Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 6024631, 12 pages
https://doi.org/10.1155/2018/6024631

Research Article

Multiobjective Order Acceptance and Scheduling on Unrelated Parallel Machines with Machine Eligibility Constraints

Bailin Wang and Haifeng Wang

Donlinks School of Economics and Management, University of Science and Technology Beijing, Beijing 100083, China

Correspondence should be addressed to Bailin Wang; wangbl@ustb.edu.cn

Received 27 October 2017; Revised 19 January 2018; Accepted 6 February 2018; Published 7 March 2018

Academic Editor: Eusebio Valero

This paper studies the order acceptance and scheduling problem on unrelated parallel machines with machine eligibility constraints. Two objectives are considered to maximize total net profit and minimize the makespan, and the mathematical model of this problem is formulated as multiobjective mixed integer linear programming. Some properties with respect to the objectives are analysed, and then a classic list scheduling (LS) rule named the first available machine rule is extended, and three new LS rules are presented, which focus on the maximization of the net profit, the minimization of the makespan, and the trade-off between the two objectives, respectively. Furthermore, a list-scheduling-based multiobjective parthenogenetic algorithm (LS-MPGA) is presented with parthenogenetic operators and Pareto-ranking and selection method. Computational experiments on randomly generated instances are carried out to assess the effectiveness and efficiency of the four LS rules under the framework of LS-MPGA and discuss their application environments. Results demonstrate that the performance of the LS-MPGA developed for trade-off is superior to the other three algorithms.

1. Introduction

In recent decades, the topic of order acceptance and scheduling (OAS) has attract considerable attention from scheduling researchers and production managers who practice it. The key issue of OAS is to make a joint decision of which orders are accepted (order acceptance decision) and how to schedule them (scheduling decision). Therefore, OAS is essentially different from traditional scheduling problem in which all jobs must be accepted, because the latter is just a special case of it. In OAS, a job has two options, to be accepted or rejected; thus the solution space can be up to 2^n times of that of traditional scheduling problem, where n is the number of orders.

Unrelated parallel machine environment is a common workshop where processing times of orders are machine dependent. In traditional OAS on unrelated parallel machines, it is usually assumed that orders are able to be processed on any machines. However, in reality, especially in the assemble lines with multivariety production, one machine is only eligible to process specified orders, which is called the machine eligibility constraint. Recently, some scholars have considered this constraint in the unrelated parallel machine scheduling models (see [1–3]). Moreover, in the assemble line with machine eligibility constraints, processing technologies used in machines are different; thus production costs of one order on different machines are often different and *do not depend on the processing times*. That is, an order on an eligible machine may have short processing time but high production cost, due to the skilled but not advanced processing technology. Therefore, order assignments on machines affect not only the productivity but also the profit.

This paper studies OAS on unrelated parallel machines with machine eligibility constraints (referred to as OAS-ME). Two objectives are considered. One is to maximize the total net profit with respect to the revenue, tardiness cost, and production cost. The other is to minimize the makespan, which is a classic scheduling criterion with respect to the productivity. Note that the two objectives are conflicting. The makespan is the completion time of the last finished orders; thus the solution with minimum makespan usually does not have the minimum total tardiness, let alone the minimum total

tardiness cost (namely, total *weighted* tardiness). Moreover, because production cost is independent of the processing time, the minimum makespan does not imply minimal total production cost. Therefore, minimizing the makespan does not mean minimizing the total net profit, and vice versa.

As shown in Section 3.1, OAS-ME is NP-hard; thus this paper presents four list scheduling (LS) rules and then develops a list-scheduling-based multiobjective parthenogenetic algorithm (LS-MPGA). The contributions of this paper are in the following areas: (a) for the OAS-ME, analyse some properties with respect to the objective of total net profit; (b) extend a classic LS rule with the consideration of the net profit, and present three new LS rules according to problem characteristics; (c) propose a LS-based algorithm (named LS-MPGA) with parthenogenetic operators and Pareto-ranking and selection method; (d) suggest application environments of the four LS rules under the framework of LS-MPGA through computational studies. The rest of this paper is organized as follows. Section 2 reviews the related works. Section 3 models the problem OAS-ME and analyses some properties. Section 4 proposes four LS rules and Section 5 presents the LS-MPGA. Experimental research in Section 6 inspects performance of these LS rules under the framework of LS-MPGA. Finally, Section 7 concludes the paper.

2. Literature Review

This paper considers an OAS problem on unrelated parallel machines with binary objectives. In the following, we will review the works related to the OAS on parallel machines and multiobjective unrelated parallel machine scheduling algorithms.

By now, extensive studies on OAS have been conducted in the production scheduling literature, and Slotnick [4] provided a comprehensive literature review on it. For parallel machine environment, there are some new studies presented after the review [4]. Wang et al. [5] studied the problem with two identical parallel machines and developed two heuristics and an exact algorithm based on optimal properties and the Lagrangian relaxation technique. Emami et al. [6] considered nonidentical parallel machine environment in which the revenue from an accepted order and processing times are uncertain and developed a Lagrangian relaxation algorithm. Moreover, a series of related papers treat OAS from the perspective of order rejection, and a comprehensive survey was made by Shabtay et al. [7]. For the unrelated parallel machine scheduling with rejection, the new studies after the survey [7] are reviewed as follows. Hsu and Chang [8] studied the problem with deteriorating jobs to minimize of the sum of total rejection cost and a scheduling criterion and proved that if the scheduling criterion is either total load or total completion time, the problem is solvable in polynomial time. Lin et al. [9] presented a deterministic 3-approximation algorithm and a randomized 3-approximation algorithm for two unrelated parallel machine scheduling problem with rejection. Jiang and Tan [10] presented a heuristic with worst-case ratio of 2 for an unrelated parallel machine scheduling with rejection and nonsimultaneous machine available time to minimize the sum of the makespan and total rejection cost. Above studies, including literature on unrelated parallel machines in the two survey papers, mostly employed the a priori optimization approach to sum objectives as an aggregated function, which cannot provide Pareto-optimal solutions for trade-off. Moreover, to the best of our knowledge, the machine eligibility constraint and time-independent production cost in our problem have not been considered in OAS previously in the literature.

Our problem is an extension of multiobjective unrelated parallel machine scheduling problem. Pareto-based metaheuristic is an effective approach for the multiobjective scheduling problem [11]. For the unrelated parallel machine environment, Lin et al. [12] proposed two heuristics and a Pareto-based genetic algorithm for unrelated parallel machine scheduling problem to minimize the makespan, total weighted completion time, and total weighted tardiness. Lin and Ying [13] presented a multiobjective multipoint simulated annealing (MOMSA) algorithm and Lin et al. [14] proposed a Tabu-enhanced iterated Pareto greedy algorithm for the same problem. Afzalirad and Rezaeian [15] studied the unrelated parallel machine scheduling problem with sequence-dependent setup times, release times, machine eligibility, and precedence constraints to minimize mean weighted flow time and mean weighted tardiness and improved two classic Pareto-based algorithms, NSGA-II and MOACO. Above literature indicates that Pareto-based metaheuristic is an effect approach for multiobjective scheduling problem on unrelated parallel machines.

3. Problem Modelling and Analysis

3.1. Problem Model. OAS-ME can be formally described as follows. Given a set of n nonpreemptive orders and a set of m unrelated parallel machines, this problem is to decide which orders should be accepted and how to schedule them. Each accepted order must be processed on one machine, and rejected orders are not allowed to be processed. Each machine can process at most one order at a time. Moreover, each order can only be processed on specific machines (machine eligibility constraints), and processing times and production costs of an order on the eligible machines are different from each other. Two objectives are considered. One is to maximize the total net profit, which is equal to revenues minus the sum of production costs and tardiness costs of accepted orders. The other is to minimize the makespan.

For convenience, following notations are introduced.

(a) Indexes and Sets

j: order index; n: the number of orders; O_j: the order with index j.

i: machine index; m: the number of machines; M_i: the machine with index i.

EM_j: the set of machines which are eligible to process O_j. $\mathrm{EM}_j \subseteq \{M_1, \ldots, M_m\}$.

(b) Problem Parameters

PC_{ij}, pt_{ij}: production cost and processing time of O_j on M_i, respectively.

RV_j, d_j, TC_j: revenue, due date, and unit tardiness cost of O_j, respectively.

M: a very large positive number.

(c) Decision Variables

x_j: binary variable for order acceptance decision. If $x_j = 1$, O_j is accepted, otherwise, rejected.

y_{ijk}: binary variable representing whether O_j is the kth processed order on M_i ($y_{ijk} = 1$) or not ($y_{ijk} = 0$).

ct_j: nonnegative continuous variable representing the completion time of O_j.

(d) Objective Parameters

PR_j, T_j: the net profit and tardiness of O_j, respectively.

$\sum \mathrm{PR}$, $C_{\max}$: total net profit and the makespan, respectively.

Mathematical model for the OAS-ME can be formulated as a multiobjective mixed integer linear programming (MILP) as follows.

$$\max \quad f_1 = \sum_{j=1}^{n} \mathrm{PR}_j = \sum_{j=1}^{n} \left\{ \left[\mathrm{RV}_j - \mathrm{TC}_j \times T_j - \sum_{i \in M_j} \left(\mathrm{PC}_{ij} \times \sum_{k \in O} y_{ijk} \right) \right] \times x_j \right\} \tag{1}$$

$$\min \quad f_2 = C_{\max} \tag{2}$$

$$\text{Subject to:} \quad x_j - \sum_{i \in \mathrm{EM}_j} \sum_{k=1}^{n} y_{ijk} = 0, \quad \forall j \tag{3}$$

$$1 - \sum_{j=1}^{n} y_{i,j,k+1} + M \sum_{j=1}^{n} y_{ijk} > 0, \quad \forall i, k = 1, \ldots, m-1 \tag{4}$$

$$\mathrm{ct}_l - \mathrm{pt}_{il} - \mathrm{ct}_j + M \left(2 - y_{il(k+1)} - y_{ijk} \right) \geq 0, \quad \forall i, k = 1, \ldots, m-1 \tag{5}$$

$$\mathrm{ct}_j - \sum_{i \in \mathrm{EM}_j} \sum_{k=1}^{n} y_{ijk} \mathrm{pt}_{ij} \geq 0, \quad \forall j \tag{6}$$

$$T_j \geq 0, \quad \forall j \tag{7}$$

$$T_j \geq \mathrm{ct}_j - d_j, \quad \forall j \tag{8}$$

$$C_{\max} \geq \mathrm{ct}_j, \quad \forall j \tag{9}$$

$$y_{ijk} = 0, \quad \forall j, k, \forall i \notin \mathrm{EM}_j \tag{10}$$

$$x_j, y_{ijk} \in \{0, 1\}, \quad \forall j, k \in O, \ \forall i \in M_j. \tag{11}$$

Objective (1) is to maximize the sum of net profits (revenue minus tardiness cost and production cost) of all accepted orders. Objective (2) is to minimize the makespan.

Constraints (3) restrict the relationship between variables x_j and y_{ijk}. If O_j is rejected ($x_j = 0$), it cannot be scheduled on any machine; otherwise, namely, O_j is accepted ($x_j = 1$), it must be processed on one and only one position of a machine. Constraints (4) present a position arrangement rule that if one position of a machine is free ($\sum_{j=1}^{n} y_{ijk} = 0$), its succeeding positions are no longer assigned to any orders as well ($\sum_{j=1}^{n} y_{i,j,k+1} = 0$). Constraints (5) indicate that each machine can process at most one order at a time. Constraints (3), (4), and (5) work together to restrict a feasible order sequence on a machine. Constraints (6) make sure that the completion time of any accepted order is no less than its processing time. Constraints (7) and (8) define the tardiness, and Constraints (9) define the makespan. Constraints (10) are machine eligibility constraints where an order is not allowed to be processed on an ineligible machine. Constraints (11) are binary constraints for x_j and y_{ijk}.

Scheduling decision in OAS-ME has a special case. While $\mathrm{pt}_{ij} = \mathrm{pt}_{i'j}$ ($\forall i, i' \in \mathrm{EM}_j$, $\forall j$), OAS-ME is equivalent to the identical parallel machine scheduling with machine eligibility constraints. Liao and Sheen [16] indicate that the latter problem to minimize the makespan is NP-hard; thus OAS-ME is NP-hard as well.

3.2. Objective Analysis. For the objective of total net profit, Theorem 1 holds.

Theorem 1. *If there is at least one accepted order with a negative net profit, the solution is certainly not optimal.*

Proof. Denote this solution by δ and its total net profit and the makespan by $\sum \mathrm{PR}$ and $C_{\max}$, respectively. Let the set O^- include all the orders with negative net profits in δ. By rejecting the orders in O^-, we can get a solution named δ' with two objectives $\sum \mathrm{PR}'$ and $C'_{\max}$.

First, consider the total net profits of δ and δ'. For the orders in O^-, their net profits in δ' (which are equal to zero) are greater than those in δ (which are negative). For the accepted orders which are not in O^-, their tardiness in δ' must not be greater than those in δ; thus their net profits in δ' are not smaller than those in δ. Therefore, $\sum \mathrm{PR}' \geq \sum \mathrm{PR}$.

Second, consider the makespan of δ and δ'. Rejecting an order will not increase the completion time of other accepted orders; thus $C'_{\max} \leq C_{\max}$.

Since $\sum \mathrm{PR}' \geq \sum \mathrm{PR}$ and $C'_{\max} \leq C_{\max}$, the solution δ is dominated by δ'; thus δ is certainly not optimal. □

Based on Theorem 1, we can get following corollaries.

Corollary 2. *The solution in which all orders are rejected is an extreme optimal solution with* $\sum PR = 0$ *and* $C_{\max} = 0$.

Corollary 3. *If an order* O_j *satisfies* $RV_j < \min_{i \in EM_j}\{PC_{ij}\}$, *then in any optimal solution, it must be rejected; otherwise, if* $RV_j = \min_{i \in EM_j}\{PC_{ij}\}$, *and* O_j *is accepted in an optimal solution, then there must exist an optimal solution in which* O_j *is rejected.*

For those orders that will surely be rejected in optimal solutions, they can be rejected in advance; therefore, according to Corollary 3, the following rule is set for preliminary rejection decision.

Rule 1. If $\mathrm{RV}_j \leq \min_{i \in \mathrm{EM}_j}\{\mathrm{PC}_{ij}\}$, then reject O_j.

Moreover, from Theorem 1 and Corollary 3, we can get the following theorem that gives the upper and lower bounds of the total net profits in optimal solutions. This theorem provides the basis for the design of the scheduling rule named IFH in Section 4.5 (see Section 4.5.1).

Theorem 4. *Denote the total net profit in an optimal solution as* $\sum PR^*$; *then*

$$0 \leq \sum PR^* \leq \sum_{j=1}^{n} \left(\max \left\{ 0, RV_j - \min_{i \in EM_j} \left\{ PC_{ij} \right\} \right\} \right). \quad (12)$$

Proof. For an order O_j satisfying $\mathrm{RV}_j \geq \min_{i \in \mathrm{EM}_j}\{\mathrm{PC}_{ij}\}$, ideal condition of its net profit is to schedule O_j to the machine with the minimal production cost, and this assignment will not cause tardiness, namely, $\mathrm{PR}_j^* \leq \mathrm{RV}_j - \min_{i \in \mathrm{EM}_j}\{\mathrm{PC}_{ij}\}$. According to Theorem 1, the worst condition is to reject O_j; then the net profit is 0, namely, $\mathrm{PR}_j^* \geq 0$. Therefore, if $\mathrm{RV}_j \geq \min_{i \in \mathrm{EM}_j}\{\mathrm{PC}_{ij}\}$, then $0 \leq \mathrm{PR}_j \leq \mathrm{RV}_j - \min_{i \in \mathrm{EM}_j}\{\mathrm{PC}_{ij}\}$; otherwise, according to Corollary 3, $\mathrm{PR}_j^* = 0$; thus $0 \leq \mathrm{PR}_j^* \leq \max\{0, \mathrm{RV}_j - \min_{i \in \mathrm{EM}_j}\{\mathrm{PC}_{ij}\}\}$; then Theorem 4 holds. □

4. List Scheduling Rules

4.1. Motivation and Algorithm Overview. As aforementioned, OAS-ME is a joint decision of order acceptance and order scheduling. In the problem model in Section 3.1, the variables $\{x_j \mid \forall j\}$ correspond to the order acceptance decision, and the variables $\{y_{ijk} \mid \forall i, j, k\}$ and $\{\mathrm{ct}_j \mid \forall j\}$ correspond to the order assignment subdecision and completion time subdecision of scheduling decision, respectively. According to Constraints (5) and (6), while $\{x_j \mid \forall j\}$ and $\{y_{ijk} \mid \forall i, j, k\}$ are determined, completion times can be calculated by (13). Therefore, the key to solve OAS-ME is how to find the optimal values of variables $\{x_j \mid \forall j\}$ and $\{y_{ijk} \mid \forall i, j, k\}$.

$$\mathrm{ct}_j = \begin{cases} 0, & \text{if } x_j = 0 \\ \mathrm{pt}_{ij}, & \text{if } y_{ij1} = 1,\ i \in \mathrm{EM}_j \\ \mathrm{ct}_{j'} + \mathrm{pt}_{ij}, & \text{if } y_{ijk} = y_{ij'k-1} = 1,\ i \in \mathrm{EM}_j \cap \mathrm{EM}_{j'},\ j' \in \{1, \ldots, n\}. \end{cases} \quad (13)$$

OAS-ME is a NP-hard multiobjective combinatorial optimization problem, where it is practically impossible to find an optimal solution in polynomial time. For this kind of problem, constructive heuristics and metaheuristics are effective in producing good solutions in a short time. For parallel machine scheduling problem, list scheduling (LS) is a classic constructive heuristic which provides an assignment rule to schedule orders one by one in a specific order list [17]. In this section, we introduce the order acceptance decision to a classic LS rule in Section 4.2 and present three new LS rules in Sections 4.3–4.5, which can make the order acceptance decision and order assignment decision simultaneously.

Unlike the traditional LS rules which are only for scheduling problem, these LS rules for OAS-ME make one of the following decisions for an order O_j:

(1) Reject O_j; then the variables x_j and y_{ijk} $(\forall i, k)$ are all assigned as 0.

(2) Accept O_j, assign it to a machine M_i, and insert it at the end of the order sequence on M_i. Suppose O_j is the kth order scheduled to M_i; then $x_j = 1$, $y_{ijk} = 1$, and $y_{ijk'} = 0$ where $k' \neq k$, and $y_{i'jk''} = 0$ where $i' \neq i$, $\forall k''$.

Hereinafter, the above two decisions are briefly described as "reject O_j" and "schedule O_j to M_i" in the LS rules. The calculation of the variables x_j, y_{ijk}, *and* ct_j corresponding to

each decision has been described in detail herein, therefore not further described in the subsections.

For ease of algorithm description, some notations are defined.

AV_i: available time of M_i, which is equal to the total processing time of scheduled orders on M_i;

PR_{ij}, ct_{ij}: the net profit and completion time of O_j while it is scheduled to M_i, respectively;

$C_{\max}^{j-1}$: the makespan of scheduled jobs $\{O_1, O_2, \ldots, O_{j-1}\}$;

$C_{\max}^{j}(i)$: the makespan of partial schedule $\{O_1, O_2, \ldots, O_j\}$ in which O_j is scheduled to M_i.

4.2. Extended FAM (EFAM). One of the most popular LS rules is the first available machine (FAM) rule, which schedules the next job of the list on a machine which is available first [17]. We develop an extended FAM (EFAM) for OAS-ME, in which order rejection decision is complemented and machine eligibility constraint is considered.

According to Theorem 1, while scheduling O_j, EFAM first creates a candidate set of machines (CS_j). CS_j contains the eligible machines which produce positive net profit for O_j (14). EFAM then defines Rule 2 to reject O_j or select a machine for it.

$$\mathrm{CS}_j = \left\{ i \mid i \in \mathrm{EM}_j \wedge \mathrm{PR}_{ij} > 0 \right\}. \tag{14}$$

Rule 2. If $\mathrm{CS}_j = \varnothing$, then reject O_j; otherwise, schedule O_j to M_{i^*} where $i^* = \arg\min_{i \in \mathrm{CS}_j} \{\mathrm{AV}_i\}$.

In EFAM, for one order, creating the candidate set and selecting the first available machine both take $O(m)$. There are n orders; thus the complexity of EFAM is $O(nm)$.

4.3. Highest Profit First (HPF). This subsection presents a LS rule named highest profit first (HPF) rule, which prefers solutions with high net profit. HPF first finds a machine M_{i^*} satisfying (15) and then calls Rule 3.

$$i^* = \arg\max_{M_i \in \mathrm{EM}_j} \left\{ \mathrm{PR}_{ij} \right\}. \tag{15}$$

Rule 3. If $\mathrm{PR}_{i^* j} \le 0$, then reject O_j; otherwise, schedule O_j to M_{i^*}.

HPF finds i^* for an order in $O(m)$; thus its time complexity is $O(nm)$.

4.4. Smallest Makespan First (SMF). Besides total net profit, OAS-ME has the other objective, the makespan. This subsection presents a smallest makespan first (SMF) rule. SMF schedule O_j to the machine with the minimal completion time of O_j if O_j is accepted, and the difficulty to develop SMF is how to determine whether to accept O_j. Here the criterion of EFAM is adopted, namely, if none of the machines can make a positive net profit for the considered order; then reject this order. Therefore, In SMF, the candidate set CS_j is first created by (14); then Rule 6 is employed. Time complexity of SMF is $O(nm)$.

Rule 4. If $\mathrm{CS}_j = \varnothing$, then reject O_j; otherwise, schedule O_j to M_{i^*}, where

$$i^* = \arg\min_{M_i \in \mathrm{CS}_j} \left\{ \mathrm{ct}_{ij} \right\} = \arg\min_{M_i \in \mathrm{CS}_j} \left\{ \mathrm{AV}_i + \mathrm{pt}_{ij} \right\}. \tag{16}$$

4.5. Integrated-Function-Based Rule (IFH). Above LS rules either extend an existing heuristic, or only focus on one objective, which do not take into account the balance of two objectives. This subsection presents a LS rule to trade-off total net profit and the makespan, which is formalized as Rule 5.

Rule 5. If $\mathrm{CS}_j = \varnothing$, then reject O_j; otherwise, schedule O_j to M_{i^*}, where

$$i^* = \arg\min_{i \in \mathrm{EM}_j} \left\{ w_1 G_{ij}^1 + w_2 G_{ij}^2 \right\}. \tag{17}$$

In (17), G_{ij}^1 and G_{ij}^2 are the functions reflecting the influence of assigning an order on the net profit and the makespan, and w_1 and w_2 are the weights of G_{ij}^1 and G_{ij}^2, respectively. G_{ij}^1 and G_{ij}^2 should satisfy two requirements: (a) Minimizing G_{ij}^1 and G_{ij}^2 represents the same effect trend of optimization; (b) G_{ij}^1 and G_{ij}^2 should be on the same order of magnitude. The following gives the ideas to design these functions.

4.5.1. Function G_{ij}^1 with respect to the Net Profit. For the net profit of O_j (PR_j), according to Theorem 4, Proposition 5 holds.

Proposition 5. *$0 \le PR_j \le PR_j^+$, where $PR_j^+ = RV_j - \min_{i \in EM_j}\{PC_{ij}\}$.*

Therefore, G_{ij}^1 is defined by (18), in which $G_{ij}^1 \in [0, 1]$, and a small G_{ij}^1 corresponds to a high net profit of O_j.

$$\begin{aligned} G_{ij}^1 &= \frac{\mathrm{PR}_j^+ - \mathrm{PR}_{ij}}{\mathrm{PR}_j^+} \\ &= \frac{\mathrm{TC}_j \times \max\left\{ \mathrm{ct}_{ij} - d_j, 0 \right\} + \left(\mathrm{PC}_{ij} - \min_{i \in \mathrm{EM}_j} \left\{ \mathrm{PC}_{ij} \right\} \right)}{\mathrm{RV}_j - \min_{i \in \mathrm{EM}_j} \left\{ \mathrm{PC}_{ij} \right\}}. \end{aligned} \tag{18}$$

4.5.2. Function G_{ij}^2 with respect to the Makespan. For the makespan of partial schedule $\{O_1, O_2, \ldots, O_j\}$ ($C_{\max}^j$), Proposition 6 holds.

Proposition 6. *$C_{\max}^{j-1} \le C_{\max}^j \le C_{\max}^{j-1} + \max_{i \in EM_j}\{pt_{ij}\}$.*

Proof. If O_j is rejected, then $C_{\max}^j(i) = C_{\max}^{j-1}$. If O_j is accepted and scheduled to M_i, then $C_{\max}^j(i) = \max\{\mathrm{ct}_{ij}, C_{\max}^{j-1}\}$, and the worst condition is to schedule O_j to the machine with the latest available time and the longest processing time, that is, $C_{\max}^{j-1} \le C_{\max}^j(i) \le C_{\max}^{j-1} + \max_{i \in \mathrm{EM}_j}\{\mathrm{pt}_{ij}\}$. Therefore, Proposition 6 holds. □

Therefore, (19) defines G_{ij}^2 which belongs to $[0, 1]$. Clearly, small G_{ij}^2 represents a small makespan of $\{O_1, O_2, \ldots, O_j\}$.

$$G_{ij}^2 = \frac{C_{\max}^j(i) - C_{\max}^{j-1}}{\max_{i\in \mathrm{EM}_j}\left\{\mathrm{pt}_{ij}\right\}} = \frac{\max\left\{\mathrm{ct}_{ij} - C_{\max}^{j-1}, 0\right\}}{\max_{i\in \mathrm{EM}_j}\left\{\mathrm{pt}_{ij}\right\}}. \quad (19)$$

4.5.3. Time Complexity of IFH. In IFH, for each order O_j, to calculate G_{ij}^1 and G_{ij}^2, $\min_{i\in \mathrm{EM}_j}\{\mathrm{PC}_{ij}\}$ and $\max_{i\in \mathrm{EM}_j}\{\mathrm{pt}_{ij}\}$ should be found at first, which takes $O(m)$. Therefore, finding M_{i^*} for O_j takes $O(m)$, and time complexity of IFH is $O(nm)$.

5. Multiobjective Parthenogenetic Algorithm

List scheduling supposes that the order list has been already specified; thus the difficulty of using this method is how to determine a good order list. Genetic algorithm (GA) is a popular approach to produce near-optimal solutions with flexible encoding scheme and genetic operators, and parthenogenetic algorithm (PGA) is an improved GA which is suitable for combinatorial optimization [18]; thus this paper employs PGA with LS rules to produce order lists and then generate solutions for OAS-ME. This algorithm is named LS-based multiobjective PGA (LS-MPGA), and its key points are elaborated in the following subsections.

5.1. Chromosome Encoding and Initialization. A chromosome in LS-MPGA corresponds to an order list; that is, it contains n genes which are all different, and otherwise this chromosome is invalid.

In population initialization process, LS-MPGA randomly generates N chromosomes (order lists) to maintain diversity, where N is the population size.

5.2. Parthenogenetic Operators. A chromosome in LS-MPGA is a permutation in which all genes should be different. Traditional GA has two genetic operators as crossover and mutation, which both have a difficulty in maintaining validity of a permutation chromosome. Parthenogenetic algorithm (PGA) is an improved GA proposed by Li and Tong [19], in which each chromosome has only one parent [20]. In PGA, gene recombination operators are presented as a parthenogenesis approach instead of crossover, and it can guarantee the validity of the offspring for the chromosomes encoded as permutation. There are three types of gene recombination operators: gene shift operator, gene exchange operator, and gene inverse operators, which are employed in LS-MPGA.

(a) *Gene shift operator* evolves chromosomes by an *insertion* process with probability p_s. It inserts the gene in position h_1 into position h_2, where $h_1 \neq h_2$.

(b) *Gene exchange operator* implements a *swap* process with probability p_e. This operator swaps the genes in positions k_1 and k_2, where $k_1 \neq k_2$.

(c) *Gene inverse operator* is a *reversion* process with probability p_{inv}. This operator reverses sequence of genes between position l_1 and l_2, where $1 \leq l_1 \leq l_2 \leq n$.

Figure 1 illustrates examples of these operators, in which a parent evolves offspring chromosomes by these operators with $h_1 = k_1 = l_1 = 2$ and $h_2 = k_2 = l_2 = 5$.

One issue of parthenogenetic operation is how to compose these genetic operators. Insertion and swap are two of the most widely used neighbourhood structures in scheduling [21, 22]; thus, to keep a pure insertion or swap operation, gene shift operator and gene exchange operator are totally executed at most once in one genetic operation. Furthermore, to maintain the diversity, if neither gene shift operator nor gene exchange operator is executed, gene inverse operator must be applied. These considerations are formalized into following parthenogenetic rules, where a, b, and c are random numbers within the range of $[0, 1]$.

Rule 6. If $a \leq p_s$, then call *gene shift operator( )*.

Rule 7. If $(a > p_s) \wedge (b \leq p_e)$, then call *gene exchange operator( )*.

Rule 8. If $[(a > p_s) \wedge (b > p_e)] \vee (c \leq p_{\mathrm{inv}})$, then call *gene-inverse-operator( )*.

5.3. Pareto-Ranking and Selection. OAS-ME is a multiobjective optimization, and Pareto-optimal solutions are practical when considering real-life problems since the final solution of the decision-maker is always a trade-off. There have been many effective multiobjective GAs to approximate the true Pareto points, and a classic one is the fast nondominated sorting genetic algorithm (NSGA-II) proposed by Deb et al. [23]. NSGA-II does not need external archive to store discovered nondominated solutions and has high efficiency and low space complexity [24]; thus its ranking and selection approach is applied to LS-MPGA.

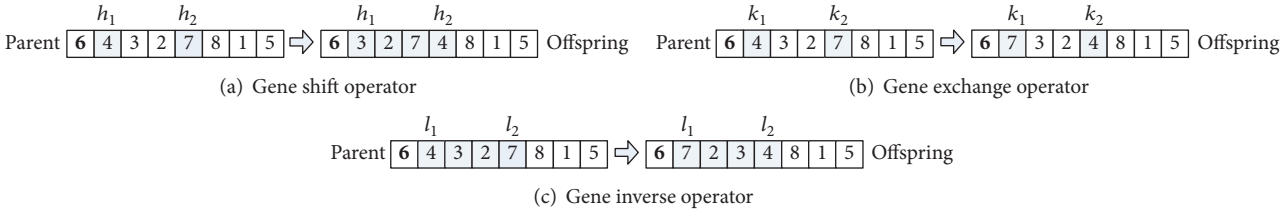
5.4. Procedure of LS-MPGA. LS-MPGA is terminated when the generation reaches the specified maximal generation $T_{\max}$. Main procedure of LS-MPGA is shown in Procedure 1.

The advantage of LS-MPGA can be explained based on three points. The first is the simple chromosome encoding scheme with only n genes due to the utility of LS rule. Second, LS-MPGA employs parthenogenetic operators to effectively avoid reproducing invalid offspring. Last, LS-MPGA can provide a variety of nondominated solutions by Pareto-ranking and selection method.

6. Computational Experiments

This section conducts an experimental study to evaluate the performance of proposed LS rules under the framework of LS-MPGA (referred to as EFAM-MPGA, HPF-MPGA, SMF-MPGA, and IFH-MPGA, respectively). Algorithms in this section are all coded in C# language and implemented on a computer with Intel Core i5-6300U/CPU 2.40 GHz 2.50 GHz and RAM 8.00 GB.

6.1. Comparison Algorithm and Parameter Settings. As mentioned in Section 5.3, NSGA-II is a classic and effective



(a) Gene shift operator

(b) Gene exchange operator

(c) Gene inverse operator

Figure 1: Genetic recombination operators of LS-MPGA.

Randomly generate N chromosomes to form an initial population P_0;
For each chromosome in P_0 **Do**
  Call a *LS rule* to produce a solution with a rejection set and a schedule;
  Calculate its total net profit and the makespan;
End For
For $t = 1$ to $T_{\max}$ **Do**
  $P_t = Q_t = \varnothing$;
  For $i = 1$ to N **Do**
    Call Rules 6–8 to evolve a new chromosome; add it to Q_t;
    Call a *LS rule* to produce a solution with a rejection set and a schedule;
    Calculate its total net profit and the makespan;
  End For
  If $(t \neq T_{\max})$ **Then**
    Call the ranking and selection approach of NSGA-II to select the best N chromosomes in $P_{t-1} \cup Q_t$ and then add them to P_t;
  Else
    Calculate ranks of all chromosomes in $P_{t-1} \cup Q_t$ by NSGA-II, and output the non-dominated set R_1;
  End If
End For

Procedure 1: LS-MPGA.

multiobjective GA [24]; thus we use it as a benchmark algorithm throughout this section. In NSGA-II, the encoding scheme and genetic operators are problem-related, which are described in detail as follows.

(a) Encoding Scheme. OAS-ME is a joint decision problem; thus a chromosome in NSGA-II contains $2n$ genes, which are divided into two subsets $\{\{a_1, \ldots, a_n\}, \{b_1, \ldots, b_n\}\}$ to make the order acceptance and order scheduling decisions simultaneously. Subset $\{a_1, \ldots, a_n\}$ corresponds to an order list, where gene a_j is the index of jth scheduled order. Subset $\{b_1, \ldots, b_n\}$ determines the accepted orders and their processing machines. Based on the idea proposed by Shabtay et al. [7], if $b_i = m+1$, then reject order j; otherwise, schedule order j to machine b_i.

(b) Genetic Operators. The two subsets in a chromosome are restricted by different constraints; thus NSGA-II employs different genetic operators for them. Subset $\{a_1, \ldots, a_n\}$ is a permutation which requires values of all genes to be different; then the one-point crossover and swap mutation are adopted [1]. Subset $\{b_1, \ldots, b_n\}$ is encoded by mutually independent positive integers, and it is evolved by the traditional crossover and mutation operators.

By extensive preliminary experimentations, parameters of LS-MPGAs are set as $p_s = p_e = 0.8$ and $p_{\text{inv}} = 0.2$. Set $w_1 = 1$ and $w_2 = 2$ for IFH-MPGA. The crossover and mutation probabilities in NSGA-II are 0.9 and $1/n$, respectively [23]. In all the LS-MPGAs and NSGA-II, the maximal generation $T_{\max} = 250$, and the size of population $N = 100$ if $n \leq 100$; otherwise, set $N = 200$.

A number of test instances are generated with the following parameter setting. Let DU$[a, b]$ represent a discrete uniform distribution with a range from a to b. Processing time p_{ij} is generated from DU$[0.5 \times p_{ij}^*, p_{ij}^*]$, where $p_{ij}^* \in$ DU$[1, 99]$. Revenue, production cost, unit tardiness cost, and due date are generated from DU$[50, 550]$, DU$[50, 100]$, DU$[0.5, 1.5]$, and DU$[50, 50 \times n/(m-1)]$, respectively.

Problem sizes of test instances are set as $n = \{20, 50, 100\}$ and $m = \{2, 4, 6, 8, 10\}$; $n = 150$ and $m = \{2, 4, 6, 8, 10, 15\}$; $n = 200$ and $m = \{2, 4, 6, 8, 10, 15, 20\}$. According to the problem size, this set of experiments are divided into 28 groups, and 50 instances are generated randomly for each group. So there are $28 \times 50 = 1400$ instances in total.

To measure the quality of obtained Pareto front, we adopt the metrics Υ and Δ presented by Deb et al. [23].

Metric Υ measures the extent of convergence to reference Pareto front PF_{ref} (20). In (20), n_A is the number of obtained Pareto front PF_A by algorithm A, and dd_i is the Euclidean distance between member i in PF_A and its nearest member in PF_{ref}. Small Υ means a good convergence.

$$\Upsilon = \frac{1}{n_A} \sum_{i=1}^{n_A} dd_i. \tag{20}$$

Metric Δ measures the maintenance of diversity (21). In (21), d_f and d_l are the Euclidean distances between the extreme solutions of PF_{ref} and the boundary solutions of PF_A, d_i is the Euclidean distance between consecutive solutions of PF_A, and $\overline{d}$ is the average of all d_i. Algorithm with better diversity has a smaller Δ.

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{n_A - 1} \left| d_i - \overline{d} \right|}{d_f + d_l + (n_A - 1)\,\overline{d}}. \tag{21}$$

These metrics both contain a reference Pareto front set PF_{ref}, which normally consists of the true Pareto front. However, true Pareto front is unknown for our problem because of its NP-hardness. To find a PF_{ref} that is approximate the true Pareto front, we analyse the respective bound of the two objectives as follows. For OAS-ME, if accepted orders are confirmed, the upper bound of the net profit $\sum PR^+$ can be calculated by (22) according to Theorem 4 and the lower bound of the makespan $C^-_{\max}$ by (23) [25]. The two bounds can form a point in the solution space as $(\sum PR^+, C^-_{\max})$, hereinafter referred as to the *bound-point* of an order acceptance solution.

$$\sum PR^+ = \sum_{j=1}^{n} \left\{ \left(RV_j - \min_{i \in M_j} PC_{ij} \right) \times x_j \right\} \tag{22}$$

$$C^-_{\max} = \frac{\sum_{j=1}^{n} \left\{ x_j \times \min_{i \in EM_j} p_{ij} \right\}}{m}. \tag{23}$$

Obviously, bound-points of all order acceptance solutions can construct an outer boundary line of the true Pareto front. However, it is hard to exhaustively list all order acceptance solutions in polynomial time, because for an OAS-ME with n orders, the number of order acceptance solutions is 2^n, which is an exponential function of the problem size. In order to obtain a reference Pareto front set PF_{ref} in a reasonable time, we present a *bound-based approach* as follows. First, extract the nondominated solutions from all solutions found by the four LS-MPGAs and NSGA-II; then, calculate the bound-points of these nondominated solutions; at last, add the bound-points which are not dominated by the other bound-points to PF_{ref}. Essentially, the *bound-based approach* produces a PF_{ref} consisting of nondominated bound-points for the solutions found by the four LS-MPGAs and NSGA-II.

6.2. Numerical Results and Analysis. Means (Avg.) and standard deviations (Std.) of the two metrics Υ and Δ for 28 test groups are listed in Tables 1 and 2, respectively.

Results in Table 1 reveal that, in terms of the average value of Υ, IFH-MPGA is the best, followed by SMF-MPGA, and the four LS-MPGAs are all significantly better than NSGA-II. It indicates the effectiveness of the framework of LS-MPGA. Moreover, the values of Υ obtained by IFH-MPGA, SMF-MPGA, and EFAM-MPGA are small. The minimal Υ values obtained by IFH-MPGA, SMF-MPGA and EFAM-MPGA are 0.91%, 0.63% and 0.73%, respectively. It implies the good convergence of the three algorithms which can produce the solutions that are close to the bound-points.

It can be observed from Table 1 that, with the increase of m, the values of Υ by all the algorithms increase obviously, while the growth of those by IFH-MPGA is the slowest. In addition, for the instances with small m, such as those with $n = 20$ and $m = 4$, $n = 50$ and $m \leq 4$, $n = 100$ and $m \leq 6$, and $n = 150, 200$ and $m \leq 10$, values of Υ obtained by SMF-MPGA are much better (lower) than those of other LS-MPGAs. For the instances with large m, values of Υ obtained by IFH-MPGA are significantly better than those of other algorithms.

Experimental results in Table 2 show that the best (smallest) values of Δ are often obtained by HPF-MPGA, followed by IFH-MPGA. HPF-MPGA produced the best values of Δ for the instances with large number of orders ($n \geq 100$), and IFH-MPGA for those with small n. There is no significant difference between EFAM-MPGA and SMF-MGPA, which are both worse than the above two algorithms but better than NSGA-II on average. Average objective values of the boundary solutions with maximal net profit obtained by the five algorithms are further collected in Table 3, and solution distributions of two randomly generated instances are shown in Figure 2. The following characteristics of these algorithms are observed from these data.

(a) The minimal makespan is often obtained by SMF-MPGA (Figure 2), and the maximal total net profit is often obtained by HPF-MPGA (Table 3 and Figure 2). (b) Through statistics of Table 3, deviations of total net profit obtained by EFAM/SMF/HPF-MPGA/NSGA-II to those by HPF-MPGA are 2.78%, 2.70%, 1.10%, and 4.10%, respectively, and deviations of the makespan are 37.98%, 39.04%, 31.79%, and 9.87%, respectively. It implies that boundary solutions of HPF-MPGA usually have better net profit than those of other algorithms but with much larger makespan. (c) In Figure 2 and Table 3, most of the nondominated solutions and boundary solutions produced by NSGA-II are dominated by the four LS-MPGAs, again demonstrating that the proposed LS-MPGA framework and LS rules are effective.

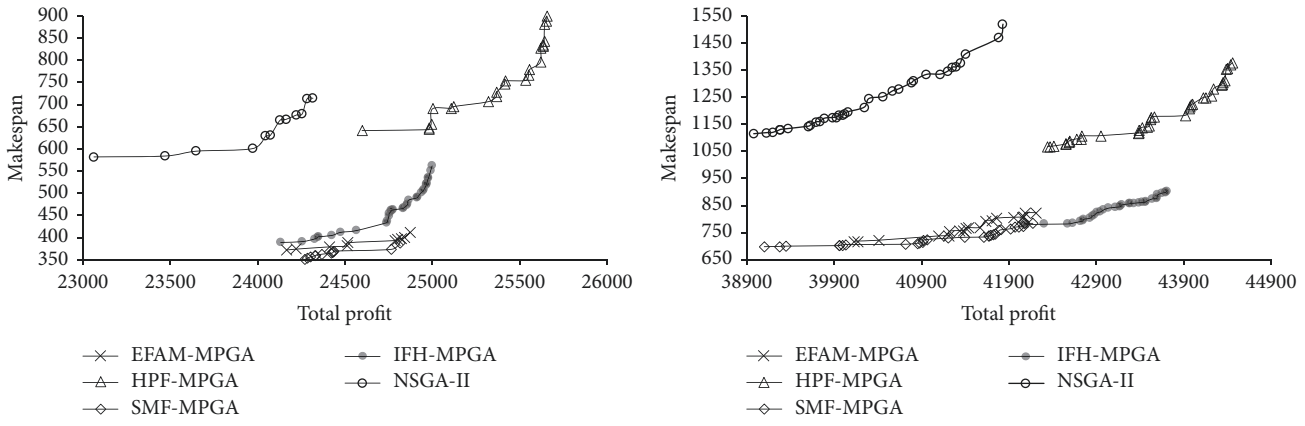
To measure the algorithm efficiency, Table 4 collects the average CPU times for all instances and for the instances with three problem sizes 20×2, 100×10, and 200×20. Note that the population size N for the problem size 200×20 is twice that for 20×2 and 100×10; thus these algorithms take much longer time to solve the former instances. Results indicate that, in terms of computational time, TGA, SMF-MPGA, HPF-MOGA, and EFAM-MPGA are very similar and very short, while IFH-MPGA is the longest one. However, for the instances with the largest problem size 200×20, the least efficient algorithm IFH-MPGA with $T_{\max} = 250$ and $N = 200$ can be terminated in 16 s, which means that the average computational time of one generation is just 0.064 s. It indicates that all these algorithms can produce solutions in a reasonable time.

6.3. Application Environments of These Algorithms. The above experimental results indicate that the performance of LS-MPGAs is much better than that of NSGA-II. According to these results, the application environments of the four LS-MPGA algorithms are suggested as follows.

(a) If a decision-maker prefers the objective of total net profit, HPF-MPGA is a good choice. However, HPF-MPGA

Table 1: Means and standard deviations of $\Upsilon \times 100$ (%).

n	m	EFAM-MPGA		HPF-MPGA		SMF-MPGA		IFH-MPGA		NSGA-II	
		Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
20	2	**1.84**	1.52	4.99	3.87	1.87	1.55	2.84	3.75	10.93	7.97
	4	6.35	2.88	18.40	14.80	**6.03**	3.13	8.58	10.96	32.25	15.78
	6	8.72	5.10	21.06	13.60	9.28	4.55	**8.14**	7.46	55.30	20.74
	8	**11.38**	5.67	26.78	18.27	14.08	8.39	11.57	13.49	65.77	19.53
	10	12.02	5.66	30.52	16.68	12.76	7.07	**9.19**	7.95	66.84	18.27
50	2	**1.15**	0.83	5.78	2.67	**1.15**	0.79	1.78	1.61	8.85	4.81
	4	2.60	1.65	13.30	6.92	**2.56**	1.55	3.42	3.49	26.35	8.99
	6	5.04	2.17	18.69	13.64	4.35	2.55	**3.99**	2.29	39.77	11.25
	8	9.66	5.53	24.64	13.06	8.79	5.38	**5.82**	2.46	56.88	14.28
	10	11.85	6.26	24.96	14.04	10.28	5.56	**5.93**	2.85	59.37	13.49
100	2	1.14	0.98	6.14	2.81	**1.04**	0.88	1.45	1.18	9.44	2.99
	4	1.95	1.13	7.45	3.68	**1.80**	0.93	2.76	2.55	18.67	3.90
	6	3.95	2.18	10.67	7.32	**2.91**	2.62	3.28	1.99	31.19	9.02
	8	5.60	2.26	11.80	6.56	4.06	2.60	**3.46**	1.94	38.93	6.74
	10	9.91	10.36	16.38	12.76	7.69	9.46	**6.43**	6.72	47.52	10.36
150	2	0.91	0.61	4.95	2.01	**0.87**	0.58	1.24	0.78	8.50	2.71
	4	1.47	0.55	6.64	2.93	**1.13**	0.53	1.68	0.99	16.52	3.59
	6	3.74	6.00	8.76	4.54	**3.10**	6.54	3.66	4.45	26.84	5.56
	8	4.49	2.79	8.87	4.03	**2.77**	2.33	3.43	1.97	36.40	7.08
	10	5.89	3.14	10.10	5.62	**3.57**	2.53	3.69	3.43	45.02	7.29
	15	11.58	8.08	13.70	11.45	8.15	7.20	**3.21**	1.71	53.55	9.53
200	2	0.73	0.46	5.37	1.97	**0.63**	0.39	0.91	0.52	9.48	2.61
	4	1.35	0.65	6.26	2.83	**0.99**	0.54	2.23	1.25	15.74	3.62
	6	2.76	2.27	8.21	4.26	**1.80**	1.95	3.24	1.89	27.18	5.47
	8	4.24	2.25	7.96	4.90	**2.14**	1.88	4.40	3.45	33.95	5.07
	10	5.89	2.87	9.56	4.91	**3.12**	2.21	5.02	5.38	42.98	7.45
	15	9.91	6.72	9.45	5.76	7.02	5.50	**3.33**	2.97	48.93	8.24
	20	13.02	10.43	8.18	4.98	9.60	9.46	**2.69**	2.61	51.99	10.75
Avg.		5.68		12.48		4.77		**4.19**		35.18	



(a) Solutions for an instance with 100 × 10

(b) Solutions for an instance with 200 × 10

Figure 2: Solutions of the four algorithms for instances with problem sizes of 100 × 10 and 200 × 10.

Table 2: Means and standard deviations of Δ.

n	m	EFAM-MPGA		HPF-MPGA		SMF-MPGA		IFH-MPGA		NSGA-II	
		Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
20	2	**0.693**	0.205	0.716	0.185	0.695	0.207	0.699	0.221	0.867	0.134
	4	0.711	0.165	0.793	0.115	0.712	0.148	**0.697**	0.158	0.899	0.109
	6	0.685	0.197	0.779	0.135	0.688	0.165	**0.682**	0.187	0.884	0.110
	8	0.728	0.122	0.784	0.144	0.726	0.112	**0.689**	0.138	0.927	0.083
	10	**0.691**	0.132	0.835	0.127	0.758	0.117	0.692	0.161	0.943	0.081
50	2	0.734	0.119	**0.715**	0.114	0.740	0.114	0.742	0.117	0.779	0.092
	4	0.804	0.105	0.816	0.091	0.816	0.099	**0.788**	0.109	0.815	0.101
	6	0.780	0.106	0.829	0.095	0.781	0.103	**0.748**	0.095	0.862	0.064
	8	0.782	0.098	0.823	0.099	0.784	0.093	**0.727**	0.123	0.881	0.067
	10	0.810	0.074	0.830	0.087	0.825	0.077	**0.730**	0.102	0.908	0.067
100	2	0.849	0.070	**0.777**	0.093	0.854	0.055	0.851	0.060	0.808	0.083
	4	0.895	0.053	**0.812**	0.064	0.886	0.054	0.862	0.059	0.817	0.078
	6	0.873	0.070	**0.824**	0.085	0.898	0.058	0.876	0.064	0.836	0.070
	8	0.874	0.058	**0.811**	0.075	0.874	0.053	0.853	0.065	0.854	0.078
	10	0.890	0.071	**0.829**	0.079	0.886	0.070	0.837	0.078	0.878	0.057
150	2	0.867	0.044	**0.785**	0.064	0.870	0.041	0.866	0.039	0.813	0.061
	4	0.902	0.042	**0.795**	0.049	0.888	0.057	0.873	0.036	0.814	0.059
	6	0.885	0.060	**0.804**	0.071	0.889	0.082	0.875	0.056	0.831	0.066
	8	0.893	0.049	**0.819**	0.073	0.896	0.067	0.874	0.055	0.846	0.055
	10	0.895	0.078	**0.819**	0.064	0.892	0.061	0.883	0.065	0.854	0.048
	15	0.918	0.043	**0.831**	0.065	0.903	0.058	0.873	0.053	0.877	0.051
200	2	0.877	0.039	**0.776**	0.046	0.881	0.037	0.877	0.041	0.808	0.054
	4	0.897	0.038	**0.807**	0.049	0.889	0.041	0.878	0.032	0.811	0.044
	6	0.885	0.056	**0.802**	0.067	0.886	0.044	0.887	0.047	0.823	0.044
	8	0.907	0.046	**0.814**	0.053	0.910	0.049	0.894	0.054	0.842	0.054
	10	0.910	0.055	**0.818**	0.067	0.906	0.040	0.895	0.044	0.857	0.044
	15	0.924	0.054	**0.833**	0.078	0.911	0.048	0.894	0.057	0.850	0.053
	20	0.941	0.039	**0.846**	0.065	0.931	0.038	0.895	0.056	0.883	0.043
Avg.		0.839		**0.804**		0.842		0.819		0.852	

has two significant drawbacks: its convergence is the worst and the obtained makespan is much larger than those of other algorithms.

(b) If a decision-maker prefers the objective of the makespan, they can adopt SMF-MPGA or EFAM-MPGA. In particular, SMF-MPGA can achieve the best convergence for the instances with small number of machines. The drawback of the two algorithms is the difficulty to produce a solution with high total net profit.

(c) If a decision-maker would like to find some trade-off solutions, especially for the instances with large number of machines, IFH-MPGA is the best. In terms of convergence, IFH-MPGA is able to produce the lowest value of Υ for most of instances, especially for those with large m. In terms of the maintenance of diversity, the value of Δ is just behind HPF-MPGA. Therefore, the overall performance of IFH-MPGA is superior to the other rules.

7. Conclusions

This paper studied the order acceptance and scheduling problem on unrelated parallel machines with machine eligibility constraints (OAS-ME), which is a NP-hard problem. Two objectives are considered as total net profit and the makespan. This paper analyses some properties of the objectives and presents four list scheduling rules named EFAM, HPF, SMF, and IFH. EFAM extends the classic first available machine (FAM) rule. HPF and SMF are the heuristics mainly designed for one objective as total net profit and the makespan, respectively. IFH presents an integrated function to balance the two objectives. Time complexities of these rules are all $O(nm)$. Based on the list scheduling method, we develop a multiobjective parthenogenetic algorithm (LS-MPGA) with parthenogenetic operators and Pareto-ranking and selection method. Computational experiments compared the performances of the four LS rules under the framework of LS-MPGA, analysed their strengths and shortcomings and suggested their application environments.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Table 3: Average values of objectives in boundary solutions with maximal net profit.

n	m	EFAM-MPGA		HPF-MPGA		SMFG-MPGA		IFH-MPGA		NSGA-II	
		$\sum$ PR	$C_{\max}$	$\sum$ PR	$C_{\max}$	$\sum$ PR	$C_{\max}$	$\sum$ PR	$C_{\max}$	$\sum$ PR	$C_{\max}$
20	2	4753	497	**4760**	545	4752	**496**	4726	499	4684	510
	4	4744	293	**4764**	369	4737	**282**	4729	302	4598	342
	6	4613	220	**4649**	297	4602	**216**	4629	233	4445	283
	8	4594	168	**4642**	246	4572	**161**	4617	182	4391	231
	10	4785	156	**4835**	219	4757	**146**	4809	162	4549	199
50	2	11531	**1145**	**11564**	1339	11531	1147	11470	1158	11404	1266
	4	11312	591	**11450**	899	11315	**588**	11346	630	11094	791
	6	11588	422	**11814**	711	11592	**415**	11740	482	11324	611
	8	11453	348	**11694**	574	11457	**340**	11649	401	11121	518
	10	11558	299	**11854**	524	11567	**291**	11786	353	11206	438
100	2	23094	1992	**23267**	2628	23082	**1977**	23012	1993	22944	2451
	4	22881	**1053**	**23442**	1861	22892	1061	23110	1181	22705	1587
	6	22915	736	**23644**	1397	22930	**726**	23356	844	22656	1163
	8	23031	570	**23915**	1082	23094	**562**	23663	668	22754	933
	10	23313	473	**24255**	961	23350	**466**	23980	561	22982	803
150	2	34764	2910	**35141**	4028	34758	**2899**	34628	2922	34603	3737
	4	34032	1465	**35052**	2571	34061	**1462**	34525	1637	33959	2292
	6	34419	1072	**35732**	1929	34510	**1062**	35219	1203	34353	1680
	8	34356	794	**35908**	1471	34457	**785**	35417	925	34242	1343
	10	34224	684	**35856**	1292	34273	**660**	35421	795	33933	1183
	15	33892	487	**35761**	990	34050	**474**	35411	582	33543	886
200	2	45615	3781	**46166**	5303	45612	**3761**	45458	3768	45485	4963
	4	45333	1921	**46847**	3226	45388	**1918**	45997	2133	45513	2990
	6	45051	1364	**46977**	2346	45131	**1344**	46229	1538	45072	2190
	8	45376	1031	**47679**	1884	45527	**1015**	46909	1201	45427	1698
	10	45133	861	**47666**	1619	45308	**845**	46938	1007	45100	1502
	15	44955	609	**47725**	1184	45147	**591**	47091	720	44755	1084
	20	44872	490	**47921**	1023	45138	**469**	47382	588	44657	900

Table 4: CPU time (in seconds) of the four algorithms.

$n \times m$	N	EFAM-MPGA	HPF-MPGA	SMF-MPGA	IFH-MPGA	NSGA-II
20×2	100	0.87	0.86	**0.84**	0.89	0.88
100×10	100	2.15	2.08	1.99	2.96	**1.95**
200×20	200	9.74	9.78	9.25	15.52	**8.07**
Avg.		3.92	3.73	3.63	5.04	**3.62**

Acknowledgments

This work was supported by the Humanity and Social Science Youth Foundation of Ministry of Education of China (no. 17YJC630143), the National Natural Science Foundation of China (nos. 71701016 and 71471015), the Beijing Natural Science Foundation (no. 9174038), and the Fundamental Research Funds for the Central Universities (no. FRF-BD-16-006A).

References

[1] C. M. Joo and B. S. Kim, "Hybrid genetic algorithms with dispatching rules for unrelated parallel machine scheduling with setup time and production availability," *Computers & Industrial Engineering*, vol. 85, article no. 3974, pp. 102–109, 2015.

[2] M. Afzalirad and J. Rezaeian, "Resource-constrained unrelated parallel machine scheduling problem with sequence dependent setup times, precedence constraints and machine eligibility restrictions," *Computers & Industrial Engineering*, vol. 98, pp. 40–52, 2016.

[3] 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.

[4] S. A. Slotnick, "Order acceptance and scheduling: a taxonomy and review," *European Journal of Operational Research*, vol. 212, no. 1, pp. 1–11, 2011.

[5] X. Wang, G. Huang, X. Hu, and T. C. Edwin Cheng, "Order acceptance and scheduling on two identical parallel machines," *Journal of the Operational Research Society*, vol. 66, no. 10, pp. 1755–1767, 2015.

[6] S. Emami, M. Sabbagh, and G. Moslehi, "A Lagrangian relaxation algorithm for order acceptance and scheduling problem: A globalised robust optimisation approach," *International Journal of Computer Integrated Manufacturing*, vol. 29, no. 5, pp. 535–560, 2016.

[7] D. Shabtay, N. Gaspar, and M. Kaspi, "A survey on offline scheduling with rejection," *Journal of Scheduling*, vol. 16, no. 1, pp. 3–28, 2013.

[8] C.-J. Hsu and C.-W. Chang, "Unrelated parallel-machine scheduling with deteriorating jobs and rejection," *Applied Mechanics and Materials*, vol. 263-266, no. 1, pp. 655–659, 2013.

[9] F. Lin, X. Zhang, and Z. Cai, "Approximation algorithms for scheduling with rejection on two unrelated parallel machines," *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 11, 2015.

[10] D. Jiang and J. Tan, "Scheduling with job rejection and nonsimultaneous machine available time on unrelated parallel machines," *Theoretical Computer Science*, vol. 616, pp. 94–99, 2016.

[11] H. Tian, K. Li, and W. Liu, "A pareto-based adaptive variable neighborhood search for biobjective hybrid flow shop scheduling problem with sequence-dependent setup time," *Mathematical Problems in Engineering*, vol. 2016, Article ID 1257060, 2016.

[12] Y.-K. Lin, J. W. Fowler, and M. E. Pfund, "Multiple-objective heuristics for scheduling unrelated parallel machines," *European Journal of Operational Research*, vol. 227, no. 2, pp. 239–253, 2013.

[13] S.-W. Lin and K.-C. Ying, "A multi-point simulated annealing heuristic for solving multiple objective unrelated parallel machine scheduling problems," *International Journal of Production Research*, vol. 53, no. 4, pp. 1065–1076, 2015.

[14] S.-W. Lin, K.-C. Ying, W.-J. Wu, and Y.-I. Chiang, "Multi-objective unrelated parallel machine scheduling: A Tabu-enhanced iterated Pareto greedy algorithm," *International Journal of Production Research*, vol. 54, no. 4, pp. 1110–1121, 2016.

[15] M. Afzalirad and J. Rezaeian, "A realistic variant of bi-objective unrelated parallel machine scheduling problem: NSGA-II and MOACO approaches," *Applied Soft Computing*, vol. 50, pp. 109–123, 2017.

[16] L.-W. Liao and G.-J. Sheen, "Parallel machine scheduling with machine availability and eligibility constraints," *European Journal of Operational Research*, vol. 184, no. 2, pp. 458–467, 2008.

[17] J. Hurink and S. Knust, "List scheduling in a parallel machine environment with precedence constraints and setup times," *Operations Research Letters*, vol. 29, no. 5, pp. 231–239, 2001.

[18] B. Wang, H. Wang, and T. Li, "Gene exchange operators of partheno-genetic algorithm for permutation flowshop scheduling with maximum and minimum time lag constraints," in *Proceedings of the International Conference on Materials Engineering and Information Technology Applications (MEITA 2015)*, pp. 596–600, Guilin, China, August 2015.

[19] M. J. Li and T. S. Tong, "A partheno-genetic algorithm and analysis of its global convergence," *Zidonghua Xuebao/Acta Automatica Sinica*, vol. 25, no. 1, pp. 68–72, 1999.

[20] J. Wang, W. Huang, G. Ma, and S. Chen, "An improved partheno genetic algorithm for multi-objective economic dispatch in cascaded hydropower systems," *International Journal of Electrical Power & Energy Systems*, vol. 67, pp. 591–597, 2015.

[21] S. Reza Hejazi and S. Saghafian, "Flowshop-scheduling problems with makespan criterion: a review," *International Journal of Production Research*, vol. 43, no. 14, pp. 2895–2929, 2005.

[22] P. Ramezani, M. Rabiee, and F. Jolai, "No-wait flexible flowshop with uniform parallel machines and sequence-dependent setup time: a hybrid meta-heuristic approach," *Journal of Intelligent Manufacturing*, vol. 26, no. 4, pp. 731–744, 2015.

[23] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[24] A. Konak, D. W. Coit, and A. E. Smith, "Multi-objective optimization using genetic algorithms: a tutorial," *Reliability Engineering & System Safety*, vol. 91, no. 9, pp. 992–1007, 2006.

[25] M. Pinedo, *Scheduling: Theory, Algorithms and Systems*, Prentice-Hall, New York, NY, USA, 1995.