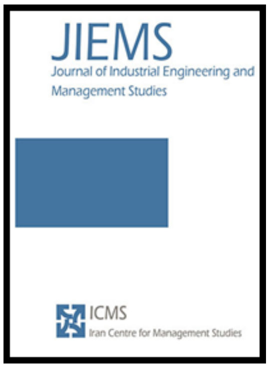
JIEMS

Journal of Industrial Engineering and Management Studies

Vol. 5, No. 2, 2018, pp. 97-112

DOI: 10.22116/JIEMS.2018.80687

www.jiems.icms.ac.ir



Investigation of the two-machine flow shop scheduling problem to minimize total energy costs with time-dependent energy prices

Amir Ebrahimzadeh Pilerood[1], Mehdi Heydari[1,*], Mohammad Mahdavi Mazdeh[1]

Abstract

In this paper, the problem of two-machine flow shop scheduling to minimize total energy costs under time-of-use tariffs is investigated. As the objective function of this study is not a regular measure, allowing intentional idle-time can be advantageous. So this study considers two approaches, one for non-delay version of the problem and the other one for a situation when inserting intentional idle time is permitted. A mixed integer linear programming is formulated to determine the timing of jobs in order to minimize total energy costs while idle time insertion is allowed. For the non-delay version of the problem, a branch-and-bound algorithm is presented. A lower bound and several dominance properties are used to increase the speed of the branch-and-bound algorithm. Computational experiments are also given to evaluate the performance of the algorithm. Based on results, the proposed algorithms can optimally schedule jobs in small size samples but by increasing the number of jobs from 15 and cost periods from 3, the performance of branch-and-bound has been decreased.

Keywords: Flow shop scheduling; mixed integer programming; branch-and-bound; time-of-use energy costs; idle time insertion.

Received: June 2018-08

Revised: July 2018-05

Accepted: September 2018-22

1. Introduction

All the manufacturing units and service providers will need energy to deliver their products and services. In the recent years, there has been a growing concern over energy consumption of traditional manufacturing units, and many countries are implementing different policies and strategies to control energy consumption. Besides using new equipment, production scheduling can play a key role in reducing the energy consumption. For example, in some countries such as Iran, time-of-use (TOU) tariffs are implemented in order to shift electricity use from peak hours to off-peak hours. Under this pricing strategy, costs of using energy are variable based on time of use.

* Corresponding author; Mheydari@iust.ac.ir

[1] Department of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran.

TOU tariffs provide an opportunity for electricity users to implement economic load dispatch (ELD), i.e., reduce electricity costs by cutting power loads during on-peak periods and shifting loads from on-peak to off-peak periods (Tan et al., 2016). Then by shifting highly energy consumed jobs to off-peak hours of the day, a manufacturer can play an essential role in saving energy for other users and at the same time, saving energy costs for itself.

In this paper, we are investigating the problem of two-machine flow shop scheduling to minimize the total energy costs (TEC) under TOU situation that will be addressed here as TFTT. The single machine version of this problem has been proved to be NP-hard by Fang et al. (2016).

It is important to note that TEC under TOU tariffs is not a regular measure, i.e. the measures that are non-decreasing in job completion times. So allowing intentional idle-time can be advantageous. Kanet and Sridharan (2000) defined an inserted idle time (IIT) schedule as a feasible schedule in which a machine is kept idle at a time when it could begin processing an operation.

Although IIT may improve non-regular measures in scheduling problems but there are some cases where idle time cannot be inserted in real life settings. Such cases are generally motivated by the prohibitively high cost of idling the machines, the technological infeasibility of idling them, or undesirable effects in certain production environments (see e.g. Józefowska, 2007 and Schaller and Valente, 2013).

In this paper the TFTT problem will be investigated under two scenarios. First, when an inserted idle time schedule is permitted, and the second with non-delay scheduling. By non-delay we mean operations on machines must be carried out without any inserted idle times which never permit a delay (via inserted idle time) when the machine becomes available and work is waiting. For the first case a new time-interval based mixed integer mathematical model is presented, while for the non-delay version of the problem a branch-and-bound algorithm is suggested. So, the main contribution of this work is two-fold. First, a new continuous-time mixed-integer linear programming (MILP) model is proposed for the problem. Second, a branch-and-bound algorithm is developed.

In the next section, a literature review is presented. In this section, we firstly review some most important energy related scheduling research and in the second part, scheduling research with concentration on time-dependent-costs, e.g. TOU prices, are introduced. In the third section, the problem TFTT is introduced and a small example is provided. In the fourth section a mixed integer linear programming is formulated for solving the problem while inserted idle time is permitted. In the fifth section, different dominance properties and special cases of the non-delay version of the problem are studied. In section six, procedure and elements of our suggested branch-and-bound is introduced which consists an approach for calculating lower bound. In section seven, computational results are presented. Finally, in section eight, and as conclusion some further studies are suggested.

2. Literature review

Actually most of the research about energy related scheduling are about machine on/off strategies. In these problems, by proposing a turn-on and turn-off scheduling framework, overall energy consumption can be reduced (Mouzon et al, 2007). For example, Mouzon and Yildrim (2008) analyzed a single machine problem which aims to minimize both the total tardiness and the total energy consumption. Liu et al. (2014) studied a bi-objective minimization of total weighted tardiness and total energy consumption in job shops. They considered different energy consumption rate depending on the jobs and the machines and tried to find the best solutions based on Pareto front. For solving the problem, they proposed a NSGA-II metaheuristic to gain optimal Pareto fronts.

When the on/off control framework is not applicable, an alternative for energy saving can be based on machine speed scaling. For example, Liu and Huang (2014) examined carbon footprint within the context of production scheduling. They considered a batch-processing machine scheduling problem to minimize both the total weighted tardiness and energy related criteria. Che et al. (2015) investigated a single machine scheduling problem to minimize energy consumption with bounded maximum tardiness. They assumed that energy consumption is associated with the processing speed and they proposed two MILP models to make two optimal decisions about job sequencing and speed choosing.

Zhang and Chiong (2016) proposed a multi-objective genetic algorithm to minimize the total energy consumption and the total weighted tardiness in job shops while energy consumption is based on speed scaling framework.

From the energy cost perspective, time-of-use (TOU) electricity prices is getting more attention. Under this pricing strategy, cost of energy will be different from time to time and finding the optimal decision will be completely different from the situation with static prices.

As one of the first studies in this regard, Wan and Qi (2010) presented the concept of "timeslot" in scheduling literature and that is when different time frames can have different costs whenever a job is scheduled on that time frame. Although they did not study the problem of energy cost, but they mentioned that TOU costs can be seen as a special application for their problem. They studied complexity of different classic scheduling problem when they are mixed with "timeslot costs" in order to minimize a combination of the classic objective functions and total slotcost minimization. They actually proved that for some of the problems such as combination of total flow time and total slotcost, the problem is NP-hard. Also some other research has been conducted based on the slotcosts concept (see Zhong and Liu (2012), Chen et al. (2015), Zhao et al. (2015)).

By considering the TOU concept in scheduling, Luo et al. (2013) studied the problem of Flexible Flow Shop scheduling with time-of-use tariffs to minimize both make span and total energy costs. In their problem, each machine consumes a specific amount of energy depending on their state and speed. A bi-objective Ant Colony algorithm is presented to solve the problem in large scale examples.

Moon et al. (2013) presented a metaheuristic algorithm to minimize two objectives of make span and total energy costs in the environment of parallel machine scheduling with TOU costs. They could minimize the energy costs by inserting idle time for machines and at the same time they were considering the trade-off between make span deterioration and energy costs improvement. Shrouf et al. (2014) considered a single machine scheduling problem, when machine could stay at three different states of process, idle and turned-off. Then they proposed a mathematical model and a metaheuristic algorithm to minimize the total energy costs with TOU costs. Later, Aghelinejad et al. (2016) proposed an improved version of Shrouf et al. (2014) mathematical model. They presented two new mathematical models to reduce total energy consumption cost of a single machine manufacturing system. The problem consists of optimizing simultaneously the processing of the jobs and utilization of the machine.

Zhang et al. (2014) studied a Flow Shop scheduling problem to find a Pareto front for total energy costs and total CO_2 emissions under TOU costs. They presented a time-indexed integer programming model and implemented their model on a case study with only one product. Actually their mathematical model is not flexible enough to schedule a variety of products and it can perform well for serial production planning of production types with specific number of throughput. Contrary to their research, we will introduce a new time-interval based mathematical model and a branch-and-bound algorithm for finding exact scheduling of various jobs. Gong et al. (2015) studied a single machine total energy costs minimization considering the TOU costs.

A mixed integer mathematical model is presented while make span is not allowed to exceed the desired level of decision maker.

Fang et al. (2016) studied the single machine scheduling problem with TOU costs to minimize total energy costs. In their problem under study, each job has a different resource consumption and they studied five classes of problems. One of the most important results of their study is proving that the non-preemptive version of the problem is NP-hard. Later Che et al. (2016) proposed a mathematical model and a greedy heuristic algorithm to minimize total energy costs for the same problem, also Ding et al. (2016) studied the same problem but in the parallel machine environment with two approaches: 1) Presenting a time-interval-based mixed integer linear programming 2) Using Dantzig-Wolfe technique and column generation algorithm.

Wang et al. (2017) studied a two-machine permutation flow shop scheduling problem to minimize the total electricity cost of processing jobs under time-of-use electricity tariffs. Two heuristic algorithms based on Johnson's rule and dynamic programming method are designed and they investigated how to find an optimal schedule using dynamic programming when the processing sequence is fixed.

Most of the studies in this field, investigated the single machine environment while the flow shop scheduling studies are rare. To best of our knowledge, this study is among the first studies that consider both approaches when an inserted idle time schedule is permitted, and the one with non-delay scheduling. In the next section, our suggested problem will be defined and formulated.

3. Problem definition

We are studying a two machine flow shop problem with J= (1,2,..., n) that is a given set of two-operation jobs available to be processed at time zero. Each job requires the first operation on machine 1 and the second operation on machine 2. At any time, each machine can process at most one job and each job can be processed on at most one machine. Once the processing of a job on a machine has started, it must be completed without interruption. Also, each job must be processed in the same order at every machine. Let $p_{j,1}$ and $p_{j,2}$ be the processing times of job j on operation 1 and 2 respectively.

On the other hand, the processing of jobs consumes energy resources and incurs corresponding resource cost, e.g. electricity costs. Let $q_{j,1}$ and $q_{j,2}$ be the resource demand for the first and the second operation of job j, respectively. The resource price varies in time horizon and obeys the time-of-use tariffs. A set of P consists of K time periods, is used to formulate the time dependent pricing scheme. Period k, $k \in P$ is characterized by its starting time t_k, duration a_k and resource price f_k. Main aim of the problem is to assign the jobs to the available cost periods with different resource prices in the time horizon $[0,T]$ so as to minimize the total resource cost required for processing them. In fact, when an inserted idle time schedule is permitted, the solution is to determine the start time and completion time of each operation while in the non-delay version of the problem a solution is a sequence of operations. As mentioned earlier the single machine version of this problem has been proven to be strongly NP-hard by Fang et al. (2016).

To clarify the problem of TFTT, let's consider a small example with six jobs and three cost periods. Suppose each job has a specific processing time and energy demand on each machine which is based on table 1. Table 2 gives the TOU tariffs throughout a day. Figure 1 shows one sequence of jobs and their corresponding energy cost (EC) in non-delay version of the problem.

Table 1. Processing time and energy demand of each job on each machine

Jobs	Processing time on the first machine	Processing time on the second machine	Energy demand on the first machine	Energy demand on the second machine
1	2	3	1	1.5
2	3	3	2	1.5
3	1	3	1.5	1.5
4	5	3	2	2.5
5	4	4	1	1.5
6	4	2	1.5	1.5

Table 2. TOU tariffs

Cost Period	Start time	Duration	Energy Price
1	0	10	1
2	10	6	4
3	16	8	2

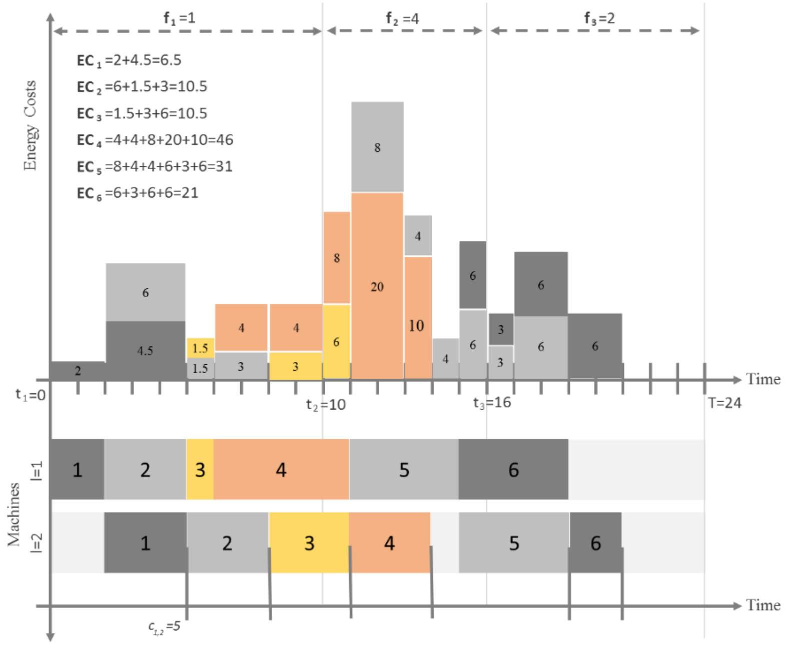


Figure 1. One sequence of jobs (lower part) and the corresponding energy cost (upper part)

It worth to mention that T is seen as an upper bound for make span of the problem. Because if there exist some sequences with completion time later than T, they will be infeasible. So T is a parameter which is defined by decision maker. In this paper T is a number more than C_u which is the upper bound of make span. Kim (1993) developed a procedure to generate a sequence with the maximum make span, which we show by C_u. Then in our problem it is assumed that $T \geq C_u$ and this implies that a feasible schedule always exists.

4. MILP of TFTT with idle time

To develop a mixed integer mathematical model, two types of variables are needed. The first type will be used to sequence jobs and determine a feasible schedule while the latter will be used to calculate TEC.

For defining a feasible schedule an approach inspired by Manne (1960) is suggested based on whether a job occurs before others. Let $x_{j,i}$ be a binary variable equal to one if job j precedes job i in the sequence (not necessarily immediately before it), and zero otherwise. Also let $s_{j,l}$ and $c_{j,l}$

be the start time and completion time of each job *j* on the l^{th} operation, respectively. Finally, constant M_1 used in the formulation, represents a large positive number. Then constraints (1-6) can be used for sequencing jobs in two machine flow shop and also determining start time and completion time of each operation. For more clarification about the model, all the indices and parameters are defined as the following:

$x_{j,i}$: a binary variable equal to one if job *j* precedes job *i* in the sequence
$p_{j,l}$: processing times of job *j* on the l^{th} operation
$s_{j,l}$: start time of each job *j* on the l^{th} operation
$c_{j,l}$: completion time of each job *j* on the l^{th} operation
M_1 : a large positive number
T : an upper bound for make span of the problem, i.e. a fixed number

$$c_{j,1} \geq p_{j,1} \quad \text{for } j=1,...,n \tag{1}$$
$$c_{j,2} \geq c_{j,1} + p_{j,2} \quad \text{for } j=1,...,n \tag{2}$$
$$c_{j,l} \geq p_{j,l} + c_{i,l} - M_1 x_{j,i} \quad \text{for } j=1,...,n-1;\ i=j+1,...,n;\ l=1,2 \tag{3}$$
$$c_{i,l} \geq p_{i,l} + c_{j,l} - M_1(1-x_{j,i}) \quad \text{for } j=1,...,n-1;\ i=j+1,...,n;\ l=1,2 \tag{4}$$
$$s_{j,l} = c_{j,l} - p_{j,l} \quad \text{for } j=1,...,n;\ l=1,2 \tag{5}$$
$$c_{j,2} \leq T \quad \text{for } j=1,...,n \tag{6}$$

Before talking about the second type of variables, it should be noted that an operation might be processed in more than one cost period because either its processing time is sufficiently greater than the duration of a period or it is processed across two or more adjacent periods. For this reason, three cases can be distinguished to calculate energy cost of operations. The first one is when an operation is completely processed within a cost period. The second case happens when an operation is processed among two adjacent cost periods. Finally, the third case happens when an operation is done within more than two cost periods. Figure 2 shows an example with all the aforementioned cases.

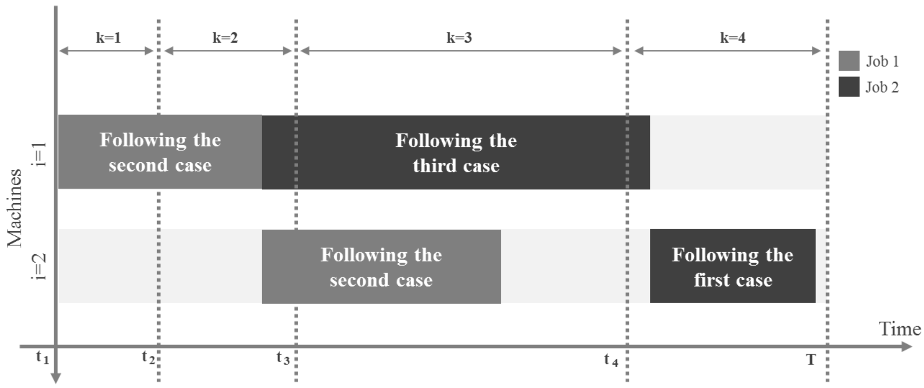


Figure 2. An example for three conditions of operation placements

For calculating energy costs of each job, two types of binary variables are defined as $S_{j,k,l}$ a binary variable is equal to one if job *j* starts its l^{th} operation at time period *k* and $F_{j,k,l}$ that is equal to one when job *j* completes its l^{th} operation at time period *k*. Finally $y_{j,k,l}$ shows how much time of each operation l^{th} of job *j* is processed at time period *k*. For more clarification about the model, all the indices and parameters are defined as the following:

$S_{j,k,l}$:	a binary variable equal to one if job j starts its l^{th} operation at time period k
$F_{j,k,l}$:	a binary variable equal to one when job j completes its l^{th} operation at time period k
$y_{j,k,l}$:	shows how much time of each operation l^{th} of job j is processed at time period k
$q_{j,l}$:	resource demand of job j on the l^{th} operation
t_k :	starting time of period k
a_k :	duration of period k
f_k :	resource price of period k
M_2 :	a large positive number

With the notation given above, the investigated problem can be formulated as the following MILP model:

$$\text{Minimize } TEC = \sum_{j=1}^{n}\sum_{k=1}^{K}\sum_{l=1}^{2}(y_{j,k,l})(q_{j,l})(f_k)$$

Subject to:

(1-6)

$$y_{j,k,l} \geq p_{j,l} - M_2(2 - S_{j,k,l} - F_{j,k,l}) \qquad \text{for } j=1,...,n; \quad k=1,...,K; \quad l=1,2 \tag{7}$$

$$y_{j,k,l} \geq (t_{k+1} - s_{j,l}) - M_2(2 - S_{j,k,l} - F_{j,k+1,l}) \qquad \text{for } j=1,...,n; \quad k=1,...,K-1; \quad l=1,2 \tag{8}$$

$$y_{j,k,l} \geq (c_{j,l} - t_k) - M_2(2 - S_{j,k-1,l} - F_{j,k,l}) \qquad \text{for } j=1,...,n; \quad k=2,...,K; \quad l=1,2 \tag{9}$$

$$y_{j,k,l} \geq a_k\left((\sum_{v=1}^{k-1} S_{j,v,l} + \sum_{v=k+1}^{K} F_{j,v,l}) - 1 \right) \qquad \text{for } j=1,...,n; \quad k=2,...,K-1; \quad l=1,2 \tag{10}$$

$$s_{j,l} \leq t_{k+1} + M_2(1 - S_{j,k,l}); \quad t_{K+1} = T \qquad \text{for } j=1,...,n; \quad k=1,...,K; \quad l=1,2 \tag{11}$$

$$s_{j,l} \geq t_k - M_2(1 - S_{j,k,l}) \qquad \text{for } j=1,...,n; \quad k=1,...,K; \quad l=1,2 \tag{12}$$

$$\sum_{k=1}^{K} S_{j,k,l} = 1 \qquad \text{for } j=1,...,n; \quad l=1,2 \tag{13}$$

$$c_{j,l} \leq t_{k+1} + M_2(1 - F_{j,k,l}); \quad t_{K+1} = T \qquad \text{for } j=1,...,n; \quad k=1,...,K; \quad l=1,2 \tag{14}$$

$$c_{j,l} \geq t_k - M_2(1 - F_{j,k,l}) \qquad \text{for } j=1,...,n; \quad k=1,...,K; \quad l=1,2 \tag{15}$$

$$\sum_{k=1}^{K} F_{j,k,l} = 1 \qquad \text{for } j=1,...,n; \quad l=1,2 \tag{16}$$

$$\begin{aligned} & s_{j,l}, c_{j,l} \geq 0 && \text{for } j=1,...,n; \quad l=1,2 \\ & S_{j,k,l}, F_{j,k,l}, y_{j,k,l}, x_{j,i} \quad \textit{binary} && \text{for } i, j=1,...,n; \quad k=1,...,K; \quad l=1,2 \end{aligned} \tag{17}$$

Constraints (7-10) calculate amount of time each operation occupies at different cost periods. Figure 2 and aforementioned explanation about the three different cases of job-period engagement can clarify these constraints. Constraints (11-13) deal with assigning binary variable of $S_{j,k,l}$ and similarly constraints (14-16) calculate $F_{j,k,l}$ that shows if an operation is completed in a period. Finally, to accelerate the speed of solvers, it can be useful to define a value for $M_1 = T + \max_{j,l}(p_{j,l})$ and $M_2 = T$ .Some studies in the literature use a discrete-time formulation for scheduling problems under time dependent costs. With discrete-time formulation, the number of binary variables can be really huge due to the division of time horizon. In this work, a continuous-time formulation is proposed.

The proposed MILP can find optimum solutions for small size instances while for solving large size instances, heuristic algorithms can be useful.

5. Dominance properties of non-delay TFTT

In this section, some lemmas, which are the basis of the proposed branch-and-bound (B&B) algorithm, are presented and special cases are investigated for the non-delay version of the problem. By non-delay we mean operations on machines must be carried out without any inserted idle times which never permit a delay (via inserted idle time) when the machine becomes available and work is waiting. As a result the solution space of the problem will be finite and there will be $n!$ different combinatorial solutions to be searched.

By finding dominance rules, some sequences can be pruned without checking lower bounds if they are dominated by others and as a result they can be excluded from consideration to accelerate the search procedure of our proposed B&B.

We employ the method of pairwise interchange of two adjacent jobs *i* and *j*. Let *S* be the current schedule while *S'* be identical with *S* except that the jobs *i* and *j* are interchanged. Also let *h* be the job immediately following $\{i, j\}$ either before or after pairwise interchange.

On the other hand, because of the property of flow-shop scheduling, by interchanging two adjacent jobs, the next jobs after $\{i, j\}$ in the sequence might shift forward or backward in the sequence results in changing objective function. So before investigating the dominance rules, it is important to understand the conditions in which interchanging $\{i, j\}$ does not affect any shift for the operations of job *h*.

Let $EC_i(S)$ and $EC_j(S)$ be the energy cost values for jobs *i* and *j*, respectively in the schedule *S*. Also let *TEC(S)* and *TEC(S')* represent total energy costs for the schedules *S* and *S'*. By inspiring the formulation of Sen et al. (1989), let I_i and I_j denote idle times in *S* occurring on the second machine immediately prior to processing of jobs *i* and *j*. Similarly, let I'_i and I'_j denote idle times in *S'*.

If $C_i(S)$ denotes the completion time of job *i* in the schedule *S*, then:

$$C_i(S) = \sum_{b=1}^{i} p_{b,2} + \sum_{b=1}^{i} I_b = \sum_{b=1}^{i-1} p_{b,2} + \sum_{b=1}^{i-1} I_b + p_{i,2} + I_i \tag{18}$$

For ease of further calculations, letting *N*:

$$N = \sum_{b=1}^{i-1} p_{b,1} - \sum_{b=1}^{i-1} p_{b,2} - \sum_{b=1}^{i-1} I_b \tag{19}$$

It can be concluded:

$$I_i = \max\{0, N + p_{i,1}\} \tag{20}$$

$$I_j = \max\{0, N + p_{i,1} + p_{j,1} - p_{i,2} - I_i\} \tag{21}$$

$$I'_j = \max\{0, N + p_{j,1}\} \tag{22}$$

$$I'_i = \max\{0, N + p_{i,1} + p_{j,1} - p_{j,2} - I'_j\} \tag{23}$$

The only condition in which the position of job h remains unchanged in both schedules of S and S', is when (see figure 3):

$$I_i + I_j = I'_i + I'_j \tag{24}$$

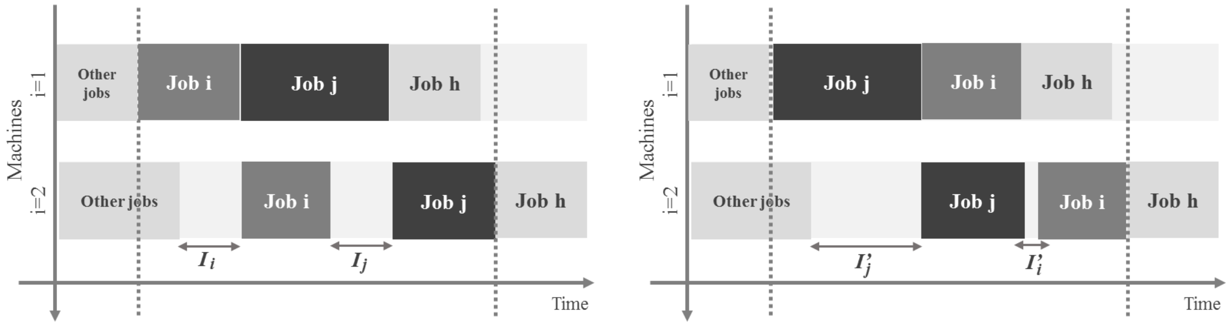


Figure 3. Sequence S (left) and sequence S' (right)

Then the following equations can be presented:

$$I_i + I_j = I_i + \max\{0, N + p_{i,1} + p_{j,1} - p_{i,2} - I_i\} = \max\{I_i, N + p_{i,1} + p_{j,1} - p_{i,2}\}$$

$$= \max\{0, N + p_{i,1}, N + p_{i,1} + p_{j,1} - p_{i,2}\} \tag{25}$$

Similarly, $I'_i + I'_j = \max\{0, N + p_{j,1}, N + p_{i,1} + p_{j,1} - p_{j,2}\}$

Eight cases can be considered if $I_i + I_j = I'_i + I'_j$. These cases are:

(a) $p_{i,2} \geq p_{j,1}, p_{j,2} \geq p_{i,1}, p_{i,1} = p_{j,1}$ (26)

(b) $p_{i,2} \leq p_{j,1}, p_{j,2} \leq p_{i,1}, p_{i,2} = p_{j,2}$ (27)

(c) $p_{i,2} \geq p_{j,1}, p_{j,2} \leq p_{i,1}, p_{j,1} = p_{j,2}$ (28)

(d) $p_{i,2} \leq p_{j,1}, p_{j,2} \geq p_{i,1}, p_{i,1} = p_{i,2}$ (29)

(e) $p_{i,2} \geq p_{j,1}, p_{j,2} \geq p_{i,1}, p_{i,1} + N \leq 0, p_{j,1} + N \leq 0$ (30)

(f) $p_{i,2} \leq p_{j,1}, p_{j,2} \leq p_{i,1}, p_{i,1} + p_{j,1} - p_{i,2} + N \leq 0, p_{i,1} + p_{j,1} - p_{j,2} + N \leq 0$ (31)

(g) $p_{i,2} \geq p_{j,1}, p_{j,2} \leq p_{i,1}, p_{i,1} + N \leq 0, p_{i,1} + p_{j,1} - p_{j,2} + N \leq 0$ (32)

(h) $p_{i,2} \leq p_{j,1}, p_{j,2} \geq p_{i,1}, p_{j,1} + N \leq 0, p_{i,1} + p_{j,1} - p_{i,2} + N \leq 0$ (33)

Therefore, $I_i + I_j = I'_i + I'_j$ will hold if at least one of the (26)-(33) is true. Then we will be sure that by interchanging $\{i, j\}$, the next jobs will not change. Also it is trivial to understand by interchanging $\{i, j\}$, the completion time of the previous jobs will not change too. As a result for comparing $TEC(S)$ and $TEC(S')$, it is enough if we only compare $EC_i(S) + EC_j(S)$ with $EC_i(S') + EC_j(S')$.

As described before, for any operation, it might be processed in one cost period or in more than one period. Then it is important to consider two scenarios for two adjacent jobs $\{i, j\}$.

In the first scenario, all operations of the jobs *i* and *j* will be processed within one cost period. In the latter scenario, at least some operations will be processed across two adjacent cost periods.

Lemma 1. Under condition (24), if a pair of adjacent jobs $\{i, j\}$ are processed completely in one cost period, it means there exist one cost period $\exists k \in P \mid t_k \leq c_{i,1} - p_{i,1}, c_{j,2} \leq t_{k+1}$, then $TEC(S) = TEC(S')$.

Proof. Under condition (24), for comparing $TEC(S)$ and $TEC(S')$, it is enough if we only compare $EC_i(S) + EC_j(S)$ with $EC_i(S') + EC_j(S')$. As the processing time and resource demand of jobs are not dependent on their position, then if $\{i, j\}$ are processed completely in one cost period, the resource cost will remain unchanged during processing. As a result $TEC(S) = TEC(S')$.

Lemma 2. Under condition (24), if for a pair of adjacent jobs $\{i, j\}$, at least some operations will be processed across two adjacent cost periods, i.e. there exist one cost period $\exists k \in P \mid t_k \leq c_{i,1} - p_{i,1}, c_{j,2} \geq t_{k+1}, c_{j,2} \leq t_{k+2}$, then $TEC(S) \leq TEC(S')$ if $q_{i,1} \geq q_{j,1}$, $q_{i,2} \geq q_{j,2}$ and $f_{k+1} \geq f_k$.

Proof. As the energy demand of both operations of job *i* are more than the energy demand of both operations of job *j*, then by processing high consuming operations in the less expensive cost period, total energy cost will be less, then $TEC(S) \leq TEC(S')$.

Lemma 3. Under condition (24), if for a pair of adjacent jobs $\{i, j\}$, at least some operations will be processed across two adjacent cost periods, then $TEC(S) \leq TEC(S')$ if and only if $EC_i(S) + EC_j(S) \leq EC_i(S') + EC_j(S')$.

Proof. Under condition (24), for comparing $TEC(S)$ and $TEC(S')$, it is enough if we only compare $EC_i(S) + EC_j(S)$ with $EC_i(S') + EC_j(S')$.

Lemma 4. If for a pair of adjacent jobs $\{i, j\}$, they both are needed to process in the last cost period, it means $t_K \leq c_{i,1} - p_{i,1}$, then $TEC(S) = TEC(S')$. It means interchanging jobs which are placed at the last cost period, does not change total resource cost, even if condition (24) does not hold.

Proof. At the last cost period, even if by interchanging jobs $\{i, j\}$ the next jobs move forward or backward, they will not effect on total energy cost because they will be again in a same cost period.

Lemma 5. Based on Fang et al. (2016), on single machine, if cost periods are sorted in non-decreasing order of resource costs, it means $f_1 \leq f_2 \leq \ldots \leq f_K$, then the sequence which is sorted by non-increasing order of resource demand, i.e. Highest Resource Demand (HRD Rule), have the least total energy cost on the machine.

6. Branch and Bound procedure

Calculating Lower bound

In this subsection, a lower bound is established to accelerate the search process of the branch-and-bound algorithm. Let $S^{(t)} = (\pi, \pi^c)$ denote a sequence in which π contains a set of scheduled jobs and π^c considers the remaining unscheduled jobs at time *t*. For each scheduled job $j \in \pi$, energy cost can be easily calculated by considering processing time and energy demand of operations and energy price of the periods which job *j* is processed in them.

For calculating lower bound of unscheduled jobs, the basic idea is to generate two new time horizons one for each machine.

By sorting new time horizons in non-decreasing order of their resource price, we can make the prerequisite conditions of lemma 5. Then for each machine, operations will be sorted according to HRD rule and their corresponding energy costs will be calculated. Let $tt_1^{(t)} = \sum_{j \in \pi} p_{j,1}$ and $tt_2^{(t)} = \max_{j \in \pi}[c_{j,2}]$. Also let $k_1^{(t)}$ and $k_2^{(t)}$ to be the first $k \in P$ that $t_{k_1^{(t)}} > tt_1^{(t)}$ and $t_{k_2^{(t)}} > tt_2^{(t)}$, respectively. These are the first cost periods after scheduled jobs on the first and second machine. Now two new cost periods of k_1^{new} and k_2^{new} will be defined with a start times of $t_{k_1^{new}} = tt_1^{(t)}$ and $t_{k_2^{new}} = tt_2^{(t)}$, resource cost of $f_{k_1^{new}} = f_{(k_1^{(t)}-1)}$ and $f_{k_2^{new}} = f_{(k_2^{(t)}-1)}$.

According to Lemma 5, if we assume cost periods in increasing form of their energy price (i.e. f_k), a sequence with HRD rule is optimal. So for computing lower bound of remaining unscheduled jobs on each machine, two new time horizons are defined as $[k \in P | t_k > tt_1^{(t)}, k_1^{new}]$ and $[k \in P | t_k > tt_2^{(t)}, k_2^{new}]$ for the first and second machine respectively. Finally, cost periods are sorted in non-decreasing order of cost periods' resource price.

Then the energy cost (EC) of each job on each machine in the new time horizon ($EC_{j,l}^{new}$) will be calculated according to HRD rule. This will be the least possible resource cost for unscheduled operations, and can be seen as a lower bound.

Thus, the final lower bound on the total resource costs of sequence $S^{(t)}$ based on π scheduled jobs is:

$$LB(S^{(t)}) = \sum_{j \in \pi} EC_j + \sum_{j \in \pi^c} EC_{j,1}^{new} + \sum_{j \in \pi^c} EC_{j,2}^{new} \tag{34}$$

Figure 4 shows how new cost periods for machine one and two can be created. These new cost periods are made based on an existing partial sequence of 1-2-3 for the aforementioned example, i.e. the one with six jobs and three cost periods.

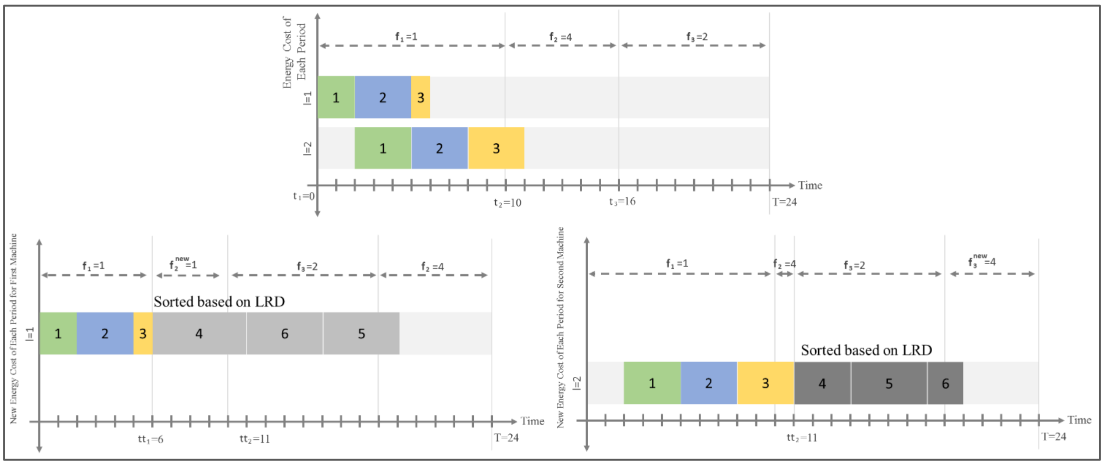


Figure 4. Existing partial sequence (up) and new cost periods for jobs on machine one (down-left) and new cost periods for jobs on machine two (down-right) for calculating lower bounds

Branch and Bound steps

This section describes the B&B algorithm suggested in this study. In the B&B algorithm, a node represents a partial schedule that is to be placed at the front part of a complete schedule. In fact, jobs are sequenced from front to back, i.e., the one that is in the first position of an optimal schedule is first determined, then the one immediately after it, and so on so the contribution of the scheduled jobs to the overall total resource cost will be known exactly.

Each node in the B&B tree corresponds to a partial schedule, and a lower bound is computed for each node. The B&B algorithm also uses the dominance properties given in Lemma (1-5), and nodes will be fathomed by partial schedules that are dominated by other partial schedule(s) (corresponding to other nodes) as follows:

For each node generated in the B&B algorithm, we check whether the last two jobs of partial sequence are within the last cost period or not. If they are placed in the last cost period, we just add other remaining jobs to the sequence in an arbitrary order, and calculate total energy cost. In fact, we will treat to that node similar to a leaf node.

If such node does not exist, we will check whether one of the conditions (26-33) is satisfied. If the condition is satisfied, we will check if conditions of lemma 1 is satisfied, then to break ties, the node will be fathomed if the processing time of the new added job on the first machine is more than the processing time of the previous job on the first machine.

If conditions of lemma 2 is satisfied, then we will fathom the node if $q_{i,1} \geq q_{j,1}$, $q_{i,2} \geq q_{j,2}$ and $f_{k+1} \geq f_k$ while indices of j and i correspond to the new added job and the previous job, respectively.

Finally if conditions of lemma 3 is satisfied, we will calculate $EC_i + EC_j$ then we will interchange jobs i and j and again calculate $EC'_i + EC'_j$ while indices of j and i correspond to the new added job and the previous job, respectively. The node will be fathomed if $EC_i + EC_j \geq EC'_i + EC'_j$.

To select a node from which to generate branches, the depth-first rule is employed in the algorithm. In other words, a node with the most jobs in the corresponding partial schedule is selected for branching. The branch systematically is worked down until it is either eliminated by virtue of dominance properties, the lower bound or reaches a leaf node. Figure 5 shows overall structure of branching for the aforementioned example.

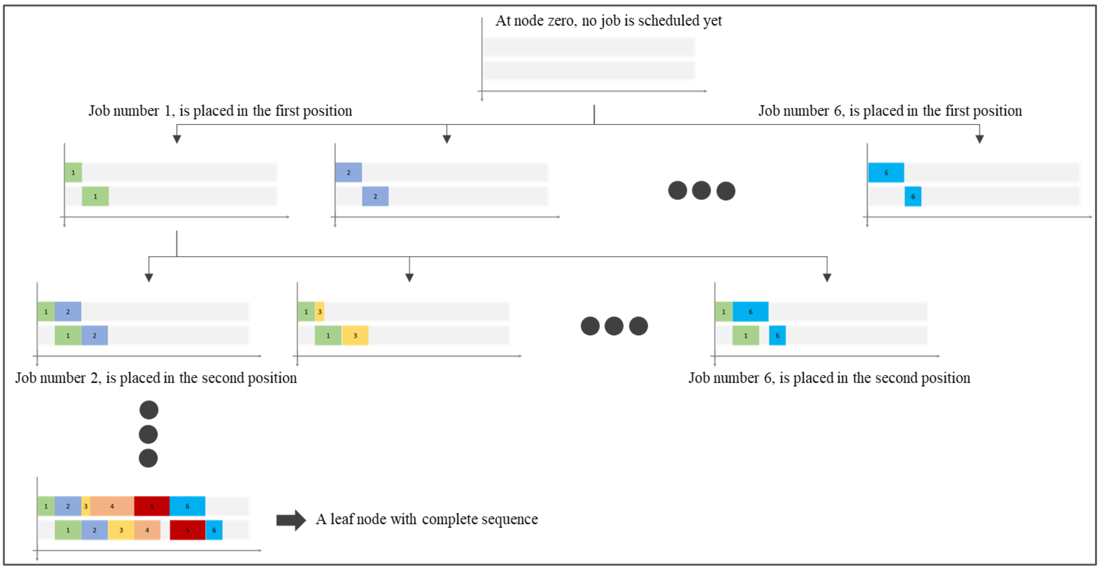


Figure 5. An overall structure of branching for the example with six jobs

The initial incumbent sequence will be generated by a greedy heuristic algorithm which assigns jobs to cost periods with the lowest cost. It will be replaced whenever a new better solution is found in the procedure of B&B.

7. Computational experiments

A computational experiment is conducted to test the effectiveness of the proposed B&B algorithm. The experiment is conducted over various problem sizes of n and K. The processing times and resource demands of jobs are generated randomly from discrete uniform distribution with a range of one to ten and with a range of one to five, respectively. Then based on the procedure suggested by Kim (1993), the maximum make span is calculated and put as the value for T. Then the time horizon is split into K time periods with same duration. Energy price of each period is generated from discrete uniform distribution with a range of one to five. For testing performance of MILP model, different problem size as a combination of job numbers for two cost period numbers of 3 and 5, are considered. Ten replications are produced for each combination of n and K results in a total of 120 problems to be tested on CPLEX solver and run on a PC 2.4 GHz CPU with 4 GB RAM. The computational results on the average and maximum CPU time are shown in table 3. A problem is considered not solved if CPLEX takes more than 3,600 seconds of CPU time.

Table 3. Results of MILP for different problem sizes

Job Numbers	Cost Period Numbers	Average CPU Time [Sec]	Max CPU Time [Sec]
5	3	0.70	1.22
	5	5.28	10.84
6	3	2.51	4.20
	5	36.86	81.60
7	3	10.34	20.50
	5	278.65	484.00
8	3	35.36	60.00
	5	Not Solved	Not Solved
9	3	76.75	100.00
	5	Not Solved	Not Solved
10	3	1,322.00	2,800.00
	5	Not Solved	Not Solved

For testing performance of B&B algorithm, different problem size as a combination of job numbers and cost period numbers are considered. Ten replications are produced for each combination of n and K and as a result, a total of 180 problems are tested. The algorithms are coded in MATLAB 2012a and run on a PC 2.4 GHz CPU with 4 GB RAM. The computational results on the number of problems solved, the average CPU times and the average number of nodes processed are shown in table 4. Also average percentage of total processed nodes to total complete nodes, i.e. number of nodes in search without fathoming, average percentage of fathomed nodes to total processed nodes and average percentage of nodes fathomed by dominance rules are presented in table 4. A reasonable time for solving a problem is set at 3,600 seconds. A problem is considered not solved if an algorithm takes more than 3,600 seconds of CPU time.

Table 4. Results of B&B algorithm for different problem sizes

Job Numbers	Cost Period Numbers	Number Solved	Average CPU Time	Average Total Nodes	Average Percentage of		
					Total Nodes to Complete Nodes	Fathomed Nodes to Total Nodes	Fathomed by Domination Rules
5	3	10	0.049	104	50.0%	67.0%	9.7%
	5	10	0.092	164	80.0%	53.3%	4.7%
	8	10	0.154	201	95.0%	52.0%	0.0%
7	3	10	1.358	2,728	31.5%	69.0%	21.3%
	5	10	1.211	1,682	19.4%	70.0%	8.2%
	8	10	4.063	4,120	47.6%	61.5%	0.4%
10	3	10	129.0	205,180	3.3%	83.8%	26.6%
	5	10	662.4	720,300	11.6%	73.0%	17.2%
	8	10	475.6	345,832	5.6%	74.6%	2.7%
12	3	8	434.2	703,378	0.1%	87.0%	33.9%
	5	4	197.0	190,170	0.0%	87.7%	17.8%
	8	1	231.1	129,122	0.0%	85.6%	0.4%
14	3	5	70.4	95,135	0.0%	94.3%	20.4%
	5	2	172.6	171,264	0.0%	90.0%	28.7%
	8	0					
15	3	7	690.1	959,185	0.0%	91.4%	35.6%
	5	4	530.6	444,287	0.0%	85.7%	21.9%
	8	1	427.4	236,172	0.0%	85.2%	15.0%

8. Conclusion

This paper addresses two-machine flow shop scheduling problem where cost of energy use depends on time intervals at which operations are processed. The objective is to minimize the total energy cost of producing all jobs. The problem is reduced to a mixed integer linear programming (MILP). Two types of the schedule are considered: (a) the inserted idle time between operations is permitted; (b) the operation on machine must start without delay as soon as the machine and the operation are available. For the type (a), a new time-interval based mixed integer mathematical model is presented, while for type (b) a branch-and-bound algorithm has been suggested. An efficient mechanism to calculate lower bound based on given partial sequence of jobs and several dominance properties have been suggested to accelerate the search procedure of the algorithm. Based on results, the proposed algorithms can optimally schedule jobs in small size samples but by increasing the number of jobs from 15 and cost periods from 3, the performance of branch-and-bound has been decreased.

The time interval based integer programming adopted here guarantees the global optimum but is time consuming. It is worthwhile to investigate heuristic algorithms which can identify close-to-optimal solutions with much shorter computational times. Also efforts are needed to advance the problem formulation to study other objective functions beside total energy costs and consider a multi objective approach for solving the problem.

References

Aghelinejad, M., Ouazene, Y., and Yalaoui, A., (2016). "Machine and production scheduling under electricity time varying prices", In Industrial Engineering and Engineering Management (IEEM), IEEE International Conference on, pp. 992-996.

Che, A., Lv, K., Levner, E., and Kats, V., (2015). "Energy consumption minimization for single machine scheduling with bounded maximum tardiness". In Networking, Sensing and Control (ICNSC), IEEE 12th International Conference on, pp. 146-150.

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*, Vol. 129, pp. 565-577.

Chen, L., Megow, N., Rischke, R., Stougie, L., and Verschae, J., (2015). "Optimal algorithms and a PTAS for cost-aware scheduling". In International Symposium on Mathematical Foundations of Computer Science (pp. 211-222). Springer Berlin Heidelberg.

Ding, J. Y., Song, S., Zhang, R., Chiong, R., and Wu, C. (2016). "Parallel Machine Scheduling Under Time-of-Use Electricity Prices: New Models and Optimization Approaches. IEEE Transactions on Automation Science and Engineering, Vol. 13, No. 2, pp. 1138-1154.

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*, Vol. 238, pp. 199-227.

Gong, X., De Pessemier, T., Joseph, W., and Martens, L., (2015). "An energy-cost-aware scheduling methodology for sustainable manufacturing", Procedia CIRP, 29, pp.185-190.

González, M. A., Oddi, A., and Rasconi, R., (2015). "A multi-objective memetic algorithm for solving job shops with a non-regular energy cost", *aint S ing an*, Vol. 15.

Jozefowska, J., (2007). *Just-in-time scheduling: models and algorithms for computer and manufacturing systems* (Vol. 106). Springer Science & Business Media.

Kanet, J. J., and Sridharan, V., (2000). "Scheduling with inserted idle time: problem taxonomy and literature review", *Operations Research*, Vol. 48, No. 1, pp. 99-110.

Kim, Y. D., (1993). "A new branch and bound algorithm for minimizing mean tardiness in two-machine flow shops", *Computers & Operations Research*, Vol. 20, No. 4, pp. 391-401.

Liu, Y., Dong, H., Lohse, N., Petrovic, S., and Gindy, N., (2014). "An investigation into minimising total energy consumption and total weighted tardiness in job shops", *Journal of Cleaner Production*, Vol. 65, pp. 87-96.

Liu, C. H., and Huang, D. H., (2014). "Reduction of power consumption and carbon footprints by applying multi-objective optimisation via genetic algorithms", *International Journal of Production Research*, Vol. 52, No. 2, pp. 337-352.

Luo, H., Du, B., Huang, G. Q., Chen, H., and Li, X., (2013). "Hybrid flow shop scheduling considering machine electricity consumption cost", *International Journal of Production Economics*, Vol. 146, No. 2, pp. 423-439.

Manne, A. S., (1960). “On the job-shop scheduling problem”, *Operations Research*, Vol. 8, No. 2, pp. 219-223.

Moon, J. Y., Shin, K., and Park, J., (2013). “Optimization of production scheduling with time-dependent and machine-dependent electricity cost for industrial energy efficiency”, *The International Journal of Advanced Manufacturing Technology*, Vol. 68, pp. 523-535.

Mouzon, G., and Yildirim, M.B., (2008). “A framework to minimise total energy consumption and total tardiness on a single machine”, *Int. J. Sustain. Eng.*, Vol. 1, No. 2, pp. 105-116.

Mouzon, G., Yildirim, M.B., Twomey, J., (2007). Operational methods for minimization of energy consumption of manufacturing equipment. Int. J. Prod. Res. 45 (18-19), 4247-4271.

Schaller, J., and Valente, J. M., (2013). “A comparison of metaheuristic procedures to schedule jobs in a permutation flow shop to minimise total earliness and tardiness”, *International Journal of Production Research*, Vol. 51, No. 3, pp. 772-779.

Sen, T., Dileepan, P., and Gupia, J. N., (1989). “The two-machine flow shop scheduling problem with total tardiness”, *Computers & operations research*, Vol. 16, No. 4, pp. 333-340.

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

Tan, M., Duan, B., and Su, Y., (2016). “Economic batch sizing and scheduling on parallel machines under time-of-use electricity pricing”, *Operational Research*, pp. 1-18.

Wan, G., and Qi, X., (2010). “Scheduling with variable time slot costs”, *Naval Research Logistics (NRL)*, Vol. 57, No. 2, pp. 159-171.

Wang, S., Zhu, Z., Fang, K., Chu, F., and Chu, C., (2017). “Scheduling on a two-machine permutation flow shop under time-of-use electricity tariffs”, *International Journal of Production Research*, pp. 1-15.

Zhang, H., Zhao, F., Fang, K., and Sutherland, J. W., (2014). “Energy-conscious flow shop scheduling under time-of-use electricity tariffs”, *CIRP Annals-Manufacturing Technology*, Vol. 63, No. 1, pp. 37-40.

Zhang, R., and Chiong, R., (2016). “Solving the energy-efficient job shop scheduling problem: a multi-objective genetic algorithm with enhanced local search for minimizing the total weighted tardiness and total energy consumption”, *Journal of Cleaner Production*, Vol. 112, pp. 3361-3375.

Zhao, Y., Qi, X., and Li, M., (2015). “On scheduling with non-increasing time slot cost to minimize total weighted completion time”, *Journal of Scheduling*, pp. 1-9.

Zhong, W., and Liu, X., (2012). “A Single Machine Scheduling Problem with Time Slot Costs”, In Recent Advances in Computer Science and Information Engineering (pp. 677-681). Springer Berlin Heidelberg.

This article can be cited: Ebrahimzadeh Pilerood, A., Heydari, M., and Mahdavi Mazdeh, M., (2018). "Investigation of the two-machine flow shop scheduling problem to minimize total energy costs with time-dependent energy prices", *Journal of Industrial Engineering and Management Studies*, Vol. 5, No. 2, pp. 97-112.