algorithms

MDPI

Article

A New Greedy Insertion Heuristic Algorithm with a Multi-Stage Filtering Mechanism for Energy-Efficient Single Machine Scheduling Problems

Hongliang Zhang *, Youcai Fang, Ruilin Pan and Chuanming Ge

School of Management Science and Engineering, Anhui University of Technology, Ma'anshan 243032, China; ahutfangyoucai@163.com (Y.F.); rlpan9@ahut.edu.cn (R.P.); 13083209689@163.com (C.G.)

* Correspondence: zhanghongliang_17@126.com; Tel.: +86-555-231-5379

Received: 25 December 2017; Accepted: 6 February 2018; Published: 9 February 2018

Abstract: To improve energy efficiency and maintain the stability of the power grid, time-of-use (TOU) electricity tariffs have been widely used around the world, which bring both opportunities and challenges to the energy-efficient scheduling problems. Single machine scheduling problems under TOU electricity tariffs are of great significance both in theory and practice. Although methods based on discrete-time or continuous-time models have been put forward for addressing this problem, they are deficient in solution quality or time complexity, especially when dealing with large-size instances. To address large-scale problems more efficiently, a new greedy insertion heuristic algorithm with a multi-stage filtering mechanism including coarse granularity and fine granularity filtering is developed in this paper. Based on the concentration and diffusion strategy, the algorithm can quickly filter out many impossible positions in the coarse granularity filtering stage, and then, each job can find its optimal position in a relatively large space in the fine granularity filtering stage. To show the effectiveness and computational process of the proposed algorithm, a real case study is provided. Furthermore, two sets of contrast experiments are conducted, aiming to demonstrate the good application of the algorithm. The experiments indicate that the small-size instances can be solved within 0.02 s using our algorithm, and the accuracy is further improved. For the large-size instances, the computation speed of our algorithm is improved greatly compared with the classic greedy insertion heuristic algorithm.

Keywords: energy-conscious single machine scheduling; time-of-use electricity tariffs; greedy insertion heuristic; coarse granularity and fine granularity filtering mechanism; concentration and diffusion strategy

1. Introduction

Driven by the rapid development of global economy and civilization, there will be a consistent growth in energy consumption in the years ahead. According to a survey of the International Energy Agency (IEA), the world-wide demand for energy will increase by 37% by 2040 [1]. Non-renewable energy resources such as coal, oil, and gas are diminishing day-by-day, which is threatening the sustainable development of many countries. Meanwhile, greenhouse gas emissions generated from inappropriate usage of fossil fuels have taken a heavy toll on the global climate as well as the atmospheric environment [2]. Therefore, how to save energy and then improve the environment quality has become a pressing matter of the moment.

As the backbone of many countries, the industrial sector consumes about half of the world's total energy and emits the most greenhouse gases [3,4]. Hence, energy saving in the industry sector has priority in promoting sustainable economic development. As we all know, most of the energy is converted into the form of electricity that numerous industrial sectors use as their main energy [5,6].

Nevertheless, electricity is hard to store effectively, and thus, must be produced and delivered to its customers at once [7]. In addition, the electricity demand is always uneven, which leads to an increase in generating cost owing to the utilization of backup power facilities during peak hours [8]. In order to maintain balance between electricity supply and demand, electricity providers usually implement demand-side management programs [9], which are an essential component of realizing the goals of a smart grid and rationalizing the allocation of power resources [10].

One of the demand-side management programs is time-of-use (TOU) electricity tariffs, which have been widely used around the world. Usually, a common TOU tariff scheme can be divided into three types of periods: off-peak, mid-peak, and on-peak periods. The basic nature of the TOU scheme is that the retail prices set by electricity providers vary hourly throughout the day according to the amount of electricity demands; when there is an increase in demand, the electricity cost goes up correspondingly, and vice versa [11]. The practice of TOU electricity tariffs not only provides significant opportunities for the industrial sector to enhance energy efficiency, but also avoids power rationing during on-peak periods, and improves the stability of the power grid [7].

Using low-energy equipment and improving the efficiency of production management are two important methods to save energy [12]. As a widely used production management method, scheduling can effectively control energy consumption [13], which brings a lower cost of operation. However, the studies about energy-saving scheduling are still limited [14]. Over recent years, energy-efficient scheduling problems have gradually aroused the attention of scholars. To achieve the goal of energy saving during the production process, some researchers have investigated the problems with various energy-efficient mechanisms to reduce electricity costs by minimizing overall energy consumption, such as speed-scaling [15–18] and power-down [19–21], while others have studied the problems from the perspective of TOU electricity tariffs, which has become a frontier issue in this field.

As for researching scheduling problems under TOU electricity tariffs, there has been a growing interest recently. Considering both production and energy efficiency, Luo et al. [22] proposed an ant colony optimization meta-heuristic algorithm for hybrid flow shop scheduling problems under TOU electricity tariffs. Zhang et al. [12] studied a flow shop scheduling problem with production throughput constraints to minimize electricity cost and the carbon footprint simultaneously. Sharma et al. [23] presented a so called "econological scheduling" model for a speed-scaling multi-machine scheduling problem aimed to minimize the electricity cost and environmental impact. Moon et al. [24] examined the unrelated parallel machine scheduling problem under TOU electricity tariffs to optimize the weighted sum of makespan and electricity cost. Ding et al. [7] and Che et al. [25] addressed a similar parallel machine scheduling problem under TOU electricity tariffs to minimize the total electricity cost. The former developed a time-interval-based mixed-integer linear programming (MILP) model and a column generation heuristic algorithm. The latter improved the former model by providing a linear programming relaxation and a two-stage heuristic algorithm.

Single machine scheduling problems are of great significance both in theory and practice. On one hand, there are many single machine scheduling problems in the real industrial environment. For example, a Computer Numerical Control (CNC for short) planer horizontal milling and boring machine can be regarded as a single machine. On the other hand, the research results and methods of single machine scheduling problems can provide reference for other scheduling problems, such as flow shop, job shop, and parallel machine scheduling problems. For single machine scheduling problems under TOU electricity tariffs, Wang et al. [26] investigated a single-machine batch scheduling problem to minimize the makespan and the total energy costs simultaneously. Considering the TOU electricity tariffs and the power-down mechanism, Shrouf et al. [27] proposed a model that enables the operations manager to determine the "turning on" time, "turning off" time, and idle time at machine level, leading to a significant reduction in electricity cost by avoiding on-peak periods. Gong et al. [28] developed a mixed integer linear programming model and a genetic algorithm for the same problem, reducing electricity cost and greenhouse gas emissions effectively during peak time periods. Without considering a power-down mechanism, Fang et al. [29] studied the single machine scheduling problem under TOU

electricity tariffs systematically to minimize the total electricity costs. They divided the problem into the two cases of uniform-speed and speed-scalable, in which a preemptive version and a non-preemptive version were investigated respectively. For the uniform-speed case with non-preemptive assumption (Problem U-pyr), they demonstrated that the problem is strongly non-deterministic polynomial-time hard (NP-Hard). Note that the Problem U-pyr is the same as our problem. Based on Fang et al. [29], Che et al. [9] investigated a continuous-time MILP model for Problem U-pyr and developed a greedy insertion heuristic algorithm(GIH) that is the most classic method for this problem until now, according to our best knowledge. In their algorithm, the jobs are inserted into available periods with lower electricity prices in sequence, and the jobs with higher power consumption rates are mostly assigned to periods with lower electricity prices by traversing all non-full "forward blocks" and "backward blocks".

However, Fang et al. [29] did not establish a complete mathematical model for the single machine scheduling problem (Problem U-pyr) under TOU electricity tariffs, and their algorithm is only feasible in the condition that all the jobs have the same workload and the TOU tariffs follow a so-called pyramidal structure. Regrettably, the TOU electricity tariffs rarely follow a complete pyramidal structure in most provinces in China. To perfect the theory of Fang et al. [29], Che et al. [9] developed a new model and algorithm, but their algorithm requires that all the jobs must traverse all non-full "forward blocks" and "backward blocks", causing a strong high-time complexity. Especially when the processing times of the jobs are relatively short, it usually takes several jobs to fill one period and generates uneven remaining idle times, which leads to an increase in the number of forward and backward blocks required to be calculated. In addition, the generation process of "forward (backward) block" is limited to the job movements that do not incur additional electricity cost, which may cause some jobs to miss the optimum positions.

Thus, by focusing on the jobs with short processing times, this paper proposes a more efficient greedy insertion algorithm with a multi-stage filtering mechanism (GIH-F) based on the continuous-time MILP model to address these issues. The proposed algorithm mainly consists of two stages: coarse granularity filtering and fine granularity filtering. In the coarse granularity filtering stage, all the possible positions are first divided into three levels (i.e., three layers) according to the price of electricity, corresponding to high-price, medium-price, and low-price layers. Then, all the jobs are sorted in non-increasing order of their electricity consumption rates and assigned to the layer with a lower price successively. Based on the concentration and diffusion strategy, once the layer to which a job belongs is determined, the hunting zone of possible positions of a job is concentrated in a certain layer. To find the optimal position, the job to be inserted can search for its position in a relatively large space in the selected layer. Then, considering processing times, electricity consumption rates of the jobs and electricity prices, the electricity cost with respect to each possible position can be compared using characteristic polynomials. Based on these, several judgment conditions can be set up in for a fine granularity filtering stage to determine the position of each job to be inserted.

To summarize, the proposed algorithm can filter out impossible layers, and then judge each condition that belongs to the selected layer successively. Once the condition is satisfied, the job is just inserted into the corresponding position. Through coarse granularity filtering and fine granularity filtering, our algorithm does not have to traverse all possible positions, which leads to a great reduction in the time complexity. A real case study and two sets of randomly generated instances are used to test the performance of the proposed algorithm in this paper.

The rest of this paper is organized as follows. Section 2 presents a description of the single machine scheduling problem under TOU electricity tariffs. In Section 3, an efficient GIH-F is developed. Next, a real case study and two sets of experimental tests are provided in Section 4. Finally, the conclusions and prospects are presented in Section 5.

2. MILP Formulation for the Problem

This paper studies an energy-efficient single machine scheduling problem under TOU electricity tariffs. The problem can also be called a single machine scheduling problem with electricity costs (SMSEC). Consider a set of jobs $N = \{1, 2, \ldots, n\}$ that need to be processed on a single machine with the objective of minimizing the electricity cost. It is assumed that all the jobs must be processed at a uniform speed. Each job $i \in N$ has its unique processing time t_i and power consumption per hour p_i. A machine can process, at most, one job at a time, and when it is processing a job, no preemption is allowed. Each job and the machine are available for processing at time instant 0. Machine breakdown and preventive maintenance are not considered in this paper.

The machine is mainly powered by electricity. The electricity price follows a TOU pricing scheme represented by a set of time periods $M = \{1, 2, \ldots, m\}$, with each period $k \in M$, having an electricity price c_k and a starting time b_k. The interval of period k is represented by $[b_k, b_{k+1}]$, $k \in M$, and $b_1 = 0$ is always established. It is assumed that the C_{max} is the given makespan and $b_{m+1} \geq C_{max}$. This means that a feasible solution always exists.

The main work of this problem is to assign a set of jobs to periods with different electricity prices in the time horizon $[0, b_{m+1}]$ to minimize total electricity cost, and the main task is to determine to which period(s) a job is assigned and how long a job is processed in each period. Hence, two decision variables are given as follows. Note that the starting time of each job can be determined by the decision variables (i.e., $x_{i,k}$ and $y_{i,k}$).

$x_{i,k}$: assigned processing time of job i in period k, $i \in N$, $k \in M$;

$$y_{i,k} = \begin{cases} 1, \text{if job } i \text{ or part of job } i \text{ is processed in period } k \\ 0, \text{othertwise} \end{cases}, i \in N, k \in M.$$

In addition, a job is called processed within a period if both its starting time and completion time are within the same period. Otherwise, it is called processed across periods [9]. Let d_k and X_k, $k \in M$, represent the duration of period k and the total already assigned processing times in period k, respectively. The difference between d_k and X_k is defined as the remaining idle time of period k which is represented by I_k, $k \in M$. If $I_k = 0$, the period k is called *full*.

The MILP model for the single machine scheduling problem can be presented as follows:

$$\text{Min } TEC = \sum_{i=1}^{n} \sum_{k=1}^{m} p_i x_{i,k} c_i \tag{1}$$

s.t.

$$\sum_{k=1}^{m} x_{i,k} = t_i, i \in N; \tag{2}$$

$$\sum_{i=1}^{n} x_{i,k} \leq b_{k+1} - b_k, k \in M; \tag{3}$$

$$x_{i,k} \leq t_i y_{i,k}, i \in N, k \in M; \tag{4}$$

$$\sum_{k=j+1}^{l-1} y_{i,k} \geq (l - j - 1)(y_{i,l} + y_{i,j} - 1), i \in N, 3 \leq l \leq m, 1 \leq j \leq l - 2; \tag{5}$$

$$x_{i,k} \geq (y_{i,k-1} + y_{i,k+1} - 1)(b_{k+1} - b_k), i \in N, 2 \leq k \leq m - 1; \tag{6}$$

$$y_{i,k} + y_{i,k+1} + y_{j,k} + y_{j,k+1} \leq 3, i \in N, j \in N, 1 \leq k \leq m - 1, i \neq j. \tag{7}$$

Equation (1), the objective is to minimize the total electricity cost (*TEC*). Constraints (2)–(4) are associated with the processing time assigned to periods. Constraints (5) and (6) are used to guarantee the non-preemptive assumption. Specifically, constraint (5) guarantees that if a job is processed across

more than one period, it must occupy several continuous periods. Constraint (6) ensures that if a job is processed across three periods, the middle period must be fully occupied by the job. Constraint (7) ensures that, at most, one job is processed across any pair of adjacent periods. An illustration to the MILP model is given by an example of a 2-job single machine scheduling problem under a 3-period TOU scheme, as shown in Figure 1. Note that more detailed explanations of the formulas can be seen in Che et al. [9].

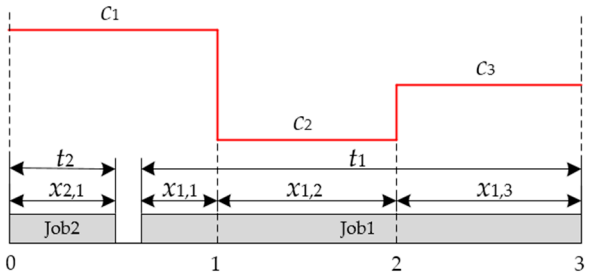


Figure 1. An example of the mixed-integer linear programming (MILP) model.

3. A Greedy Insertion Heuristic Algorithm with Multi-Stage Filtering Mechanism

3.1. The Characteristics of TOU Electricity Tariffs

In China, the TOU electricity tariffs can be mainly divided into two types according to the relative position of the off-peak period: (1) the off-peak period lies between an on-peak period and a mid-peak period; (2) the off-peak period lies between two mid-peak periods. In addition, there is only one off-peak period in a day, and the duration of the off-peak period is longest. This paper will investigate the single-machine scheduling problem under the first type of TOU electricity tariffs, which are being implemented in many places in China, such as Shanxi, Guangxi, and Jiangxi Provinces and so on. Next, let A, B, and Γ represent the off-peak, mid-peak, and on-peak periods, respectively. That is, A, B, $\Gamma \subseteq M$, $A \cup B \cup \Gamma = M$, $A \cap B = \varnothing$, $A \cap \Gamma = \varnothing$, and $B \cap \Gamma = \varnothing$. An illustration is given in Figure 2. Accordingly, $M = \{1, 2, \ldots, 10\}$, $A = \{5, 10\}$, $B = \{1, 3, 6, 8\}$, and $\Gamma = \{2, 4, 7, 9\}$.

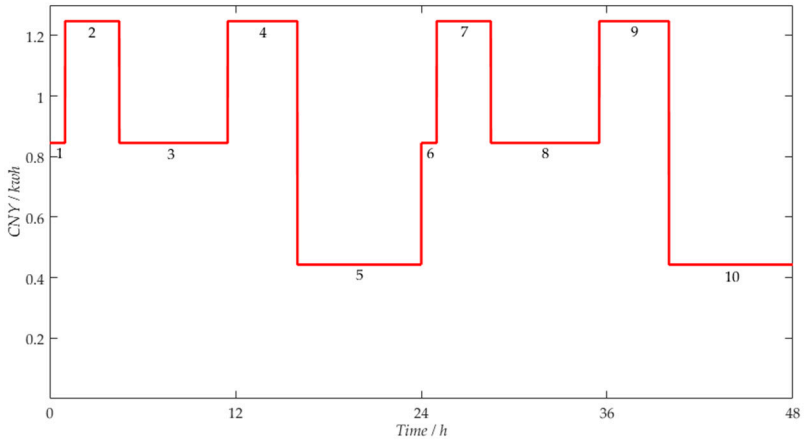


Figure 2. An illustration of the first type of time-of-use (TOU) electricity tariffs.

3.2. Multi-Stage Filtering Mechanism Design

The proposed algorithm is built on the concentration and diffusion strategy. Firstly, all the possible positions are divided into three layers based on the price of electricity. Specifically, in layer 1, all the jobs are processed within an off-peak period or processed across a pair of periods consisting of an off-peak period and a mid-peak period. In layer 2, the vast majority of the jobs are processed within a mid-peak

period or processed across a set of adjacent periods consisting of an off-peak period, a mid-peak period, and an on-peak period. Obviously, the electricity prices corresponding to the positions in this layer are relatively higher than layer 1. In layer 3, the vast majority of the jobs are processed across a pair of periods consisting of a mid-peak period and an on-peak period or processed within an on-peak period. The electricity prices of the positions corresponding to 3rd layer are the highest.

Then, all the jobs are sorted in non-increasing order of their electricity consumption rates and assigned to the layer with a lower price successively to achieve the preliminary coarse granularity filtering target. Once the layer to which a job belongs is determined, the hunting zone of possible positions of a job is concentrated in a certain layer. In the fine granularity filtering stage, several judgment conditions are set up to determine the position of each job to be inserted using characteristic polynomials based on considering processing time, electricity consumption rate of the job, and electricity price. What is more, aiming to find the optimal position, the hunting zone of possible positions of a job is properly expanded in this stage.

Assuming that a job, say job i, whose processing time does not exceed the duration of the shortest on-peak period, is to be inserted, the idea of the proposed algorithm is given as follows.

Layer 1 includes conditions 1–2. If $\exists k \in A, t_i \leq \max_{k\in A}\{I_k\} + I_{k+1}$, job i is assigned to layer 1. Obviously, at the very start, the off-peak periods are preferentially occupied by the jobs with high power consumption rates, since all the jobs are sorted in advance.

Condition 1: $\exists k \in A, t_i \leq I_k$.

If Condition 1 is satisfied, job i can be directly processed within an off-peak period with the lowest electricity price. Note that each job processed within a period is placed to the leftmost side of the period. An illustration is given in Figure 3 (i.e., job 3 is a job to be inserted, and $t_3 < I_4$). It is noteworthy that the smaller the number of the job, the greater the power consumption rate.

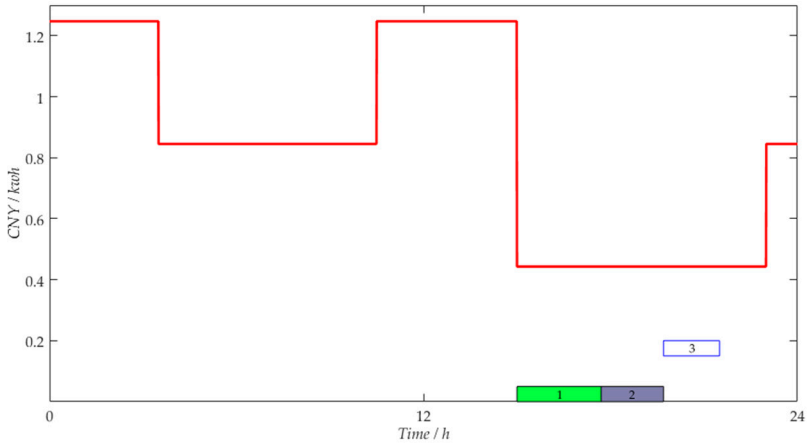


Figure 3. Illustration of Condition 1.

Condition 2: $t_i > \max_{k\in A}\{I_k\} > 0$ and $d_{k+1} > 0$.

When Condition 2 is satisfied, it means that job i cannot be processed within an off-peak period. As a second-best choice, job i can be processed across periods k and $k + 1$ in such a condition. An illustration is given in Figure 4. Note that when the job 5 is inserted into the position, it should be adjacent to job 2.

Layer 2 includes Conditions 3–5. If $t_i > \max_{k\in A}\{I_k\} + I_{k+1}$ and $\exists k' \in B, t_i \leq I_{k'}$, job i is assigned to layer 2.

Condition 3: $\max_{k\in A}\{I_k\} > 0$ and $d_{k+2} > 0$.

Since $d_{k+2} > 0$, it follows that period $k + 2$ always exists. To minimize the electricity cost for processing job i, the off-peak period with maximal remaining idle time (period k) should be given

priority, compared with other off-peak, mid-peak, and on-peak periods. Thus, job i is preferred to be processed across periods k, $k+1$, and $k+2$. Meanwhile, the corresponding electricity cost is named as $cost1$. However, if job i is processed within a mid-peak period (i.e., the corresponding electricity cost is named as $cost2$), the electricity cost may be lower. Hence, two positions are considered and an illustration is given in Figure 5. Let c_A, c_B, and c_Γ represent the electricity prices of off-peak, mid-peak, and on-peak periods, respectively. To select the optimal position, the key property 1 is given as follows.

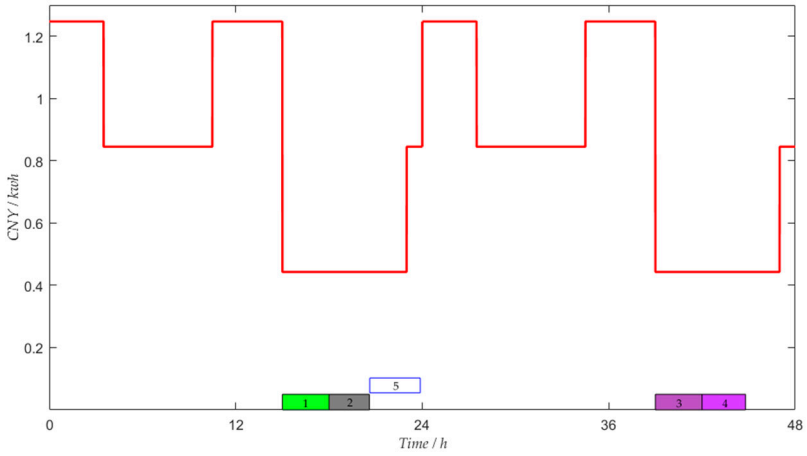


Figure 4. Illustration of Condition 2.

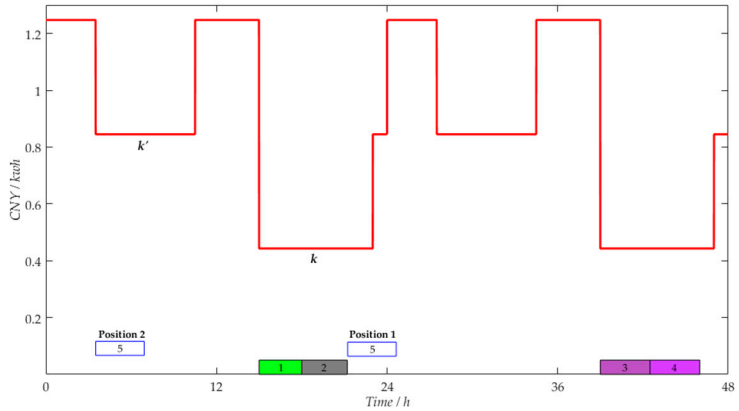


Figure 5. Illustration of Condition 3.

Property 1. *If Condition 3 is satisfied and* $x_{i,k} \times (c_B - c_A) < x_{i,k+2} \times (c_\Gamma - c_B)$ *holds, the best choice for job i is to be processed within a mid-peak period.*

Proof. $cost1 = p_i \times (x_{i,k} \times c_A + x_{i,k+1} \times c_B + x_{i,k+2} \times c_\Gamma)$ and $cost2 = p_i \times t_i \times c_B$. It is assumed that $cost1 > cost2$, that is, $cost1 - cost2 = p_i \times (x_{i,k} \times c_A + x_{i,k+1} \times c_B + x_{i,k+2} \times c_\Gamma) - p_i \times t_i \times c_B > 0$. Since $t_i = x_{i,k} + x_{i,k+1} + x_{i,k+2}$, it follows that:

$$x_{i,k} \times (c_B - c_A) < x_{i,k+2} \times (c_\Gamma - c_B). \tag{8}$$

Therefore, when inequality (8) holds, job i can be directly processed within a mid-peak period. Otherwise, job i must be processed across periods k, $k+1$, and $k+2$. This ends the proof. □

Condition 4: $\max_{k \in A}\{I_k\} > 0$ and $d_{k+2} = 0$.

The only difference between Conditions 3 and 4 is whether $d_{k+2} = 0$ or not. This implies that period k is the last off-peak period and period $k + 2$ does not exist. Hence, period k must be in the last cycle (i.e., on the last day) and there are two scenarios that should be considered as follows (the zero points of the two scenarios are different).

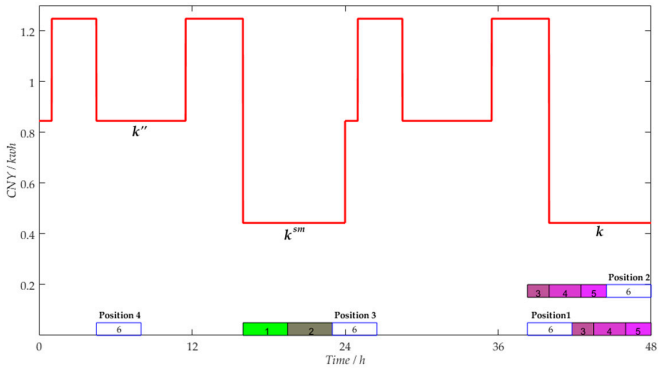
Scenario 1: In this scenario, neither period $k + 1$ nor period $k + 2$ exists and four possible positions marked in Figure 6a are given. Similar to Condition 3, period k is critical due to its lowest price and the longest remaining idle time. To insert job i into period k, the other jobs processed within period k should be moved right or left. Hence, two positions are analyzed first.

1. To occupy off-peak periods as much as possible, a set of already inserted jobs in period k should be moved to the rightmost side, and then job i can be processed across periods k and $k - 1$.
2. All inserted jobs in period k should be moved left so that job i can be processed within period k.

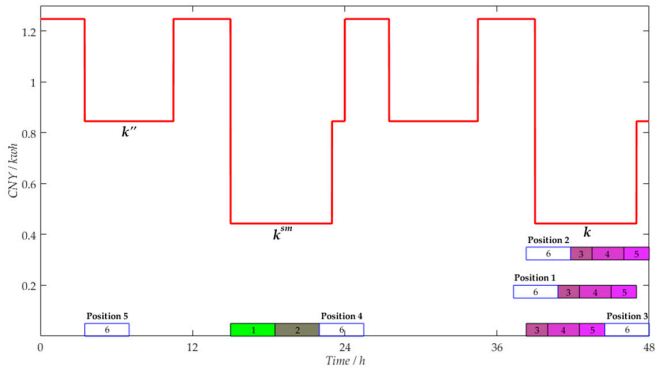
Note that the movement of jobs may lead to a change in their electricity costs, defined as movement cost. The insertion cost of job i is equal to the electricity cost for processing job i plus the corresponding movement cost if some already inserted jobs must be moved to enlarge the idle time of period(s). If the movement cost is zero, then the insertion cost is equal to the electricity cost for processing the job.

Let $cost1$ and $cost2$, respectively, denote the insertion costs of the above corresponding positions. Since the power consumption rate of job i is not higher than any one of the already inserted jobs, it follows that $cost1 \leq cost2$. That is, the Position 1 is the only one that should be considered when job i is inserted into period k. Next, let k^{sm} denote the off-peak period with submaximal remaining idle time. Positions 3 and 4 are given as follows.

3. Suppose that job i is processed across periods k^{sm}, $k^{sm} + 1$, and $k^{sm} + 2$. If I_k^{sm} is slightly smaller than I_k, $cost3$ may be less than $cost1$ as period $k^{sm} + 1$ is a mid-peak period. Hence, Position 3 needs to be considered.
4. Similar to Condition 3, job i can be processed within a mid-peak period.



(**a**) Illustration of Scenario 1



(**b**) Illustration of Scenario 2

Figure 6. Illustration of Condition 4.

To ensure that $cost1 = p_i \times (x_{i,k} \times c_A + x_i x_{i,k-1} \times c_\Gamma)$ always holds, Property 2 is given first.

Property 2. *When Condition 4 is satisfied and $d_{k+1} = 0$, job i can be directly inserted into Position 1 without moving any already inserted jobs in period $k-1$.*

Proof. Since $\exists k \in A$, $I_k > 0$, there must be no jobs processed within period $k-1$. It is only possible that a job, say job j, $j < i$, is processed across periods $k-2$ and $k-1$. Therefore, $x_{j,k-2}$ may be the value of the maximal remaining idle time of all mid-peak periods before inserting job j. Since $\exists k' \in B$, $t_i \leq I_{k'}$ and $j < i$, it is understandable that $t_i \leq I_{k'} \leq x_{j,k-2}$. Now, job i is to be inserted, it follows that $t_i + x_{j,k-1} \leq x_{j,k-2} + x_{j,k-1} = t_j$. As mentioned earlier, the processing times of all the jobs do not exceed the duration of the shortest on-peak period, that is, $t_j \leq d_{k-1}$. Hence, $t_i + x_{j,k-1} \leq d_{k-1}$. If job i is processed across periods k and $k-1$, then $x_{i,k-1} + x_{j,k-1} \leq x_{i,k-1} + x_{i,k} + x_{j,k-1} = t_i + x_{j,k-1} \leq d_{k+1}$. That is, $x_{i,k-1} + x_{j,k-1} \leq d_{k-1}$. Thus, $d_{k-1} - x_{j,k-1} - x_{i,k-1} = I_{k-1} \geq 0$. This suggests when job i is inserted into Position 1, period $k-1$ cannot be full. Hence, job i can be directly inserted into Position 1 without moving any already inserted jobs in period $k-1$. Note that this property applies to Scenario 2 as well. □

According to Property 2, $cost1 = p_i \times (x_{i,k} \times c_A + x_{i,k-1} \times c_\Gamma)$ is always satisfied. In the following part, three formulas for calculating the insertion costs of Positions 1, 3, and 4 are given.

$$cost1 = p_i \times (x_{i,k} \times c_A + x_{i,k-1} \times c_\Gamma); \tag{9}$$

$$cost3 = p_i \times (x_{i,k^{sm}} \times c_A + x_{i,k^{sm}+1} \times c_B + x_{i,k^{sm}+2} \times c_\Gamma); \tag{10}$$

$$cost4 = p_i \times t_i \times c_B. \tag{11}$$

Since $cost1$ is always less than $cost2$, there is no need to compute $cost2$. Eventually, the insertion costs of all the possible positions that job i can be inserted into can be easily and directly calculated by the above formulas, and then the position with minimum insertion cost will be chosen.

Scenario 2: It can be seen from Figure 6b that the Positions 1, 4, and 5 in Scenario 2 are the same as the Positions 1, 3, and 4 in Scenario 1. The only difference between Scenarios 1 and 2 is whether $d_{k+1} > 0$ or not (i.e., period $k+1$ exists). Since period $k+1$ is a mid-peak period, two additional positions need to be considered in comparison with Scenario 1.

1. A set of already inserted jobs in period k are moved to the rightmost side of the period $k+1$, and then job i is processed across periods k and $k-1$.
2. A set of already inserted jobs in period k should be moved to the left until job i can be processed across periods k and $k+1$.

The size of the two insertion costs (i.e., the processing electricity cost of job 6 plus the movement costs of jobs 3, 4, and 5) corresponding to Positions 2 and 3 is uncertain, because the electricity cost for processing job i is greater at Position 2 than at Position 3, while the movement costs are just the opposite. Eventually, it should calculate insertion costs of five possible positions, and then choose the position with minimum cost.

Condition 5: $\forall k \in A$, $I_k = 0$.

When Condition 5 is satisfied, this means all the off-peak periods are full and job i can be directly processed within a mid-peak period.

Layer 3 includes Conditions 6 and 7. Most of the jobs are processed across a pair of periods consisting of a mid-peak period and an on-peak period or processed within an on-peak period in this layer.

Condition 6: $t_i > \max_{k' \in B}\{I_{k'}\} > 0$.

If Condition 6 is satisfied, it means job i cannot be processed within any one of the mid-peak periods, let alone off-peak periods. In other words, job i can only be used to fill the remaining idle time of a certain period with lower electricity price. There is an obvious regularity that the more the remaining idle time of the off-peak or mid-peak period job i occupy, the lower is its processing electricity cost. Figure 7 shows all the possible positions for a scenario. The analysis process of the possible positions that job i can be inserted into is similar to Condition 4, and, therefore, will not be repeated here.

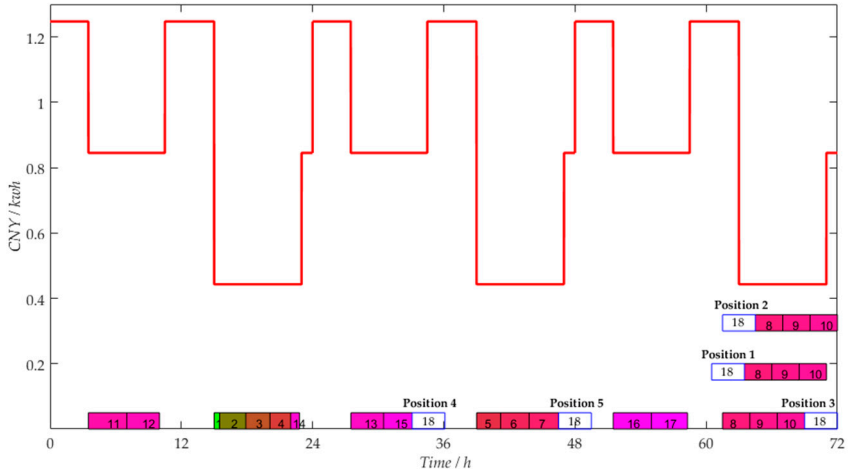


Figure 7. Illustration of Condition 6.

Condition 7: $\max_{k' \in B}\{I_{k'}\} = 0$.

Condition 7 implies that all the off-peak periods and mid-peak periods are full, and job i can only be inserted into on-peak periods. If $\max_{k'' \in \Gamma}\left\{I_{k''}\right\} < t_i$, job i should be inserted into all non-full on-peak periods by moving a set of already inserted jobs, and then the position with the lowest insertion cost can be chosen. The movement method can refer to Che et al. [9].

The core component of the heuristic algorithm is described as a pseudo code, shown in Algorithm 1. Note that the argmin$(I_k \geq t_i)$ denotes the minimal index k for $I_k \geq t_i$.

Theorem 1. *The proposed heuristic algorithm runs in $O(n^2 m \mid \Gamma \mid)$ in the worst case.*

Proof. Step 1 runs in O($n$log$n$) time for sorting n numbers and Step 2 requires O(m) time. For each given job, Step C1 runs in O($\mid A \mid$) in the worst case and Step C2 requires O(1). Steps C3 and C5 both require O($\mid B \mid$) operations in the worst case. Steps C4 and C6 demand O(nm) to compute the movement cost when calculating the insertion cost. Step C7 includes steps C7.1 and C7.2, wherein the complexity of step C7.1 is O($\mid \Gamma \mid$), and the complexity of step C7.2 is O($nm \mid \Gamma \mid$). Therefore, step C7 requires O($nm \mid \Gamma \mid$) operations in the worst case. To summarize, the proposed algorithm runs in O($n^2 m \mid \Gamma \mid$) in the worst case. □

Now, assume that no jobs need to traverse all non-full on-peak periods, that is, the Step C7.2 has never been used. In this case, the time complexity of the proposed algorithm is O($n^2 m$). However, the classic greedy insertion algorithm proposed by Che et al. [9] requires O($n^2 m^2$) operations in the worst case when dealing with the same problem, because their algorithm requires all the jobs to traverse all non-full periods to find an optimum position.

Algorithm 1: Greedy insertion heuristic algorithm with multi-stage filtering mechanism

1. Sort all jobs in non-increasing order of their power consumption rates
2. Initialization: $I_k = b_{k+1} - b_k$, for all $1 \leq k \leq m$
3. **For** $i = 1$ to n **do**
   3.1. **If** layer 1
      C1. **If** Condition 1 is satisfied
      Initial the period index $kk = \text{argmin}_{k \in A}(I_k \geq t_i)$
      //Job i is processed within period kk.
      C2. **Else if** Condition 2 is satisfied
      //Job i is processed across periods k and $k + 1$.
   3.2. **Else if** layer 2
      C3. **If** Condition 3 is satisfied
         C3.1. **If** inequality (8) is not satisfied
         //Job i is processed across periods k, $k + 1$, and $k + 2$.
         C3.2. **Else**
         Initial the period index $kk' = \text{argmin}_{k' \in B}(I_{k'} \geq t_i)$
         //Job i is processed within period kk'.
      C4. **Else if** Condition 4 is satisfied
         C4.1. **If** $d_{k+1} = 0$
         //Calculate $cost1$, $cost3$, and $cost4$ and insert job i into the position with minimal insertion cost.
         C4.2. **Else if** $d_{k+2} = 0$ and $d_{k+1} > 0$
         //Calculate $cost1$, $cost2$, $cost3$, $cost4$, and $cost5$ and insert job i into the position with minimal insertion cost.
      C5. **Else if** Condition 5 is satisfied
      Initial the period index $kk' = \text{argmin}_{k' \in B}(I_{k'} \geq t_i)$
      //Job i is processed within period kk'.
   3.3. **Else if** layer 3
      C6. **If** Condition 6 is satisfied
      //Similarly to Condition 4, it needs to calculate the insertion cost of several possible positions and insert job i into the position with minimal insertion cost.
      C7. **Else if** Condition 7 is satisfied
         C7.1. **If** $\max_{k'' \in \Gamma}\left\{ I_{k''} \right\} > t_i$
         Initial the period index $kk'' = \text{argmin}_{k'' \in \Gamma}\left(I_{k''} \geq t_i \right)$
         //Job i is processed within period kk''.
         C7.2. **Else**
         //Job i traverses all non-full on-peak periods and insert job i into the position with minimal insertion cost.

4. Computational Results

In this section, a real-life instance from a machinery manufacturing company in China is provided to further illustrate the MILP model and the proposed algorithm. Then, two sets of contrasting experiments with randomly generated instances are conducted, aiming to show the good performance of the algorithm. The algorithm is coded in MATLAB R2015b and the experiments are run on an Intel(R) Core(TM) i7-4790 3.60 GHz processor with 16 GB of memory under the Windows 7 operating system. For benchmarking, the greedy insertion heuristic algorithm proposed by Che et al. [9] is adopted for our contrast tests.

The TOU electricity tariffs used for all instances are those implemented in Shanxi Province, China, as given in Table 1. It can be seen from Table 1 that the off-peak period is between an on-peak period and a mid-peak period, which means that the actual electricity prices meets the first type of TOU electricity tariffs. Assume that the time horizon starts at 8:00 a.m. of the first day.

Table 1. The TOU tariffs used for all instances.

Period Type	Electricity Price (CNY/kwh)	Time Periods
On-peak	1.2473	8:00–11:30 18:30–23:00
Mid-peak	0.8451	7:00–8:00 11:30–18:30
Off-peak	0.4430	23:00–7:00

C_{max} and C_B denote the given makespan and the total processing times of all the jobs, respectively. The parameter $e = C_{max}/C_B$ ($e \geq 1$) is used to measure the degree of time tightness. In these instances, C_{max} can be obtained by the formula $C_{max} = e \times C_B$ as long as the parameter e is set. Obviously, as e increases, C_{max} increases. Note that b_{m+1} is calculated by $b_{m+1} = \lceil C_{max}/24) \rceil \times 24$. Let TEC_F and TEC_H be the total electricity cost calculated by our algorithm (GIH-F)and Che et al.'s algorithm (GIH), respectively. The runtimes of the two algorithms are represented by CT_F and CT_H, respectively. The gaps between TEC_F and TEC_H are represented by G, $G = (TEC_F - TEC_H)/TEC_H \times 100\%$. The ratio of CT_H/CT_F is represented by R.

4.1. A Real Case Study

The MILP model and the proposed algorithm are applied to a vertical machining center (SMTCL VMC 1600P) from a machinery manufacturing company shown in Figure 8. In this real-life instance, the company receives some orders of processing rectangular parts with three product models for continuous casting machines of the steel plants. Figure 9 is the picture of the rectangular part, and the complex cavity surfaces including the planes, camber surfaces, and spherical surfaces are to be processed on the VMC. The power consumption per hour required by the VMC is related to the material of the job, cutting depth, feed speed, and so on. To obtain the average power consumption rate of the machine, a power measurement is performed using a power meter (UNI-T UT232), and the measurement data is presented by Table 2. In addition, the order quantity of the three models is 15, 35, and 10 parts, respectively.

Figure 8. Vertical machining center.

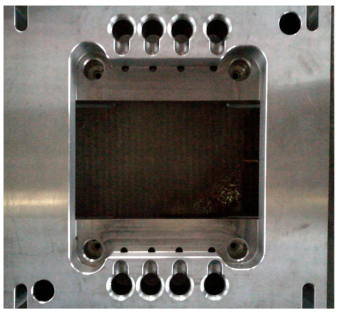


Figure 9. The geometry of the rectangular part.

Table 2. The measurement data of the real-life instance.

Product Model	Average Power Consumption Rate (kW)	Processing Time (h)	The Number of Parts
40	4.4	2.4	15
70	4.7	2.6	35
100	5.3	3.1	10

Let us temporarily put aside the above real-life instance, and talk about solving a hypothetical instance given in Tables 3 and 4 based on the above data. Specifically, it is assumed that there are twelve parts which need to be processed within two days. That is, the number of periods is ten (i.e., $m = 10$). According to this example, we will demonstrate the MILP model and the main computational process of GIH-F.

Table 3. A hypothetical instance based on real-life data.

Part (Job)	1	2	3	4	5	6	7	8	9	10	11	12
Processing time (h)	2.4	2.4	2.4	2.4	2.6	2.6	3.1	3.1	3.1	3.1	3.1	3.1
Power consumption rate (kW)	4.4	4.4	4.4	4.4	4.7	4.7	5.3	5.3	5.3	5.3	5.3	5.3

Table 4. TOU tariffs for the hypothetical instance.

Period	1	2	3	4	5	6	7	8	9	10
Duration (h)	3.5	7	4.5	8	1	3.5	7	4.5	8	1
Price (CNY/kwh)	1.2473	0.8451	1.2473	0.443	0.8451	1.2473	0.8451	1.2473	0.443	0.8451

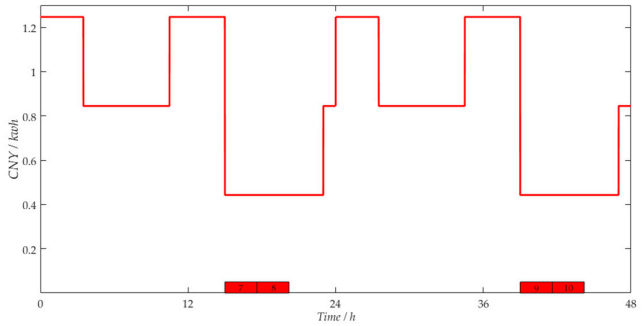
First, the twelve jobs are sorted in non-increasing order according to their power consumption rates, that is, job 7, job 8, job 9, job 10, job 11, job 12, job 5, job 6, job 1, job 2, job 3, and job 4. Second, the remaining idle time of all the periods are initialized. Obviously, $t_7 \leq \max_{k \in A}\{I_k\} + I_{k+1}$ and $t_7 \leq I_4$(i.e., $k = 4$), hence job 7 can be inserted into the low-price layer 1 and is placed in the position corresponding to the Condition 1. The same applies to jobs 8, 9, and 10. In this way, the off-peak periods are fully utilized by the jobs with high power consumption rates, resulting in lower total electricity costs. At this stage, the remaining idle time of each period is as follows: $I_1 = 3.5$, $I_2 = 7$, $I_3 = 4.5$, $I_4 = 1.8$, $I_5 = 1$, $I_6 = 3.5$, $I_7 = 7$, $I_8 = 4.5$, $I_9 = 1.8$, $I_{10} = 1$. An illustration is given in Figure 10a.

Now, $t_{11} > \max_{k \in A}\{I_k\} + I_{k+1}$ and $\exists k, \in B, t_{11} \leq I_{k,}$ (i.e., $k = 4$ and $k' = 2$), hence job 11 is assigned to medium-price layer 2, and is placed in the position corresponding to the Condition 3 because $\max_{k \in A}\{I_k\} > 0$ and $d_{k+2} = 3.5 > 0$. Moreover, $x_{i,k} \times (c_B - c_A) > x_{i,k+2} \times (c_\Gamma - c_B)$ where $k = 4$ and $i = 11$, thus, job 11 is to be inserted into the position across periods 4–6. At this stage, the remaining idle time of each period is as follows: $I_1 = 3.5$, $I_2 = 7$, $I_3 = 4.5$, $I_4 = 0$, $I_5 = 0$, $I_6 = 3.2$, $I_7 = 7$, $I_8 = 4.5$, $I_9 = 1.8$, $I_{10} = 1$. An illustration is given in Figure 10b.

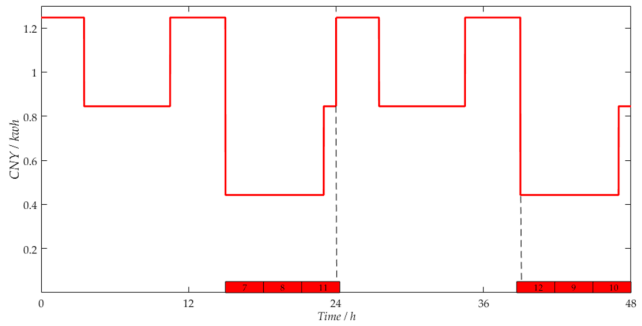
Let us continue, and it is not hard to check that the job 12 is to be inserted into the position corresponding to Condition 4. As mentioned earlier, there will be five positions waiting for selection at this moment. The insertion costs for these five positions are 12.8, 10.7, 10.7, Inf, and 13.9, respectively. Therefore, job 12 is to be inserted into the Position 2 which crosses periods 8 and 9. Note that $I_k^{sm} = 0$ (i.e., $I_4 = 0$), hence job 12 cannot be processed across periods k^{sm}, $k^{sm} + 1$, and $k^{sm} + 2$, and the corresponding insertion cost is infinity. At this stage, the remaining idle time of each period is as follows: $I_1 = 3.5$, $I_2 = 7$, $I_3 = 4.5$, $I_4 = 0$, $I_5 = 0$, $I_6 = 3.2$, $I_7 = 7$, $I_8 = 4.2$, $I_9 = 0$, $I_{10} = 0$. An illustration is given in Figure 10b.

Next, let us explain the insertion process of other jobs concisely. Jobs 5 and 6 are assigned to layer 2, and they satisfy Condition 5. Therefore, these two jobs are to be inserted into mid-peak period 2. Similarly, jobs 1 and 2 are to be inserted into period 7. At this stage, the remaining idle time of each period is as follows: $I_1 = 3.5$, $I_2 = 1.8$, $I_3 = 4.5$, $I_4 = 0$, $I_5 = 0$, $I_6 = 3.2$, $I_7 = 2.2$, $I_8 = 4.2$, $I_9 = 0$, $I_{10} = 0$. An illustration is given in Figure 10c.

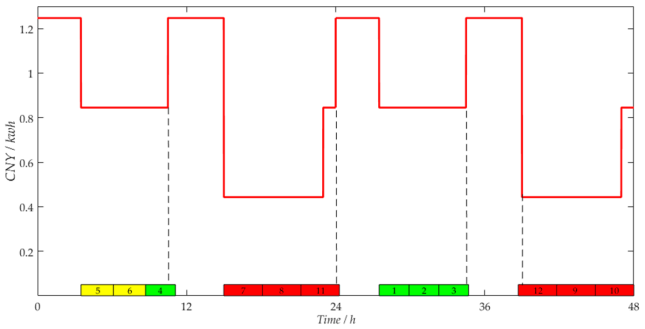
Finally, jobs 3 and 4 are assigned to high-price layer 3, and they both satisfy Condition 6. A final complete scheduling diagram is given in Figure 10c.



(**a**) Computational process diagram 1



(**b**) Computational process diagram 2



(**c**) Computational process diagram 3

Figure 10. Illustration of the computational process of GIH-F.

Returning to the real-life instance, let us now talk about the efficiency of the MILP model and the proposed algorithm. Currently, the company plans to produce 70-Model, 40-Model, and 100-Model rectangular parts in 7, 3, and 2 days respectively. That is, five parts are to be produced every day from 8:00 to 24:00. This suggests that it needs 12 days to process all the parts and the total electricity cost (TEC) can be computed as follows:

$$TEC_{70} = 4.7 \times (3.5 \times 1.2473 + 7 \times 0.8451 + 2.5 \times 1.2473) \times 7 = 440.8 \text{ CNY};$$

$$TEC_{40} = 4.4 \times (3.5 \times 1.2473 + 7 \times 0.8451 + 1.5 \times 1.2473) \times 3 = 160.4 \text{ CNY};$$

$$TEC_{100} = 5.3 \times (3.5 \times 1.2473 + 7 \times 0.8451 + 4.5 \times 1.2473 + 0.5 \times 0.4430) \times 2 = 170.8 \text{ CNY};$$

$$TEC = TEC_{70} + TEC_{40} + TEC_{100} = 772.0 \text{ CNY}.$$

Figure 11 is the scheduling result diagram of the real-life instance. It can be seen from Figure 11 that with our scheduling, the total electricity cost for processing all the parts is 447.9 CNY, which can be reduced by 42.0%.

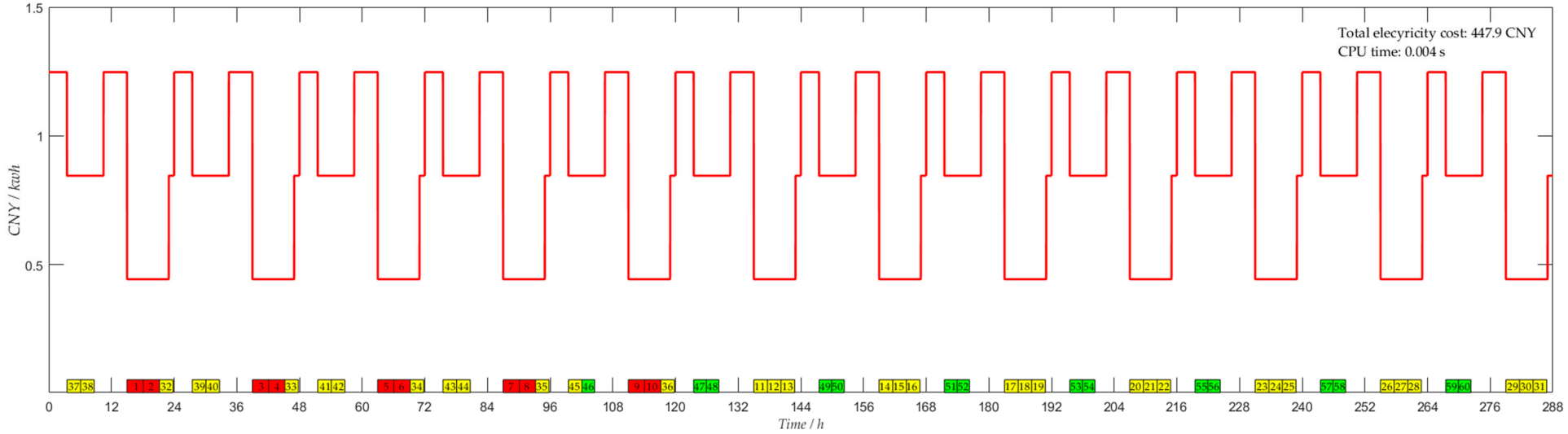


Figure 11. The scheduling result of the real-life instance.

4.2. Randomly Generated Instances Studies

In these tests, we utilize two sets of randomly generated instances to evaluate the performance of the proposed algorithm. For the first set of small-size instances, the range of jobs is [20, 100]. For the second set of large-size instances, the number of jobs is set as 500, 1000, 2000, 3000, 4000, and 5000. The processing time t_i is randomly generated from a uniform distribution (30, 210) min and the power consumption per hour p_i is randomly generated in (30, 100) kW. To measure the effect of the proposed algorithm, parameter e is set as e = 1.2, 1.5, 2, 3.

For each group of n ($n \leq 2000$) and e, 10 random instances are generated, then the average values of 10 tests are calculated and recorded. When the number of jobs is set as 3000, 4000, and 5000, GIH has to run for more than 4 h (the longest is nearly two days) to find a feasible solution. Thus, considering the feasibility of the experiment, only 3 random instances are generated in such a group of tests. All the average values are recorded in Tables 5 and 6. Meanwhile, for the large-size instances, we add two rules to GIH to reduce the computation time without changing the computational accuracy. The improved algorithm is named GIH2.

Table 5. Computational results for the small-size instances.

Instance			GIH		GIH-F			
n	e	m	TEC_H	CT_H (s)	TEC_F	CT_F (s)	G (%)	R
20	1.2	12.0	1634.1	0.034	1632.5	0.002	−0.10%	17.0
	1.5	15.0	1370.1	0.037	1370.1	0.002	0.00%	18.5
	2.0	19.0	1295.7	0.041	1295.1	0.002	−0.05%	20.5
	3.0	28.5	1168.0	0.056	1168.0	0.001	0.00%	56.0
30	1.2	18.0	2414.6	0.064	2415.7	0.002	0.05%	32.0
	1.5	20.0	2274.6	0.065	2274.1	0.002	−0.02%	32.5
	2.0	28.5	2005.0	0.083	2005.0	0.002	0.00%	41.5
	3.0	39.5	1741.0	0.119	1741.0	0.002	0.00%	59.5
40	1.2	23.0	3342.1	0.096	3342.0	0.004	0.00%	24.0
	1.5	28.0	2900.1	0.109	2899.3	0.003	−0.03%	36.3
	2.0	36.0	2775.6	0.143	2775.0	0.003	−0.02%	47.7
	3.0	52.0	2380.3	0.194	2380.3	0.002	0.00%	97.0
50	1.2	27.5	4242.5	0.137	4242.4	0.005	0.00%	27.4
	1.5	34.0	3733.0	0.164	3732.6	0.003	−0.01%	54.7
	2.0	43.0	3243.8	0.212	3243.2	0.004	−0.02%	53.0
	3.0	64.5	2940.6	0.315	2940.6	0.003	0.00%	105.0
60	1.2	34.0	4820.8	0.204	4819.7	0.006	−0.02%	34.0
	1.5	40.0	4536.5	0.224	4536.3	0.004	0.00%	56.0
	2.0	52.0	4029.0	0.293	4028.9	0.004	0.00%	73.3
	3.0	78.0	3544.1	0.464	3544.1	0.004	0.00%	116.0
70	1.2	37.5	6133.5	0.249	6132.2	0.007	−0.02%	35.6
	1.5	46.0	5416.3	0.303	5416.2	0.007	0.00%	43.3
	2.0	61.0	4676.0	0.413	4675.8	0.004	0.00%	103.3
	3.0	90.0	4024.9	0.643	4024.9	0.005	0.00%	128.6
80	1.2	43.0	7073.1	0.321	7072.9	0.009	0.00%	35.7
	1.5	53.0	6049.6	0.401	6049.6	0.006	0.00%	66.8
	2.0	68.5	5348.1	0.554	5348.1	0.007	0.00%	79.1
	3.0	101.5	4514.4	0.868	4514.3	0.005	0.00%	173.6
90	1.2	48.0	8128.5	0.399	8128.4	0.009	0.00%	44.3
	1.5	58.0	6772.5	0.501	6772.4	0.011	0.00%	45.5
	2.0	77.5	6172.7	0.697	6172.6	0.008	0.00%	87.1
	3.0	104.1	5228.2	1.196	5228.2	0.009	0.00%	132.9
100	1.2	53.5	8623.5	0.509	8622.9	0.017	−0.01%	29.9
	1.5	64.0	7607.1	0.614	7607.0	0.011	0.00%	55.8
	2.0	86.5	6896.8	0.927	6896.8	0.014	0.00%	66.2
	3.0	128.0	5815.2	1.482	5815.1	0.009	0.00%	164.7

Table 6. Computational results for the large-size instances.

Instance			GIH2		GIH-F		
n	e	m	TEC_H	CT_{H2} (s)	TEC_F	CT_F (s)	R
500	1.2	250.5	43,909.1	53.0	43,909.1	0.219	242.0
	1.5	315.0	38,637.8	52.2	38,637.9	0.187	279.1
	2.0	417.0	34,417.5	56.3	34,417.5	0.082	686.6
	3.0	628.5	28,948.1	56.9	28,948.0	0.093	611.8
1000	1.2	504.0	87,500.3	244.2	87,500.3	1.802	135.5
	1.5	628.5	77,598.3	230.3	77,597.0	0.873	263.8
	2.0	840.0	69,199.2	294.1	69,199.2	0.432	680.8
	3.0	1256.5	57,923.1	250.7	57,923.1	0.485	516.9
2000	1.2	1002.5	176,681.2	3910.8	176,680.7	15.701	249.1
	1.5	1255.5	155,205.3	3114.3	155,206.2	6.503	478.9
	2.0	1669.0	137,774.1	4316.2	137,774.1	3.346	1290.0
	3.0	2501.7	115,661.0	1785.8	115,661.0	3.574	499.7
3000	1.2	1511.7	263,511.1	19,136.9	263,511.6	46.551	411.1
	1.5	1880.0	231,954.6	14,429.7	231,954.9	25.560	564.5
	2.0	2483.3	205,368.8	19,759.8	205,368.8	11.219	1761.3
	3.0	3780.0	173,630.6	6571.9	173,630.6	12.432	528.6
4000	1.2	1991.7	352,975.8	59,016.2	352,977.0	107.610	548.4
	1.5	2511.7	306,983.3	43,971.5	306,983.3	66.669	659.5
	2.0	3335.0	275,694.8	60,539.8	275,694.8	26.281	2303.6
	3.0	4986.7	231,148.4	17,014.0	231,148.4	29.728	572.3
5000	1.2	2498.3	438,717.9	136,314.9	438,718.6	168.581	808.6
	1.5	3131.1	386,546.1	101,071.7	386,548.1	106.764	946.7
	2.0	4161.1	341,504.3	139,122.7	341,504.3	50.931	2731.6
	3.0	6257.8	291,685.7	51,471.0	291,685.7	58.821	875.0

Rule 1: If $t_i \leq \max_{k\in A}\{I_k\}$ and $kk = \text{argmin}_{k\in A}\{t_i \leq I_k\}$, where $\text{argmin}_{k\in A}\{t_i \leq I_k\}$ denotes the minimal index k for $t_i \leq I_k$, then job i can be directly inserted into the off-peak period kk and the job no longer traverses all the non-full periods.

Rule 2: If $t_i \leq \min_{k_{''}\in\Gamma}\left\{I_{k_{''}}\right\}$, $\max_{k\in A}\{I_k\} > 0$ or $\max_{k'\in B}\{I_{k'}\} > 0$, then job i no longer traverses on-peak periods.

As mentioned above, when there are no jobs that need to traverse non-full on-peak periods, the time complexity of GIH-F is $O(n^2m)$ and GIH is $O(n^2m^2)$. This implies that GIH-F is m times faster than GIH, theoretically, and the experimental data of the small-size instances in Table 5 can verify this conclusion. From Table 5, we can see that R and m are almost the same order of magnitude. In addition, all the small-size instances can be solved within 0.02 s using GIH-F. By and large, the computation time increases slightly with n, and parameter e has no significant effect on the computation time, which indicates that the algorithm is quite stable. In addition, it can be seen from Table 5 that the smaller the parameter e (i.e., the shorter the makespan), the higher the total electricity cost. Therefore, in a specific instance, the decision-makers can obtain a set of Pareto solutions by adjusting the makespan, and they can choose a solution according to actual needs. What is more, it is amazing to see that our algorithm not only greatly improves computation speed but also further improves the accuracy.

Table 6 shows that the number of periods increases quickly with the number of jobs. Since the time complexity of GIH is $O(n^2m^2)$, it's runtime will rise sharply. To ensure the feasibility of the contrast tests, we add two rules (i.e., Rule 1 and Rule 2) to improve the computational speed of GIH without changing the computational accuracy.

Intuitively, the CT_F is significantly less than CT_{H2}, which means that the designed filtering mechanism is efficient in dealing with large-scale instances. Specially, as n = 5000 and e = 2.0, our algorithm can solve a randomly generated instance within 1 min and maintain the same accuracy as GIH2, while GIH2 takes nearly 39 h, let alone GIH. Note that when e is set as 3.0, the given makespan is very abundant and there is no job processed within an on-peak period in our experimental

environments. Thus, according to Rule 2, all the jobs do not have to traverse on-peak periods, and then CT_{H2} is greatly reduced. Conversely, when e is set as 1.2, the number of periods decreases and the jobs are arranged very tightly. There will be many jobs inserted into the periods with higher electricity prices. Therefore, our algorithm should filter more positions with lower electricity prices and constantly judge whether the job needs to be moved. Obviously, all these operations may increase the computation time. Thus, when dealing with large-size instances and setting e to 1.2 or 1.5, our algorithm runs longer, but the computation time is still far less than GIH2.

5. Conclusions and Prospects

This paper develops a new greedy insertion heuristic algorithm with a multi-stage filtering mechanism for single machine scheduling problems under TOU electricity tariffs. The algorithm can quickly filter out many impossible positions in the coarse granularity filtering stage and then each job to be inserted can search for its optimal position in a relatively large space in the fine granularity filtering stage. Compared with the classic greedy insertion algorithm, the greatest advantage of our algorithm is that it no longer needs to traverse all non-full periods, so the time complexity of the algorithm is quite low, and it can easily address the large-scale single machine scheduling problems under TOU electricity tariffs. The real case study demonstrates that with our scheduling, the total electricity cost for processing all the parts can be reduced by 42.0%. In addition, two sets of experimental instances are provided. The computational results demonstrate that the small-size instances can be solved within 0.02 s using our algorithm, and the accuracy of the algorithm is further improved. For the large-size instances, we add two rules to the classic greedy insertion algorithm, which reduces the computation time without changing the calculation precision, but the results show that our algorithm still outperforms it. Specifically, when addressing the large-scale instances with 5000 jobs, the computation speed of our algorithm improves by nearly 2700 times. Computational experiments also reveal that the smaller the parameter e, the more significant the filtering mechanism is.

This paper focuses on the single machine scheduling problems under the first type of TOU electricity tariffs. In our future research, we will continue to study the problem under the second type of TOU tariffs (i.e., the off-peak period lies between two mid-peak periods). In addition, we will also strive to improve our algorithm and extend it to other machine environments, such as parallel machines and flow shop.

Acknowledgments: This research is supported by the National Natural Science Foundation of China (Grant No. 71772002).

Author Contributions: Hongliang Zhang contributed to the overall idea, algorithm, and writing of the manuscript; Youcai Fang coded the algorithm in MATLAB and contributed to the detailed writing; Ruilin Pan contributed to the ideas and discussions on the scheduling problem under TOU electricity tariffs, as well as the revision, preparation, and publishing of the paper; Chuanming Ge analyzed the characteristics of the single machine scheduling problem under TOU electricity tariffs. All authors have read and approved the final manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. International Energy Agency. *World Energy Investment Outlook*; International Energy Agency (IEA): Paris, France, 2015.
2. Li, C.; Tang, Y.; Cui, L.; Li, P. A quantitative approach to analyze carbon emissions of CNC-based machining systems. *J. Intell. Manuf.* **2015**, *26*, 911–922. [CrossRef]
3. Jovane, F.; Yoshikawa, H.; Alting, L.; Boër, C.R.; Westkamper, E.; Williams, D.; Tseng, M.; Seliger, G.; Paci, A.M. The incoming global technological and industrial revolution towards competitive sustainable manufacturing. *CIRP Ann. Manuf. Technol.* **2008**, *57*, 641–659. [CrossRef]
4. Lu, C.; Gao, L.; Li, X.; Pan, Q.; Wang, Q. Energy-efficient permutation flow shop scheduling problem using a hybrid multi-objective backtracking search algorithm. *J. Clean. Prod.* **2017**, *144*, 228–238. [CrossRef]

5. Sun, Z.; Li, L. Opportunity estimation for real-time energy control of sustainable manufacturing systems. *IEEE Trans. Autom. Sci. Eng.* **2013**, *10*, 38–44. [CrossRef]
6. Park, C.W.; Kwon, K.S.; Kim, W.B.; Min, B.K.; Park, S.J.; Sung, I.H.; Yoon, Y.S.; Lee, K.S.; Lee, J.H.; Seok, J. Energy consumption reduction technology in manufacturing—A selective review of policies, standards, and research. *Int. J. Precis. Eng. Manuf.* **2009**, *10*, 151–173. [CrossRef]
7. Ding, J.Y.; Song, S.; Zhang, R.; Chiong, R.; Wu, C. Parallel machine scheduling under time-of-use electricity prices: New models and optimization approaches. *IEEE Trans. Autom. Sci. Eng.* **2016**, *13*, 1138–1154. [CrossRef]
8. Merkert, L.; Harjunkoski, I.; Isaksson, A.; Säynevirta, S.; Saarela, A.; Sand, G. Scheduling and energy—Industrial challenges and opportunities. *Comput. Chem. Eng.* **2015**, *72*, 183–198. [CrossRef]
9. Che, A.; Zeng, Y.; Ke, L. An efficient greedy insertion heuristic for energy-conscious single machine scheduling problem under time-of-use electricity tariffs. *J. Clean. Prod.* **2016**, *129*, 565–577. [CrossRef]
10. Longe, O.; Ouahada, K.; Rimer, S.; Harutyunyan, A.; Ferreira, H. Distributed demand side management with battery storage for smart home energy scheduling. *Sustainability* **2017**, *9*, 120. [CrossRef]
11. Shapiro, S.A.; Tomain, J.P. Rethinking reform of electricity markets. *Wake For. Law Rev.* **2005**, *40*, 497–543.
12. Zhang, H.; Zhao, F.; Fang, K.; Sutherland, J.W. Energy-conscious flow shop scheduling under time-of-use electricity tariffs. *CIRP Ann. Manuf. Technol.* **2014**, *63*, 37–40. [CrossRef]
13. Che, A.; Wu, X.; Peng, J.; Yan, P. Energy-efficient bi-objective single-machine scheduling with power-down mechanism. *Comput. Oper. Res.* **2017**, *85*, 172–183. [CrossRef]
14. He, F.; Shen, K.; Guan, L.; Jiang, M. Research on energy-saving scheduling of a forging stock charging furnace based on an improved SPEA2 algorithm. *Sustainability* **2017**, *9*, 2154. [CrossRef]
15. Pruhs, K.; Stee, R.V.; Uthaisombut, P. Speed scaling of tasks with precedence constraints. In Proceedings of the 3rd International Workshop on Approximation and Online Algorithms, Palma de Mallorca, Spain, 6–7 October 2005; Erlebach, T., Persiano, G., Eds.; Springer: Berlin, Germany, 2005; pp. 307–319.
16. Fang, K.; Uhan, N.; Zhao, F.; Sutherland, J.W. A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction. *J. Manuf. Syst.* **2011**, *30*, 234–240. [CrossRef]
17. Dai, M.; Tang, D.; Giret, A.; Salido, M.A.; Li, W.D. Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm. *Robot. Comput. Integr. Manuf.* **2013**, *29*, 418–429. [CrossRef]
18. Liu, C.H.; Huang, D.H. Reduction of power consumption and carbon footprints by applying multi-objective optimisation via genetic algorithms. *Int. J. Prod. Res.* **2014**, *52*, 337–352. [CrossRef]
19. Mouzon, G.; Yildirim, M.B.; Twomey, J. Operational methods for minimization of energy consumption of manufacturing equipment. *Int. J. Prod. Res.* **2007**, *45*, 4247–4271. [CrossRef]
20. Mouzon, G.; Yildirim, M. A framework to minimise total energy consumption and total tardiness on a single machine. *Int. J. Sustain. Eng.* **2008**, *1*, 105–116. [CrossRef]
21. Liu, C.; Yang, J.; Lian, J.; Li, W.; Evans, S.; Yin, Y. Sustainable performance oriented operational decision-making of single machine systems with deterministic product arrival time. *J. Clean. Prod.* **2014**, *85*, 318–330. [CrossRef]
22. Luo, H.; Du, B.; Huang, G.Q.; Chen, H.; Li, X. Hybrid flow shop scheduling considering machine electricity consumption cost. *Int. J. Prod. Econ.* **2013**, *146*, 423–439. [CrossRef]
23. Sharma, A.; Zhao, F.; Sutherland, J.W. Econological scheduling of a manufacturing enterprise operating under a time-of-use electricity tariff. *J. Clean. Prod.* **2015**, *108*, 256–270. [CrossRef]
24. Moon, J.-Y.; Shin, K.; Park, J. Optimization of production scheduling with time-dependent and machine-dependent electricity cost for industrial energy efficiency. *Int. J. Adv. Manuf. Technol.* **2013**, *68*, 523–535. [CrossRef]
25. Che, A.; Zhang, S.; Wu, X. Energy-conscious unrelated parallel machine scheduling under time-of-use electricity tariffs. *J. Clean. Prod.* **2017**, *156*, 688–697. [CrossRef]
26. Wang, S.; Liu, M.; Chu, F.; Chu, C. Bi-objective optimization of a single machine batch scheduling problem with energy cost consideration. *J. Clean. Prod.* **2016**, *137*, 1205–1215. [CrossRef]
27. Shrouf, F.; Ordieres-Meré, J.; García-Sánchez, A.; Ortega-Mier, M. Optimizing the production scheduling of a single machine to minimize total energy consumption costs. *J. Clean. Prod.* **2014**, *67*, 197–207. [CrossRef]

28. Gong, X.; Pessemier, T.D.; Joseph, W.; Martens, L. An energy-cost-aware scheduling methodology for sustainable manufacturing. In Proceedings of the 22nd CIRP Conference on Life Cycle Engineering (LCE) Univ New S Wales, Sydney, Australia, 7–9 April 2015; Kara, S., Ed.; Elsevier: Amsterdam, The Netherlands, 2015; pp. 185–190.
29. Fang, K.; Uhan, N.A.; Zhao, F.; Sutherland, J.W. Scheduling on a single machine under time-of-use electricity tariffs. *Ann. Oper. Res.* **2016**, *238*, 199–227. [CrossRef]