

Research Article

Robust Parallel Machine Scheduling Problem with Uncertainties and Sequence-Dependent Setup Time

Hongtao Hu,¹ K. K. H. Ng,² and Yichen Qin²

¹Logistics Engineering College, Shanghai Maritime University, Shanghai, China

²Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong

Correspondence should be addressed to K. K. H. Ng; kkh.ng@connect.polyu.hk

Received 6 September 2016; Accepted 11 October 2016

Academic Editor: Si Zhang

Copyright © 2016 Hongtao Hu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A parallel machine scheduling problem in plastic production is studied in this paper. In this problem, the processing time and arrival time are uncertain but lie in their respective intervals. In addition, each job must be processed together with a mold while jobs which belong to one family can share the same mold. Therefore, time changing mold is required for two consecutive jobs that belong to different families, which is known as sequence-dependent setup time. This paper aims to identify a robust schedule by min-max regret criterion. It is proved that the scenario incurring maximal regret for each feasible solution lies in finite extreme scenarios. A mixed integer linear programming formulation and an exact algorithm are proposed to solve the problem. Moreover, a modified artificial bee colony algorithm is developed to solve large-scale problems. The performance of the presented algorithm is evaluated through extensive computational experiments and the results show that the proposed algorithm surpasses the exact method in terms of objective value and computational time.

1. Introduction

Parallel machine systems are widely adopted in a variety of manufacturing environments, such as the semiconductor manufacturing industry [1] and the electronics industry [2]. Minimizing makespan is one of the commonly used objectives in manufacturing scheduling problems [3, 4]. Parallel machine scheduling problems have both theoretical and practical importance. Many literatures assume that the parameters of the problem (i.e., the processing time and job release time of jobs) are known in advance precisely before production process begins. However, the results of the solution derived under the deterministic assumption may deviate considerably in real situations [5]. In practice, it is difficult to get the exact parameters before production process begins due to some uncertainties, such as machine conditions, production environments, and jobs' characteristics [6–9]. To characterize and overcome the impact of uncertainties, several robust scheduling approaches are proposed to enhance the quality and stability of the derived solution in real situations.

Unlike the typical parallel machine system, a parallel machine system for plastic product involves mold changes during the process of production, which is also known as setup requirements. In a plastic production system, jobs are allocated to one of parallel injection machines and the corresponding injection mold is required to be installed onto the injection machine before the injection process. Jobs belonging to one family can be processed with the same mold and the mold changing process costs a period of time. The production planner would try to arrange the jobs such that jobs that belong to the same family are processed together in order to avoid extra mold change process and improve efficiency as well. For two jobs that are scheduled to be processed consecutively but belong to different families, mold change must be conducted before the next job can be performed. Therefore, reducing mold changing time is crucial in plastic production scheduling.

This paper investigates the uniform parallel machine problem in the plastic production system, which involves uncertain processing time, job release time, and setup time for mold. It is an extended problem studied by [5]. The

preliminary objective is to minimize the makespan. A scenario is firstly defined as a possible realization of processing times and job release time for all jobs. Due to the uncertainty in job processing time and release time, the makespan of a given solution can be different under various scenarios. It is proposed to identify an optimal solution with the strongest stability across all scenarios. The performance (or stability) of a given solution under its worst-case scenario is the major concern in robust scheduling. For generality, it is assumed that the processing times and job release time lie in their respective intervals with lower bound and upper bound value. The intervals of the corresponding parameters are considered to be known but not fall into any statistical distribution due to asymmetric information [5, 9]. And then the robust deviation criterion is adopted to evaluate the robustness of each candidate schedule. The robust deviation criterion is also known as min-max regret criterion, which aims to find out a solution with a minimum maximal deviation across all possible scenarios. This criterion has its advantages in the highly competitive environment for the reason that the robust decision performs well in any set of potential realizable scenarios [5]. The robust deviation criterion confines the magnitude of missed opportunities by identifying a schedule that possesses a performance close to that of the optimal (or near optimal) decision in any scenarios.

To the best knowledge of the authors, parallel machine scheduling problem with uncertain processing time, ready time, and mold change consideration has not been covered by the other researchers in the parallel machine scheduling problem. In this paper, a mixed integer linear programming (MILP) formulation is introduced to identify a robust schedule with minimum maximal regret across all scenarios. The concept of critical machine proposed in [5] is revised and adopted in the problem to eliminate the worst-case scenario into a finite number of extreme point scenarios in order to evaluate the robust deviation of given solution by calculating the maximal regret. Exact algorithm based iterative relaxation procedures are presented, and a modified artificial bee colony algorithm is proposed for the research problem. To demonstrate the effectiveness and efficiency of the proposed heuristic algorithm, a set of testing problem is carried out.

The contribution of this paper is shown as follows:

- (1) A parallel machine scheduling problem with consideration of uncertain processing time, job release time, and mold changing time is studied. For generality, the uncertain data is assumed to lie in intervals, which capture the situation in real environment.
- (2) Swarm intelligence algorithm is proposed to solve the NP-hard problem. The computational results demonstrate the stability and effectiveness of the proposed algorithm. To the best of our knowledge, this research is the first to adopt artificial bee colony algorithm in robust optimization for a parallel machine scheduling problem.

The rest of this paper is organized as follows. Section 2 provides the literature review of works related to parallel machine as well as robust scheduling approaches. Section 3 presents

the description and the mathematical model of the proposed problem. The exact algorithm based on iterative relaxation is provided in Section 4. The modified artificial bee colony algorithm for large-scale problem is proposed in Section 5. Results of extensive computational experiment and comparison are provided in Section 6. Section 7 provides the conclusion and direction for further works in related fields.

2. Literature Review

Stochastic approaches for tackling scheduling problem under uncertainties are available [10], but some of these approaches have their limitation due to the strict prerequisite and assumption [5]. For example, stochastic approaches require certain information on probability distribution of processing time or release time of each job, which can be inferred on the condition that a substantial amount of historical data is available [9, 11]. However, such amount of historical data is unavailable in highly uncertain environment and the only information is an educated guess of the lower bound and upper bound of some parameters, such as processing time and ready time [12]. For some one-time jobs, decision-makers are more interested in obtaining a robust schedule, which is against the worst-case performance across all scenarios, rather than obtaining an expected optimal performance under an expected situation. Robust deviation approach, which is known as min-max regret [13], is suitable for these circumstances to obtain such solution [14, 15]. The robust deviation approach is widely adopted in various combinational optimization problems when the input data are presented as intervals, such as the shortest path [16, 17], spanning tree [14, 17–19], and production problems [5, 20]. This approach yields a satisfactory result in an uncertain environment. Moreover, robust scheduling approaches are also adopted in other problems, such as maritime transportation problem [21, 22], routing problem [23], and scheduling problem in public health service department [24]. Most recently, a min-max regret makespan minimization in an identical parallel machine scheduling environment with interval data is studied [5]; in particular, they considered the processing time of jobs lies in respective intervals. To solve this nondeterministic polynomial-time hard (NP-hard) problem, the concept of critical machine and extreme point scenarios and two properties are proposed to avoid visiting an infinite number of possible scenarios and eliminating worst-case scenario into finite number of extreme point scenarios. A makespan minimization problem with interval job processing time on identical parallel machines is addressed in [5] for the first time in related field. Furthermore, the approach of robust deviation on a uniform parallel machine scheduling problem is adopted in [12] to minimize the total flow time with uncertain processing time which lies in respective intervals. The concept of worst-case scenario used to identify the maximal regret for a feasible solution is adopted in [5, 12] and then the exact and heuristic algorithms are proposed to find the robust schedule across all scenarios.

Scheduling problem with sequence-dependent setup time is a very active research area [25–28]. In the plastic manufacturing process, injection operation is a typical single-stage

manufacturing process, which requires the plastic to be injected into specified mold to produce plastic products in different shape. Several jobs are manufactured by a single machine with their respective molds [29]. A setup of mold is required for a product to be produced in injection operation, while setup is a sequence-dependent operation and parameters such as time for mold change and install time vary depending on two consecutive jobs [30]. If two consecutive jobs are processed by different mold, cost of setup is induced during the manual mold change operation. Therefore, the time for mold change as well as reinstall cannot be ignored in a mass production situation and planner would try to arrange those jobs by the same mold to be processed together in order to reduce the number of mold changes during production. The sequence-dependent family setup time represented by interval data on a single machine scheduling problem to minimize the total flow time is considered in [20].

Exact and heuristic algorithms have been proposed to solve min–max regret problem. A min–max regret problem of minimizing the total flow time on a single machine in [6] and then this problem are investigated further by the other researchers. In order to reduce the computational effort, different approaches have been proposed for the min–max regret model. It is proved that the optimal schedule under the mid-point scenario guarantees a 2-approximation of the optimal solution [15]. Heuristics method is an approximation algorithm to obtain near optimal result for min–max regret model. Typical example is shown in [5]. Some heuristic algorithms based on job swap moves and insert moves are presented by researchers. A hybrid tabu search algorithm for batching and sequencing decision-making in a single machine scheduling problem is proposed by Suppiah and Omar [31]. Job swapping and insertion approaches are applied to generate neighborhood solution and employed arcs in which the solution appears in the form of arcs in the tabu list to represent the sequence of job on a single machine; in addition, they also implemented a search depth strategy in the process of neighborhood generation to eliminate noneffective moves so as to reduce the computational burden while obtaining final schedule with outperformed quality. The work of Bilge et al. [32] points out that, for the situation where neighborhood of solution is in large number or its elements are expensive to evaluate, it is essential to restrict the number of solutions examined on a given iteration to screen the neighborhood so as to concentrate on promising moves at each iteration. Three candidate list strategies are proposed to confine the number of neighborhood solution for calculation efficiency.

Various studies on parallel machine problem have, respectively, considered uncertain processing time, arbitrary ready time, and mold changing time in the model development stage while there are no studies concerning these factors altogether in an integrated robust scheduling model. Therefore, this paper investigates these factors altogether and formulates a novel model in a plastic production environment. The presented paper is an extended version of the previous paper, containing more uncertain factors and features in a practical manufacturing system.

3. Problem Formulation and Mathematical Model

It is started by describing the problem and the definition of the maximal regret of a feasible schedule. Then the MILP is presented.

3.1. Problem Description. The problem under consideration deals with the scheduling of parallel machine over an assigned planning horizon in order to minimize the makespan. The job is processed by machine and mold and each job can be processed only once. Under the previous assumptions, the problem can be modelled as a mixed integer linear programming formulation as stated below.

3.2. MILP Formulation

Notations

i : job, $i = 1, 2, \dots, n$, where n is the total number of jobs, $i \in J$

m : machine, $m = 1, 2, \dots, q$, where q is the total number of machines, $m \in M$

A_i : mold installation time for job i

D_i : mold removal time for job i

p_i^s : job i 's processing time under scenario s , $p_i^s \in [\underline{p}_i, \bar{p}_i]$

r_i^s : job i 's arrival time under scenario s , $r_i^s \in [\underline{r}_i, \bar{r}_i]$

q_m : processing speed of machine m

s : scenario $s = \{p_1^s, p_2^s, \dots, p_n^s; r_1^s, r_2^s, \dots, r_n^s\}$ which is a possible realization of the processing times and arrival time of jobs, $s \in S$

M : a large positive number

Decision Variables

x_{im} : 1 if job i is scheduled to be processed on machine m and 0 otherwise

y_{jim} : 1 if job i is scheduled to be processed after job j on the same machine m and 0 otherwise

$B_{im}^s(X)$: the beginning time of job i to be processed on machine m in schedule X under scenario s

$Z_{im}^s(X)$: the completion time of job i scheduled on machine m in schedule X under scenario s

$C_m^s(X)$: the completion time of machine m in schedule X under scenario s

A feasible solution should satisfy $\sum_{m=1}^q x_{im} = 1$ ($i = 1, 2, 3, \dots, n$) and $y_{ijm} + y_{jim} \geq x_{im} + x_{jm} - 1$, $\forall i, j \in J$, $i \neq j; m \in M$. That is, each job should be processed exactly once in parallel machine system and there exists a processing sequence for jobs to be processed on the same machine. Let Φ be the set of feasible schedules.

Completion time of job i in schedule X under scenario s can be defined as follows:

$$Z_{im}^s(X) = B_{im}^s(X) + \left(\frac{p_i^s}{q_m}\right) - M(1 - x_{im}). \quad (1)$$

Calculation of completion time of machine m under scenario s :

The completion time of machine m in schedule X under scenario s is equal to the completion time of the job that is scheduled to be processed by the last machine

$$C_m^s(X) \geq Z_{im}^s(X), \quad \forall i \in J, m \in M. \quad (2)$$

That is,

$$C_m^s(X) \geq B_{im}^s(X) + \left(\frac{p_i^s}{q_m}\right) - M(1 - x_{im}), \quad (3)$$

$$\forall i \in J, m \in M.$$

The starting of job i on machine m is equal to or greater than the arrival time of job i under scenario s :

$$B_{im}^s(X) \geq r_i^s - M \cdot (1 - x_{im}), \quad \forall i \in J. \quad (4)$$

In addition, if job i and its preceding job j are processed by different mold, the time for mold change is involved before job i starts to be processed:

$$B_{im}^s(X) - Z_{jm}^s(X) \geq (D_j + A_i) \cdot y_{jim} - M(1 - y_{jim}), \quad (5)$$

$$\forall i, j \in J, i \neq j; m \in M.$$

The makespan of schedule X under scenario s is equal to the completion time of the machine that finished last in parallel machine system:

$$F(X, s) = \max_{m \in M} \{C_m^s(X)\} \quad (6)$$

or

$$F(X, s) \geq C_m^s(X). \quad (7)$$

We define the minimum makespan under scenario s as F_s^* and the corresponding optimal schedule as X_s^* :

$$F_s^* = F(X_s^*, s) = \min_{X \in \Phi} F(X, s). \quad (8)$$

For a given schedule $X \in \Phi$, its regret under scenario s is defined as

$$R(X, s) = F(X, s) - F_s^*. \quad (9)$$

The maximum regret of schedule X is

$$R_{\max}(X) = \max_{s \in S} R(X, s). \quad (10)$$

The robust parallel machine scheduling with interval processing time and arrival time and mold change can be formulated as

$$\min_{X \in \Phi} R_{\max}(X) = \min_{X \in \Phi} \max_{s \in S} (F(X, s) - F_s^*). \quad (11)$$

The proposed problem is a generalization of classical $P \parallel C_{\max}$ schedule problem and the optimal solution derived from (11) corresponds to the robust schedule across all possible scenarios.

The parallel machine robust scheduling with uncertain processing time and arrival time (RS) can be formulated as follows:

$$\min_X \left\{ \max_{s \in S} [F(X, s) - F_s^*] \right\}, \quad (12)$$

$$\text{s.t. } B_{im}^s(X) \geq r_i^s - M \cdot (1 - x_{im}), \quad (13)$$

$$\forall i \in J, m \in M, s \in S,$$

$$B_{im}^s(X) - Z_{jm}^s(X) \geq (D_j + A_i) \cdot y_{jim} - M(1 - y_{jim}), \quad (14)$$

$$\forall i, j \in J, i \neq j; m \in M, s \in S,$$

$$\sum_{m=1}^q x_{im} = 1, \quad \forall i \in J, \quad (15)$$

$$y_{ijm} + y_{jim} \geq x_{im} + x_{jm} - 1, \quad (16)$$

$$\forall i, j \in J, i \neq j; m \in M,$$

$$x_{im} \in \{0, 1\}, \quad \forall i, j \in J, i \neq j, \quad (17)$$

$$y_{ijm} \in \{0, 1\}, \quad \forall i, j \in J, i \neq j; m \in M. \quad (18)$$

We note that the above formulation is nonlinear due to the two operators in the objective function. However, the original model of RS can be transformed into a mixed integer linear programming model:

$$\min RV, \quad (19)$$

$$\text{s.t. } B_{im}^s(X) + \left(\frac{p_i^s}{q_m}\right) - M(1 - x_{im}) - F_s^* \leq RV,$$

$$\forall i \in J, m \in M, s \in S,$$

(13)–(18).

After linearization of the original model of RS problem, the formulation still cannot be solved directly due to an infinite number of possible scenarios. To tackle this problem, the primal goal is to identify the maximal regret of each feasible solution and then select the solution with the minimum maximal regret across all solutions. Therefore, two properties are proposed to identify and confine worst-case scenario to a finite number of extreme point scenarios, which means that we do not have to visit all possible scenarios to obtain maxima regret of a given solution. In order to capture the uncertainties

in real manufacturing environment, we assume that the processing time and arrival time of each job are uncertain by lying in their respective intervals. Therefore, the number of possible scenarios is infinite under such assumption and it is impractical to enumerate all possible scenarios to get the worst-case scenario for a given solution. The derived properties are used to narrow down the worst-case scenario to several extreme point scenarios.

Definition 1. A machine $f \in M$ is said to be critical in a schedule $X \in \Phi$ under scenario $s \in S$, if f is the machine with maximum completion time in \mathbf{X} under s :

$$C_f^s(X) = \max_{m \in M} \{C_m^s(X)\} = F(X, s). \quad (20)$$

Definition 2. An extreme point scenario s^m for schedule \mathbf{X} is defined as follows:

$$p_i^{s^m} = \begin{cases} \bar{p}_i, & \text{if } x_{im} = 1, \\ \underline{p}_i, & \text{if } x_{im} = 0, \end{cases} \quad i \in J, \quad (21)$$

$$r_i^{s^m} = \begin{cases} \bar{r}_i, & \text{if } x_{im} = 1, \\ \underline{r}_i, & \text{if } x_{im} = 0, \end{cases} \quad i \in J.$$

Property 1. For any schedule $X \in \Phi$, let s^0 be a worst-case scenario for \mathbf{X} in which machine f is critical. Then a scenario s^f exists for schedule \mathbf{X} such that

- (a) machine f is also critical in \mathbf{X} under s^f ;
- (b) scenario s^f is a worst-case scenario for \mathbf{X} .

Assume that a given solution \mathbf{X} has a worst-case scenario s^f that satisfies conditions (a) and (b) of Property 1. Since machine f is critical in \mathbf{X} under scenario s^f , the makespan and maximal regret for solution \mathbf{X} can be calculated as follows.

Note that the set containing all the jobs on machine m in schedule \mathbf{X} is denoted as $P(X, m)$, $m \in M$. If $x_{im} = 1$, then $i \in P(X, m)$.

$$F(X, s^f) = \max_{\forall i \in P(X, f)} \left\{ B_{if}^{s^f}(X) + \left(\frac{\bar{p}_i}{q_f} \right) \right\},$$

where $B_{if}^{s^f}(X) \geq \bar{r}_i, \forall i \in P(X, f)$,

$$R_{\max}(X) = F(X, s^f) - F_{s^f}^* \quad (22)$$

$$= \max_{\forall i \in P(X, f)} \left\{ B_{if}^{s^f}(X) + \left(\frac{\bar{p}_i}{q_f} \right) \right\} - F_{s^f}^*,$$

where $B_{if}^{s^f}(X) \geq \bar{r}_i, \forall i \in P(X, f)$.

Proof. Worst-case scenario s^0 can be transformed into scenario s^f by decreasing $p_i^{s^0}$ to \underline{p}_i and $r_i^{s^0}$ to \underline{r}_i for all $i \notin P(X, f)$ and by increasing $p_i^{s^0}$ to \bar{p}_i and $r_i^{s^0}$ to \bar{r}_i for all $i \in P(X, f)$, and f is also the critical machine in \mathbf{X} under s^f . Let

$\Delta = F(X, s^f) - F(X, s^0)$ denote the increase in the makespan under s^f . Let $X_{s^0}^*$ be the optimal schedule for scenario s^0 . Because the makespan in $X_{s^0}^*$ cannot increase by more than Δ under scenario s^f in comparison with s^0 , we get

$$F_{s^f}^* - F_{s^0}^* \leq F(X_{s^0}^*, s^f) - F(X_{s^0}^*, s^0) \leq \Delta$$

$$= F(X, s^f) - F(X, s^0). \quad (23)$$

From the abovementioned inequality, we have $F(X, s^0) - F_{s^0}^* \leq F(X, s^f) - F_{s^f}^*$, indicating that the maximal regret cannot decrease if we replace s^0 with s^f , which is also the worst-case scenario for \mathbf{X} . \square

Property 2. The maximal regret of a given solution \mathbf{X} can be expressed as follows:

$$R_{\max}(X) = \max_{m \in M} \left\{ \max_{\forall i \in P(X, m)} \left\{ B_{im}^s(X) + \left(\frac{\bar{p}_i}{q_m} \right) \right\} - F_{s^m}^* \right\}, \quad (24)$$

where $B_{im}^s(X) \geq \bar{r}_i, \forall i \in P(X, m)$.

Proof. Equation in Property 2 shows that

$$R_{\max}(X) \leq \max_{m \in M} \left\{ \max_{\forall i \in P(X, m)} \left\{ B_{im}^s(X) + \left(\frac{\bar{p}_i}{q_m} \right) \right\} - F_{s^m}^* \right\}. \quad (25)$$

Suppose by contradiction that there exists a machine $k \in M$ such that

$$R_{\max}(X) < B_{ik}^{s^k}(X) + \left(\frac{\bar{p}_i}{q_k} \right) - F_{s^k}^* \quad (26)$$

where $B_{ik}^{s^k}(X) \geq \bar{r}_i, \forall i \in P(X, k)$.

Since $F(X, s^k) \geq B_{ik}^{s^k}(X) + (\bar{p}_i/q_k)$, where $B_{im}^s(X) \geq \bar{r}_i, \forall i \in P(X, k)$,

$$R_{\max}(X) < F(X, s^k) - F_{s^k}^*. \quad (27)$$

\square

4. Exact Algorithm

RS is a min-max problem that can be solved using a general iterative relaxation (IR) procedure proposed by [33–35]. First, the set of all possible scenarios S is replaced in RS with a finite

set of scenarios $\Gamma = \{s_1, s_2, \dots, s_h\}$, resulting in relaxed mixed integer program (RS-relaxed):

$$\min \quad RV \quad (28)$$

$$\text{s.t.} \quad B_{im}^{s_k}(X) + \left(\frac{p_i^{s_k}}{q_m} \right) - M(1 - x_{im}) - F_s^* \leq RV, \quad (29)$$

$$\forall i \in J, m \in M, s_k \in \Gamma,$$

$$B_{im}^{s_k}(X) \geq r_i^{s_k} - M \cdot (1 - x_{im}), \quad (30)$$

$$\forall i \in J, m \in M, s_k \in \Gamma,$$

$$B_{im}^{s_k}(X) - Z_{jm}^{s_k}(X) \geq (D_j + A_i) \cdot y_{jim} - M(1 - y_{jim}), \quad (31)$$

$$\forall i, j \in J, i \neq j; m \in M, s_k \in \Gamma,$$

$$(15)-(18). \quad (32)$$

Note. In this iterative relaxation approach, we replace the set of all possible scenarios S ($|S| = \infty$) with $\Gamma = \{s_1, s_2, \dots, s_h\}$, and the scenario in Γ is the worst-case scenario for the feasible solution that is identified in the last iteration.

The minimal makespan $F_{s_k}^*$ can be solved by solving a mixed integer programming if the scenario is fixed.

We refer to the m constraints $B_{im}^{s_k}(X) + (p_i^{s_k}/q_m) - M(1 - x_{im}) - F_s^* \leq r; \forall i, m$ and $B_{im}^{s_k}(X) \geq r_i^{s_k} - M \cdot (1 - x_{im}), \forall i, m$ associated with s_k as regret cuts.

Iterative Relaxation Algorithm. The IR procedures stop when $LB = UB$, where UB is the maximum regret for current relaxed solution \widehat{X} .

Step 0. Set $LB = 0$ (LB is lower bound for regret value) and $UB^* = +\infty$ (UB is upper bound for regret value); choose an initial solution \widehat{X} .

Step 1. Identify solution \widehat{X} 's worst-case scenario \widehat{s} and its respective $R_{\max}(\widehat{X})$ through Property 1 and Property 2. If $UB^* \geq R_{\max}(\widehat{X})$, then $UB^* = R_{\max}(\widehat{X}), X^* = \widehat{X}$. If $UB^* \leq LB$, go to Step 4.

Step 2. Add regret cuts $B_{im}^{s_k}(X) + (p_i^{s_k}/q_m) - M(1 - x_{im}) - F_s^* \leq r; \forall i, m$ and $B_{im}^{s_k}(X) \geq r_i^{s_k} - M \cdot (1 - x_{im}), \forall i, m$; to RS-relaxed scenario Γ .

Step 3. \widehat{X} and \widehat{r} are identified by solving RS-relaxed. Set $LB = \widehat{r}$; go to Step 1.

Step 4. Stop.

5. Swarm Intelligence Approach

5.1. Modified Artificial Bee Colony Algorithm

5.1.1. Description of Modified Artificial Bee Colony Algorithm. ABC algorithm has been well-studied in parallel machine

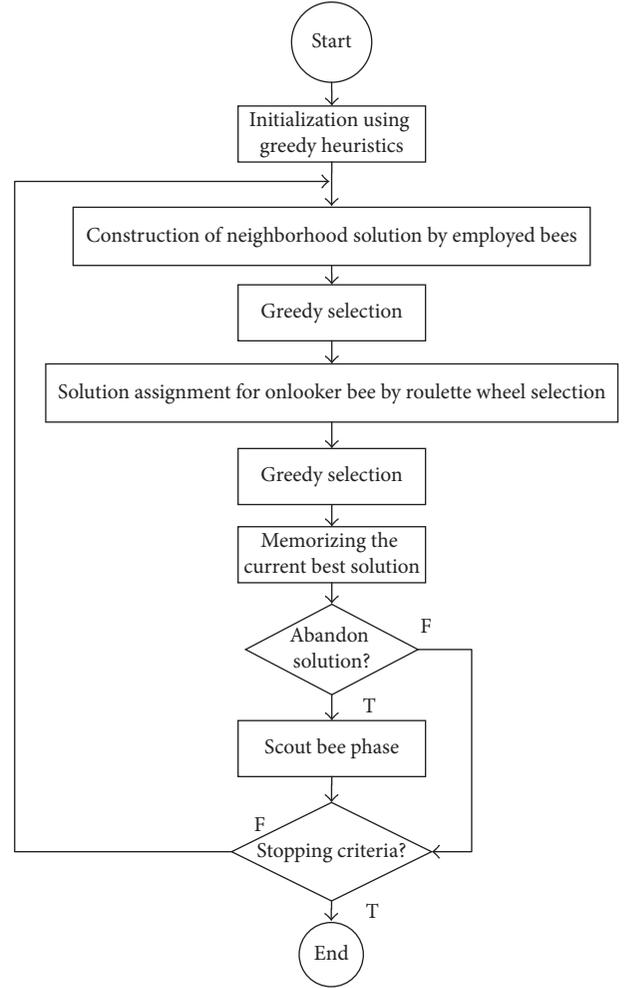


FIGURE 1: Process flowchart of artificial bee colony algorithm.

scheduling [36–38]. ABC algorithm is a promising swarm intelligence algorithm for optimization problems, which simulates the foraging behavior of honeybees. Three types of bees serve different functions to contribute to the forage acquisition for the bee social structure in the hive. The collected nectar will be stored as a food supply during the shortage of nectar. Employed bees and onlooker bees respond to obtain nectar from the nearby plants, while the main force of scout bee is to gather nectar information to guide the foraging behavior of the employed bees and onlooker bees. The design of the ABC algorithm follows a recursive search procedure with four major phases, including initialization phase, employed bee phase, onlooker bee phase, and scout bee phase. The process flowchart is shown in Figure 1.

The size of the bee colony is equal to CS , which includes employed bees, onlooker bees, and one scout bee. Approximately half of honeybees are as employed bees, and the remaining are onlooker bees. The number of solutions in ABC algorithm is equal to SN , where $SN = CS/2$. Each solution c_i represents an occupied food source by one employed bee. Hence, the population of the solution is represented by $c_i, i = 1, 2, \dots, SN$. The role of employed bee

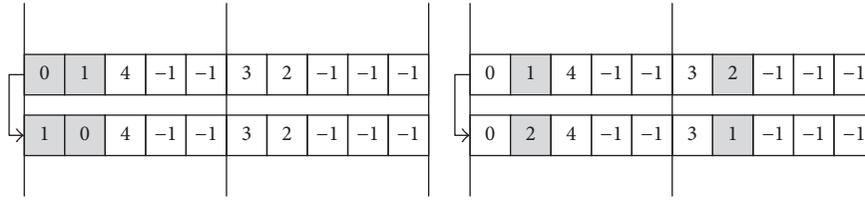


FIGURE 2: Swap operator in employed bee phase (same machine and different machine).

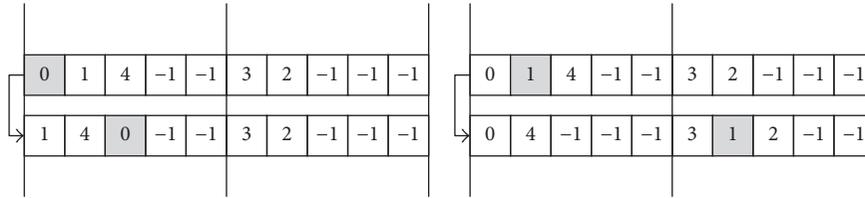


FIGURE 3: Insert operator in employed bee phase (same machine and different machines).

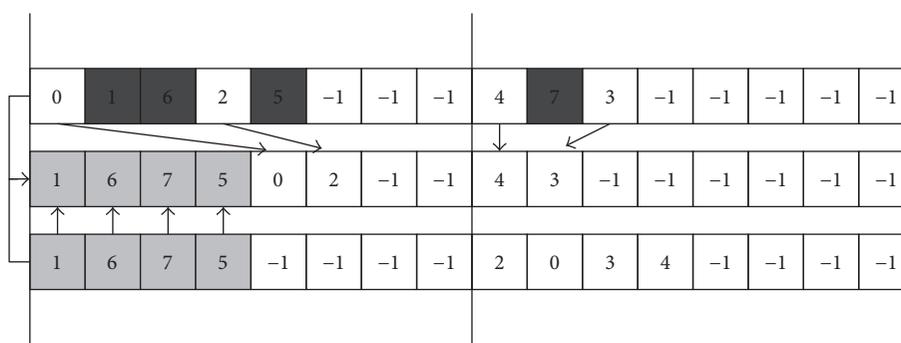


FIGURE 4: Crossover operator in employed bee phase.

is to gather nectar back to the hive; then the information of the food source will be shared among the hives. Onlooker bees will select one food source to further exploit based on the abundance level of the known nectar. Scout bee will seek for a new solution once a food source is exhausted.

5.1.2. Initialization Phase. For the initialization phase, the conventional approach is to generate a random assignment and sequence for an optimization problem. This prolongs the convergence to the global optimum and fails to achieve the global optimum, as the process of converging to the promising regions involves certain unsuccessful neighborhood searching under the large size solution space. A solution will be abandoned when there is no improvement after certain unsuccessful search. Therefore, a constructive heuristic for the initialization phase insists on developing the best possible solution at the initial stage in order to reduce the searching effort of the ABC algorithm in converging to the global optimum. The composition of the minimal makespan in PMS model aims to obtain a solution with the least total length of the parallel machine schedule to complete all the available jobs. The solution quality of an initial solution by sorting the ready time of each job in ascending order is expected to be the best alternative in timely fashion compared to

random assignment. While maintaining a certain quality of initial solution, the diversity between solutions should be considered. Otherwise, certain solution space could not be visited. At the same time, it is difficult for the population-based metaheuristics to escape from the premature convergence [39]. One way of reducing the possibility of converging to an identical but not a global optimal solution for the population-based metaheuristics relies on randomness in the construction of initial solution. In our preliminary study, it is effective to develop a fair quality of solution by random machine assignment for a sorted sequence.

5.1.3. Employed Bee Phase. Each employed bee performs a neighborhood search on a solution c_i to generate a neighborhood solution \tilde{c}_i by three neighborhood operators to enhance the solution quality in terms of its objective value. A greedy selection is applied herein to continuously improve a solution by comparing the objective values between $fit(c_i)$ and $fit(\tilde{c}_i)$. If the solution quality of neighborhood solution is better than the original one, the neighborhood solution will replace the previous solution $c_i \leftarrow \tilde{c}_i$. The neighborhood operators include swap operator, insert operator, and crossover operator as shown in Figures 2, 3, and 4, correspondingly. The neighborhood search is not restricted on

TABLE 1: List of algorithms for robust parallel machine scheduling with uncertainties processing time and arrival time.

Iterative relaxation approach	Algorithm for RS	Algorithm for $P \parallel C_{\max}$
Approach 1	Mixed integer programming	Mixed integer programming
Approach 2	Proposed artificial bee colony algorithm	Proposed artificial bee colony algorithm

either the same or another runway by randomly selecting two or more elements. The accumulative trial of unsuccessful improvement on a solution trial(c_i) will be increased by 1 if the objective value of a neighborhood solution is worse than the previous solution. The crossover operator will select a single runway solution with the longest completion time and choose another solution by fitness selective probability to combine and develop an offspring solution.

5.1.4. Onlooker Bee Phase. Each onlooker bee will further act as neighborhood operators to a specified solution based on a probabilistic selection with regard to the fitness values. The fitness value of a solution $\text{fit}(c_i)$ is measured by (33). A larger number of fitness values imply a better quality of a solution c_i . After the computation of fitness value for all the solutions, (34) is able to obtain the individual selective probability of each solution IndiProb_i . A cumulative probability CumuProb_i is computed by aggregating the preceding individual probabilities. The selection criterion follows the approach of roulette wheel selection. A random number p will be generated for each onlooker bee for selecting a solution to further search by neighborhood operators:

$$\text{fit}(c_i) = \begin{cases} \frac{1}{1 + \text{fun}(c_i)}, & \text{if } \text{fun}(c_i) \geq 0, \\ \frac{1}{1 + \text{abs}(\text{fun}(c_i))}, & \text{if } \text{fun}(c_i) \leq 0, \end{cases} \quad (33)$$

$\forall i,$

$$\text{IndiProb}_i = \frac{\text{fit}(c_i)}{\sum_{i=1}^{\text{SN}} \text{fit}(c_i)}, \quad \forall i. \quad (34)$$

5.1.5. Scout Bee Phase. During the neighborhood search, the cumulative number of unsuccessful update will be recorded as trial(c_i). A large number of trial(c_i) imply a possibility that solution c_i is trapped in a local optimum. The employment of scout bee phase is to abandon a solution once it reaches the maximum tolerance of searching. If one solution cannot be further improved by several iterations and reaching the maximum tolerance *limit*, the scout bee will be replaced by an exhausted solution by an initial solution.

5.1.6. Min–Max Regret Approach Using Artificial Bee Colony Algorithm. The design of the min–max regret approach is to obtain a robust PMS schedule to minimize the maximum regret value by considering a number of scenarios. Two main stages require optimization techniques to resolve the min–max regret model, which are optimal makespan under the worst-case scenario and makespan under robust schedule. The process workflows using the ABC algorithm in deriving the worst-case optimal makespan and robust makespan are the same, except the computation of objective value and

fitness value. The objective function in optimal makespan under the worst-case scenario is $F(X, s^{\zeta})$, while the objective function in robust ALP makespan by considering all the worst-case scenarios follows (28) and (29).

6. Numerical Experiments and Results

In this section, the performance of the modified ABC algorithm is evaluated by random robust PMS instances and compared with the results using mixed integer programming. The MIP is computed by *IBM ILOG CPLEX Optimization Studio 12.6.3* for comparison. All algorithms are written in *C# language with visual studio 2015* on a computer with *Intel Core i7 3.60 GHz CPU and 16.0 GB ram* under the *Window 7 Enterprise 64-bit* operating environment. In order to evaluate the algorithm performance, two approaches have been considered in the numerical experiments, as shown in Table 1. Approach 1 using mixed integer programming in resolving RS and $P \parallel C_{\max}$ is considered as a baseline for comparison. In approach 2, the optimization technique for RS and $P \parallel C_{\max}$ is the proposed artificial bee colony algorithm. In this regard, the performance of convergence and computational speed of the proposed artificial bee colony algorithm can be measured by comparing the values obtained in exact method.

The test running is conducted in preliminary study to obtain the parameter setting. The maximum iteration *MaxIter* is equal to $1000 \times n \times q$. The maximum tolerance of unsuccessful update is equal to $\text{SN} \times n \times q \times 2$. The colony size is set to be 80, and the number of solutions is 40. These results are also evaluated together with the optimal solution from MIP to measure the deviation between exact algorithm and swarm intelligence. The maximum computational time for MIP and proposed ABC algorithm is 7200 seconds.

Both algorithms are evaluated by test instances with $n = 21, 27, 33$ jobs and $q = 3, 4, 5$ machines. For each job, the lower bound and upper bound of ready time and processing time fall into a random uniform distribution integer with intervals of $r_i \in [0, 15n]$, $\bar{r}_i \in [r_i, 1.1r_i]$, $p_i \in [10, 50]$, and $\bar{p}_i \in [p_i, p_i(1 + \beta)]$ correspondingly, where β value is considered to be (0.2, 0.4, 0.6, 0.8, 1.0). The total number of test instances is 45 with any combination of n, q , and β . The processing speed q_m is equal to a random uniform distribution from 1 to 10, where $m = 1, 2, \dots, k$. Assuming there are only three types of injection molds, the mold installation time D_i and mold removal time A_i of injection molds for each group are defined as (30, 60, 90) and (10, 20, 30). The number of jobs for each group is equally distributed.

The regret upper bound value and regret lower bound value have been recorded using MIP to indicate the performance of the iterative relaxation algorithm. If the regret gap is equal to 0%, the recorded robust optimal value implies a

true optimum. Otherwise, the model has not been able to determine the optimal solution within 7200 seconds if the regret gap is larger than 0%. In order to compare between the performance of MIP algorithm and the performance of MABC algorithm, the deviation from the best-known solution indicates the deviation between average optimal and the optimal value using Cplex with limited computational time.

Table 2 shows the average computational performance for robust parallel machine scheduling using modified ABC algorithm in 10 run times. The results indicated that the computational time is exponentially increased when the problem size grows for both algorithms. For the problem size of $n = 21$, the robust optimal solution obtained by MIP performs better than the result using MABC algorithm in terms of optimal value and computational time. This implies the certain computational overhead for MABC algorithm in small size instance. When the problem size increases, MIP was not able to determine the optimal solution within a reasonable time, while MABC algorithm was able to obtain near optimal value. In order to measure the variability of algorithm performance, the number of best solutions denotes the number of optimal values (maximum number of 10) that surpasses the result obtained by MIP in Table 3. These results can provide evidence that the variability of MABC algorithm performance for the test instance with $\beta = 20, 40, 60, 80$ is in low level with respect to the quality of the solutions. The number of better solutions #best represents the number of solutions found by modified ABC algorithm in 10 run times which outperforms the solution obtained by MIP with time limit. As for the large size instance, the modified ABC algorithm is able to obtain better solution with lower computational effort than the exact method.

7. Concluding Remarks

Parallel machine model has been widely adopted in the manufacturing environment and its scheduling problem is extensively studied in literatures and minimizing makespan is the common objective in parallel machine scheduling problem. However, most of the literatures have the assumption that the parameters of job such as processing time and ready time are precisely known, which is counter to the practical situation for the set of jobs that are processed for the first time. Uncertain processing time and ready time in parallel machine scheduling are the common phenomenon in the production scheduling. In practice, the ready time of a job is uncertain and cannot be estimated by an exact probability distribution due to the effect of external factors, such as delivery variability, in-bound logistics delays, labour issues, and transportation schedules. The processing time of a job may deviate from the standard operation time, which is caused by corrupting operational events, corrective action, and reworking.

In order to characterize the uncertainties of parameters, it is assumed that the exact value of processing time and ready time of each job is unknown and the available information is the bound of each parameter. To resolve this robust version of $P \parallel C_{\max}$ problem, the min-max regret approach

is introduced herein to obtain a robust parallel machine schedule that considers all possible scenarios. We first introduce a mixed integer linear programming formulation to characterize the problem and then propose two properties to eliminate worst-case scenario. The proposed two properties allow the model obtain the maximal regret for each feasible schedule with a finite number of extreme point scenarios.

An exact algorithm based on general iterative relaxation algorithm is developed to solve the proposed problem. However, it is difficult to obtain a solution from large size instances by the exact algorithm since identifying worst-case scenario for each feasible is computationally expensive. Therefore, a modified artificial bee colony algorithm is proposed to solve the large-scale parallel machine scheduling with uncertainty. The experimental result indicates that a robust near optimal solution can be obtained with a reasonable time. The strength of artificial bee colony algorithm follows the three main features to obtain a near optimal solution by utilizing decentralization of honey bee, self-organizing in swarm performance, and collective behavior in algorithm structure. The proposed artificial bee colony algorithm is widely adopted in a static parallel machine scheduling and other related problems. In order to enhance the exploitation and exploration of the proposed algorithm to achieve a high level of optimization technique for robust modelling, certain modification on artificial bee colony algorithm has been proposed.

Future work can be concluded as follows: (1) extension of parallel machine scheduling with realistic uncertainties, constraints, and objectives to retain feasible solution in actual operation and (2) optimizing the algorithm structure and iteration process with other known optimization techniques to achieve optimum with less computational effort.

Notation of Artificial Bee Colony Algorithm

CS:	The size of bee colony
SN:	The number of colony solutions
MaxIter:	The maximum number of iterations
Dim:	The dimension of an independent solution
$c_i, i = 1, 2, \dots, SN$:	The position of each solution in bee colony
$f_{\text{un}}(c_i)$:	The objective value of solution c_i
$fit(c_i)$:	The fitness value of solution c_i
$IndiProb_i$:	The probability of an individual solution c_i among the entire colony in terms of fitness value
$CumuProb_i$:	The cumulative probability of an individual solution c_i in ascending order among the entire colony in terms of fitness value
\tilde{c}_i :	The neighbor solution of an individual solution c_i
$trial(c_i)$:	The accumulated trial value of an individual solution c_i , which cannot enhance the quality of solution in terms of its objective value

TABLE 2: The average computational performance for robust parallel machine scheduling using modified ABC algorithm.

n	q	β	MIP w/Cplex					Modified ABC algorithm				
			Optimal	Iteration	CPU	Regret UB	Regret LB	Regret gap (%) ^a	Average optimal ^c	Average iteration ^c	CPU	DFB (%) ^b
21	3	20	368.4	2	1.46	0.00	0.00	0.00	371.43	3.50	350.26	0.82
		40	363.6	2	TL	41.50	6.30	84.82	374.20	4.40	477.13	2.83
		60	346.2	0	TL	119.90	0.00	100.00	396.44	7.50	859.81	12.67
		80	400.2	1	TL	96.10	0.00	100.00	419.52	9.70	1172.62	4.61
		100	378	0	TL	174.50	0.00	100.00	432.20	9.00	1062.08	12.54
21	4	20	332	3	63.15	0.00	0.00	0.00	341.64	3.40	558.23	2.82
		40	338.4	3	58.35	0.00	0.00	0.00	356.80	6.00	961.19	5.16
		60	346.2	5	119.16	15.90	36.40	0.00	357.14	9.70	1762.27	3.06
		80	342.6	4	298.16	22.00	45.00	0.00	360.06	9.80	1854.48	4.85
		100	350.6	4	TL	43.90	3.30	92.48	354.80	10.00	1871.19	1.18
21	5	20	332	0	7.43	0.00	0.00	0.00	344.32	0.70	357.86	3.58
		40	332	9	91.40	21.50	31.60	0.00	348.28	4.30	966.30	4.67
		60	332	8	82.64	27.20	36.40	0.00	353.90	7.70	2085.18	6.19
		80	342.6	8	110.42	8.20	22.00	0.00	359.52	9.80	2700.58	4.71
		100	332	8	90.58	56.20	62.00	0.00	354.54	10.00	2468.68	6.36
27	3	20	442	1	TL	67.70	0.00	100.00	488.56	4.40	794.83	9.53
		40	446.8	0	TL	102.80	0.00	100.00	521.12	7.60	1450.59	14.26
		60	487.4	0	TL	118.40	0.00	100.00	519.66	9.80	2049.22	6.21
		80	498.8	0	TL	163.20	0.00	100.00	519.92	10.00	2076.67	4.06
		100	514.6	0	TL	218.80	0.00	100.00	546.42	10.00	2085.48	5.82
27	4	20	463	2	TL	28.80	10.00	65.28	427.02	0.00	253.22	-8.43
		40	469.4	2	TL	55.10	10.00	81.85	436.88	0.00	253.35	-7.44
		60	435.4	1	TL	84.30	0.00	100.00	436.08	10.00	3574.79	0.16
		80	482.2	2	TL	108.30	0.00	100.00	445.92	10.00	3530.21	-8.14
		100	488.6	2	TL	127.70	0.00	100.00	453.70	10.00	3566.66	-7.69
27	5	20	418	3	TL	22.10	2.70	87.78	422.50	5.30	2038.03	1.07
		40	418	2	TL	32.30	6.00	81.42	435.22	9.20	3544.58	3.96
		60	418	3	TL	31.20	4.40	85.90	419.92	8.80	3614.32	0.46
		80	435.4	3	TL	57.20	21.30	62.76	439.56	9.80	4098.94	0.95
		100	429	10	TL	59.50	70.60	0.00	447.08	9.80	4139.47	4.04
33	3	20	575.6	0	3541.30	0.00	0.00	0.00	589.19	4.30	1513.64	2.31
		40	585	1	TL	118.50	26.20	77.89	609.84	7.40	2857.19	4.07
		60	560.2	0	TL	0.00	0.00	0.00	627.76	9.40	4102.80	10.76
		80	618.9	1	TL	215.20	17.50	91.87	642.50	7.80	4419.67	3.67
		100	602	0	TL	293.50	0.00	100.00	627.36	7.80	4412.68	4.04
33	4	20	556.5	8	TL	18.80	0.00	100.00	555.92	3.40	2763.64	-0.10
		40	562.9	4	TL	89.30	66.00	26.09	538.80	5.40	4641.83	-4.47
		60	540.2	2	TL	115.80	13.20	88.60	561.04	6.00	4680.89	3.71
		80	519	2	TL	194.90	0.00	100.00	542.20	6.00	4662.75	4.28
		100	519	2	TL	152.00	34.70	77.17	563.08	6.00	3566.53	7.83
33	5	20	556.5	2	463.82	0.00	0.00	0.00	556.50	0.90	1872.39	0.00
		40	562.9	8	1614.77	15.40	33.80	0.00	566.20	2.60	3383.73	0.58
		60	519	11	2449.81	48.40	50.30	0.00	526.72	5.40	4764.68	1.47
		80	519	9	TL	88.00	41.40	52.95	519.54	6.00	5264.46	0.10
		100	519	8	TL	58.50	14.90	74.53	560.98	5.40	5088.87	7.48

^aThe percentage gap between regret lower bound value and regret upper bound value.^bThe deviation in percentage between the average optimal value using MABC algorithm and the optimal value using MIP.^cThe average solution obtained by MABC algorithm in 10 run times.

TABLE 3: The best computational performance for robust parallel machine scheduling using modified ABC algorithm.

n	q	β	MIP w/Cplex					Modified ABC algorithm					
			Optimal	Iteration	CPU	Regret UB	Regret LB	Regret gap (%) ^a	Best optimal ^c	Best iteration ^c	CPU	# best	DFB (%) ^b
21	3	20	368.4	2	1.46	0.00	0.00	0.00	368.40	4.00	381.22	7	0.00
		40	363.6	2	TL	41.50	6.30	84.82	363.60	2.00	246.51	3	0.00
		60	346.2	0	TL	119.90	0.00	100.00	368.40	7.00	702.52	1	6.03
		80	400.2	1	TL	96.10	0.00	100.00	404.60	10.00	1225.68	2	1.09
		100	378	0	TL	174.50	0.00	100.00	394.00	10.00	1167.61	1	4.06
21	4	20	332	3	63.15	0.00	0.00	0.00	332.00	1.00	126.41	7	0.00
		40	338.4	3	58.35	0.00	0.00	0.00	338.40	1.00	258.02	4	0.00
		60	346.2	5	119.16	15.90	36.40	0.00	346.20	7.00	1359.59	5	0.00
		80	342.6	4	298.16	22.00	45.00	0.00	342.60	10.00	1730.55	2	0.00
		100	350.6	4	TL	43.90	3.30	92.48	350.60	10.00	1821.72	7	0.00
21	5	20	332	0	7.43	0.00	0.00	0.00	332.00	0.00	184.60	5	0.00
		40	332	9	91.40	21.50	31.60	0.00	332.00	0.00	215.59	3	0.00
		60	332	8	82.64	27.20	36.40	0.00	334.60	10.00	3020.11	4	0.78
		80	342.6	8	110.42	8.20	22.00	0.00	342.60	10.00	2351.84	4	0.00
		100	332	8	90.58	56.20	62.00	0.00	332.00	10.00	2704.27	1	0.00
27	3	20	442	1	TL	67.70	0.00	100.00	474.20	4.00	733.30	0	6.79
		40	446.8	0	TL	102.80	0.00	100.00	511.00	6.00	1112.00	0	12.56
		60	487.4	0	TL	118.40	0.00	100.00	465.80	2.00	2140.45	2	-4.64
		80	498.8	0	TL	163.20	0.00	100.00	501.60	10.00	2084.83	0	0.56
		100	514.6	0	TL	218.80	0.00	100.00	499.00	10.00	2084.33	2	-3.13
27	4	20	463	2	TL	28.80	10.00	65.28	418.00	0.00	252.50	6	-10.77
		40	469.4	2	TL	55.10	10.00	81.85	434.00	0.00	253.65	10	-8.16
		60	435.4	1	TL	84.30	0.00	100.00	426.00	10.00	3576.53	4	-2.21
		80	482.2	2	TL	108.30	0.00	100.00	440.40	10.00	3570.41	10	-9.49
		100	488.6	2	TL	127.70	0.00	100.00	439.30	10.00	3588.61	10	-11.22
27	5	20	418	3	TL	22.10	2.70	87.78	418.00	0.00	296.45	9	0.00
		40	418	2	TL	32.30	6.00	81.42	418.00	10.00	3511.84	3	0.00
		60	418	3	TL	31.20	4.40	85.90	418.00	10.00	4067.50	8	0.00
		80	435.4	3	TL	57.20	21.30	62.76	422.40	10.00	4393.53	4	-3.08
		100	429	10	TL	59.50	70.60	0.00	418.00	10.00	4433.44	2	-2.63
33	3	20	575.6	0	3541.30	0.00	0.00	0.00	575.60	6.00	1930.36	3	0.00
		40	585	1	TL	118.50	26.20	77.89	596.50	5.00	1866.59	0	1.93
		60	560.2	0	TL	0.00	0.00	0.00	584.80	10.00	4416.73	0	4.21
		80	618.9	1	TL	215.20	17.50	91.87	614.80	7.00	3997.03	2	-0.67
		100	602	0	TL	293.50	0.00	100.00	575.90	8.00	4318.77	2	-4.53
33	4	20	556.5	8	TL	18.80	0.00	100.00	519.00	1.00	1209.28	8	-7.23
		40	562.9	4	TL	89.30	66.00	26.09	527.00	5.00	4797.14	6	-6.81
		60	540.2	2	TL	115.80	13.20	88.60	557.60	6.00	4299.88	6	3.12
		80	519	2	TL	194.90	0.00	100.00	524.20	6.00	4707.69	2	0.99
		100	519	2	TL	152.00	34.70	77.17	542.00	6.00	4430.14	0	4.24
33	5	20	556.5	2	463.82	0.00	0.00	0.00	556.50	0.00	925.34	10	0.00
		40	562.9	8	1614.77	15.40	33.80	0.00	566.20	0.00	976.70	10	0.58
		60	519	11	2449.81	48.40	50.30	0.00	519.00	5.00	4530.66	8	0.00
		80	519	9	TL	88.00	41.40	52.95	519.00	5.00	4811.97	8	0.00
		100	519	8	TL	58.50	14.90	74.53	523.50	5.00	5248.92	0	0.86

^aThe percentage gap between regret lower bound value and regret upper bound value.

^bThe deviation in percentage between the average optimal value using modified ABC algorithm and the optimal value using MIP.

^cThe best solution (in terms of objective value and CPU time) obtained by MABC algorithm in 10 run times.

limit: The maximum tolerance trial(c_i)

p : Random number, $0 \leq p \leq 1$.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

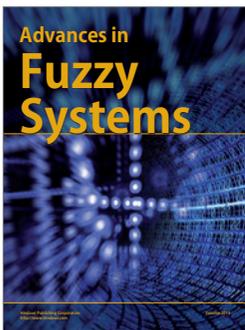
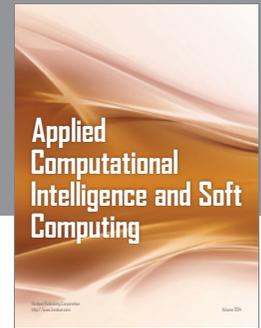
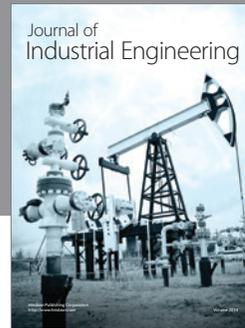
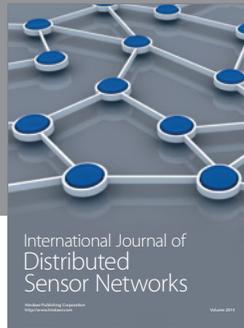
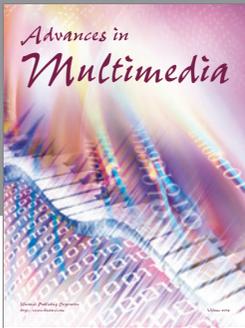
Acknowledgments

The research is supported by The National Natural Science Foundation of China [no. 71201099], Innovation Program of Shanghai Municipal Education Commission [no. 14YZ111], Shanghai Young Eastern Scholar Programme [QD2015041], Shanghai Pu Jiang Program (no. 13PJC066), Shanghai Youth Teacher Foundation (no. ZZshhs13021), and the Hong Kong Polytechnic University. The authors' gratitude is also extended to the research committee and the Department of Industrial and Systems Engineering of the Hong Kong Polytechnic University for support of this project (RUF1 and RU8H).

References

- [1] C.-F. Chien and C.-H. Chen, "Using genetic algorithms (GA) and a coloured timed Petri net (CTPN) for modelling the optimization-based schedule generator of a generic production scheduling system," *International Journal of Production Research*, vol. 45, no. 8, pp. 1763–1789, 2007.
- [2] W. Kaczmarczyk, "Proportional lot-sizing and scheduling problem with identical parallel machines," *International Journal of Production Research*, vol. 49, no. 9, pp. 2605–2623, 2011.
- [3] H. Gong and L. Tang, "Two-machine flowshop scheduling with intermediate transportation under job physical space consideration," *Computers and Operations Research*, vol. 38, no. 9, pp. 1267–1274, 2011.
- [4] X. Xie, L. X. Tang, and Y. P. Li, "Scheduling of a hub reentrant job shop to minimize makespan," *International Journal of Advanced Manufacturing Technology*, vol. 56, no. 5–8, pp. 743–753, 2011.
- [5] X. Q. Xu, W. T. Cui, J. Lin, and Y. J. Qian, "Robust makespan minimisation in identical parallel machine scheduling problem with interval data," *International Journal of Production Research*, vol. 51, no. 12, pp. 3532–3548, 2013.
- [6] R. L. Daniels and P. Kouvelis, "Robust scheduling to hedge against processing time uncertainty in single-stage production," *Management Science*, vol. 41, no. 2, pp. 363–376, 1995.
- [7] J. Yang and G. Yu, "On the robust single machine scheduling problem," *Journal of Combinatorial Optimization*, vol. 6, no. 1, pp. 17–33, 2002.
- [8] R. Montemanni, "A mixed integer programming formulation for the total flow time single machine robust scheduling problem with interval data," *Journal of Mathematical Modelling and Algorithms*, vol. 6, no. 2, pp. 287–296, 2007.
- [9] A. Allahverdi and H. Aydilek, "Heuristics for the two-machine flowshop scheduling problem to minimise makespan with bounded processing times," *International Journal of Production Research*, vol. 48, no. 21, pp. 6367–6385, 2010.
- [10] Q. Meng and X. Qu, "Bus dwell time estimation at bus bays: a probabilistic approach," *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 61–71, 2013.
- [11] A. Anglani, A. Grieco, E. Guerriero, and R. Musmanno, "Robust scheduling of parallel machines with sequence-dependent setup costs," *European Journal of Operational Research*, vol. 161, no. 3, pp. 704–720, 2005.
- [12] X. Q. Xu, J. Lin, and W. T. Cui, "Hedge against total flow time uncertainty of the uniform parallel machine scheduling problem with interval data," *International Journal of Production Research*, vol. 52, no. 19, pp. 5611–5625, 2014.
- [13] H. Aissi, C. Bazgan, and D. Vanderpooten, "Min-max and min-max regret versions of combinatorial optimization problems: a survey," *European Journal of Operational Research*, vol. 197, no. 2, pp. 427–438, 2009.
- [14] A. Kasperski and P. Zieliński, "An approximation algorithm for interval data minmax regret combinatorial optimization problems," *Information Processing Letters*, vol. 97, no. 5, pp. 177–180, 2006.
- [15] A. Kasperski and P. Zielinski, "A2-approximation algorithm for interval data minmax regret sequencing problems with the total flow time criterion," *Operations Research Letters*, vol. 36, no. 3, pp. 343–344, 2008.
- [16] R. Montemanni and L. M. Gambardella, "An exact algorithm for the robust shortest path problem with interval data," *Computers & Operations Research*, vol. 31, no. 10, pp. 1667–1680, 2004.
- [17] R. Montemanni and L. M. Gambardella, "The robust shortest path problem with interval data via Benders decomposition," *4OR*, vol. 3, no. 4, pp. 315–328, 2005.
- [18] H. Aissi, C. Bazgan, and D. Vanderpooten, "Complexity of the min-max and min-max regret assignment problems," *Operations Research Letters*, vol. 33, no. 6, pp. 634–640, 2005.
- [19] R. Montemanni, "A Benders decomposition approach for the robust spanning tree problem with interval data," *European Journal of Operational Research*, vol. 174, no. 3, pp. 1479–1490, 2006.
- [20] C.-C. Lu, S.-W. Lin, and K.-C. Ying, "Robust scheduling on a single machine to minimize total flow time," *Computers and Operations Research*, vol. 39, no. 7, pp. 1682–1691, 2012.
- [21] S. Wang and Q. Meng, "Robust schedule design for liner shipping services," *Transportation Research Part E: Logistics and Transportation Review*, vol. 48, no. 6, pp. 1093–1106, 2012.
- [22] X. Wang, "Stochastic resource allocation for containerized cargo transportation networks when capacities are uncertain," *Transportation Research Part E: Logistics and Transportation Review*, vol. 93, pp. 334–357, 2016.
- [23] Y. Yan, Q. Meng, S. Wang, and X. Guo, "Robust optimization model of schedule design for a fixed bus route," *Transportation Research Part C: Emerging Technologies*, vol. 25, pp. 113–121, 2012.
- [24] F. Meng, J. Qi, M. Zhang, J. Ang, S. Chu, and M. Sim, "A robust optimization model for managing elective admission in a public hospital," *Operations Research*, vol. 63, no. 6, pp. 1452–1467, 2015.
- [25] R. Driessel and L. Mönch, "Variable neighborhood search approaches for scheduling jobs on parallel machines with sequence-dependent setup times, precedence constraints, and ready times," *Computers & Industrial Engineering*, vol. 61, no. 2, pp. 336–345, 2011.
- [26] S. Wang and M. Liu, "Multi-objective optimization of parallel machine scheduling integrated with multi-resources preventive maintenance planning," *Journal of Manufacturing Systems*, vol. 37, pp. 182–192, 2015.

- [27] P. Liu, X. Y. Zhou, and L. X. Tang, "Two-agent single-machine scheduling with position-dependent processing times," *International Journal of Advanced Manufacturing Technology*, vol. 48, no. 1–4, pp. 325–331, 2010.
- [28] X. P. Wang and L. X. Tang, "A hybrid metaheuristic for the prize-collecting single machine scheduling problem with sequence-dependent setup times," *Computers and Operations Research*, vol. 37, no. 9, pp. 1624–1640, 2010.
- [29] S. Ghosh Dastidar and R. Nagi, "Scheduling injection molding operations with multiple resource constraints and sequence dependent setup times and costs," *Computers & Operations Research*, vol. 32, no. 11, pp. 2987–3005, 2005.
- [30] S. Wang, "Bi-objective optimisation for integrated scheduling of single machine with setup times and preventive maintenance planning," *International Journal of Production Research*, vol. 51, no. 12, pp. 3719–3733, 2013.
- [31] Y. Suppiah and M. K. Omar, "A hybrid tabu search for batching and sequencing decisions in a single machine environment," *Computers & Industrial Engineering*, vol. 78, pp. 135–147, 2014.
- [32] Ü. Bilge, F. Kiraç, M. Kurtulan, and P. Pekkün, "A tabu search algorithm for parallel machine total tardiness problem," *Computers and Operations Research*, vol. 31, no. 3, pp. 397–414, 2004.
- [33] M. Inuiguchi and M. Sakawa, "Minimax regret solution to linear programming problems with an interval objective function," *European Journal of Operational Research*, vol. 86, no. 3, pp. 526–536, 1995.
- [34] H. E. Mausser and M. Laguna, "A new mixed integer formulation for the maximum regret problem," *International Transactions in Operational Research*, vol. 5, no. 5, pp. 389–403, 1998.
- [35] H. E. Mausser and M. Laguna, "Heuristic to minimax absolute regret for linear programs with interval objective function coefficients," *European Journal of Operational Research*, vol. 117, no. 1, pp. 157–174, 1999.
- [36] F. J. Rodriguez, C. García-Martínez, C. Blum, and M. Lozano, "An artificial bee colony algorithm for the unrelated parallel machines scheduling problem," in *Parallel Problem Solving from Nature—PPSN XII: 12th International Conference, Taormina, Italy, September 1–5, 2012, Proceedings, Part II*, vol. 7492 of *Lecture Notes in Computer Science*, pp. 143–152, Springer, Berlin, Germany, 2012.
- [37] E. Caniyilmaz, B. Benli, and M. S. Ilkay, "An artificial bee colony algorithm approach for unrelated parallel machine scheduling with processing set restrictions, job sequence-dependent setup times, and due date," *The International Journal of Advanced Manufacturing Technology*, vol. 77, no. 9–12, pp. 2105–2115, 2015.
- [38] M. Yazdani, S. Gohari, and B. Naderi, "Multi-factory parallel machine problems: improved mathematical models and artificial bee colony algorithm," *Computers & Industrial Engineering*, vol. 81, pp. 36–45, 2015.
- [39] D. E. Goldberg, *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*, vol. 7, Springer, 2013.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

