Available online at www.ijpe-online.com

**International Journal of Performability Engineering**

vol. 15, no. 7, July 2019, pp. 1849-1859
DOI: 10.23940/ijpe.19.07.p11.18491859

# Heuristic for Hot-Rolled Batch Scheduling of Seamless Steel Tubes with Machine Maintenance and Tardiness

Yang Wang[a,b,*], Tieke Li[a,b], and Bailin Wang[a,b]

[a]*Donlinks School of Economics and Management, University of Science and Technology Beijing, Beijing, 100083, China*
[b]*Engineering Research Center of MES Technology for Iron & Steel Production, Ministry of Education, Beijing, 100083, China*

**Abstract**

Machine maintenance is an indispensable management activity for companies to maintain stability and safety in the process of production. In this paper, the batch scheduling of hot-rolled steel tubes with maintenance and tardiness are considered and abstracted into a single machine scheduling problem with maintenance and tardiness. Combined with the constraint of sequence-dependent setup times, a multi-objective integer programming model is established to minimize the total idle time, total setup time, and total tardiness, and a two-stage local reordering heuristic based on optimization strategy is designed. Finally, comparative experiments are carried out based on actual production data, and the results show that the model and algorithm help alleviate this kind of problem.

*Keywords*: sequence-dependent setup times; machine maintenance; tardiness; local reordering; heuristic algorithm

(Submitted on March 27, 2019; Revised on April 16, 2019; Accepted on June 25, 2019)



## 1. Introduction

The seamless steel tube is an important production material that is widely used in ships, aviation, petrochemical, automotive, and other fields. Machines are extremely worn out in the production process due to the high temperature and fast speeds, so regular maintenance of machines is necessary to ensure product quality and production safety. In the current steel market, customer orders are characterized with small batches and multiple varieties. Therefore, the hole system, thermal tools, and other related parameters of machines are frequently switched because of numerous specifications, so the setup times increase greatly. This affects the continuity and stability of production. At the same time, due to the high requirements for on-time delivery of orders, enterprises must deliver orders on time under the premise of ensuring product quality. In summary, the above factors put forward higher requirements for the production management level of enterprises.

Lee [1] proved that the single machine scheduling problem with maintenance is NP hard. For the scheduling problem with machine maintenance and sequence-dependent setup times, Nesello et al. [2] designed an exact algorithm based on the iterative solution of three alternative arc-time-indexed models to minimize the makespan. Francisco et al. [3] proposed an improved heuristic algorithm based on greedy randomized adaptive search procedures to minimize the makespan. Wu [4] proposed a minimum rolling mill setup times rule and designed a two stage algorithm to minimize the idle time and setup times. Pacheco et al. [5] proposed an improved formulation and designed a skewed variable neighborhood search algorithm that allowed for the movement of the current solution to a worse solution and the incorporation of memory in the search process. Pacheco et al. [6]. hybridized multi-start strategies with tabu search to solve the problem. In terms of the problems of machine maintenance and tardiness, Yin [7] discussed the approximability of the problem and then developed two pseudo-polynomial dynamic programming algorithms and a fully polynomial-time approximation scheme to solve this kind of problem. Benmansour [8] considered two situations, scheduling without availability constraints and under maintenance constraints, and different mixed integer programming methods were designed to minimize the weighted sum of maximum earliness and tardiness costs. Lee [9] designed a two-stage heuristic algorithm to minimize the number of tardiness jobs: at the first stage, an initial solution was obtained with a method modified from Moore's algorithm, and then at the second

* Corresponding author.
*E-mail address*: 1297791019@qq.com

stage, the solution was improved. Sbihi [10] considered two situations with maintenance, fixed maintenance periods and flexible maintenance periods, and proposed a branch-and-bound algorithm to minimize the maximum tardiness. Considering the problem with sequence-dependent setup times and tardiness, Luo [11] designed an iterated filter-and-fan algorithm that minimized the total weighted tardiness and searched the solution space as a manner of a branch tree. Luo also proposed a branching moving strategy with sequence segment recomposing and referenced local search to generate branch nodes. Xu [12] presented an iterated local search algorithm to minimize the total weighted tardiness and designed a new neighborhood structure and a fast incremental evaluation technique for evaluating neighborhood solutions. Guo and Tang [13] proposed a scatter search algorithm with improved component modules. Subramanian [14] proposed a simple effective limitation strategy to speed up the local search procedure, and it was embedded in a local search based meta-heuristic. The above papers analyzed the problem from different aspects and proposed corresponding solutions and methods, which have important reference values for research. Due to the sequence-dependent setup times and order delivery on-time existing in seamless steel tubes and the management requirements of machine maintenance, the existing research results are difficult to directly apply to such problem.

In summary, for the actual production characteristics of seamless steel tubes, they are abstracted into a multi-objective single machine scheduling problem, and a mathematical model for minimizing the setup time, idle time, and tardiness is established. According to the relationship between setup times and tardiness, a local optimization strategy is proposed, and then a two-stage local reordering heuristic algorithm based on optimization strategy is designed. Finally, the feasibility and effectiveness of the algorithm are verified by multiple sets of comparison experiments.

## 2. Problem Description and Formulation

### *2.1. Problem Description*

This study considers the single machine scheduling problem with machine maintenance, which is an important factor that influences the overall production rhythm. In this paper, we assume that no emergency equipment failure or unexpected disruptions happen on accident, the maintenance will be arranged at fixed time, and its duration time is also fixed. We define the time period from the current end time of maintenance to the next start time of maintenance as a rolling planning period (RPP).

Furthermore, sequence-dependent setup times and tardiness are also key factors in the production process. For sequence-dependent setup times, Wu [4] analyzed the characteristic of sequence-dependent setup times and abstracted it as a linear function that is the absolute value of the specification difference between adjacent batches; the setup times can be calculated by $W_{i,i+1} = b + a\,|\,d_{i+1} - d_i\,|$, where $d_i$ and $d_{i+1}$ are the respective specifications of $J_i$ and $J_{i+1}$, and $a$ and $b$ are the model parameters that can be adjusted by the actual situation. For tardiness, the formula can be defined as $\max(0, C_i - dl_i)$, where $C_i$ and $dl_i$ are the completion time and due date of $J_i$, respectively.

Therefore, considering the abovementioned factors, the objective of this problem is to find an optimal solution to minimize the total setup time, idle time, and tardiness. Figure 1 illustrates the representation of a schedule with maintenance, and the parameter information of the batch is given in Table 1. It can be seen from the table that there are two kinds of scheduling methods. The upper part scheduling is sequenced by natural order; through analysis and calculation, the objectives of the setup time, idle time, and tardiness are 9, 7, and 86, respectively. The lower scheduling part is sorted by the scheduling rule, and the calculations show that the setup time, idle time, and tardiness are 5, 3, and 47, respectively. From these two kinds of data results, relevant indicators can be optimized and the economic efficiency can be improved by using different scheduling strategies.

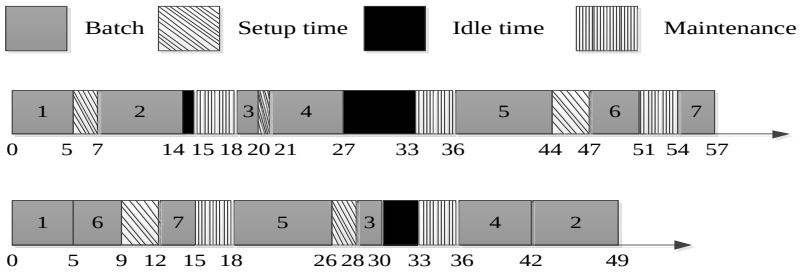


Figure 1. Illustration of batch scheduling

Table 1. Parameters of the batch

| $i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| $p_i$ | 5 | 7 | 2 | 6 | 8 | 4 | 3 |
| $d_i$ | 1 | 3 | 2 | 3 | 4 | 1 | 4 |
| $dl_i$ | 15 | 28 | 21 | 29 | 22 | 25 | 19 |

### *2.2. Notation Definition*

This problem assumes that the job has already been grouped into batches, all the information of the batch is given, and all batches are available at time zero.

There are $n$ batches $J_1, J_2, \cdots, J_n$ to be processed to generate a solution of batch plan $\pi$, and each batch $J_i$ has a processing time $p_i$, a due date $dl_i$, and a specification $d_i$. $D_i$ is the start time of $J_i$, $w_{i,k}$ represents setup times between $J_i$ and $J_k$, and there are $m$ RPPs $Q_1, Q_2, \cdots, Q_m$, where $r$ is the number of RPP. $t$ is the duration time of the machine, $U_r$ and $I_r$ are the respective start time and idle time of the rth RPP, $x_{i0}$ is the starting maintenance after $J_i$ is completed, $x_{0i}$ is the processing batch $J_i$ after maintenance is completed, and 0 indicates machine maintenance. If $J_i$ is the first batch in each RPP, then $x_{0i} = 1$; otherwise, $x_{0i} = 0$. If $J_i$ is the last batch in each RPP, then $x_{i0} = 1$; otherwise, $x_{i0} = 0$. If the specification of $J_i$ is equal to $J_k$, then $y_{i,k} = 1$; otherwise, $y_{i,k} = 0$. If $J_k$ is adjacent to $J_i$, then $z_{i,k} = 1$; otherwise, $z_{i,k} = 0$. Finally, if there is maintenance between $J_i$ and $J_k$, then $t_{i,k} = 1$; otherwise $t_{i,k} = 0$.

### *2.3. Mathematical Model*

Equations (1) to (3) are objective functions. Equation (1) represents minimizing total setup times. Equation (2) represents minimizing total tardiness. Equation (3) represents minimizing total idle time. Equations (4) and (5) indicate that there is only one batch to process before or after maintenance. Equations (6) and (7) calculate the setup times and complete time, respectively. Equation (8) represents that there is only one batch can be processed on the machine at the same time and no setup times are considered after maintenance. Equation (9) shows that the total completion time in each RPP is not greater than the start time of corresponding maintenance.

$$\text{Min} \quad f_1 = \sum_{i=1}^{n-1} \sum_{k=i+1}^{n} w_{i,k} \tag{1}$$

$$\text{Min} \quad f_2 = \sum_{i=1}^{n} \max(0, C_i - dl_i) \tag{2}$$

$$\text{Min} \quad f_3 = \sum_{r=1}^{m} I_r \tag{3}$$

*s.t.*

$$\sum_{i=1}^{n} x_{i0} = r \tag{4}$$

$$\sum_{i=1}^{n} x_{0i} = r \tag{5}$$

$$w_{i,k} = (1 - z_{i,k}) \cdot (1 - y_{i,k}) \cdot (b + a \cdot | d_k - d_i |), \quad i,k \in J \tag{6}$$

$$C_i = \sum_{k=0}^{n-1} \sum_{i=k+1}^{n} ((1 - x_{0k}) \cdot w_{k,i} + p_i), \quad i,k \in J \tag{7}$$

$$D_i \geq (1-t_{i,k}) \cdot (C_i + w_{i,k}) + t_{i,k} \cdot (C_i + t), \quad i,k \in J \tag{8}$$

$$[p_1 + \sum_{i=2}^{n} (p_i + w_{k,i})]_r \leq U_r, \ r \ \in Q \tag{9}$$

## 3. Analysis of Problem Property

Considering the characteristics of sequence-dependent setup times between adjacent batches in seamless steel tubes, the different sequences between batches greatly affect the value of setup times. Combined with the constraint of tardiness, we establish a theorem to locally optimize the problem, as follows:

**Theorem** In sequence $\pi$, the earliest due date (EDD) is used to sort continuous batches with the same specification, and the delay of the new sequence is inevitably inferior to the original sequence.

**Proof** We assume that at least one local sequence exists and is grouped by continuous batch with the same specification in $\pi$. We define it as *G*, and it contains *t* batches ($1 < t < n$), where the batches are not all sorted by the EDD rule. There are at least two adjacent batches $j_i$ and $j_k$, $j_k$ is behind $j_i$, and $dl_i > dl_k$. If the tardiness of the new sequence is not inferior to $\pi$ when the positions of $j_i$ and $j_k$ are exchanged, then the theorem will be proven.

(1) $p_i > p_k$;

a) Both $j_i$ and $j_k$ are tardy, and the tardiness is $E = E_i + E_k = (t + p_i - dl_i) + (t + p_i + p_k - dl_k)$. Only exchange the position of $j_i$ and $j_k$, and the new tardiness is $E' = E_i' + E_k' = (t + p_k - dl_k) + (t + p_i + p_k - dl_i)$. Therefore, $\Delta = E - E' = p_i - p_k$ because of $p_i > p_k$, so $\Delta > 0$;

b) $j_i$ is tardy and $j_k$ is not tardy because of $dl_i > dl_k$, and $j_k$ is behind $j_i$, so this situation does not exist;

c) $j_i$ is not tardy, $j_k$ is tardy, the tardiness of $j_k$ is $E = E_k = t + p_i + p_k - dl_k$, and the tardiness is $E' = E_i' + E_k' = (t + p_k - dl_k) + (t + p_i + p_k - dl_i)$ when the positions of $j_i$ and $j_k$ are exchanged. Then, $\Delta = E - E' = dl_i - (p_k + t)$ because $j_i$ is not tardy, so $dl_i > p_i + t$ and $p_i > p_k$. Therefore, $dl_i > p_k + t$, so $\Delta > 0$;

d) Both $j_i$ and $j_k$ are not tardy, so $dl_i > dl_k \geq p_i + p_k + t$ because the total completion time remains unchanged after changing the position of $j_i$ and $j_k$. Therefore, there is still no tardiness.

(2) $p_i < p_k$ and $p_i = p_k$;

The process of the proof is similar to (1), so it will not be repeated here.

## 4. Heuristic Algorithm

### *4.1. Solution Strategy*

The single machine scheduling problem studied in this paper is a kind of multi-objective integer programming problem. It is difficult to be solved by an exact algorithm to find the solution in effective time. Combining the characteristics of machine maintenance and batch processing, which will not be interfered by each other, a local reordering heuristic algorithm based on optimization strategy (LRHOS) is designed. LRHOS contains two stages: the initial sorting stage and the local reordering optimization stage. Details are shown in Figure 2.

### *4.2. Initial Sorting Stage*

At this stage, batches are sorted according to the EDD rule to generate an initial set of batches, which is then sequentially inserted into the corresponding RPP to form an initial batch planning according to the maintenance time. The pseudo-code

is as follows:

Input: batch sets $J$, machine maintenance planning sets $Q$;
Set $i=1$, $j=1$, $r=1$, $\pi_r=\pi=\varnothing$, $T=0$;
Sort $J$ using EDD to get $J'$, $J \leftarrow J'$;
While ($i \le n$)
  Insert $J_i$ into $\pi_r$;
  If ($j>1$)
    $T=p_i$;
  Else $T=p_i+w_{i-1,i}$;
  If ($T \ge U_r$)
    $j=1$, $r=r+1$, $i=i-1$;
    $\pi \leftarrow \pi \cup \pi_r$, $\pi_r=\varnothing$;
  Else
    Insert $J_i$ into $\pi_r$, delete $J_i$ from $J$;
    $j=j+1$, $i=i+1$;
  If ($J=\varnothing$)
    Output $\pi$, break;
  Else continue;

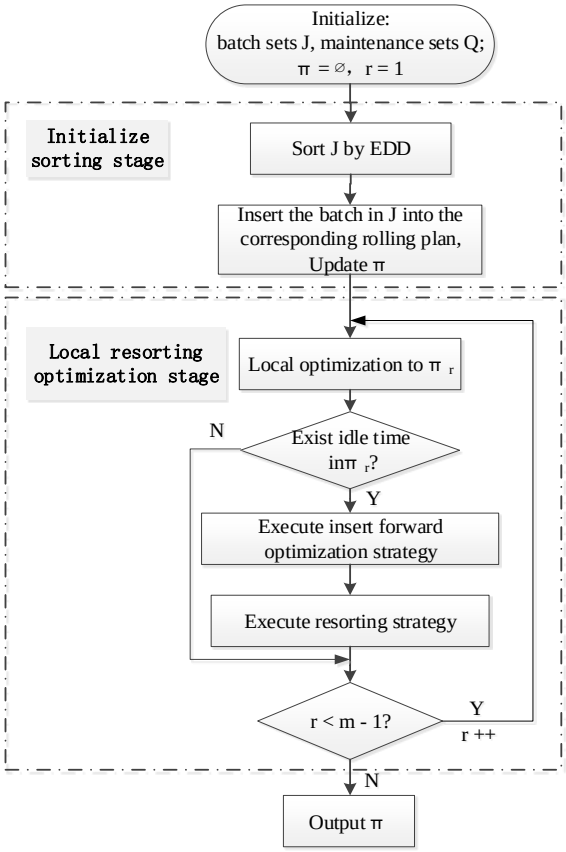



Figure 2. The diagram of algorithm

### *4.3. Local Reordering Optimization Stage*

The local reordering optimization stage is based on the initial sorting stage that generates the batch planning, and it reorders batches according to the idle time. Its main purpose is to reduce the idle time in each RPP, which will maximize the efficiency of the machine, and reduce the tardiness penalty.

At this stage, three operation processes are designed: insert forward in group (IFG), local reordering (LR), and local

optimization (LO).

(1) IFG: Idle time in RPP is the basis for the execution of the insertion forward operation. If there is a batch that satisfies the requirement, the insertion forward operation is triggered, and it will be inserted into the idle location. The general insert operation is composed by one batch; however, it has the disadvantage of repeated traversal and omission of optimal batch, so IFG is designed.

IFG refers to insert forward the batch, which satisfies the requirement (call it the batch pool to be selected, BPS) in the group in order to reduce the searching complexity. Set the group length to three, which means a maximum of three batches will be grouped to insert forward. At the same time, the batches that are obviously not in BPS are excluded according to the remaining idle time, which will be helpful for speeding up the search. The details of the pseudo-code are as follows:

---

Set $r=1$, $l=1$, $length=3$, $J_{best}=\varnothing$, $F_{best}=0$,
  $I_r$ represents the idle time in $\pi_r$;
While ( $l \le length$ )
  For $t = r + 1 \cdots m$
    If ( $J_i$ satisfies the requirement)
      $J_{best} = J_{best} \cup J_i$;
      $F_{best} = F_{best} + p_i + w_{k,i}$;
  If ( $I_r - F_{best} \ge 0$ )
    If ( $J_{best} \ne \varnothing$? )
      Insert $J_{best}$ into the idle location;
      Update $\pi$;
  Else
    $l=l+1$;

---

(2) LR: If there are batches that are inserted into the current RPP, they will disturb the batch sequence in the subsequent RPP, so LR is designed. There are two things to do for LR: first, the sequences in remaining batches will be reordered, and then the batches will be inserted into the corresponding RPP again.

---

While ( $r < m-1$ )
  If (There exists $J_i$ in $\pi_r$ be inserted into $\pi_{r-1}$ )
    For $t = r + 1 \cdots m$
      Reordering $\pi_t$;
      $\pi \leftarrow \pi \cup \pi_t$;
    Updating $\pi$ according to machine maintenance plan;
  Else $r = r+1$;

---

(3) LO: Two rules are designed in LO based on the theorem: the same specification delivery time priority rule (SSDP) and the minimum specification distance rule between batches (MSDB). SSDP aims to reorder continuous batches with the same specification by EDD to optimize the total tardiness and keep the total setup times stable. For MSDB, if there are continuous batches with the same delivery time, they will be selected as the first batch, which has the smallest distance from the previous batch specification in current continuous batches. Remaining batches are sorted according to the smallest distance rule, which can optimize setup times and speed up the convergence of the algorithm under the premise that the total tardiness is not increased. The pseudo-codes of SSDP and MSDB are as follows:

---

**SSDP:**

---

Set $r=1$, $i=2$, *spec* represents the specification of the first batch, $J'=\varnothing$, $J''=\varnothing$, $\pi'=\varnothing$;
For $i$ =1$\cdots n$
  If ( $J_i \in \pi_r$ )
    $J' \leftarrow J' \cup J_i$;
  Else
    For $t = r + 1 \cdots count\ J'$
      If ( $d_i$=*spec* )

$J'' \leftarrow J'' \cup J_t'$;
Else
Sorting $J''$ according to EDD;
$\pi' \leftarrow \pi' \cup J''$;
$r = r+1$, $J' = \varnothing$, $J'' = \varnothing$, $spec \leftarrow d_i$;

**MSDB:**

Initialize: *rank* represents grade, set $rank = 1$, $\pi' = \varnothing$, $\pi'' = \varnothing$, $h = 1$, $k = 1$;
For $r$ =1,⋯,$m$
Sequence $\pi_r$ order by specification in ascending order where $rank = h$;
$\pi' \leftarrow \pi_r$, set the specification of last batch in $\pi'$ is $d$, $h = h+1$;
If ( $h > k$ )
$k = 1$, $h = 1$;
Continue;
Else
Set the quantity of batch in $\pi'$ is $t$, $a = 1$;
Calculate the distance between *rank* and $d$;
Insert the batch $J_i$ to $\pi''$ where $J_i$ has the smallest distance, $d = d_a$, delete $J_i$ from $\pi'$;
If ( $a = t$ )
Output $\pi \leftarrow \pi'$
Else $a = a+1$;

## 5. Computational Experiment

### *5.1. Experimental Data*

In order to verify the effectiveness and feasibility of LRHOS, an experiment is carried out with actual data from a large seamless steel tube factory. We choose one month of batch data in a database and use seven samples of batches (50, 80, 100, 120, 150, 180, and 200). The basic properties of batches contain the batch ID, rolling count, batch specification, batch due date, and so on. The specific parameters are as follows: the rolling count ranges from [20, 250], the specification ranges from [1, 20], and the due date ranges from [1, 31] ×24.

This paper focuses on a multi-objective optimization problem with sequence-dependent setup times, tardiness, and machine maintenance. We have not yet found similar publications, so we design two algorithms for comparison: a local reordering heuristic algorithm without local optimization strategy (LRH) and a non-dominated sorting genetic algorithm-II (NSGA-II) [15]. LRH is based on the framework of LRHOS, but local optimization strategy is not used. NSGA-II is an effective algorithm for solving multi-objective problems, and the parameters are set as follows: the population size is 20, the probabilities of crossover and mutation are 0.4 and 0.6, respectively, the maximum iteration is 100, the number of elite populations is 20, each scale of batch runs 20 times independently, and we take the average as the output.

The algorithm is coded on Visual C# compiler, and its hardware environment is Intel Core i5-3470 CPU /3.20GHz/8GB/Window 7.

### *5.2. Experimental Results*

The experimental results are shown in Table 2.

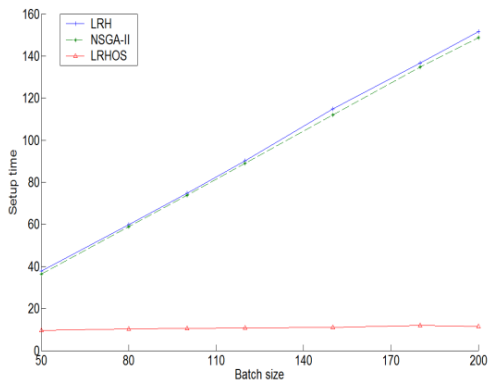
Table 2. Experimental results

| Batch size | Total processing time | | | Setup time | | | Idle time | | |
|---|---|---|---|---|---|---|---|---|---|
| Algorithm | LRH | NSGA-II | LRHOS | LRH | NSGA-II | LRHOS | LRH | NSGA-II | LRHOS |
| 50 | 184.90 | 248.84 | 156.71 | 37.91 | 36.50 | 9.57 | 0.20 | 1.07 | 0.36 |
| 80 | 294.77 | 479.14 | 245.40 | 59.93 | 58.87 | 10.37 | 0.52 | 3.23 | 0.72 |
| 100 | 372.08 | 636.73 | 307.05 | 74.77 | 73.90 | 10.48 | 4.96 | 1.34 | 0.61 |
| 120 | 449.51 | 676.31 | 370.67 | 90.11 | 89.05 | 10.80 | 0.55 | 5.09 | 1.01 |
| 150 | 564.46 | 852.16 | 460.70 | 114.76 | 111.89 | 11.14 | 1.10 | 6.64 | 0.97 |
| 180 | 679.43 | 1044.88 | 554.38 | 136.54 | 134.67 | 11.92 | 1.32 | 8.60 | 0.89 |
| 200 | 750.14 | 1182.46 | 610.14 | 151.62 | 148.79 | 11.61 | 1.25 | 10.48 | 1.26 |
| Avg | 470.75 | 731.50 | 386.44 | 95.09 | 93.38 | 10.84 | 1.42 | 5.21 | 0.83 |

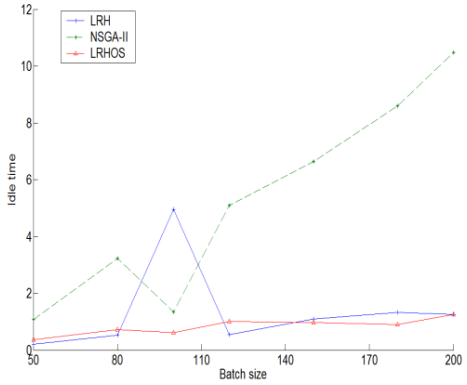
According to the data from Table 3, we make a comparison chart for the performance index of the setup time, idle time, tardiness, and total processing time, as shown in Figure 3. The horizontal axis represents the batch size, and the vertical axis represents the performance index.

Table 3. Experimental results (continued)

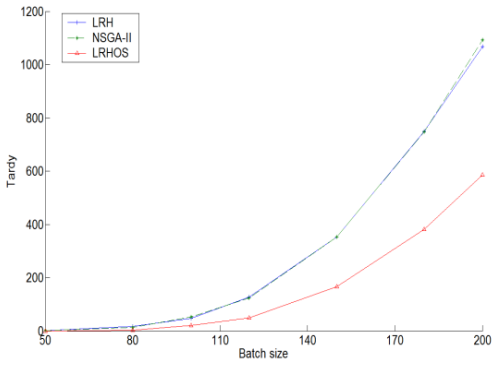
| Batch size | Tardiness | | | RPP | | |
|---|---|---|---|---|---|---|
| Algorithm | LRH | NSGA-II | LRHOS | LRH | NSGA-II | LRHOS |
| 50 | 0.10 | 0.10 | 0.00 | 2 | 2.05 | 2 |
| 80 | 17.72 | 12.66 | 2.57 | 3 | 3 | 2.7 |
| 100 | 47.34 | 52.33 | 20.88 | 3.8 | 3.8 | 3 |
| 120 | 126.96 | 124.40 | 48.88 | 4.1 | 4.2 | 3.7 |
| 150 | 353.18 | 352.52 | 166.00 | 5.05 | 5 | 4.15 |
| 180 | 749.66 | 747.40 | 381.01 | 6.05 | 6.05 | 5.05 |
| 200 | 1068.43 | 1092.54 | 586.53 | 6.95 | 7 | 5.65 |
| Avg | 337.63 | 340.28 | 172.27 | 4.42 | 4.44 | 3.75 |



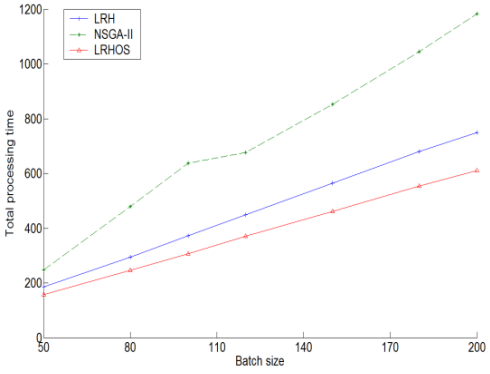

(a) Diagram of setup time




(b) Diagram of idle time




(c) Diagram of tardiness




(d) Diagram of total processing time

Figure 3. Performance index of algorithms

Considering the above table and figures, we obtain several conclusions, as follows:

(1) Comparison between LRHOS and NSGA-II.

a) In terms of the total setup time, LRHOS maintains a rising trend with an increase in batch size and is optimized by 73.79%, 82.39%, 85.82%, 87.87%, 90.04%, 91.15%, and 92.2%. In Figure 3(a), we observe that the curve of LRHOS is always below that of NSGA-II, and the distance between the two curves gradually increases because LRHOS rises slowly while NSGA-II rises fast. LRHOS has a significant effect on the optimization of setup time.

b) In terms of tardiness, LRHOS is optimized by 100%, 79.68%, 60.1%, 60.71%, 52.91%, 49.02%, and 46.32%. Combined with the trends of curves in Figure 3(c), NSGA-II rises faster when the batch size exceeds 120, while LRHOS becomes faster when batch size exceeds 150. This indicates that LRHOS achieves better performance in scheduling problems with large batch sizes. Overall, limited by machine processing capabilities and resources, when the batch size exceeds the critical point of the machine, the tardiness will increase correspondingly.

c) As for idle time, the curve of LRHOS rises slowly and is always lower than that of NSGA-II in Figure 3(b). NSGA-II maintains a rising trend but is not stable when the batch size is small. Comparison between these two curves shows that LRHOS outperforms NSGA-II in solving idle time.

d) In terms of the total processing time, LRHOS is optimized by 37.02%, 48.78%, 51.78%, 45.19%, 45.94%, 46.94%, and 48.4%. As can be seen from Figure 3(d), although both curves are rising, LRHOS is always lower than NSGA-II, and the distance between them is gradually increasing. The number of RPP is less than that of NSGA-II at different examples in the table. This demonstrates the stability and efficiency of LRHOS in this problem.

(2) Comparison between LRHOS and LRH.

a) From the perspective of setup times, LRHOS is optimized by 74.77%, 82.71%, 85.99%, 88.01%, 90.29%, 91.27%, and 92.34%. It can be seen that the optimization range of LRHOS generally rises with an increase in batch size. From Figure 3(a), the curve of LRHOS slowly rises, but the upward trending of LRH is faster, indicating that MSDB is effective at optimizing setup times.

b) For the objective of tardiness, LRHOS is optimized by 100%, 85.49%, 55.9%, 61.5%, 53%, 49.18%, and 45.1%. It can be seen that the optimization range is roughly inversely proportional to the batch size. Limited by the capabilities of the machine, the optimization effect of SSDP will decline when batch size increases.

c) Aiming at the idle time, we can see that the difference between LRHOS and LRH is small in Table 2. For some batch sizes, the result of LRHOS is inferior to LRH; this is because the local optimization strategy aims at the setup time and tardiness, but the difference is small and has little impact on the effects, so it can be accepted.

d) For the total processing time, LRHOS is optimized by 15.24%, 16.75%, 17.48%, 17.54%, 18.38%, 18.41%, and 18.66%. The optimization range is consistent with the increasing batch size. The LRHOS curve is located under the LRH curve, and the distance between these two curves increases with an increase in batch size. This confirms the validity of SSDP via the influential factor setup time.

In order to evaluate the efficiency of the above algorithms, we calculate the running time for each algorithm, as shown in Table 4, and draw a graph to highlight this contrast. The unit of calculation is seconds(s) in Table 4 and Figure 4.

From Table 4, we can see that LRHOS and LRH run faster than NSGA-II. This indicates the higher efficiency of our algorithm. Comparing the efficiency between LRHOS and LRH in Figure 4, when the batch size is less than 120, the solution time of LRHOS is longer than that of LRH. However, when the batch size is more than 120, its solution time becomes lower than that of LRH, which suggests that SSDP and MSDB are more suitable and effective for scheduling problems with large batch sizes.

Table 4. CPU time

| Batch size | LRH | LRHOS | NSGA-II |
|---|---|---|---|
| 50 | 0.003 | 0.007 | 18.999 |
| 80 | 0.010 | 0.011 | 31.979 |
| 100 | 0.013 | 0.016 | 40.177 |
| 120 | 0.028 | 0.025 | 45.636 |
| 150 | 0.049 | 0.045 | 60.314 |
| 180 | 0.090 | 0.068 | 70.339 |
| 200 | 0.095 | 0.082 | 77.736 |
| Avg | 0.041 | 0.036 | 49.311 |

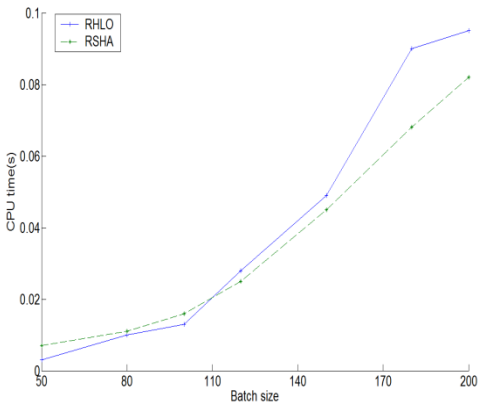



Figure 4. Diagram of CPU time

## 6. Conclusions

Machine maintenance and sequence-dependent setup times are important factors affecting the overall production rhythm in the process of seamless steel tubes, and they influence the on-time delivery of orders and production stability. In this paper, two local optimization rules are designed based on the relationship between setup time and tardiness, and a two-stage local reordering heuristic algorithm based on optimization strategy is proposed. At the first stage, the initial sequence is generated by the sorting rule, and the local optimization strategy is used to optimize setup times and tardiness. At the second stage, a local reordering algorithm based on idle time is designed, and it combines the optimization strategy to optimize the initial sequence. The experimental results show that the proposed algorithm is feasible and effective, and it can guide the formulation of the scheduling plan in seamless steel tube production.

## Acknowledgements

This research was supported by the National Natural Science Foundation of China (No. 71701016, 71471015), the Beijing Natural Science Foundation (No. 9174038), the Humanity and Social Science Youth Foundation of Ministry of Education of China (No. 17YJC630143), and the Fundamental Research Funds for Central Universities (No. FRF-BD-17-009A).

## References

1. C. Y. Lee and S. D. Liman, "Single Machine Flow-Time Scheduling with Scheduled Maintenance," *Acta Informatica*, Vol. 29, No. 4, pp. 375-382, 1992
2. V. Nesello, A. Subramanian, M. Battarra, and G. Laporte, "Exact Solution of the Single-Machine Scheduling Problem with Periodic Maintenances and Sequence-Dependent Setup Times," *European Journal of Operational Research*, Vol. 266, No. 2, pp. 498-507, 2018
3. F. ángel-Bello, A. Alvarez, J. A. Pacheco, and I. Mart ínez-Salazar, "A Heuristic Approach for a Scheduling Problem with Periodic Maintenance and Sequence-Dependent Setup Times," *Computers & Mathematics with Applications*, Vol. 61, No. 4, pp. 797-808, 2011
4. Z. X. Wu, T. K. Li, W. X. Zhang, and B. L. Wang, "Methods of Hot-Rolled Batch Planning for Seamless Steel Tube with Machine Maintenance," *Control Theory & Applications*, Vol. 9, pp. 1250-1259, 2017
5. J. Pacheco, S. Porras, S. Casado, and B. Baruque, "Variable Neighborhood Search with Memory for a Single-Machine Scheduling Problem with Periodic Maintenance and Sequence-Dependent Set-up Times," *Knowledge-based Systems*, Vol. 145, pp. 236-249, 2018
6. J. Pacheco, F. ángel-Bello, and A. Álvarez, "A Multi-Start Tabu Search Method for a Single-Machine Scheduling Problem with Periodic Maintenance and Sequence-Dependent Set-up Times," *Journal of Scheduling*, Vol. 16, No. 6, pp. 661-673, 2013
7. Y. Q. Yin, J. Y. Xu, T. C. E. Cheng, and C. C. Wu, "Approximation Schemes for Single-Machine Scheduling with a Fixed Maintenance Activity to Minimize the Total Amount of Late Work," *Naval Research Logistics*, Vol. 63, No. 2, pp. 172-183, 2016
8. B. Rachid, H. Allaoui, A. Artiba, and S. Hanafi, "Minimizing the Weighted Sum of Maximum Earliness and Maximum Tardiness Costs on a Single Machine with Periodic Preventive Maintenance," *Computers & Operations Research*, Vol. 47, pp. 106-113, 2014
9. J. Y. Lee and Y. D. Kim, "Minimizing the Number of Tardiness Jobs in a Single-Machine Scheduling Problem with Periodic Maintenance," *Computers & Operations Research*, Vol. 39, No. 9, pp. 2196-2205, 2012
10. M. Sbihi and C. Varnier, "Single-Machine Scheduling with Periodic and Flexible Periodic Maintenance to Minimize Maximum Tardiness," *Computers & Industrial Engineering*, Vol. 55, No. 4, pp. 830-840, 2008

11. J. X. Luo, H. M. Liu, and Y. M. Hu, "IFF Algorithm for the Single Machine Total Weighted Tardiness Scheduling Problem with Sequence Dependent Setup Times," *System Engineering- Theory & Practice*, Vol. 32, No. 12, pp. 2802-2808, 2012
12. H. Xu, Z. Lü, and T. C. Cheng, "Iterated Local Search for Single-Machine Scheduling with Sequence-Dependent Setup Times to Minimize Total Weighted Tardiness," *Journal of Scheduling*, Vol. 17, No. 3, pp. 271-287, 2014
13. Q. Guo and L. Tang, "An Improved Scatter Search Algorithm for the Single Machine Total Weighted Tardiness Scheduling Problem with Sequence-Dependent Setup Times," *Applied Soft Computing*, Vol. 29, pp. 184-195, 2015
14. A. Subramanian and K. Farias, "Efficient Local Search Limitation Strategy for Single Machine Total Weighted Tardiness Scheduling with Sequence-Dependent Setup Times," *Computers & Operations Research*, Vol. 79, pp. 190-206, 2017
15. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan,"A Fast and Elitist Multi Objective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, pp. 182-197, 2002