaili, H.A. Shayanfar and N. Amjady, Multi-objective congestion management incorporating voltage and transient stabilities. *Energy* **34** (2009) 1401–1412.

[7] K. Fang, N.A. Uhan, F. Zhao and J.W. Sutherland, Scheduling on a single machine under time-of-use electricity tariffs. *Ann. Oper. Res.* **238** (2016) 199–227.

[8] R.L. Graham, E.L. Lawler, J.K. Lenstra and A.R. Kan, Optimization and approximation in deterministic sequencing and scheduling: a survey. *Ann. Discrete Math.* **5** (1979) 287–326.

[9] H. Hadera and L. Harjunkoski, Continuous-time batch scheduling approach for optimizing electricity consumption cost. In vol. 32 of *23rd European Symposium on Computer Aided Process Engineering*. Elsevier (2013) 403.

[10] Y.Y. Haimes, L.S. Lasdon and D.A. Wismer, On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Trans. Syst. Man Cybernet.* (1971) 296–297.

[11] Cheng He, Xiumei Wang, Yixun Lin and Yundong Mu, An improved algorithm for a bicriteria batching scheduling problem. *RAIRO: OR* **47** (2013) 1–8.

[12] Yoshiro Ikura and M. Gimple, Efficient scheduling algorithms for a single batch processing machine. *Oper. Res. Lett.* **5** (1986) 61–65.

[13] C.-Y. Lee, R. Uzsoy and L.A. Martin-Vega, Efficient algorithms for scheduling semiconductor burn-in operations. *Oper. Res.* **40** (1992) 764–775.

[14] M. Leitner, I. Ljubić and M. Sinnl, Solving the bi-objective prize-collecting steiner tree problem with the ε-constraint method. *Electron. Notes Discrete Math.* **41** (2013) 181–188.

[15] H. Luo, B. Du, G.Q. Huang, Huaping Chen and Xiaolin Li, Hybrid flow shop scheduling considering machine electricity consumption cost. *Int. J. Prod. Econ.* **146** (2013) 423–439.

[16] G. Mavrotas, Effective implementation of the ε-constraint method in multi-objective mathematical programming problems. *Appl. Math. Comput.* **213** (2009) 455–465.

[17] C.A. Méndez, J. Cerdá, I.E. Grossmann, I. Harjunkoski and M Fahl, State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Comput. Chem. Eng.* **30** (2006) 913–946.

[18] J.-Y. Moon, K. Shin and J. Park, Optimization of production scheduling with time-dependent and machine-dependent electricity cost for industrial energy efficiency. *Int. J. Adv. Manufact. Technol.* **68** (2013) 523–535.

[19] A.K. Ojha and R.R. Ota, Multi-objective geometric programming problem with karushkuhnucker condition using ε-constraint method. *RAIRO: OR* **48** (2014) 429–453.

[20] C.N. Potts and M.Y. Kovalyov, Scheduling with batching: a review. *Eur. J. Oper. Res.* **120** (2000) 228–249.

[21] P. Reiter and W.J. Gutjahr, Exact hybrid algorithms for solving a bi-objective vehicle routing problem. *Central Eur. J. Oper. Res.* **20** (2012) 19–43.

[22] F. Shrouf, J. Ordieres-Meré, A. García-Sánchez and M. Ortega-Mier, Optimizing the production scheduling of a single machine to minimize total energy consumption costs. *J. Clean. Prod.* **67** (2014) 197–207.

[23] C.S. Sung and Y. Choung, Minimizing makespan on a single burn-in oven in semiconductor manufacturing. *Eur. J. Oper. Res.* **120** (2000) 559–574.

[24] Lixin Tang, Jiyin Liu, Aiying Rong, and Zihou Yang. A review of planning and scheduling systems and methods for integrated steel production. *Eur. J. Oper Res.* **133** (2001) 1–20.

[25] V. T'kindt and J.-C. Billaut, Multicriteria scheduling problems: a survey. *RAIRO: OR* **35** (2001) 143–163.

[26] R. Uzsoy, C.-Y. Lee and L.A. Martin-Vega, A review of production planning and scheduling models in the semiconductor industry part i: system characteristics, performance evaluation and production planning. *IIE Trans.* **24** (1992) 47–60.

[27] R. Uzsoy, C.-Y. Lee and L.A. Martin-Vega, A review of production planning and scheduling models in the semiconductor industry part ii: shop-floor control. *IIE Trans.* **26** (1994) 44–55.

[28] D.J. Van De Rzee, A. Van Harten and P.C. Schuur, Dynamic job assignment heuristics for multi-server batch operations-a cost based approach. *Int. J. Prod. Res.* **35** (1997) 3063–3094.

[29] Y. Wang and L. Li, Time-of-use based electricity demand response for sustainable manufacturing systems. *Energy* **63** (2013) 233–244.

[30] P. Wu, A. Che, F. Chu and M. Zhou, An improved exact ε-constraint and cut-and-solve combined method for biobjective robust lane reservation. *IEEE Trans. Intell. Transp. Syst.* **16** (2015) 1479–1492.

[31] R. Xu, H. Chen and X. Li, Makespan minimization on single batch-processing machine *via* ant colony optimization. *Comput. Oper. Res.* **39** (2012) 582–593.

[32] R. Xu, H. Chen and X. Li, A bi-objective scheduling problem on batch machines *via* a pareto-based ant colony system. *Int. J. Prod. Econ.* **145** (2013) 371–386.

[33] H. Zhang, F. Zhao, K. Fang and J.W. Sutherland, Energy-conscious flow shop scheduling under time-of-use electricity tariffs. *CIRP Ann. Manufact. Technol.* **63** (2014) 37–40.

[34] Z. Zhou, F. Chu, A. Che and M. Zhou, ε-constraint and fuzzy logic-based optimization of hazardous material transportation *via* lane reservation. *IEEE Trans. Intell. Transp. Syst.* **14** (2013) 847–857.