

An elitist nondominated sorting hybrid algorithm for multi-objective flexible job-shop scheduling problem with sequence-dependent setups

Z.C. Li ^a, B. Qian ^{a,*}, R. Hu ^a, L.L. Chang ^b, J.B. Yang ^c

^a School of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China

^b High-Tech Institute of Xi'an, Xi'an 710025, China

^c Manchester Business School, University of Manchester, M15 6PB, UK



ARTICLE INFO

Article history:

Received 28 June 2018

Received in revised form 31 December 2018

Accepted 21 February 2019

Available online 2 March 2019

Keywords:

Multi-objective optimization
Nondominated sorting method
Estimation of distribution algorithm
Flexible job shop scheduling problem
Sequence-dependent setups

ABSTRACT

In this paper, an elitist nondominated sorting hybrid algorithm, namely ENSHA, is proposed to solve the multi-objective flexible job-shop scheduling problem (MOFJSSP) with sequence-dependent setup times/costs (MOFJSSP_SDST/C). The objectives to be minimized are the maximal completion time (i.e., makespan) and the total setup costs (TSC). The makespan is an efficiency-focused objective whereas the TSC is an economic focused one. Existing works mainly consider the efficiency-focused multiple criteria. The main highlights of this paper are threefold, i.e., the operation-based sequence model, the problem-dependent job assignment rules and the novel evolutionary framework of ENSHA. For the operation-based sequence model, this is the first time that the sequence model of MOFJSSPs has been proposed and the TSC has been treated as an independent objective in MOFJSSPs. For the job assignment rules, the solution representation is first proposed, and then three job assignment rules are specifically designed to decode solutions or sequences into feasible scheduling schemes. For the novel evolutionary framework, it works with two populations, i.e., the main population (MP) and the auxiliary population (AP). First, ENSHA adopts the elitist nondominated sorting method for evolving MP to maintain high-quality solutions regarding both the convergence and diversity. Next, a machine learning strategy based on the estimation of distribution algorithm (EDA) is proposed to learn the valuable information from nondominated solutions in MP for building a probabilistic model. This model is then used to generate the offspring of AP. Furthermore, a simple yet effective cooperation-based refinement mechanism is raised to combine MP and AP, so as to generate MP of the next generation. Finally, experimental results on 39 benchmark instances and a real-life case study demonstrate the effectiveness and application values of the proposed ENSHA.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

In modern competitive and global markets, the production scheduling plays a vital role in the manufacturing and flow industry systems. The flexible job-shop scheduling problem (FJSSP), which divides each job into several operations, is concerned with arranging multiple operations to suitable parallel machines (assignment problem) and determining the sequences of operations on all machines (sequencing problem). FJSSP is different from the classical job shop scheduling problem (JSSP) in the sense that its operations are allowed to be assigned to any machine of a given set instead of a prescribed one. Therefore, FJSSP is a generalized version of the conventional JSSP. Since JSSP is already known to

be NP-hard [1–3], it can be naturally concluded that FJSSP belongs to NP-hard.

In real-life applications, managers always desire to enhance the overall performance of the production systems, such that the philosophy of the multi-objective optimization has been extensively advocated in many practical areas. In this trend, several seminal works on multi-objective flexible manufacturing system (MOFMS) [4–6] are pioneered. Afterwards, its simplified and general form, i.e., the multi-objective FJSSP (MOFJSSP), has gradually gained much attention in the field of production scheduling [7–12]. As we know, success in practical production systems depends largely on the ability to reasonably balance the efficiency and economic profit, which in turn requires managers to adjust their ideas in routine decision making. These facts motivate us to consider the MOFJSSP where the efficient and economic criteria are optimized simultaneously.

* Corresponding author.

E-mail address: bin.qian@vip.163.com (B. Qian).

Table 1
Example of processing constraints^a.

| Product | Available machines of each operation | | | |
|---------|--------------------------------------|-------------|-------------|-------------|
| | Operation 1 | Operation 2 | Operation 3 | Operation 4 |
| 1 | m1/ m2 /m3 | m1/ m2 | - | - |
| 2 | m2 /m3 | - | - | - |
| 3 | m1/ m2 | m1/ m2 /m3 | m1 /m3 | - |
| 4 | m1 | m1/ m2 /m3 | m1/m2 /m3 | m2 /m3 |

^aA symbol “-”denotes that the operation is not required for the corresponding product.

Setups are another important feature of MOFJSSP and have aroused wide concerns of researchers [13–18]. In practical factories, frequent adjustments of processing environments and parameters are usually unavoidable which may result in many setups, especially for chemical and pharmaceutical plants. For example, in the process of batch productions, setups are often needed to clean the device (machine) or change its temperature to a certain range for two different and consecutive products. Setups are of two types: sequence-independent setups and sequence-dependent setups. The latter one is of absolute necessity for both continuous and discrete production processes. This is because, in most cases, setups are not only determined by the job itself but also strongly depend on its predecessor on the same machine [15]. Generally, there are two primary kinds of sequence-dependent setups, i.e., sequence-dependent setup times (SDSTs) and sequence-dependent setup costs (SDSCs). SDSTs represent the time used for setup, which directly affects the efficiency-focused objective. SDSCs measure the material and labor costs as well as the production loss of setup, which belongs to the economic focused objective. Very often, there is no positive correlation between SDSTs and SDSCs. Smaller SDSTs may correspond to larger SDSCs due to the potential conflict between them. In many cases, undesirable SDSCs appear to be remarkable for the overall profits of enterprises. Thus, it is meaningful to consider both SDSTs and SDSCs in the production scheduling problems.

In view of the above considerations, this paper deals with the MOFJSSP considering both SDSTs and SDSCs (MOFJSSP_SDST/C), in which the objectives are to minimize the maximum completion time (makespan) and the total setup costs (TSC). The former objective usually reflects the efficiency rates of machines, and the latter is associated with the economic costs of setups. Obviously, MOFJSSP_SDST/C is quite different from the other existing MOFJSSPs in that it treats the TSC as a separate objective. Thereby, the decision making of MOFJSSP_SDST/C is related to more difficult scenarios compared with the existing MOFJSSPs. The considered MOFJSSP_SDST/C arises from many industrial applications. Next, we will share a real word implementation that can be modeled by MOFJSSP_SDST/C.

In recent years, the authors visited several real-life factories and observed that the basic model of MOFJSSP_SDST/C has been widely adopted in production management practices. For instance, the crystallization process (CP) for crude materials is a very common chemical process in pharmaceutical industry. When selecting machines for processing certain products (crude materials), it not only depends on the capacity constraints of machines, but also relies on the physicochemical properties of these products. Each product may have multiple operations. Each operation can be processed by using any machine of the given set that satisfies the processing constraints. Hence, there exists a typical partial flexibility feature in CP. The processing time of a product is determined not only by the current operation that belongs to, but also by the assigned machine of the operation. As shown in Fig. 1, we illustrate a feasible scheduling scheme for the CP while considering the processing constraints listed in Table 1.

Apparently, some products are produced through a single CP (e.g., Product 2 in Fig. 1), and the others are produced through repetitive CPs (e.g., Products 1, 3, and 4 in Fig. 1). To prevent cross-contamination among different materials, when one operation is completed on one machine, it should take certain setup time to clean or adjust such machine before performing successive operations. The setup time depends on both the current and the immediately performing operations on each machine. Meanwhile, the industrial wastes or the losses of materials on each machine are inevitable. For example, in Fig. 2, the cleaning and adjustment of a machine should be performed before executing the next operation on the machine, so the setup cost of each operation on this machine is not only proportional to the height of fluid level (i.e., h), but also depends on the corresponding materials of the current operation and its successive one. In this paper, SDSTs as well as SDSCs among operations that belong to the same product are set to 0, which is also one of the basic assumptions in production scheduling problems [13–18].

To date, many efforts have been devoted to addressing MOFJSSP with different optimization objectives, such as the makespan, total workload (TW), total workload of critical machine (TWCM), and maximal workload (MW). Due to the NP-hardness of MOFJSSP, some solution methods based on evolutionary algorithms (EAs) have been widely adopted. Shahsavari-Poura and Ghasemishabankarehb [7] reported a hybrid evolutionary algorithm that efficiently combined GA and simulated annealing (SA) for MOFJSSP with the objective to minimize the makespan, TWCM and TW. Besides, the other EA-based approaches are also developed to solve MOFJSSP, including artificial bee colony algorithm (ABC) [8], estimation of distribution algorithm (EDA) [9], harmony search (HS) [10], and particle swarm optimization (PSO) [11,12]. Nevertheless, the previous works primarily consider the efficiency-oriented criteria, which may be not suitable for the case of MOFJSSP_SDST/C. To the best of our knowledge, studies that simultaneously consider SDSTs and SDSCs are still relatively unexplored. Therefore, it is both meaningful and practical to design an effective solution method for MOFJSSP_SDST/C.

Solution methods based on the multi-objective evolutionary algorithm (MOEA) have become one of the most important techniques for solving MOFJSSP. On the one hand, the parallel search framework of MOEA can obtain a number of nondominated solutions in reasonable computational time. On the other hand, the evolutionary model of MOEA is helpful in maintaining the diversity of population as well as avoiding undesirable local optima. Therefore, it is highly advisable to utilize MOEAs for solving MOFJSSP_SDST/C. Based on fitness assignment strategies, MOEA can be divided into two categories: dominance-based MOEA and decomposition-based MOEA. Among the state-of-the-art MOEAs, strength Pareto evolutionary algorithm II (SPEAII) [19] and non-dominated sorting genetic algorithm II (NSGAII) [20] belong to the dominance-based MOEA, whereas MOEA based on decomposition (MOEA/D) [21] and multiple-objective genetic local search (MOGLS) [22] are classified as the decomposition-based MOEA. According to our previous tests, these state-of-the-art algorithms have strong ability to explore the solution space of multi-objective problems and can obtain very satisfactory solutions for MOFJSSP_SDST/C. Moreover, as pointed out by Cai et al. [23], the elitist nondominated sorting method can be used to maintain better solutions with good diversity. Meanwhile, they suggested that utilizing the machine learning approaches to extract information from the previous search is useful for guiding the search to promising regions. These facts motivate us to develop an elitist nondominated sorting hybrid algorithm (ENSHA) that would be more effective than the above state-of-the-art algorithms for MOFJSSP_SDST/C.

The main features of ENSHA lie in two aspects: the problem-dependent job assignment rules and the novel evolutionary

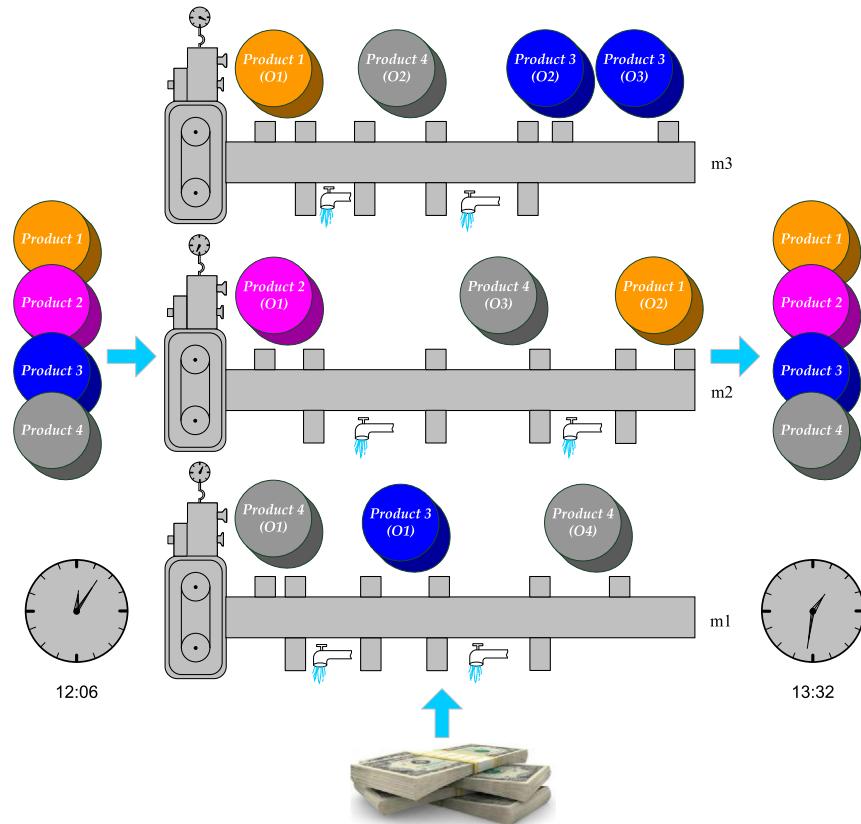


Fig. 1. Feasible scheduling scheme for the CP.

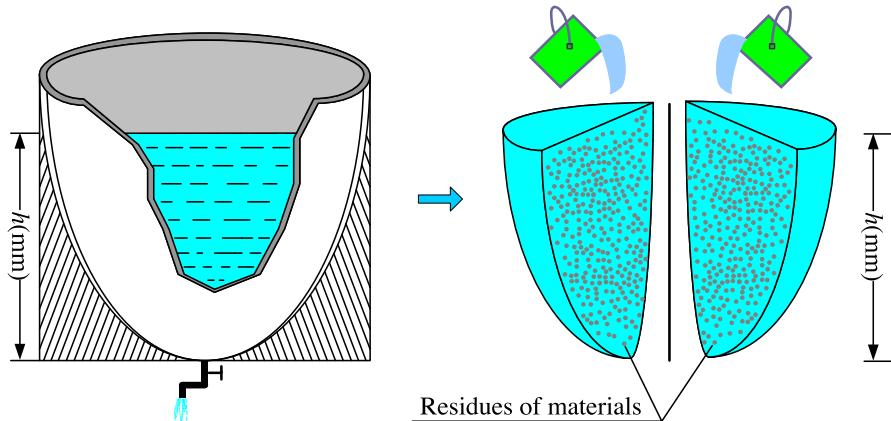


Fig. 2. Illustration of cleaning the machine.

framework of ENSHA. As for job assignment rules, after giving the operation-based solution representation, three specific job assignment rules are proposed to arrange operations in solutions or sequences to suitable parallel machines, which can locally optimize given sequences found by ENSHA. Then, they are embedded into ENSHA as decoding schemes. As for ENSHA, it uses an improved bi-population evolutionary model which consists of a main population (MP) as well as an auxiliary population (AP). In ENSHA, the genetic operators with the elitist nondominated sorting method are adopted for evolving MP with the aim to keep solutions with better convergence and diversity. Moreover, a machine learning mechanism based on the estimation of distribution algorithm (EDA) is used to build a probabilistic model for generating new individuals in AP. Furthermore, an effective cooperation-based refinement strategy is designed to combine

MP and AP to generate MP of the new generation. In such a way, MP and AP can supplement and promote each other during the search process. The novelties of this work are summarized in Table 2.

The remaining contents of the paper are organized as follows. In the next section, the operation-based sequence model of MOFJSSP_SDST/C is formulated. In Section 3, the solution representation and three problem-dependent job assignment rules are introduced. In Section 4, ENSHA is proposed and described in detail. In Section 5, experimental results and comparisons are presented and discussed. Finally, we end the paper with some conclusions and future work in Section 6.

Table 2
Differences between this work and the previous literature.

| Difference | The previous literature | This work |
|---------------------|--|---|
| Problem formulation | Existing formulations of FJSSPs and MOFJSSPs mainly consist of disjunctive graphs [24,25], mathematical models [7–10,12,13,26] and concept descriptions [27–32]. These formulations of MOFJSSPs take account of efficiency-dependent objectives [7–10,12,13,26,28–32]. | The operation-based sequence model of MOFJSSP_SDST/C is established by minimizing both the efficiency-focused and the economic focused objectives. This is the first time that the operation-based sequence model of MOFJSSPs has been proposed and the TSC (i.e., an economic focused objective) has been treated as a separate objective in MOFJSSPs. |
| Job assignment rule | The traditional rules (e.g., SPT, EST, ECT, RANDOM) are only based on one efficiency-focused objective [33–35]. | Three problem-dependent job assignment rules are constructed by considering the efficiency-focused and the economic focused objectives simultaneously. |
| Search framework | Most of MOEAs only employ conventional genetic operators (i.e., the selection, crossover and mutation) to generate offspring and perform search [14–17]. | A new bi-population-oriented evolutionary framework uses both the genetic operators and an EDA-based machine learning mechanism to effectively execute the search in different solution regions. |

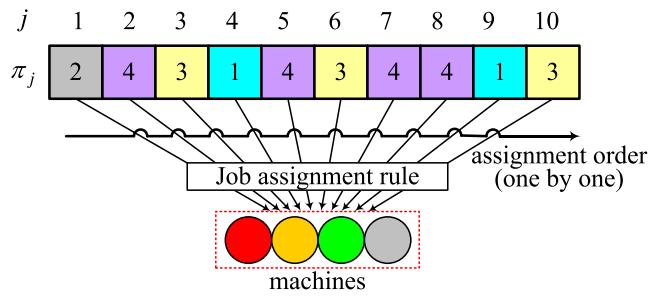


Fig. 3. Illustration of solution representation.

2. MOFJSSP_SDST/C

The researches on the evolutionary algorithms for the flow shop scheduling problems always give the corresponding job-based or operation-based sequence models, which consist of several equations to calculate the start or complete time of each job or operation on each machine. The optimization variables of a sequence model are represented as a job or operation sequence. However, due to the complexity of representation, the studies on the job shop scheduling problems and their variants (e.g., FJSSPs and MOFJSSPs) seldom propose the sequence models. For example, many studies on the evolutionary algorithms for the MOFJSSPs like to provide mixed-integer or nonlinear mathematical models [7–10,12,32], but in fact the individual or solution in each evolutionary algorithm corresponds to the variables of the unprovided sequence model, rather than the decision variables of the established mathematical model. That is, evolutionary algorithms actually optimize the variables of the sequence models and they are independent of the mathematical models. This phenomenon is easy to confuse new researchers. Thus, this section not only gives the concept description of MOFJSSP_SDST/C but also proposes an operation-based sequence model of it.

The MOFJSSP_SDST/C can be described as follows. There are n products (or jobs), m machines, and each product i ($i = 1, \dots, n$) needs opr_i operations to complete the processing. The scheduling environment is partially flexible, so that the opr_i operations of product i can only be processed by certain machines that satisfy the processing constraints. Different operations of the same product must satisfy the sequencing priority. That is, only if a previous operation is completed can the current operation be processed. The processing time of each operation depends on the assigned machine. At any time, preemption is forbidden and each machine can process at most one job at the same time. There exist SDSTs and SDSCs between the completion time of one operation and the start time of another consecutive operation on each machine.

When two operations belong to the same product, the SDSTs and SDSCs between them are set to 0. The problem is to find the sequence of products on all machines, so as to minimize the given criteria (i.e., makespan and TSC). Next, we propose the operation-based sequence model of MOFJSSP_SDST/C based on the real-life CP mentioned in Section 1.

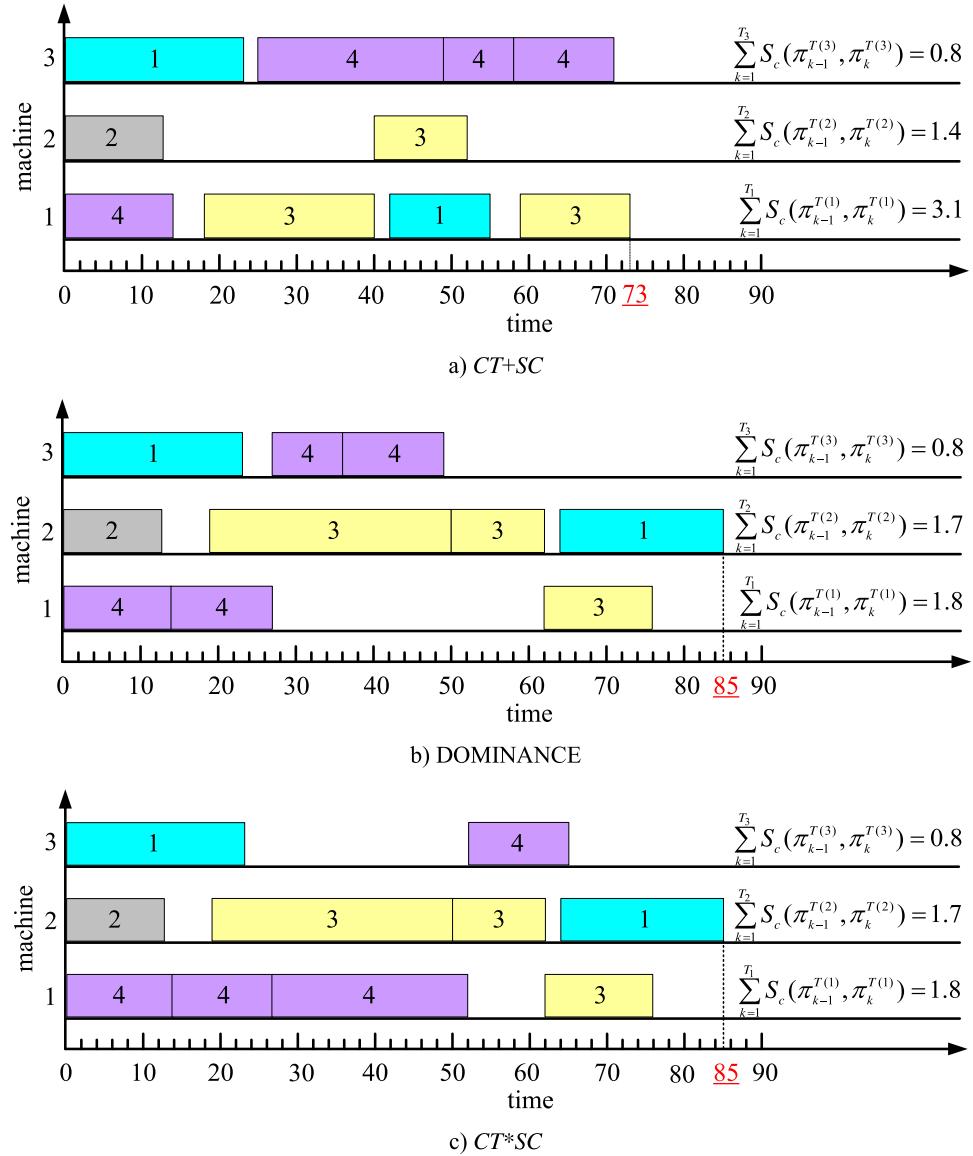
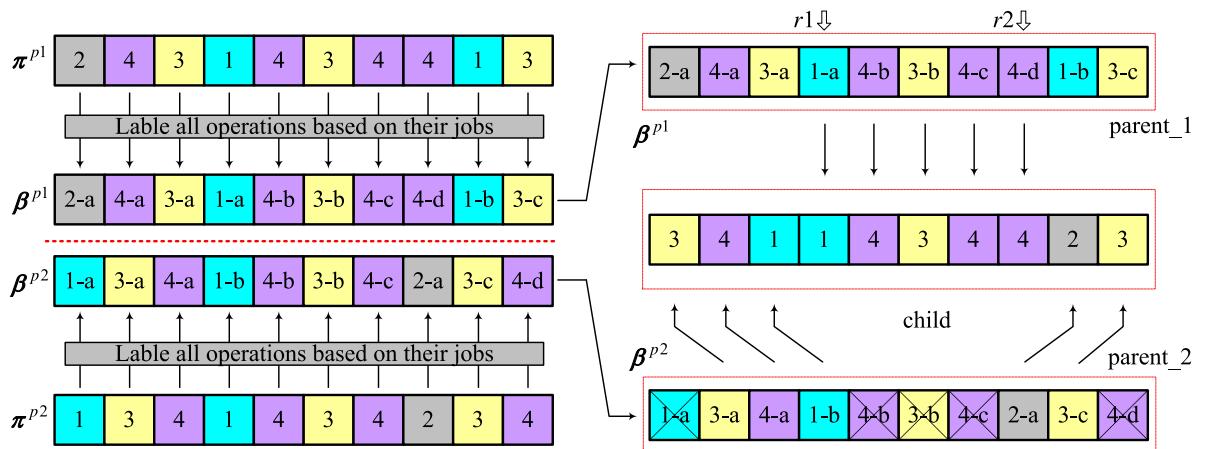
Let $TO = \sum_{i=1}^n opr_i$ denote the total number of operations, $\pi = [\pi_1, \dots, \pi_k, \dots, \pi_{TO}]$ ($\pi_k \in \{1, \dots, n\}$ for $k = 1, \dots, TO$) the operation-based sequence or schedule of the products (the operations in π are assigned to different machines from left to right by using the proposed job assignment rules in Section 3.2), T_j ($TO = \sum_{j=1}^m T_j$) the total number of the operations performed on machine j , $\pi^{T(j)} = [\pi_1^{T(j)}, \dots, \pi_l^{T(j)}, \dots, \pi_{T_j}^{T(j)}]$ the sequence of the operations performed on the j th machine ($\pi_l^{T(j)} \in \{1, \dots, n\}$ for $l = 1, \dots, T_j$), $P(\pi_l^{T(j)})$ the processing time of operation $\pi_l^{T(j)}$ ($l = 0, \dots, T_j$), $P(\pi_0^{T(j)}) = 0$, $S_t(\pi_{l-1}^{T(j)}, \pi_l^{T(j)})$ the setup time between $\pi_{l-1}^{T(j)}$ and $\pi_l^{T(j)}$ ($S_t(\pi_{l-1}^{T(j)}, \pi_l^{T(j)}) = 0$ when $l = 1$, $S_t(\pi_{l-1}^{T(j)}, \pi_l^{T(j)}) = 0$ when $\pi_{l-1}^{T(j)} = \pi_l^{T(j)}$ for $l = 2, \dots, T_j$), $Start(\pi_l^{T(j)})$ the start time of $\pi_l^{T(j)}$ ($l = 0, \dots, T_j$), $Start(\pi_0^{T(j)}) = 0$, $prod(\pi_l^{T(j)})$ the product owning $\pi_l^{T(j)}$, $pre_m(\pi_l^{T(j)})$ the machine that processes the operation of $prod(\pi_l^{T(j)})$ right before machine j ($l = 1, \dots, T_j$), $pre_m(\pi_l^{T(j)}) = 0$ when machine j is the first machine to process $prod(\pi_l^{T(j)})$, $pre_k(\pi_l^{T(j)})$ the position of $\pi_l^{T(j)}$ in $\pi^{T(pre_m(\pi_l^{T(j)}))}$ from its left side ($l = 1, \dots, T_j$), $pre_k(\pi_l^{T(j)}) = 0$ when machine j is the first machine to process $prod(\pi_l^{T(j)})$). Then, the makespan can be calculated as follows:

$$\begin{aligned} C_{\max}(\pi) = & \max\{Start(\pi_{T_1}^{T(1)}) + P(\pi_{T_1}^{T(1)}), Start(\pi_{T_2}^{T(2)}) \\ & + P(\pi_{T_2}^{T(2)}), \dots, \\ & Start(\pi_{T_m}^{T(m)}) + P(\pi_{T_m}^{T(m)})\} \end{aligned} \quad (1)$$

$$Start(\pi_l^{T(j)}) = Start(\pi_{l-1}^{T(j)}) + P(\pi_{l-1}^{T(j)}) + S_t(\pi_{l-1}^{T(j)}, \pi_l^{T(j)}), \\ pre_k(\pi_l^{T(j)}) = 0, l = 1, \dots, T_j, j = 1, \dots, m \quad (2)$$

$$\begin{aligned} Start(\pi_l^{T(j)}) = & \max\{Start(\pi_{l-1}^{T(j)}) + P(\pi_{l-1}^{T(j)}) + S_t(\pi_{l-1}^{T(j)}, \pi_l^{T(j)}), \\ & Start(\pi_{pre_m(\pi_l^{T(j)})}^{T(pre_m(\pi_l^{T(j)}))}) + P(\pi_{pre_m(\pi_l^{T(j)})}^{T(pre_m(\pi_l^{T(j)}))}), \\ & pre_k(\pi_l^{T(j)})\}, \\ pre_k(\pi_l^{T(j)}) \neq 0, l = & 1, \dots, T_j, j = 1, \dots, m. \end{aligned} \quad (3)$$

Let $S_c(\pi_{l-1}^{T(j)}, \pi_l^{T(j)})$ denote the setup cost between $\pi_{l-1}^{T(j)}$ and $\pi_l^{T(j)}$, $h(\pi_l^{T(j)})$ the average height of fluid level in terms of processing $prod(\pi_l^{T(j)})$ on all satisfied machines, $var_c(h(\pi_l^{T(j)}))$ the variable cost of waste disposal and product loss per unit $h(\pi_l^{T(j)})$

**Fig. 4.** Gantt charts of $\pi = [2, 4, 3, 1, 4, 3, 4, 4, 1, 3]$ using proposed job assignment rules.**Fig. 5.** Illustration of adopted crossover operator.

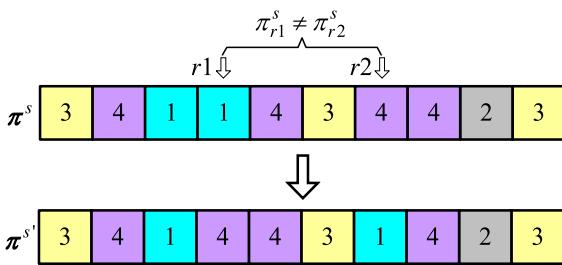


Fig. 6. Illustration of adopted mutation operator.

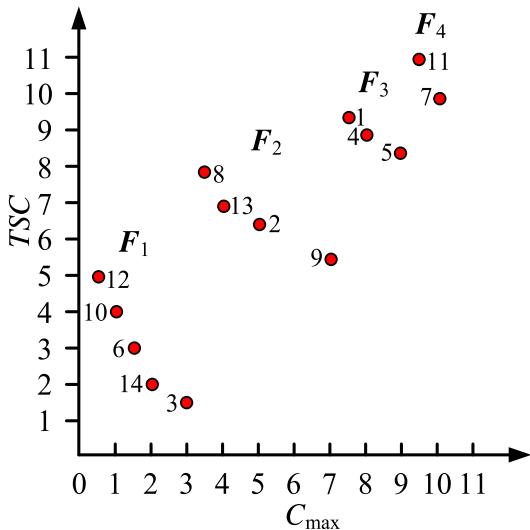


Fig. 7. Illustration of nondominated sorting method.

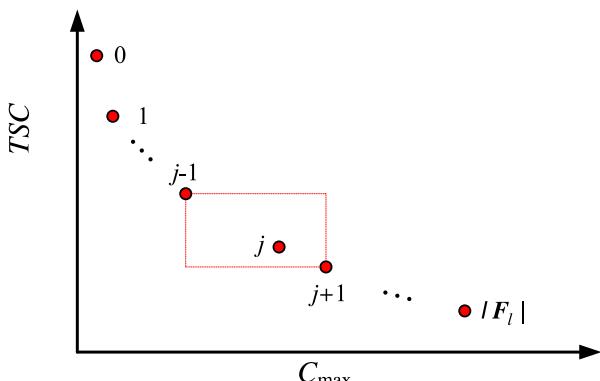


Fig. 8. Illustration of crowding distance.

$(var_c(h(\pi_l^{T(j)}))) = 0$ when $l = 0$ or T_j , $fix_c(\pi_{l-1}^{T(j)}, \pi_l^{T(j)})$ the fixed cost of adjusting machine j and performing the chemical wash between $\pi_{l-1}^{T(j)}$ and $\pi_l^{T(j)}$ ($fix_c(\pi_{l-1}^{T(j)}, \pi_l^{T(j)}) = 0$ when $l = 1$, and $fix_c(\pi_{l-1}^{T(j)}, \pi_l^{T(j)}) = 0$ when $\pi_{l-1}^{T(j)} = \pi_l^{T(j)}$ for $l = 2, \dots, T_j$), and $w(prod(\pi_l^{T(j)}))$ the cost coefficient of $var_c(h(\pi_l^{T(j)}))$. The TSC can be calculated as follow:

$$TSC = \sum_{j=1}^m \sum_{l=1}^{T_j} S_c(\pi_{l-1}^{T(j)}, \pi_l^{T(j)}), \quad (4)$$

$$S_c(\pi_{l-1}^{T(j)}, \pi_l^{T(j)}) = w(prod(\pi_{l-1}^{T(j)})) \cdot var_c(\bar{h}(\pi_{l-1}^{T(j)})) + fix_c(\pi_{l-1}^{T(j)}, \pi_l^{T(j)}), \quad (5)$$

where $S_c(\pi_{l-1}^{T(j)}, \pi_l^{T(j)}) = 0$ when $l = 1$ and $\pi_{l-1}^{T(j)} = \pi_l^{T(j)}$ for $l = 2, \dots, T_j$. For MOFJSSP_SDST/C, the makespan and TSC need to be considered simultaneously, however, they are often conflicting with each other. That is, an improvement in one objective may worsen another. Since there is usually no such a solution that is the best to all objectives, it is required to find a set of nondominated solutions in decision making [36,37]. When each operation of π has been assigned to one machine, the computational complexity (CC) of calculating $C_{\max}(\pi)$ and TSC is $O(2 * TO)$, where 2 is the number of objectives.

Unlike the mathematical models of production scheduling problems, the sequence models of these problems have no explicitly expressed constraints. As for the sequence model of MOFJSSP_SDST/C, the operational constraints are implicitly included in its equations and an operation assignment table. These constraints can be divided into two groups. The first group contains the general constraints for both the JSSPs and the FJSSPs. These general constraints are included in the Eqs. (2) and (3), which determine that each operation $\pi_l^{T(j)}$ can only be executed after the operations $\pi_{l-1}^{T(j)}$ and $\pi_{pre_k(\pi_l^{T(j)})}^{T(pre_m(\pi_l^{T(j)}))}$ are completed. The second group contains the specific processing constraints for the FJSSPs. These specific constraints are usually given in an operation assignment table (see Table 1), which requires that each operation must be executed on one of several specified machines. The difference between JSSPs and FJSSPs is that the latter one needs to assign each operation to a machine (selected from the set of specified machines in Table 1) before calculating the start times of all the operations. By adding the appropriate processing constraints, the sequence model can represent not only the common FJSSPs but also the reentrant FJSSPs (see Table 1), which widely exist in process industries and microelectronic industries. Hence, the specific processing constraints can extend the application scope of the model.

Compared with the mathematical model, the constraints of the sequence model are easy to handle. To the operation-based sequence or solution π ($\pi = [\pi_1, \dots, \pi_k, \dots, \pi_{TO}]$), the new solution or child is still feasible as long as the search operators (e.g., crossover and mutation) do not change the total number of each job's operations in the new solution. In the following proposed ENSHA, the adopted crossover and mutation can keep the total number of each job's operations unchangeable (see Section 4.1). When an operation-based sequence $\pi^{T(j)}$ ($j = 1, \dots, m$) on each machine is determined by applying a decoding scheme (i.e., a job assignment rule) to π , the start time of each operation is calculated with the equations of the sequence model and no constraints are violated. This means that any solution π generated in ENSHA is feasible. In fact, almost all the existing evolutionary scheduling algorithms choose to optimize the variables of the sequence model, since the researchers adopting the sequence model can focus on the algorithm design rather than the constraint handling.

3. Problem-dependent job assignment rules

3.1. Solution representation

Based on MOFJSSP_SDST/C's features, the operation-based encoding scheme is adopted. For example, a sequence $\pi = [\pi_1, \dots, \pi_{TO=10}] = [2, 4, 3, 1, 4, 3, 4, 4, 1, 3]$ is a solution or individual of ENSHA, when the problem size $n \times m$ and $[opr_1, \dots, opr_4]$ are set to 4×4 and $[2, 1, 3, 4]$. As shown in Fig. 3, π denotes that the unique operation of job 2 is firstly assigned to a machine based on certain job assignment rules, and then the first operation of job 4 is assigned, and so on. Apparently, the objectives of individuals are determined by both the sequence and the job assignment

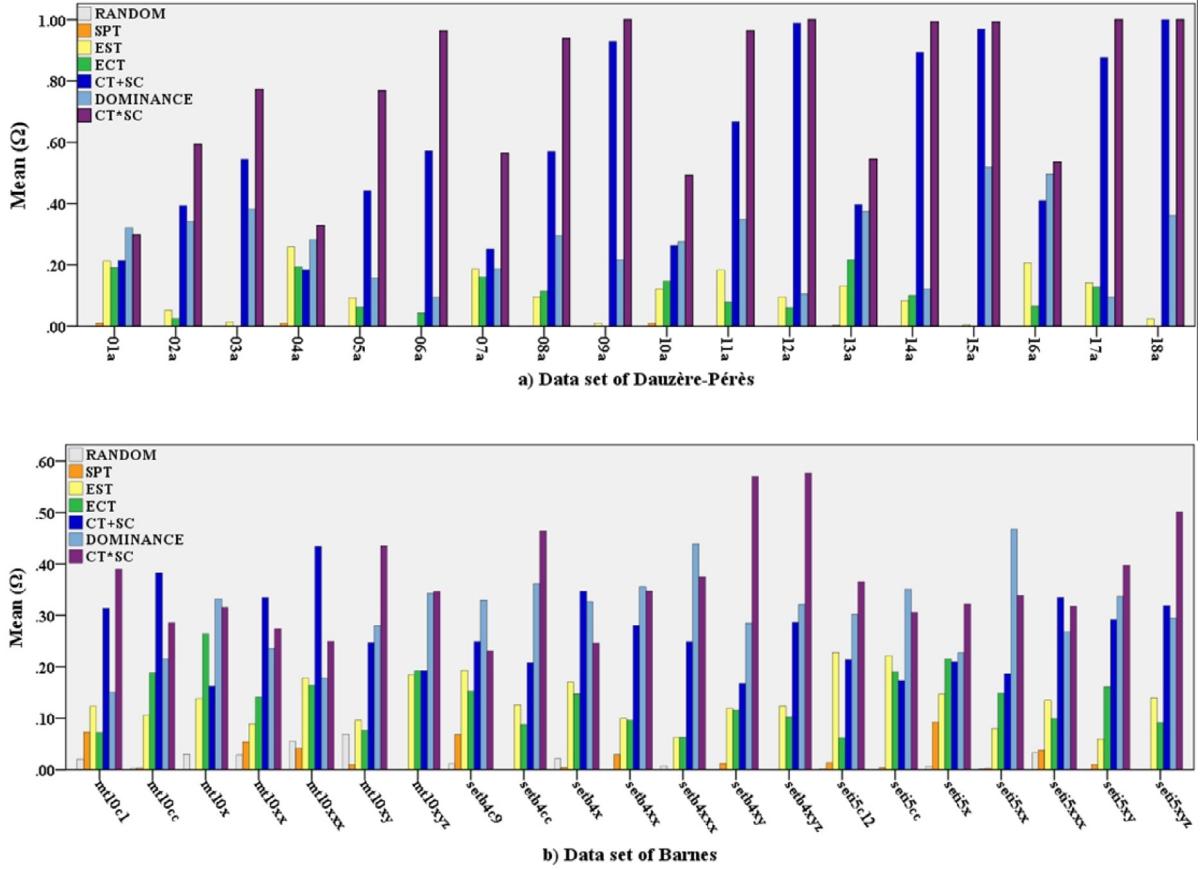


Fig. 9. Histograms of ENSHAs with different job assignment rules.

rule. Arguably, a suitable assignment rule may result in high-quality scheduling scheme. Therefore, we will put forward three problem-dependent job assignment rules in Section 3.2. And the Gantt charts for the solution [2, 4, 3, 1, 4, 3, 4, 4, 1, 3] in Fig. 3 are then given in Fig. 4, in terms of the proposed three job assignment rules.

3.2. Proposed job assignment rules

As mentioned in Section 3.1, ENSHA's solution should be converted into the feasible scheduling scheme by using job assignment rules. Nevertheless, due to the intractable partial flexibility and sequence-dependent setups in MOFJSSP_SDST/C, it may be a challenge to design effective decoding rules for such a problem. Here, the decoding rule is referred to as the job assignment rule, which is important for evaluating the objectives and improving the search quality. In existing papers, traditional job assignment rules including RANDOM [33] (select an operation at random), SPT [33–35] (shortest processing time), EST (earliest start time) [34,35] and ECT [34,35] (earliest completion time) are widely used for various kinds of scheduling problems. However, for MOFJSSP_SDST/C, the makespan and TSC are considered simultaneously, so the traditional ones may lead to poor performances. In this work, three new job assignment rules are presented to decode the solution or sequence.

For an operation π_k ($k = 1, \dots, TO$) of solution π , the set $AVA(\pi_k) = \{j_1, \dots, j_l, \dots, j_{l(\pi_k)}\}$ ($j_l \in \{1, \dots, m\}$ for $l = 1, \dots, l(\pi_k)$), which is constituted by $l(\pi_k)$ machines that satisfy the processing constraints for π_k , needs to be identified first. Thereafter, the sub-criteria of completing time (CT) and setup cost (SC) associated with π_k on each machine of $AVA(\pi_k)$ can be calculated. Let $CT_{\pi_k} = \{CT_{\pi_k j_1}, \dots, CT_{\pi_k j_{l(\pi_k)}}\}$ and $SC_{\pi_k} = \{SC_{\pi_k j_1}, \dots, SC_{\pi_k j_{l(\pi_k)}}\}$ denote the obtained sub-criteria of CTs and SCs respectively. The principles of proposed Rules 1 to 3 are given as follows:

Rule 1 (CT + SC): Select one machine $j^* \in AVA(\pi_k)$ for processing π_k that satisfies $\neg \exists j_l \in AVA(\pi_k) : (CT_{\pi_k j_l} + SC_{\pi_k j_l}) < (CT_{\pi_k j^*} + SC_{\pi_k j^*})$, $l = 1, \dots, l(\pi_k)$;

Rule 2 (CT * SC): Select one machine $j^* \in AVA(\pi_k)$ for processing π_k that satisfies $\neg \exists j_l \in AVA(\pi_k) : (CT_{\pi_k j_l} * SC_{\pi_k j_l}) < (CT_{\pi_k j^*} * SC_{\pi_k j^*})$, $l = 1, \dots, l(\pi_k)$;

Rule 3 (DOMINANCE): Select one machine $j^* \in AVA(\pi_k)$ for processing π_k that satisfies $\neg \exists j_l \in AVA(\pi_k) : ((CT_{\pi_k j_l} < CT_{\pi_k j^*}) \wedge (SC_{\pi_k j_l} < SC_{\pi_k j^*})) \vee ((CT_{\pi_k j_l} = CT_{\pi_k j^*}) \wedge (SC_{\pi_k j_l} < SC_{\pi_k j^*})) \vee ((CT_{\pi_k j_l} < CT_{\pi_k j^*}) \wedge (SC_{\pi_k j_l} = SC_{\pi_k j^*}))$, $l = 1, \dots, l(\pi_k)$.

Rules 1 to 3 are all in pursuit of finding a tradeoff between CT_{π_k} and SC_{π_k} when selecting machines, since these two sub-criteria may locally influence the objectives of MOFJSSP_SDST/C. The performance of these rules will be tested in Section 5, due to the complexity of MOFJSSP_SDST/C. The pseudo code of decoding a solution $\pi = [\pi_1, \dots, \pi_k, \dots, \pi_{TO}]$ by using any one of the proposed rules is given in Algorithm 1. Since the maximum number of machines that can process each operation π_k is m , the CC of calculating $C_{\max}(\pi)$ and TSC with one proposed job assignment rule is $O(2 * m * TO)$, where 2 is the number of objectives. That is, the CC of Algorithm 1 is $O(2 * m * TO)$.

Here, we illustrate the proposed assignment rules through the example mentioned in Fig. 3, where the operation processing constraints, processing times, and setup times/costs are listed in Tables 1, 3, and 4, respectively. For example, by using CS*TC, we can get $T_1 = 4$, $T_2 = 4$, $T_3 = 2$, $\pi^{T(1)} = [4, 4, 4, 3]$, $\pi^{T(2)} = [2, 3, 3, 1]$, and $\pi^{T(3)} = [1, 4]$. The Gantt charts of π , in terms of

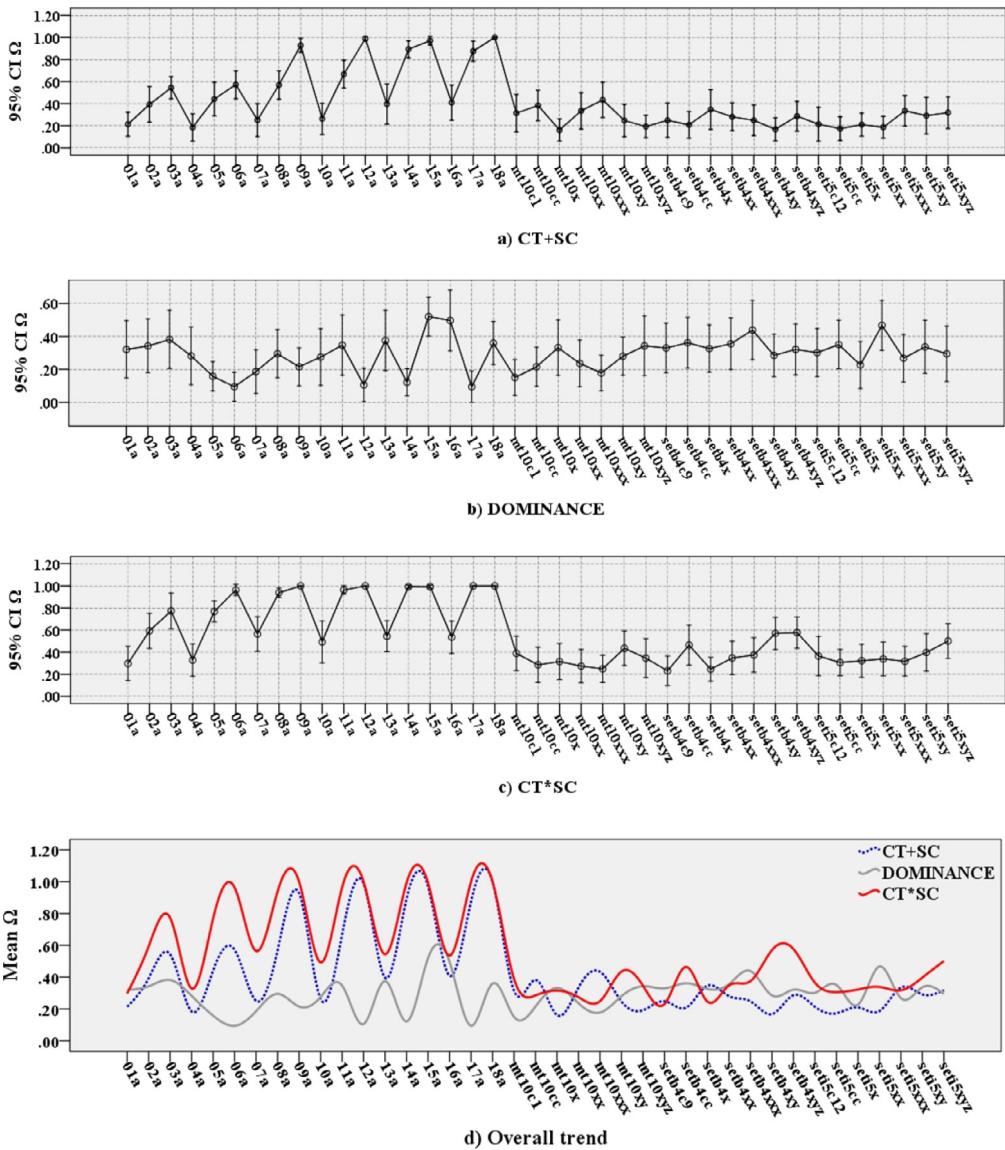


Fig. 10. 95% CI's and overall trend for proposed assignment rules.

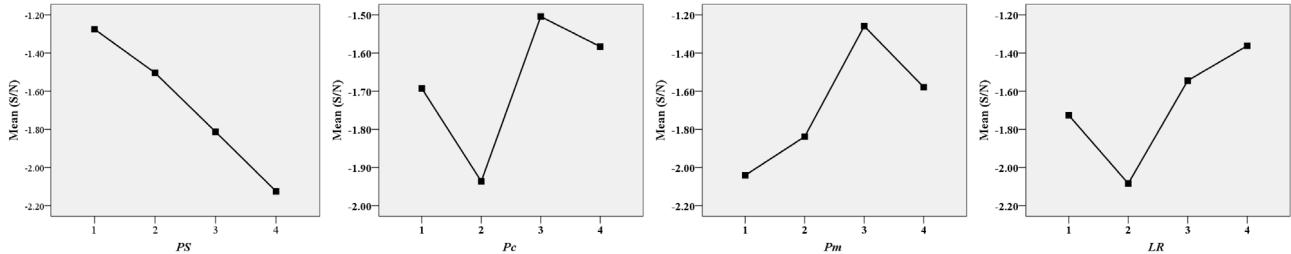


Fig. 11. Level trends of each parameter.

the three assignment rules, are illustrated in Fig. 4, in which the objectives (C_{\max} , TSC) are (73, 5.3), (85, 4.3) and (85, 4.3).

4. ENSHA for MOFJSSP_SDST/C

In this section, we will detail the proposed ENSHA for MH-PMSP_MOSST/C after explaining the genetic operators, the elitist nondominated sorting method, the EDA-based machine learning mechanism, and the cooperation-based refinement strategy.

4.1. Genetic operators

ENSHA uses a novel bi-population evolutionary framework to guide the search to the promising regions in solution space, so it needs a suitable crossover to execute search nearby these regions for finding better neighbors. A two-point crossover method known as Partial-mapped crossover (PMX) [35] has been widely used in the fields of combinatorial optimization problems [38–41]. PMX is good at preserving long subsequences (i.e., building blocks) of parents [42], which means it can effectively search the

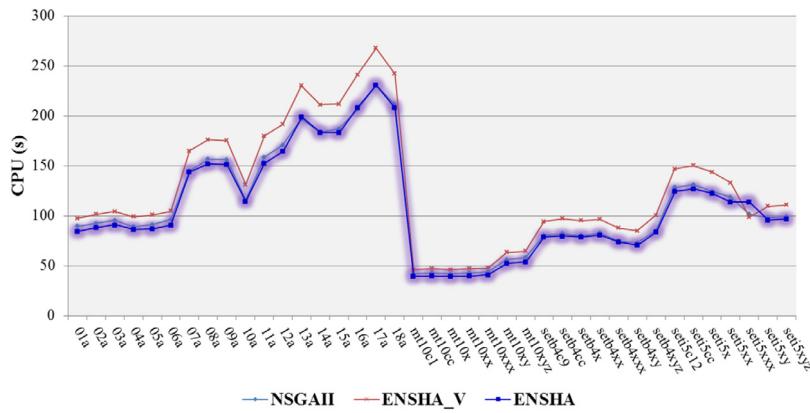


Fig. 12. Average CPU time of NSGAII, ENSHA_V and ENSHA. (ENSHA is highlighted in shadow).

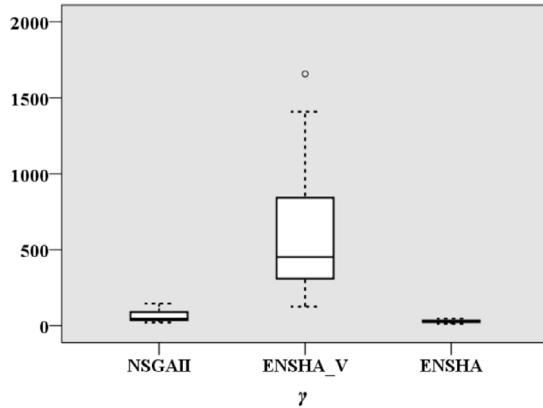


Fig. 13. Box plots of NSGAII, ENSHA_V and ENSHA for 39 instances (mean values of γ).

Table 3
Setup times/costs.

| Product | Setup time | | | | Product | Setup cost | | | |
|---------|------------|---|---|---|---------|------------|-----|-----|-----|
| | 1 | 2 | 3 | 4 | | 1 | 2 | 3 | 4 |
| 1 | 0 | 1 | 4 | 2 | 1 | 0 | 0.6 | 1 | 0.8 |
| 2 | 3 | 0 | 6 | 5 | 2 | 1.5 | 0 | 1.4 | 0.7 |
| 3 | 2 | 7 | 0 | 3 | 3 | 0.3 | 0.1 | 0 | 0.5 |
| 4 | 9 | 8 | 4 | 0 | 4 | 2.1 | 0.2 | 1.8 | 0 |

Table 4
Processing times^a.

| Product | Operation 1 | | | Operation 2 | | | Operation 3 | | | Operation 4 | | |
|---------|-------------|----|----|-------------|----|----|-------------|----|----|-------------|----|----|
| | m1 | m2 | m3 |
| 1 | 12 | 34 | 23 | 13 | 21 | — | — | — | — | — | — | — |
| 2 | — | 13 | 24 | — | — | — | — | — | — | — | — | — |
| 3 | 22 | 31 | — | 27 | 12 | 23 | 14 | — | 17 | — | — | — |
| 4 | 14 | — | — | 13 | 33 | 24 | 25 | 11 | 9 | — | 40 | 13 |

^aA symbol “—”denotes that the operation is not required for the corresponding product.

regions nearby the parents. Moreover, due to its simple repairing scheme of infeasibility, PMX can quickly search new regions (i.e., generating children) without increasing computation complexity. Hence, PMX is adopted in ENSHA. Let π^{p1} and π^{p2} denote the parent individuals selected from the current population with crossover probability P_c . To ensure the feasibility of newly recombinated children, we need to distinguish operations of the same product (or job) in an individual. To do so, we label all operations of the same product in π^{p1} and π^{p2} with symbols “a, b, c, ...” according to their order from left to right. Thereby, the labeled β^{p1}

and β^{p2} of π^{p1} and π^{p2} can be obtained. Note that the labeled β^{p1} and β^{p2} are only used to distinguish the operations that belong to the same product.

As shown in Fig. 5, for $\pi^{p1} = [2, 4, 3, 1, 4, 3, 4, 4, 1, 3]$ and $\pi^{p2} = [1, 3, 4, 1, 4, 3, 4, 2, 3, 4]$, we can get $\beta^{p1} = [2-a, 4-a, 3-a, 1-a, 4-b, 3-b, 4-c, 4-d, 1-b, 3-c]$ and $\beta^{p2} = [1-a, 3-a, 4-a, 1-b, 4-b, 3-b, 4-c, 2-a, 3-c, 4-d]$. Thereafter, two different genes r_1 and r_2 ($r_1 < r_2$) are randomly selected, and the elements in β^{p2} that belong to the subset $\{\beta_{r_1}^{p1}, \dots, \beta_{r_2}^{p1}\}$ should be locked (marked by crosses) to avoid infeasibility. Then, the subsequence $[\pi_{r_1}^{p1}, \dots, \pi_{r_2}^{p1}]$ of β^{p1} is directly copied to the corresponding positions of the child individual, and in turn, the active operations of β^{p2} (i.e., 3, 4, 1, 2 and 3) are copied to the empty positions of the child from left to right.

Besides, a simple Interchange-based mutation operator is applied in ENSHA, so as to perform perturbations for the offspring and avoid trapping into local optima in the search process. After performing the crossover operation, an individual is selected (i.e., π^s) according to the mutation probability P_m to execute mutation operator. The mutation operator is illustrated in Fig. 6. From Figs. 5 and 6, it can be seen that both the PMX and the Interchange-based mutation keep the total number of each job's operations unchangeable. Therefore, the adopted crossover and mutation can avoid the infeasibility of solutions.

4.2. Elitist nondominated sorting method

In ENSHA, MP and the cooperation-based refinement strategy are all related to the elitist nondominated sorting method. Next, we will introduce the elitist nondominated sorting method, including the nondominated sorting and the crowding distance assignment.

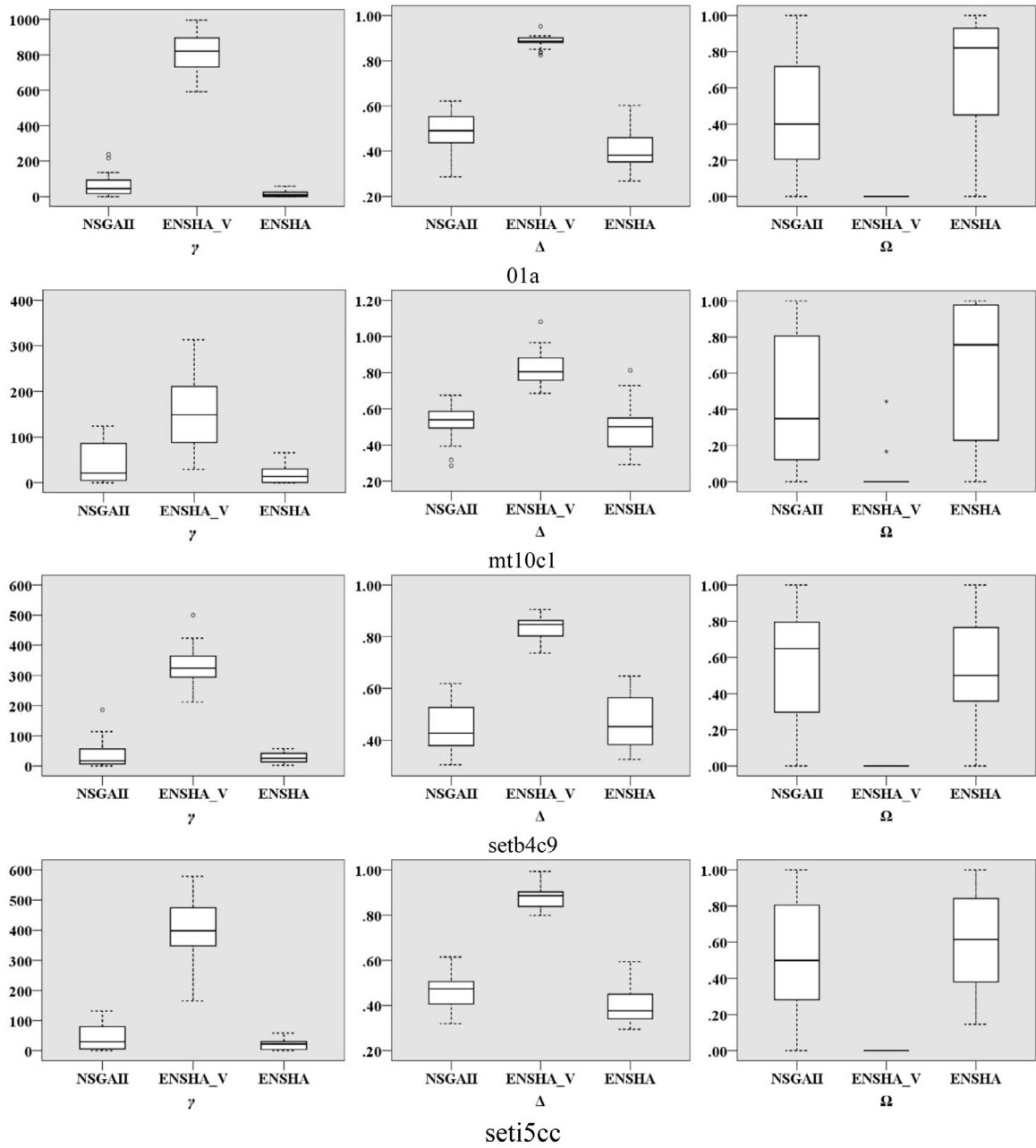


Fig. 14. Box plots of NSGAII, ENSHA_V and ENSHA.

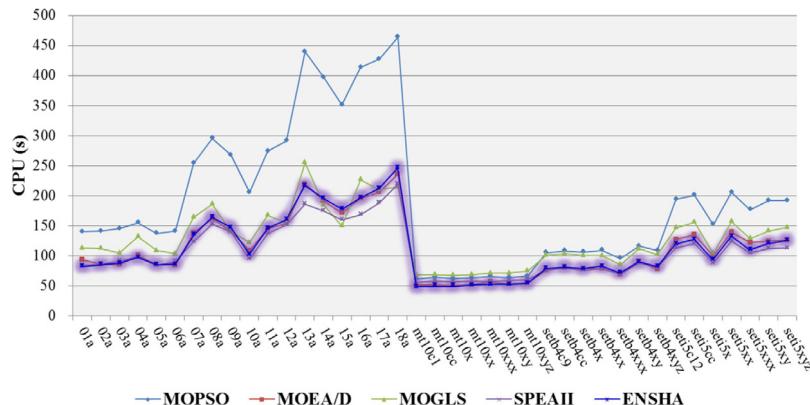


Fig. 15. Average CPU time of MOPSO, MOEA/D, MOGLS, SPEAII and ENSHA. (ENSHA is highlighted in shadow.)

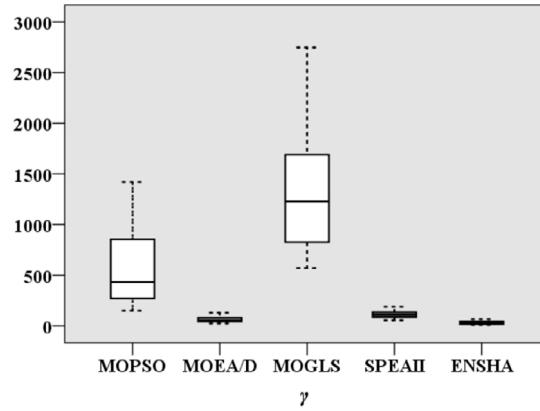


Fig. 16. Box plots of MOPSO, MOEA/D, MOGLS, SPEAII and ENSHA (mean values of γ).

4.2.1. Nondominated sorting

Let PS be the population size of MP, π_i^{MP} the i th individual of MP at generation gen , $\pi^{MP}(gen) = \{\pi_1^{MP}, \dots, \pi_{PS}^{MP}\}$ the population of MP at generation gen , n_i the number of individuals that dominate the individual π_i^{MP} , L the total number of nondominated levels of MP, R_i ($R_i \in \{1, L\}$) the nondominated level to which the i th individual is assigned, F_l ($l = 1, \dots, L$) the set of individuals that belong to the l th nondominated level, $\pi_{i,j}^{NF}(gen)$ ($j = 1, \dots, |F_l|$) the j th individual of F_l at generation gen , $n(\pi_{i,j}^{NF}(gen))$ the number of individuals in $\pi^{MP}(gen)$ that dominates individual $\pi_{i,j}^{NF}(gen)$. Then, by means of the “fast nondominated sorting method” [20], the individual $\pi_i^{MP}(gen)$ can be assigned to the corresponding nondominated level R_i according to the value of n_i . Moreover, we can obtain a set $F_1 = \{\pi_{1,1}^{NF}(gen), \pi_{1,2}^{NF}(gen), \dots, \pi_{1,|F_1|}^{NF}(gen)\}$ from MP, where the individual $\pi_{1,j}^{NF}(gen)$ ($j \in \{1, \dots, |F_1|\}$) absolutely satisfies the condition that $n(\pi_{1,j}^{NF}(gen)) = 0$.

For example, there are 14 individuals in MP with the objectives $f_i(C_{max}, TSC)$, i.e., $f_1(7.4, 9.5), f_2(5.0, 6.3), f_3(2.9, 1.2), f_4(8.0, 9.0), f_5(9.0, 8.2), f_6(1.5, 3.0), f_7(10.1, 9.9), f_8(3.5, 7.9), f_9(7.0, 5.4), f_{10}(1.0, 4.0), f_{11}(9.4, 10.9), f_{12}(0.5, 5.0), f_{13}(4.0, 6.9)$, and $f_{14}(2.0, 2.0)$. As illustrated in Fig. 7, these individuals can be divided into four nondominated levels (i.e., levels 1 to 4) via the fast nondominated sorting method. Then, the nondominated set F_1 that includes nondominated individuals $\pi_{1,1}^{NF}(gen) = \pi_3^{MP}(gen)$, $\pi_{1,2}^{NF}(gen) = \pi_{14}^{MP}(gen)$, $\pi_{1,3}^{NF}(gen) = \pi_6^{MP}(gen)$, $\pi_{1,4}^{NF}(gen) = \pi_{10}^{MP}(gen)$ and $\pi_{1,5}^{NF}(gen) = \pi_{12}^{MP}(gen)$ can be identified. Obviously, individuals of the same nondominated level are nondominated, and individuals in the higher nondominated levels ($l \geq 2$) are dominated by ones that belong to the lower nondominated levels. According to [20], the worst-case CC of the nondominated sorting method is $O(M * PS^2)$, where M is the number of objectives.

4.2.2. Crowding distance assignment

To achieve a good performance, we need to consider both the convergence and the diversity of offspring when evolving MP. Since individuals in the same nondominated level l are all nondominated, a crowding distance d_{lj} needs to be assigned to individual j . That is, if $d_{lj} > d_{lk}$ ($k \in \{1, \dots, |F_l|\}$), one can regard that the individual j is “more promising” than the individual k . As illustrated in Fig. 8, the value of d_{lj} can be calculated by using the average side length of the dashed box, and more details can be found in [20].

Based on the above definitions, the crowded-comparison operator (\prec_n) [20] is used to compare the individuals distributed in different nondominated levels for performing selection operation. In doing so, a partial order is defined as $j \prec_n k$, if and only if $(R_j < R_k)$ or $((R_j = R_k) \wedge (d_{lj} > d_{lk}))$. For instance, in Fig. 7, if the binary tournament selection operation is adopted, we prefer individual

3 to individual 8 since $R_3 = 1 < R_8 = 2$. Besides, individual 2 is better than individual 13 because the former locates in a lesser crowded box than the latter and it holds $((R_2 = R_{13}) \wedge (d_{2,2} > d_{2,13}))$. According to [20], the worst-case CC of calculating the crowding distance is $O(M * PS * \log PS)$, where M is the number of objectives.

4.3. EDA-based machine learning mechanism

The machine learning technique has been extensively regarded as an effective tool for guiding the search behavior of evolutionary algorithms [43–46]. Generally, the main methodology of machine learning follows two aspects: (1) Learning a probabilistic model to depict variable distribution from training samples; (2) Generating or predicting the testing samples from the probabilistic model. At this point, the ideas behind the machine learning are very similar to the search behavior of EDA [9,47–49], which undergoes two operations, i.e., estimating (learning) the probabilistic model and sampling (generating) new individuals from the model. Thus, an EDA-based machine learning mechanism is embedded into the framework of ENSHA to generate AP and search different promising regions. To be specific, the probabilistic model of the nondominated solutions of MP is estimated first, and then the candidate individuals of AP can be produced by sampling the probabilistic model. Finally, the AP of ENSHA can be obtained by utilizing the PMX-based method presented below.

4.3.1. Probabilistic model and estimating approach

For MOFJSSP_SDST/C, the marginal probability density of discrete random variables can be described by a set of marginal distribution laws, and in turn the corresponding marginal distribution functions could be obtained. Let $\mathbf{pr}_1(gen), \dots, \mathbf{pr}_d(gen), \dots, \mathbf{pr}_D(gen)$ denote the marginal distribution laws of the D -dimensional discrete random variables at generation gen , and then we have

$$\mathbf{pr}_d(gen) = [pr_{1d}(gen), \dots, pr_{nd}(gen)]^T, d = 1, \dots, D, \quad (6)$$

where $D = TO$, $\sum_{w=1}^n pr_{wd}(gen) = 1$, and $pr_{wd}(gen)$ is the probability of operation w appearing at the d th dimension at generation gen . Then, the marginal distribution functions can be calculated from Eq. (7).

$$F_d^{gen}(z) = \begin{cases} 0 & 0 \leq z < 1 \\ \sum_{w=1}^v pr_{wd}(gen) & v \leq z < v + 1, \text{ for } v = 1, \dots, n - 1 \\ 1 & z \geq n. \end{cases} \quad (7)$$

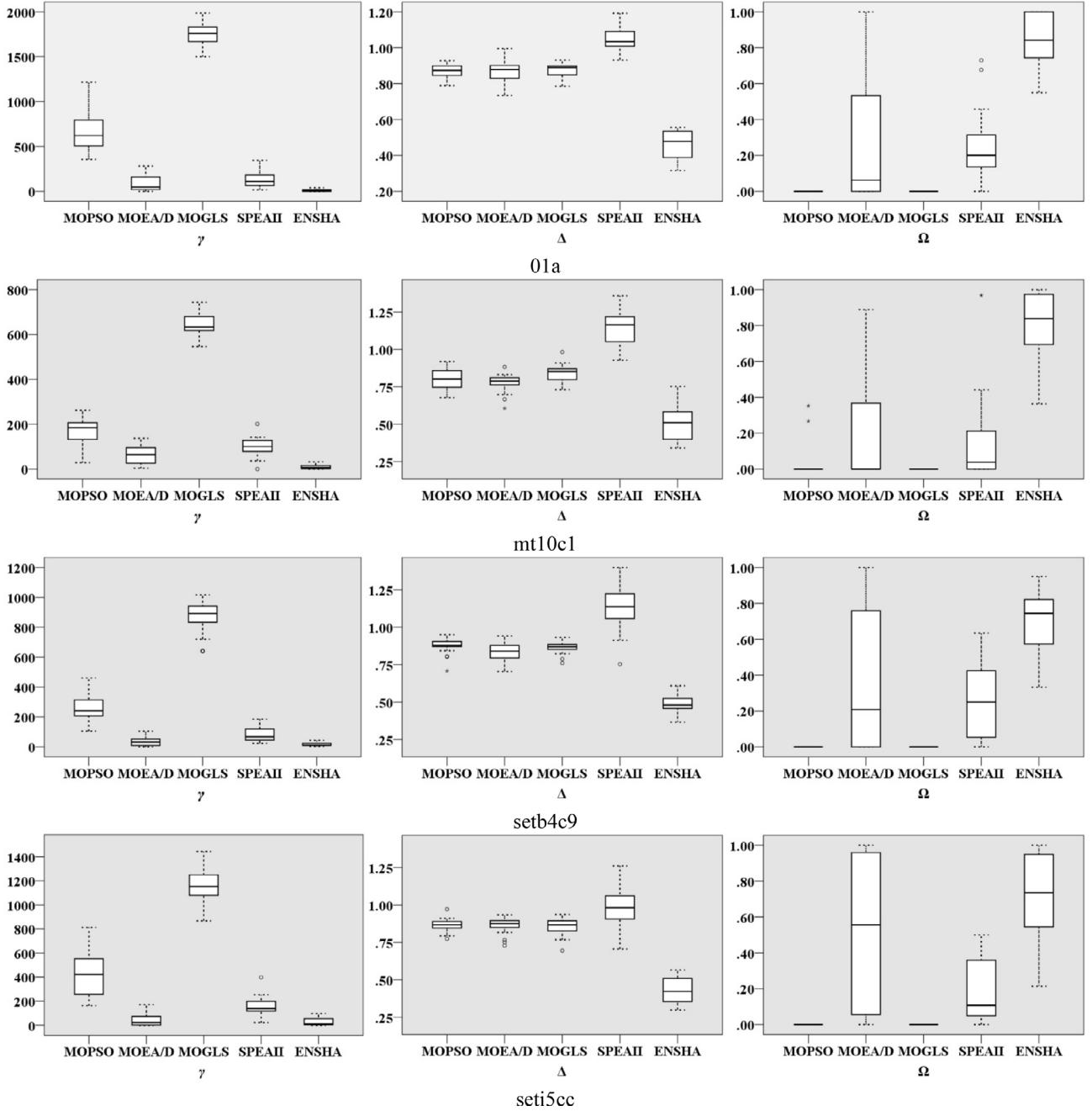


Fig. 17. Box plots of MOPSO, MOEA/D, MOGLS, SPEAII and ENSHA.

Note that the matrix $\mathbf{PM}(\text{gen})$ is used to record the marginal distribution laws of solution variables along with the iteration of ENSHA and it can be written as

$$\mathbf{PM}(\text{gen}) = \begin{pmatrix} pr_{11}(\text{gen}) & \cdots & pr_{1TO}(\text{gen}) \\ \vdots & \ddots & \vdots \\ pr_{n1}(\text{gen}) & \cdots & pr_{nTO}(\text{gen}) \end{pmatrix}_{n \times TO}. \quad (8)$$

In ENSHA, $\mathbf{PM}(\text{gen})$ accumulates useful information from excellent individuals by using the incremental learning mechanism. That is, at each generation, an individual included in \mathbf{F}_1 of MP (i.e., $\mathbf{F}_1 = \{\pi_{1,j}^{\text{NF}}(\text{gen}), \dots, \pi_{1,|F_1|}^{\text{NF}}(\text{gen})\}$) is randomly selected as the excellent individual for updating $\mathbf{PM}(\text{gen})$. Let LR denote the learning rate and $\pi_{1,j}^{\text{NF},d}(\text{gen})$ the d th element in $\pi_{1,j}^{\text{NF}}(\text{gen})$ ($j = 1, \dots, |F_1|$). Then, $\mathbf{PM}(\text{gen})$ can be estimated (updated) in Algorithm 2. When $\text{gen} = 0$, each element in $\mathbf{PM}(0)$ is set to $1/n$.

After doing so, the marginal distribution functions $F_d^{\text{gen}+1}(z)$ for $d = 1, \dots, D$ can be obtained from $\mathbf{PM}(\text{gen})$ by using Eq. (7). The CC of Algorithm 2 is $O(TO * n)$.

4.3.2. Generating candidate individuals of AP

After estimating the probabilistic model of excellent individuals of MP, the next step is to sample it for generating candidate individuals of AP. We define the quasi-inverses of marginal distribution function as

$$[F_d^{\text{gen}}(u)]^{-1} = \begin{cases} v & u \in (F_d^{\text{gen}}(v-1), F_d^{\text{gen}}(v)) \\ \min\{v \mid F_d^{\text{gen}}(v) \neq 0\} & u = 0, \\ 0 & u = 1, \dots, n \end{cases} \quad (9)$$

where $d = 1, \dots, D$, $v = 1, \dots, n$, and $D = TO$. The inverse function $[F_d^{\text{gen}}(u)]^{-1}$ utilizes roulette wheel selection scheme to

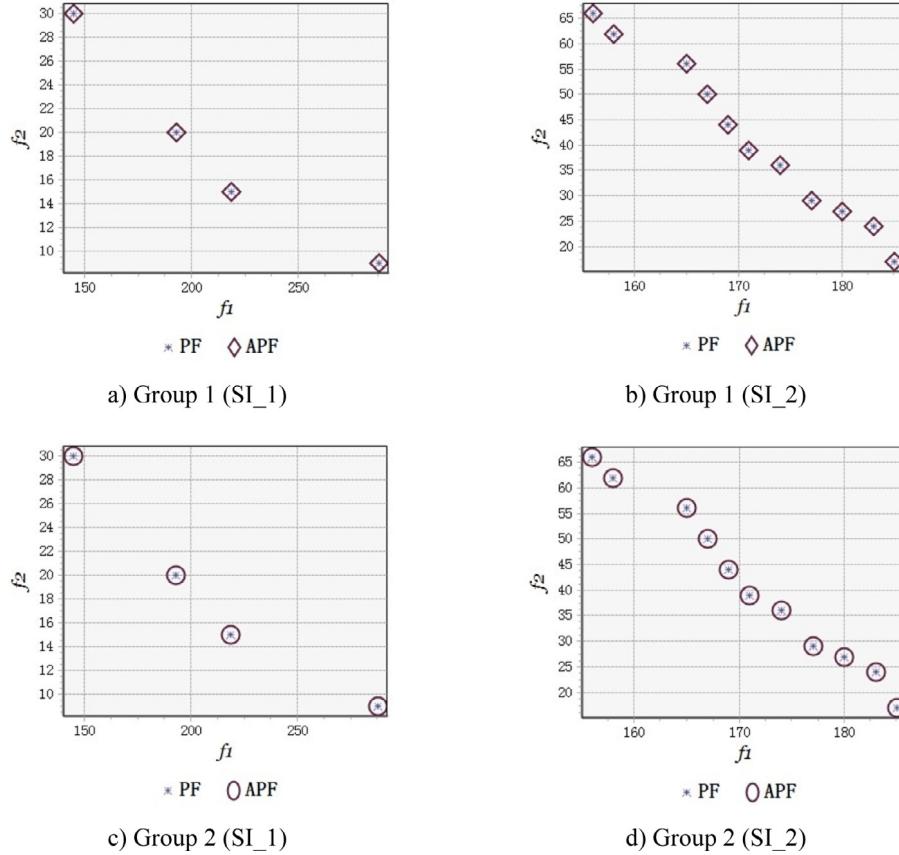


Fig. 18. Comparisons of APFs and PFs for two groups.

generate output value, in which the variable u simulates a spin of the wheel. The larger interval of $(F_d^{\text{gen}}(v-1), F_d^{\text{gen}}(v)]$ means the corresponding v can be selected and outputted with a higher probability. According to Eq. (7), the steps 3, 4 and 8 of Algorithm 2 determine the interval size of each $(F_d^{\text{gen}}(v-1), F_d^{\text{gen}}(v)]$ by the selected individual's element $\pi_{1,r}^{\text{NF},d}(\text{gen})$. Since excellent individuals are selected to update $\text{PM}(\text{gen})$, their corresponding interval sizes are enlarged during the evolutionary process. This means that candidate individuals can inherit similar structures from excellent ones.

Let $\pi^{\text{CAP}}(\text{gen}) = \{\pi_1^{\text{CAP}}(\text{gen}), \dots, \pi_i^{\text{CAP}}(\text{gen}), \dots, \pi_{PS}^{\text{CAP}}(\text{gen})\}$ be the candidate population of AP, $\pi_{i,d}^{\text{CAP}}(\text{gen})$ the d th element in $\pi_i^{\text{CAP}}(\text{gen})$, and $l(J, \pi_i^{\text{CAP}}(\text{gen}))$ the repeat times of product J in $\pi_i^{\text{CAP}}(\text{gen})$. Then, the sampling method of $\pi^{\text{CAP}}(\text{gen})$ is proposed in Algorithm 3, which generates individuals directly from $[F_d^{\text{gen}}(u)]^{-1}$. In Algorithm 3, the times that the condition $l(J, \pi_i^{\text{CAP}}(\text{gen})) = opr$ in Line 9 is met for generating a new individual, is n times. Thus, the CC of Algorithm 3 is $O(PS * n^2 * TO)$.

4.3.3. PMX-based method for generating AP

In population-based evolutionary algorithms, individuals are used to search different solution regions. Since the candidate individuals of AP totally follow the probabilistic distribution of the nondominated individuals in F_1 (see Algorithm 3), they can search the regions nearby the promising individuals or regions $\pi_{1,1}^{\text{NF}}(\text{gen}), \dots, \pi_{1,|F_1|}^{\text{NF}}(\text{gen})$. To further guide the search direction, a PMX-based method is added into ENSHA for producing AP, which executes the PMX on the individuals of $\pi^{\text{CAP}}(\text{gen})$ and F_1 to preserve partial orders within them. Let $\pi^{\text{AP}}(\text{gen}) = \{\pi_1^{\text{AP}}(\text{gen}), \dots, \pi_{PS}^{\text{AP}}(\text{gen})\}$ be the population of AP, and $\pi_i^{\text{AP}}(\text{gen})$ the i th individual in $\pi^{\text{AP}}(\text{gen})$. The PMX-based method is given in Algorithm 4. Both algorithms 3 and 4 are utilized to increase the

search ability of ENSHA. Because only Lines 2 and 3 in Algorithm 4 exist computation, the CC of Algorithm 4 is $O(PS)$.

4.4. Cooperation-based refinement strategy

When generating MP of the next generation, a cooperation-based refinement mechanism is raised to combine MP and AP for reserving more promising regions or better individuals. Denote $\pi^{\text{UP}}(\text{gen}) = \{\pi_1^{\text{UP}}(\text{gen}), \dots, \pi_{2,PS}^{\text{UP}}\}$ the union of MP and AP at generation gen , and $\pi_{l,j}^{\text{UP}}(\text{gen})$ the j th individual in the l th non-dominated level set of $\pi^{\text{UP}}(\text{gen})$. The refinement strategy can be illustrated in Algorithm 5. In Algorithm 5, the worst-case CCs of Lines 2–6, Lines 18–20, and Lines 21 are $O(M * (2PS)^2)$, $O(M * (2PS) * \log(2PS))$, and $O((2PS)^2)$, respectively. So, the worst-case CC of Algorithm 5 is $O(M * (2PS)^2)$.

4.5. Procedure of ENSHA

On the basis of Sections 4.1 to 4.4, the procedure of ENSHA can be proposed. Let $\beta^{\text{MP}}(\text{gen}) = \{\beta_1^{\text{MP}}(\text{gen}), \dots, \beta_{PS}^{\text{MP}}(\text{gen})\}$ denote the offspring (children) of $\pi^{\text{MP}}(\text{gen})$ during performing genetic operations (i.e., selection, crossover, and mutation) at generation gen , genMax the maximum generation, and ParetoSet the final set of nondominated individuals found by ENSHA. Then, the detail of ENSHA is given in Algorithm 6.

The detailed analyses of the CCs of one generation in Algorithm 6 are given in Table 5. From Algorithm 6 and Table 5, it can be concluded that the overall CC of ENSHA is $O(\text{genMax} * (PS^2 + PS * n^2 * TO + PS * m * TO))$, where the coefficients M ($M = 2$) and 2 are omitted. Obviously, the computational cost of ENSHA is not high, since the highest degree of the polynomial $\text{genMax} * (PS^2 + PS * n^2 * TO + PS * m * TO)$ is only two.

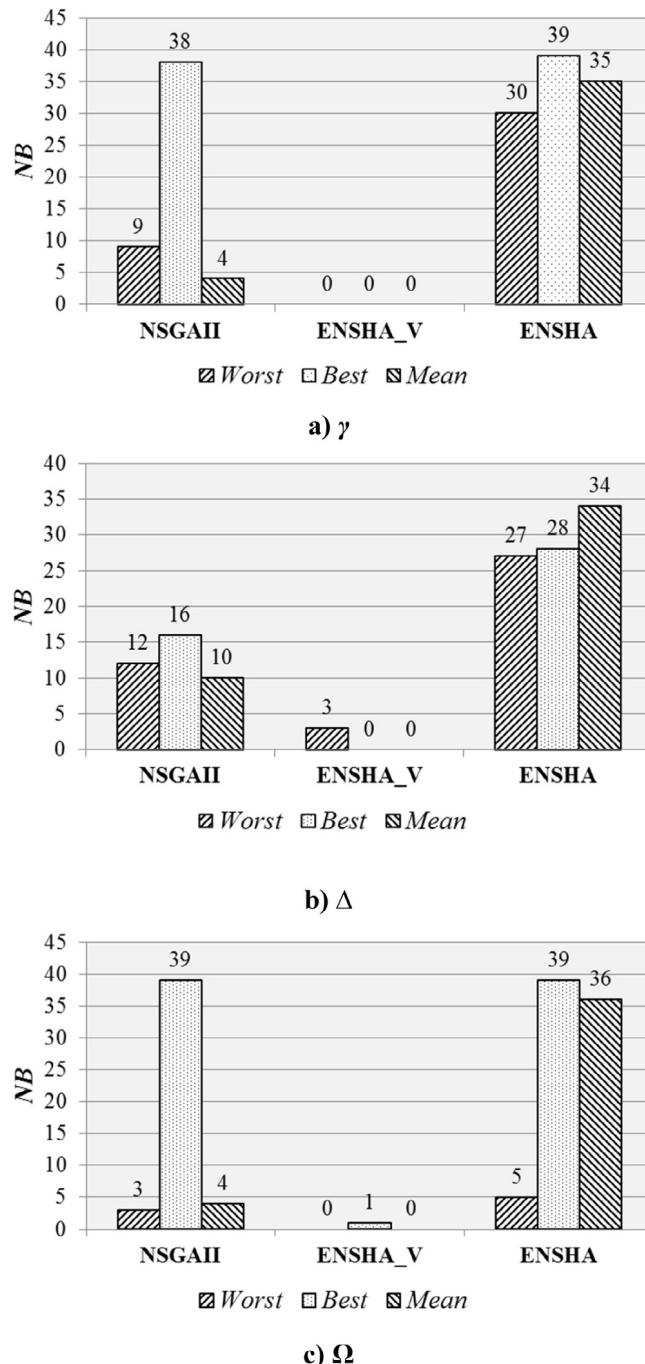


Fig. 19. Comparisons of NB associated with Group 1.

From Algorithm 6, it can be seen that ENSHA's evolutionary process is driven by the supplement and promotion of MP and AP. Through using the genetic operations with the elitist nondominated sorting scheme (Lines 17 to 38), MP can preserve the high-quality individuals in terms of both the convergence and diversity. Meanwhile, the marginal probabilistic model and the PMX-based method are applied to generate AP (Lines 13 to 16) for exploring different promising individuals nearby the nondominated individuals found so far. Furthermore, the cooperation-based refinement strategy (Lines 7–9) is used to preserve the excellent individuals in both MP and AP for generating the next generation of MP, which makes MP and AP possible to supplement and promote each other during the search process.

5. Experimental results and comparisons

5.1. Experimental design

Due to the lack of benchmark data for MOFJSSP_SDST/C, a number of instances based on well-known benchmark data for FJSSP are used to test the performance of ENSHA. In this study, we adopt two data sets of FJSPLIB¹ widely used in previous literatures, i.e., Dauzère-Pérès and Paulli [50] and Chambers and Barnes [51], and randomly generate SDSTs and SDSCs from a uniform distribution [1,50]. The original data sets of Dauzère-Pérès

¹ The original resource of these data sets is available at <http://people.idsia.ch/monaldo/fjsresults/TextData.zip>.

Algorithm 1: Decoding a Solution

Input: assignment rule, solution π
Output: makespan, TSC

```

1: For  $k := 1$  to  $TO$  do
2:   For  $j := 1$  to  $m$  do
3:     If ( $\pi_k$  satisfies processing constraints on machine  $j$ ) then
4:       Calculate  $CT$  and  $SC$  for  $\pi_k$  on machine  $j$ ;
5:     End If
6:   End For  $j$ 
7:   If (there is a tie for assignment rule) then
8:     Arbitrarily select one machine  $j^*$  from  $AVA(\pi_k)$ ;
9:   Else
10:    Select one machine  $j^*$  from  $AVA(\pi_k)$  by using adopted assignment rule;
11:   End If Else
12:   Assign  $\pi_k$  to the selected machine  $j^*$ ;
13: End For  $k$ 
14: Calculate makespan and TSC based on the model mentioned in Section 2;
15: Return makespan and TSC

```

Algorithm 2: Estimate Marginal Probability Matrix

Input: $\pi_{1,j}^{NF}(gen)$ ($j = 1, \dots, |F_1|$), LR , $PM(gen-1)$
Output: $PM(gen)$

```

1: Set  $r := random[1, |F_1|]$ ; //randomly select a nondominated individual
2: For  $d := 1$  to  $D$  do
3:   Set  $x := \pi_{1,r}^{NF,d}(gen)$ ;
4:   Set  $pr_{xd}(gen) := pr_{xd}(gen-1) + LR$ ; //accumulate information with the learning rate
5: End for  $d$ 
6: For  $d := 1$  to  $D$  do
7:   For  $w := 1$  to  $n$  do
8:     Set  $pr_{wd}(gen) := pr_{wd}(gen) / (1 + LR)$ ; //normalize the probability values
9:   End for  $w$ 
10: End for  $d$ 
11: Return  $PM(gen)$ 

```

and Barnes contain 18 instances (01a to 18a) and 21 instances (mt10c1 to seti5xyz), respectively, so that there are a total of 39 instances² used in this paper. In all results, *Worst*, *Best*, *Mean*, and *SD* denote the worst value, best value, average value, and standard derivation of relevant performance metrics, respectively. And *NB* denotes the total number of the best statistic values (i.e., *Worst*, *Best* and *Mean*) of each performance metric obtained by the related algorithms. The experiments and comparisons are performed using fixed numbers of function evaluations (FES) for each instance for all algorithms. All algorithms are coded in Delphi7.0 and executed on Intel Core i7-6700 3.40 GHz processor with 16GB memory.

5.2. Performance metrics

For multi-objective optimization problems, decision makers desire to find a satisfied solution from a number of candidate ones. According to [20,52,53], basically two main aspects should

be considered to evaluate the obtained nondominated solutions: (1) The distance of the obtained nondominated solutions to the true Pareto front (PF) or approximate Pareto front (APF); (2) The diversity and distribution of the obtained nondominated solutions. Some performance metrics were proposed in existing literature. Although the names of these metrics are different, they can be basically divided into the L1-norm-based metrics and the L2-norm-based metrics. Each L1-norm-based metric has a L1-norm item (i.e., $\|x\|_1 = \sum |x_i|$) in its numerator, and similarly each L2-norm-based metric has a L2-norm item (i.e., $\|x\|_2 = \sqrt{\sum |x_i|^2}$) in its numerator. The frequently used L1-norm-based metrics are the distance metric γ (also called the convergence metric γ) and the diversity metric Δ [20,54,55]. Meanwhile, the frequently used L2-norm-based metrics contain the generational distance (GD) metric [56], the inverted generational distance (IGD) metric [57,58], and the diversity or spacing (SP) metric [57,59]. Both those L1-norm-based metrics and L2-norm-based metrics can better evaluate the obtained nondominated solutions in most situations. Here, two L1-norm-based metrics (i.e., γ and Δ) are adopted to evaluate the performance of each compared algorithm. Moreover, since MOEAs for multi-objective scheduling

² These 39 instances used in this study can be downloaded at <http://dx.doi.org/10.17632/vtnt2kcc7s.2>.

Algorithm 3: Generate Candidate Individuals of AP

Input: $\mathbf{PM}(gen)$

Output: $\pi^{CAP}(gen)$

- 1: Set $\Psi(gen) := \mathbf{PM}(gen)$; //save the marginal probability matrix
- 2: **For** $i := 1$ to PS **do**
- 3: Set $\mathbf{PM}(gen) := \Psi(gen)$; //ensure all individuals are sampled from $\mathbf{PM}(gen)$
- 4: **For** $d := 1$ to D **do**
- 5: Calculate $F_d^{gen}(z)$ by using Eq. (7);
- 6: Set $u := random[0,1]$; //generate a random value from [0,1]
- 7: Set $J := [F_d^{gen}(u)]^{-1}$; //select a job by using Eq. (9)
- 8: Set $\pi_{id}^{CAP}(gen) := J$; //assign a sampling job
- 9: **If** $l(J, \pi_i^{CAP}(gen)) = opr_j$ **then**
- 10: **For** $w := d+1$ to D **do**
- 11: Set $pr_{Jw}(gen) := 0$;
- 12: **For** $k := 1$ to n **do**
- 13: Set $pr_{kw}(gen) := pr_{kw}(gen) / (1 - pr_{Jw}(gen))$;
- 14: **End For** k
- 15: **End For** w
- 16: **End If**
- 17: **End For** d
- 18: **End For** i
- 19: Set $\mathbf{PM}(gen) := \Psi(gen)$; //recover the marginal probability matrix
- 20: **Return** $\pi^{CAP}(gen)$

Algorithm 4: PMX-based Method for Generating AP

Input: $\pi^{CAP}(gen), \pi_{i,j}^{NF}(gen)$ ($j = 1, \dots, |\mathcal{F}_1|$)

Output: $\pi^{AP}(gen)$

- 1: **For** $i := 1$ to PS **do**
- 2: Set $r_1 := random[1, |\mathcal{F}_1|]$; //randomly select a nondominated individual
- 3: Set $r_2 := random[1, PS]$; //randomly select a candidate individual of AP
- 4: Set $\pi^{p1} := \pi_{i,r1}^{NF}(gen)$ and $\pi^{p2} := \pi_{i,r2}^{CAP}(gen)$; //randomly select the parents
- 5: Perform the PMX crossover on π^{p1} and π^{p2} ; //subsection 4.1
- 6: Set $\pi_i^{AP}(gen)$ to the child of π^{p1} and π^{p2} ;
- 7: **End For** i
- 8: **Return** $\pi^{AP}(gen)$

problems often only get several nondominated solutions, the number of them also needs to be considered in the comparisons. Thus, a dominance metric (i.e., Ω) is added for performance evaluation.

Note however, that the PF of MOFJSSP_SDST/C is not known and it might be intractable when calculating the performance metrics of compared algorithms. Same as previous studies [54, 55], we first combine the sets of nondominated solutions of K algorithms (or parameter combinations, or assignment rules) S_1, \dots, S_K together. And then the nondominated solutions are identified from $S_1 \cup \dots \cup S_K$ to construct an APF. Afterwards, the performance metrics of algorithms are calculated according to APF. Nevertheless, how to guarantee that APF is a good approximation of PF is still short of common conclusions. Hence, we perform further discussions on the approximations of APF and PF in Section 5.7, on the basis of two small instances for which the PFs can be determined exactly.

5.2.1. Convergence metric (γ)

Suppose that nondominated solution sets S_1, \dots, S_K are obtained by K algorithms (or parameter combinations, or assignment rules). Since the Pareto front of MOFJSSP_SDST/C is not known, we combine the sets S_1, \dots, S_K together and identify the nondominated solutions from $S_1 \cup \dots \cup S_K$ to approximate the Pareto front S^* . Let $d_{xy}(S_k)$ denote the shortest Euclidean distance from a solution x of S_k to the nearest solution y of S^* , and $|S_k|$ the number of solutions in S_k . Then, the convergence metric γ_k can be defined in Eq. (10). Obviously, a smaller γ_k corresponds to a better approximation to the Pareto front.

$$\gamma_k = \sum_{x=1}^{|S_k|} d_{xy}(S_k) / |S_k|, k = 1, \dots, K. \quad (10)$$

Though the values of γ_k and the L2-norm-based metric GD_k ($GD_k = \sqrt{\sum_{x=1}^{|S_k|} (d_{xy}(S_k))^2 / |S_k|}$) are different, the same ordinal

Algorithm 5: Cooperation-based Refinement Strategy

Input: $\pi^{AP}(gen-1)$, $\pi^{MP}(gen-1)$

Output: $\pi^{MP}(gen)$

- 1: Set $\pi^{UP}(gen) := \pi^{MP}(gen-1) \cup \pi^{AP}(gen-1)$; //construct the union of MP and AP
- 2: **For** $i := 1$ to $2 \cdot PS$ **do**
- 3: Calculate n_i for $\pi_i^{UP}(gen)$; //subsection 4.2
- 4: Assign $\pi_i^{UP}(gen)$ to its nondominated level R_i ; //subsection 4.2
- 5: **End For** i
- 6: Obtain L nondominated sets $F_l = \{\pi_{l,1}^{NF}(gen), \dots, \pi_{l,j}^{NF}(gen), \dots, \pi_{l,|F_l|}^{NF}(gen)\}$ ($l=1, \dots, L$) of $\pi^{UP}(gen)$ via the nondominated sorting method; //subsection 4.2
- 7: Set $w := 1$;
- 8: **For** $l := 1$ to L **do**
- 9: **If** $w > PS$ **then**
- 10: **Break**;
- 11: **End If**
- 12: **If** $w + |F_l| \leq PS + 1$ **then**
- 13: **For** $j := 1$ to $|F_l|$ **do**
- 14: $\pi_w^{MP}(gen) := \pi_{l,j}^{NF}(gen)$;
- 15: $w := w + 1$;
- 16: **End For** j
- 17: **Else**
- 18: **For** $j := 1$ to $|F_l|$ **do**
- 19: Calculate the crowding distance d_{lj} for $\pi_{l,j}^{NF}(gen)$;
- 20: **End For** j
- 21: Rank individuals in F_l in descending order according to crowding distances and obtain the ranked $F_l = \pi_{l,j_1}^{NF}(gen), \dots, \pi_{l,j_v}^{NF}(gen), \dots, \pi_{l,j_{|F_l|}}^{NF}(gen)$; //subsection 4.2
- 22: **Repeat**
- 23: Set $v := 1$;
- 24: $\pi_w^{MP}(gen) := \pi_{l,j_v}^{NF}(gen)$;
- 25: $w := w + 1$ and $v := v + 1$;
- 26: **Until** ($w = PS + 1$)
- 27: **End If Else**
- 28: **End For** l
- 29: **Return** $\pi^{MP}(gen)$

relationship among the compared sets S_1, \dots, S_K can be obtained by using these two metrics. For example, if the ordinal relationship $\gamma_3 < \gamma_1 < \gamma_2$ is obtained, the same ordinal relationship $GD_3 < GD_1 < GD_2$ also can be obtained. This means that there is no essential difference between γ_k and GD_k .

5.2.2. Diversity metric (Δ)

Let $d_{\xi k}$ denote the Euclidean distance between the highest solution of S_k and it of S^* on the vertical axis, $d_{\eta k}$ the Euclidean distance between the furthest solution of S_k and it of S^* on the horizontal axis, $d_{\lambda k}$ the Euclidean distance between the two consecutive solutions λ and $\lambda + 1$ of S_k , \bar{d}_k the average Euclidean distance among all solutions of S_k . The diversity metric is given in Eq. (11). It is clear that the smaller the Δ_k is, the better the diversity is.

$$\Delta_k = \frac{d_{\xi k} + d_{\eta k} + \sum_{\lambda=1}^{|S_k|-1} |d_{\lambda k} - \bar{d}_k|}{d_{\xi k} + d_{\eta k} + (|S_k| - 1)\bar{d}_k}, k = 1, \dots, K. \quad (11)$$

Obviously, the revised $\Delta_k^{\text{revised}}$ ($\Delta_k^{\text{revised}} = \sum_{\lambda=1}^{|S_k|-1} |d_{\lambda k} - \bar{d}_k| / (|S_k| - 1)d_k$) and the L2-norm-based metric SP_k ($SP_k = \sqrt{\sum_{\lambda=1}^{|S_k|-1} |d_{\lambda k} - \bar{d}_k|^2} / (|S_k| - 2)$) can obtain the same ordinal relationship among the compared sets S_1, \dots, S_K . By adding $d_{\xi k}$ and $d_{\eta k}$

$d_{\eta k}, \Delta_k$ can evaluate not only the diversity but also the spread of S_k . Therefore, Δ_k is a more suitable metric.

5.2.3. Dominance metric (Ω)

A dominance metric is concerned with the number of solutions in S_k that are not dominated by any solutions of S^* . This metric is written as

$$\Omega_k = |S_k \setminus \{x \in S_k : \exists y \in S^* : y \prec x\}| / |S_k|, \quad (12)$$

where $y \prec x$ denotes the solution x is dominated by solution y . Apparently, the larger the value of Ω_k is, the better the set S_k is.

5.3. Effectiveness of proposed job assignment rules

To test the effectiveness of the proposed job assignment rules (i.e., CT+SC, CT*SC and DOMINANCE), we compare them with 4 well-known assignment rules reported in existing literatures, including RANDOM [33], SPT [33–35], EST [34,35] and ECT [34,35]. Since the parameters of ENSHA have not been tuned before this comparison, we here adopt the mean values of all candidate levels of each parameter listed in Table 6. The parameters are as follows: $PS = 125$, $P_c = 0.50$, $P_m = 0.375$ and $LR = 0.04$. ENSHAs with different job assignment rules are independently run 21

Algorithm 6: ENSHA for MOFJSSP_SDST/C

Input: $genMax$, PS , Pc , Pm , LR

Output: $ParetoSet$

```

1: For  $gen := 0$  to  $genMax + 1$  do
2:   If  $gen = 0$  then
3:     Randomly initialize  $\pi^{MP}(gen=0)$  and  $\pi^{AP}(gen=0)$ ;
4:     Evaluate the individuals of  $\pi^{MP}(gen=0)$  and  $\pi^{AP}(gen=0)$ ; //Algorithm 1
5:     Initialize the marginal probability matrix  $PM(gen=0)$  where
 $pr_{wd}(gen=0) = 1/n$ ,  $d = 1, \dots, TO$  and  $w = 1, \dots, n$ ;
6:   Else
7:     /* cooperation-based refinement strategy */
8:     Perform cooperation-based refinement and obtain  $\pi^{MP}(gen)$ ; //Algorithm 5
9:     Identify  $\pi_{1,j}^{NF}(gen)$  of  $\pi^{MP}(gen)$ ,  $j = 1, \dots, |F_1|$ ; //Section 4.2
10:    If  $gen = genMax + 1$  then Break the main loop;
11:    End If
12:    /* EDA-based machine learning mechanism */
13:    Estimate the marginal probability matrix  $PM(gen)$ ; //Algorithm 2
14:    Generate the candidate population  $\pi^{CAP}(gen)$ ; //Algorithm 3
15:    Perform the PMX-based method to  $\pi^{CAP}(gen)$  and  $\pi_{1,j}^{NF}(gen)$  for
generating  $\pi^{AP}(gen)$ ; //Algorithm 4
16:    Evaluate the individuals of  $\pi^{AP}(gen)$ ; //Algorithm 1
17:    /* genetic operations for MP */
18:    Calculate the crowding distance of each individual in  $\pi^{MP}(gen)$ ;
19:    For  $i := 1$  to  $PS$  do //binary tournament selection
20:      Set  $r_1, r_2 := random[1, PS]$  where  $r_1 \neq r_2 \neq i$ ;
21:      If  $\pi_{r1}^{MP}(gen) \prec_n \pi_{r2}^{MP}(gen)$  then //Section 4.2
22:        Set  $\beta_i^{MP}(gen) := \pi_{r1}^{MP}(gen)$ ; Else Set  $\beta_i^{MP}(gen) := \pi_{r2}^{MP}(gen)$ ;
23:      End If Else
24:    End For  $i$ 
25:    For  $i := 1$  to  $PS$  do //PMX-based crossover
26:      If  $random[0,1] \leq Pc$  then
27:        Apply crossover to  $\beta_i^{MP}(gen)$  for producing  $\beta_i^{CMP}(gen)$ ; //Section 4.1
28:        Evaluate the individual  $\beta_i^{CMP}(gen)$ ; //Algorithm 1
29:        Set  $\beta_i^{MP}(gen) := \beta_i^{CMP}(gen)$ ;
30:      End If
31:    End For  $i$ 
32:    For  $i := 1$  to  $PS$  do //Interchange-based mutation
33:      If  $random[0,1] \leq Pm$  then
34:        Apply mutation to  $\beta_i^{MP}(gen)$ ; //Section 4.1
35:        Evaluate the individual  $\beta_i^{MP}(gen)$ ; //Algorithm 1
36:      End If
37:    End For  $i$ 
38:    Set  $\pi^{MP}(gen) := \beta^{MP}(gen)$ ; //update MP
39:  End If Else
40: End For  $i$ 
41: Return  $ParetoSet := \{\pi_{1,1}^{NF}(genMax+1), \dots, \pi_{1,|F_1|}^{NF}(genMax+1)\}$ 

```

times and the maximum FES (*MaxFES*) is adopted as the stopping condition. The *MaxFES* is set to 500,000. In terms of 39 instances, the histograms considering the mean values of dominance metric are shown in Fig. 9.³

From Fig. 9, it is clear that the proposed *CT + SC*, *CT*SC* and *DOMINANCE* are able to gain the top 3 performances for almost all instances. These results indicate that the proposed job assignment rules can provide relatively effective local optimization for individuals during the search process of ENSHA. This is because,

in our proposed job assignment rules, we not only consider *CT* but also *SC* which may directly impact the quality of obtained individuals. That is, our job assignment rules can find relatively better tradeoff between the multiple objectives, and, in turn, good performance can be achieved. Thus, we should consider *CT* and *SC* simultaneously for MOFJSSP_SDST/C, since conventional assignment rules (such as RANDOM, SPT, ECT and EST) concentrating on single objectives may lead to poor performances. Indeed, the design of the problem-dependent job assignment rule is important for production scheduling problem, which would be significant for the overall performance of the underlying solution methods. For multi-objective production scheduling problems with both the efficiency and economic criteria, sub-indicators

³ Details of the comparisons associated with the convergence, diversity and dominance metric can be downloaded at <http://dx.doi.org/10.17632/hzv2d8bpvb.2>.

Table 5
Detailed analyses of the CCs of one generation in Algorithm 6.

| Line | CC | Analysis |
|-------------|--|--|
| Line 2 | $O(0)$ | If $gen = 0$ |
| Lines 3–5 | $O(PS * 2 * m * TO)$ | Each initial individual of MP is evaluated by calculating the objectives. According to Algorithm 1 in Section 3.2, the CC of evaluating one individual is $O(2 * m * TO)$. Therefore, the CC of evaluating MP is $O(PS * 2 * m * TO)$. |
| Line 6 | $O(0)$ | Else |
| Lines 7–11 | $O(M * (2PS)^2)$ | Since Line 8 has already obtained the nondominated levels of the combined MP and AP by using Algorithm 5 in Section 4.4, Line 9 does not need any computation. Thus, the CC of Lines 7–11 is $O(M * (2PS)^2)$. |
| Lines 12–16 | $O(PS * n^2 * TO + PS * 2 * m * TO)$ | The CCs of Algorithms 1–4 are $O(2 * m * TO)$, $O(TO * n)$, $O(PS * n^2 * TO)$, and $O(PS)$, respectively, and the CC of evaluating $\pi^{AP}(gen)$ by using Algorithm 1 is $O(PS * 2 * m * TO)$. Thus, the CC of Lines 12–16 is $O(PS * n^2 * TO + PS * 2 * m * TO)$. |
| Lines 17–24 | $O(M * PS * \log PS)$ | Based on Section 4.2.2, the CC of Line 18 is $O(M * PS * \log PS)$. The CC of Lines 19–24 is $O(PS)$. Thus, the CC of Lines 17–24 is $O(M * PS * \log PS)$. |
| Lines 25–31 | $O(Pc * PS * 2 * m * TO)$ | Because the CC of Lines 26–30 is $O(Pc * 2 * m * TO)$, the CC of Lines 25–31 is $O(Pc * PS * 2 * m * TO)$. |
| Lines 32–37 | $O(Pm * PS * 2 * m * TO)$ | Since the CC of Lines 33–36 is $O(Pm * 2 * m * TO)$, the CC of Lines 32–37 is $O(Pm * PS * 2 * m * TO)$. |
| Line 38 | $O(0)$ | Set $\pi^{MP}(gen) := \beta^{MP}(gen)$; //update MP |
| Line 39 | $O(0)$ | End If |
| Lines 2–39 | $O(M * (2PS)^2 + PS * n^2 * TO + PS * 2 * m * TO)$ | Because the CC of Lines 7–11 is larger than that of Lines 17–24 and the CC of Lines 12–16 is larger than those of Lines 3–5, Lines 25–31, and Lines 32–37, the CC of one generation is $O(M * (2PS)^2 + PS * n^2 * TO + PS * 2 * m * TO)$. |

CT and SC may inherently determine the basic structures of job assignment rules. Therefore, the philosophy of the presented job assignment rules may be also useful for the other kinds of multi-objective scheduling problems. However, this research issue obviously beyond the scope of this paper.

It should be additionally note that CT^*SC outperforms, related to 27 instances, $CT + SC$ and DOMINANCE as shown in Fig. 9. Say that CT^*SC tends to get more reasonable tradeoff between CT and SC . Relevant observations on the 95% confidence interval (CI) of Ω are put forward in Fig. 10, in which the overall trend of the mean value of Ω is illustrated as well. It is clear that CT^*SC can obtain more effective and reliable solutions than the other two rules, in terms of both the 95% CI and the overall trend. Hence, this paper will adopt CT^*SC as the job assignment rule and embed it into the global search framework of ENSHA. We also execute ENSHA (with different rules) by using different parameter combinations; nevertheless, the conclusions are very similar to ones mentioned above.

5.4. Parameter settings

To find a suitable parameter combination for ENSHA, three performance metrics (i.e., γ , Δ and Ω) should be considered simultaneously because they are all related to ENSHA's performance. In this paper, the Taguchi method is used for parameter tuning [60–63], in which the response is the sum of normalized values of performance metrics and the factors are the parameters of ENSHA. In the Taguchi method, we consider two kinds of

factors (i.e., controllable factors S and noise factors N), and the ultimate goal is to control N and reduce the variation around the response associated with orthogonal arrays. Note that the design that is impressed less by N is selected as the robust design. As suggested by previous works [60–63], the signal to noise ratio (S/N) is used for performing the analysis on experimental results.

In the implementation of the Taguchi method, we carry out the design-of experiment (DOE) [64] based on the instance “12a”. To be specific, 4 factor levels are considered for parameters PS , Pc , Pm and LR on the performance of ENSHA, as shown in Table 6. The orthogonal array $L_{16}(4^4)$ with a total of 16 parameter combinations is adopted, which is generated by IBM SPSS Statistics 20 (the seed value is set to 1), as listed in Table 7. For all combinations, ENSHA is independently run 21 times with the stopping condition $MaxFES = 500,000$. After running ENSHA one time, the corresponding metrics γ , Δ and Ω of each combination is calculated and recorded. The obtained metrics of each combination in the first run are listed in Table 7.

As mentioned earlier, parameter combinations having the larger Ω yet the smaller γ and Δ indicate better solution quality. However, there may be a large difference among the values of three metrics, so we need to normalize them after each run. To be specific, the best result of certain metrics is assigned to 1 while the other results are assigned among [0, 1]. It should be noted that some values of metrics may be equal to 0. Therefore, we set $\gamma_k = \gamma_k + \delta_1$, $\Delta_k = \Delta_k + \delta_2$ and $\Omega_k = \Omega_k + \delta_3$ ($k = 1, \dots, 16$, $\delta_1 = \delta_2 = \delta_3 = 0.001$). We normalize the values of the performance metrics of Table 7 in Table 8, where $Sum_{k=1}$

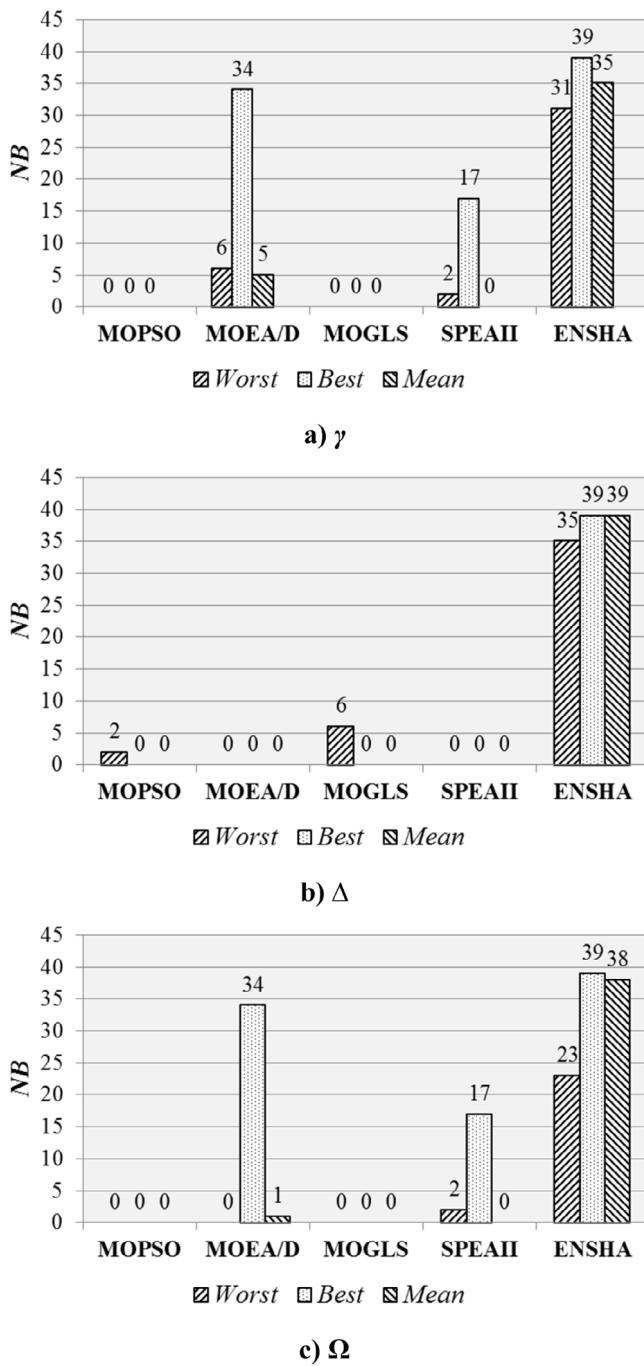


Fig. 20. Comparisons of NB associated with Group 2.

Table 6
Different combinations of parameters.

| Parameter | Factor level | | | |
|-----------|--------------|------|------|------|
| | 1 | 2 | 3 | 4 |
| PS | 50 | 100 | 150 | 200 |
| Pc | 0.20 | 0.40 | 0.60 | 0.80 |
| Pm | 0.15 | 0.30 | 0.45 | 0.60 |
| LR | 0.01 | 0.03 | 0.05 | 0.07 |

is the sum of normalized metrics in the first run. In terms of 21 runs, the sums of normalized metrics Sum_k ($k = 1, \dots, 21$) for each parameter combination are given in Table 9.

Table 7
Orthogonal array and obtained values of metrics (the first run).

| No. | Combination | | | | Metric | | |
|-----|-------------|----|----|----|----------|----------|----------|
| | PS | Pc | Pm | LR | γ | Δ | Ω |
| 1 | 2 | 2 | 1 | 2 | 164.993 | 0.838 | 0.000 |
| 2 | 1 | 2 | 4 | 4 | 98.077 | 0.920 | 0.000 |
| 3 | 3 | 2 | 3 | 1 | 471.885 | 0.870 | 0.000 |
| 4 | 4 | 3 | 4 | 1 | 203.686 | 0.880 | 0.000 |
| 5 | 3 | 1 | 2 | 4 | 13.562 | 0.759 | 0.567 |
| 6 | 3 | 4 | 4 | 2 | 194.374 | 0.798 | 0.000 |
| 7 | 4 | 1 | 3 | 2 | 249.858 | 0.912 | 0.000 |
| 8 | 3 | 3 | 1 | 3 | 332.053 | 1.017 | 0.000 |
| 9 | 1 | 1 | 1 | 1 | 41.326 | 0.891 | 0.308 |
| 10 | 4 | 2 | 2 | 3 | 88.223 | 0.915 | 0.000 |
| 11 | 1 | 4 | 3 | 3 | 233.476 | 0.856 | 0.000 |
| 12 | 4 | 4 | 1 | 4 | 384.140 | 0.902 | 0.000 |
| 13 | 1 | 3 | 2 | 2 | 178.287 | 0.837 | 0.000 |
| 14 | 2 | 3 | 3 | 4 | 6.483 | 0.897 | 0.708 |
| 15 | 2 | 1 | 4 | 3 | 57.577 | 0.794 | 0.421 |
| 16 | 2 | 4 | 2 | 1 | 34.487 | 0.916 | 0.133 |

Table 8
Normalized values of metrics (the first run).

| No. | Combination | | | | Normalized metric | | | $Sum_{k=1}$ |
|-----|-------------|----|----|----|-------------------|----------|----------|-------------|
| | PS | Pc | Pm | LR | γ | Δ | Ω | |
| 1 | 2 | 2 | 1 | 2 | 0.0393 | 0.9058 | 0.0014 | 0.947 |
| 2 | 1 | 2 | 4 | 4 | 0.0661 | 0.8252 | 0.0014 | 0.893 |
| 3 | 3 | 2 | 3 | 1 | 0.0137 | 0.8726 | 0.0014 | 0.888 |
| 4 | 4 | 3 | 4 | 1 | 0.0318 | 0.8627 | 0.0014 | 0.896 |
| 5 | 3 | 1 | 2 | 4 | 0.4781 | 1.0000 | 0.8011 | 2.279 |
| 6 | 3 | 4 | 4 | 2 | 0.0334 | 0.9512 | 0.0014 | 0.986 |
| 7 | 4 | 1 | 3 | 2 | 0.0260 | 0.8324 | 0.0014 | 0.860 |
| 8 | 3 | 3 | 1 | 3 | 0.0195 | 0.7466 | 0.0014 | 0.767 |
| 9 | 1 | 1 | 1 | 1 | 0.1569 | 0.8520 | 0.4358 | 1.445 |
| 10 | 4 | 2 | 2 | 3 | 0.0735 | 0.8297 | 0.0014 | 0.905 |
| 11 | 1 | 4 | 3 | 3 | 0.0278 | 0.8868 | 0.0014 | 0.916 |
| 12 | 4 | 4 | 1 | 4 | 0.0169 | 0.8416 | 0.0014 | 0.860 |
| 13 | 1 | 3 | 2 | 2 | 0.0364 | 0.9069 | 0.0014 | 0.945 |
| 14 | 2 | 3 | 3 | 4 | 1.0000 | 0.8463 | 1.0000 | 2.846 |
| 15 | 2 | 1 | 4 | 3 | 0.1126 | 0.9560 | 0.5952 | 1.664 |
| 16 | 2 | 4 | 2 | 1 | 0.1880 | 0.8288 | 0.1890 | 1.206 |

The aim of the Taguchi method is to find the maximum S/N . Based on the sum of normalized metrics Sum_k ($k = 1, \dots, 21$) [60, 63], S/N can be calculated as

$$S/N = -10 \log \left(\frac{1}{n} \sum_{i=1}^n Sum_i^2 \right), \quad (13)$$

where $n = 21$ is the number of replications of ENSHA. The values of S/N for all combinations are given in Table 9. Obviously, the larger the value of S/N is, the better the parameter combination is. Based on Table 9, the level trends of parameters are given in Fig. 11. Thereby, ENSHA's parameters are as follows: $PS = 50$, $Pc = 0.60$, $Pm = 0.45$ and $LR = 0.07$.

5.5. Comparisons with NSGAII and variant algorithms

For the purpose of investigating the effectiveness of ENSHA's components, we compare ENSHA with its variant (denoted as ENSHA_V). ENSHA_V is the same to ENSHA except that the genetic operators and the cooperation-based refinement strategy are not used. In ENSHA_V, we introduce an external archive to record the nondominated solutions during the search process, and the EDA-based machine learning mechanism is performed through such an external archive. That is to say, in ENSHA_V, the marginal probability matrix is estimated by using a randomly selected solution from the current external archive. Furthermore, the comparisons of ENSHA and NSGAII [20] are carried out to verify the effectiveness of the embedded EDA-based machine

Table 9

Sum of normalized metrics for each parameter combination.

| No. | Combination | Sum of normalized metrics combination | | | | | | | | | | | | | | | | | | | | S/N | | |
|-----|-------------|---------------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|----|
| | | PS | Pc | Pm | LR | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 1 | 2 2 1 2 | 0.947 | 0.686 | 0.749 | 0.721 | 0.865 | 0.576 | 0.542 | 0.688 | 0.872 | 0.826 | 1.047 | 0.744 | 0.794 | 1.086 | 0.793 | 0.781 | 0.766 | 0.874 | 0.860 | 1.295 | 0.763 | -2.176 | |
| 2 | 1 2 4 4 | 0.893 | 2.899 | 0.992 | 0.580 | 2.800 | 0.550 | 0.588 | 0.830 | 1.840 | 0.814 | 0.774 | 1.026 | 0.709 | 0.816 | 1.001 | 0.743 | 0.877 | 2.838 | 0.882 | 2.943 | 0.797 | -1.291 | |
| 3 | 3 2 3 1 | 0.888 | 0.831 | 1.116 | 0.841 | 0.786 | 0.610 | 0.589 | 0.791 | 1.026 | 0.968 | 0.967 | 0.855 | 0.813 | 0.692 | 0.660 | 0.652 | 0.870 | 1.001 | 0.899 | 1.124 | 0.770 | -1.892 | |
| 4 | 4 3 4 1 | 0.896 | 0.967 | 0.842 | 0.630 | 1.623 | 0.563 | 0.543 | 1.682 | 0.821 | 2.768 | 1.381 | 0.667 | 0.840 | 0.798 | 0.748 | 0.815 | 1.024 | 0.718 | 2.939 | 0.839 | 0.863 | -1.545 | |
| 5 | 3 1 2 4 | 2.279 | 0.929 | 0.772 | 0.725 | 0.788 | 0.571 | 3.000 | 1.689 | 0.895 | 0.637 | 0.954 | 0.690 | 1.759 | 1.200 | 0.880 | 0.715 | 0.770 | 0.888 | 0.936 | 1.029 | 0.681 | -1.317 | |
| 6 | 3 4 4 2 | 0.986 | 0.716 | 2.866 | 0.626 | 0.752 | 0.537 | 0.550 | 0.732 | 0.930 | 0.827 | 1.635 | 0.670 | 0.825 | 0.893 | 0.667 | 2.908 | 0.926 | 0.974 | 0.889 | 0.915 | 0.669 | -2.001 | |
| 7 | 4 1 3 2 | 0.860 | 0.713 | 0.803 | 0.764 | 0.983 | 0.561 | 0.508 | 0.712 | 1.754 | 0.798 | 0.854 | 0.809 | 1.024 | 0.790 | 0.646 | 0.658 | 0.936 | 0.777 | 0.906 | 0.865 | 0.791 | -2.299 | |
| 8 | 3 3 1 3 | 0.767 | 0.884 | 0.932 | 2.697 | 0.782 | 0.544 | 0.535 | 0.684 | 0.865 | 1.001 | 0.816 | 1.409 | 0.798 | 0.725 | 0.642 | 0.694 | 0.880 | 0.870 | 0.982 | 0.953 | 0.784 | -2.041 | |
| 9 | 1 1 1 1 | 1.445 | 0.782 | 0.747 | 0.671 | 0.734 | 0.561 | 0.520 | 0.874 | 2.543 | 0.804 | 0.934 | 0.927 | 1.009 | 2.716 | 1.104 | 0.760 | 0.887 | 0.705 | 0.889 | 1.021 | 0.804 | -1.677 | |
| 10 | 4 2 2 3 | 0.905 | 0.789 | 0.906 | 0.706 | 0.624 | 0.629 | 0.503 | 0.758 | 2.712 | 0.856 | 1.050 | 0.642 | 0.662 | 0.751 | 0.855 | 0.865 | 0.703 | 0.807 | 0.834 | 0.971 | 0.731 | -2.386 | |
| 11 | 1 4 3 3 | 0.916 | 0.858 | 0.919 | 1.719 | 1.013 | 0.578 | 0.633 | 1.001 | 1.257 | 0.786 | 2.940 | 2.452 | 0.801 | 0.916 | 2.806 | 1.460 | 1.745 | 1.342 | 1.004 | 0.916 | 0.729 | -0.273 | |
| 12 | 4 4 1 4 | 0.860 | 1.115 | 0.631 | 0.754 | 0.802 | 0.565 | 0.492 | 0.728 | 0.936 | 1.166 | 0.882 | 0.812 | 2.804 | 0.836 | 0.557 | 0.661 | 0.954 | 0.859 | 0.848 | 0.947 | 0.795 | -2.269 | |
| 13 | 1 3 2 2 | 0.945 | 0.888 | 1.474 | 0.792 | 0.781 | 0.543 | 0.495 | 0.651 | 1.017 | 0.856 | 0.828 | 0.906 | 0.999 | 0.758 | 0.749 | 1.119 | 0.881 | 0.711 | 0.961 | 1.007 | 1.241 | -1.860 | |
| 14 | 2 3 3 4 | 2.846 | 0.817 | 1.027 | 0.905 | 0.656 | 3.000 | 0.532 | 0.968 | 1.015 | 0.797 | 0.838 | 2.731 | 0.858 | 1.152 | 1.011 | 0.787 | 2.909 | 0.718 | 0.821 | 1.219 | 3.000 | -0.572 | |
| 15 | 2 1 4 3 | 1.664 | 0.700 | 0.736 | 1.066 | 0.800 | 0.722 | 0.510 | 0.973 | 0.899 | 0.948 | 0.934 | 0.762 | 0.956 | 0.912 | 0.899 | 0.704 | 0.835 | 0.952 | 1.345 | 1.579 | 0.759 | -1.478 | |
| 16 | 2 4 2 1 | 1.206 | 0.866 | 0.911 | 0.686 | 0.986 | 0.604 | 0.565 | 0.817 | 2.217 | 0.817 | 0.896 | 0.840 | 0.966 | 1.220 | 0.593 | 0.746 | 0.870 | 0.704 | 0.935 | 0.913 | 0.811 | -1.790 | |

Table 10Comparisons of NSGAII, ENSHA_V and ENSHA (γ).

| Ins. | NSGAII | | | | ENSHA_V | | | | ENSHA (this work) | | | |
|----------|---------------|-------------|--------------|--------|---------|---------|---------|--------|-------------------|--------------|--------------|-------|
| | Worst | Best | Mean | SD | Worst | Best | Mean | SD | Worst | Best | Mean | SD |
| 01a | 236.99 | 0.00 | 66.07 | 66.26 | 995.84 | 590.75 | 817.36 | 108.37 | 150.67 | 0.00 | 24.15 | 37.22 |
| 02a | 126.38 | 0.00 | 47.06 | 41.80 | 1017.33 | 499.61 | 735.90 | 150.13 | 106.78 | 0.00 | 25.99 | 38.46 |
| 03a | 196.70 | 0.00 | 58.88 | 62.84 | 731.77 | 425.60 | 577.43 | 94.54 | 151.89 | 0.00 | 20.31 | 37.27 |
| 04a | 235.85 | 0.00 | 62.64 | 68.77 | 1192.44 | 617.77 | 912.72 | 124.81 | 250.33 | 0.00 | 40.89 | 71.73 |
| 05a | 130.89 | 0.00 | 40.05 | 38.57 | 907.09 | 523.12 | 684.28 | 100.01 | 183.11 | 0.00 | 34.08 | 48.56 |
| 06a | 261.75 | 0.00 | 50.70 | 80.48 | 791.50 | 434.89 | 589.77 | 101.57 | 178.45 | 0.00 | 40.13 | 57.40 |
| 07a | 469.60 | 0.00 | 94.52 | 119.80 | 1536.80 | 765.71 | 1028.39 | 174.42 | 124.39 | 0.00 | 27.56 | 36.57 |
| 08a | 360.30 | 0.00 | 108.05 | 87.57 | 1544.98 | 574.93 | 908.83 | 237.05 | 180.79 | 0.00 | 18.17 | 52.22 |
| 09a | 356.54 | 0.00 | 84.87 | 112.13 | 1258.17 | 472.66 | 782.67 | 211.24 | 252.96 | 0.00 | 42.38 | 75.49 |
| 10a | 332.42 | 0.00 | 96.85 | 105.06 | 1268.80 | 629.91 | 1008.12 | 199.35 | 165.58 | 0.00 | 36.19 | 48.32 |
| 11a | 344.74 | 0.00 | 95.63 | 105.20 | 1648.35 | 729.00 | 1023.45 | 265.56 | 223.47 | 0.00 | 47.23 | 68.93 |
| 12a | 259.32 | 0.00 | 60.26 | 72.72 | 1136.32 | 394.07 | 768.26 | 233.33 | 99.67 | 0.00 | 32.12 | 36.02 |
| 13a | 382.98 | 0.00 | 118.93 | 88.82 | 1717.55 | 1020.09 | 1354.94 | 186.30 | 211.75 | 0.00 | 20.96 | 51.24 |
| 14a | 418.41 | 0.00 | 137.30 | 137.15 | 2009.62 | 602.94 | 1212.89 | 366.95 | 245.65 | 0.00 | 31.52 | 64.33 |
| 15a | 349.30 | 0.00 | 145.81 | 122.21 | 1185.64 | 247.36 | 866.10 | 245.91 | 292.20 | 0.00 | 23.52 | 69.68 |
| 16a | 463.00 | 0.00 | 108.44 | 124.76 | 1852.10 | 965.35 | 1408.69 | 191.83 | 243.42 | 0.00 | 38.56 | 63.96 |
| 17a | 414.55 | 0.00 | 145.20 | 125.72 | 2060.57 | 851.91 | 1657.32 | 302.34 | 332.56 | 0.00 | 27.53 | 75.93 |
| 18a | 537.42 | 0.00 | 121.42 | 145.87 | 1283.43 | 445.20 | 787.04 | 230.69 | 330.19 | 0.00 | 33.81 | 83.36 |
| mt10c1 | 123.87 | 0.00 | 42.58 | 43.20 | 312.75 | 29.25 | 157.31 | 79.08 | 111.32 | 0.00 | 20.14 | 31.50 |
| mt10cc | 143.15 | 0.00 | 40.68 | 37.03 | 297.88 | 69.63 | 185.72 | 50.90 | 97.44 | 0.00 | 20.82 | 27.63 |
| mt10x | 65.02 | 0.00 | 21.78 | 22.15 | 256.16 | 70.83 | 168.47 | 55.71 | 118.33 | 0.00 | 27.22 | 36.48 |
| mt10xx | 211.45 | 0.00 | 44.25 | 49.26 | 299.29 | 57.10 | 146.54 | 69.08 | 96.18 | 0.00 | 12.54 | 21.95 |
| mt10xxx | 96.55 | 96.55 | 33.95 | 30.81 | 239.18 | 239.18 | 142.62 | 71.39 | 73.17 | 73.17 | 25.54 | 23.57 |
| mt10xy | 97.88 | 0.00 | 28.17 | 26.91 | 258.62 | 113.33 | 176.44 | 45.21 | 78.82 | 0.00 | 17.52 | 21.54 |
| mt10xyz | 267.39 | 0.00 | 60.84 | 72.26 | 275.20 | 0.00 | 125.54 | 80.49 | 152.64 | 0.00 | 32.05 | 41.84 |
| setb4c9 | 186.17 | 0.00 | 37.31 | 49.52 | 500.71 | 212.03 | 329.95 | 66.21 | 126.67 | 0.00 | 29.07 | 36.40 |
| setb4cc | 208.21 | 0.00 | 34.75 | 47.92 | 434.41 | 158.22 | 282.43 | 70.99 | 147.75 | 0.00 | 33.57 | 40.31 |
| setb4x | 135.83 | 0.00 | 39.13 | 43.82 | 554.91 | 159.57 | 313.22 | 99.86 | 148.41 | 0.00 | 25.86 | 35.28 |
| setb4xx | 125.74 | 0.00 | 25.02 | 32.31 | 414.15 | 191.25 | 306.76 | 54.97 | 173.65 | 0.00 | 30.70 | 44.27 |
| setb4xxx | 92.96 | 0.00 | 26.69 | 31.24 | 508.53 | 219.99 | 325.07 | 62.34 | 78.89 | 0.00 | 27.91 | 26.80 |
| setb4xy | 92.96 | 0.00 | 26.69 | 31.24 | 508.53 | 219.99 | 325.07 | 62.34 | 78.89 | 0.00 | 27.91 | 26.80 |
| setb4xyz | 83.10 | 0.00 | 32.03 | 26.97 | 405.86 | 182.42 | 298.07 | 60.37 | 101.61 | 0.00 | 19.30 | 28.49 |
| seti5c12 | 167.19 | 0.00 | 47.79 | 56.04 | 691.62 | 319.30 | 454.39 | 100.28 | 156.44 | 0.00 | 36.77 | 46.24 |
| seti5cc | 131.80 | 0.00 | 44.17 | 43.36 | 578.57 | 165.16 | 401.61 | 104.65 | 58.59 | 0.00 | 21.49 | 18.43 |
| seti5x | 169.71 | 0.00 | 46.26 | 50.46 | 656.61 | 269.29 | 424.89 | 88.56 | 125.72 | 0.00 | 36.80 | 42.94 |
| seti5xx | 132.24 | 0.00 | 36.97 | 36.84 | 604.36 | 314.11 | 434.58 | 89.04 | 177.63 | 0.00 | 28.93 | 42.71 |
| seti5xxx | 188.71 | 0.00 | 45.85 | 51.59 | 553.96 | 262.01 | 435.91 | 82.98 | 103.22 | 0.00 | 29.06 | 28.85 |
| seti5xy | 133.99 | 0.00 | 35.70 | 41.90 | 757.60 | 218.68 | 451.81 | 117.19 | 210.05 | 0.00 | 35.52 | 54.76 |
| seti5xyz | 170.53 | | | | | | | | | | | |

Table 11Comparisons of NSGAII, ENSHA_V and ENSHA (Δ).

| Ins. | NSGAII | | | | ENSHA_V | | | | ENSHA (this work) | | | |
|----------|-------------|-------------|-------------|------|-------------|------|------|------|-------------------|-------------|-------------|------|
| | Worst | Best | Mean | SD | Worst | Best | Mean | SD | Worst | Best | Mean | SD |
| 01a | 0.62 | 0.29 | 0.49 | 0.08 | 0.95 | 0.82 | 0.88 | 0.03 | 0.60 | 0.27 | 0.40 | 0.09 |
| 02a | 0.69 | 0.31 | 0.49 | 0.11 | 0.97 | 0.78 | 0.89 | 0.05 | 0.75 | 0.31 | 0.51 | 0.11 |
| 03a | 0.90 | 0.44 | 0.70 | 0.11 | 1.00 | 0.72 | 0.91 | 0.06 | 0.82 | 0.35 | 0.56 | 0.13 |
| 04a | 0.77 | 0.30 | 0.46 | 0.11 | 0.99 | 0.84 | 0.90 | 0.04 | 0.60 | 0.31 | 0.46 | 0.08 |
| 05a | 0.84 | 0.25 | 0.54 | 0.16 | 1.04 | 0.76 | 0.90 | 0.07 | 0.67 | 0.25 | 0.47 | 0.12 |
| 06a | 0.87 | 0.51 | 0.68 | 0.10 | 0.98 | 0.81 | 0.89 | 0.04 | 0.85 | 0.44 | 0.67 | 0.12 |
| 07a | 0.66 | 0.39 | 0.51 | 0.08 | 0.97 | 0.82 | 0.92 | 0.04 | 0.51 | 0.27 | 0.40 | 0.07 |
| 08a | 0.92 | 0.39 | 0.58 | 0.11 | 0.99 | 0.87 | 0.93 | 0.03 | 0.74 | 0.34 | 0.47 | 0.11 |
| 09a | 1.03 | 0.46 | 0.78 | 0.15 | 0.97 | 0.81 | 0.89 | 0.05 | 1.14 | 0.25 | 0.66 | 0.18 |
| 10a | 0.64 | 0.29 | 0.49 | 0.09 | 0.98 | 0.82 | 0.92 | 0.04 | 0.61 | 0.29 | 0.41 | 0.09 |
| 11a | 0.97 | 0.39 | 0.58 | 0.14 | 0.98 | 0.83 | 0.89 | 0.05 | 0.73 | 0.30 | 0.48 | 0.12 |
| 12a | 1.02 | 0.36 | 0.74 | 0.15 | 0.97 | 0.78 | 0.88 | 0.05 | 1.01 | 0.38 | 0.70 | 0.17 |
| 13a | 0.67 | 0.33 | 0.53 | 0.08 | 0.97 | 0.87 | 0.93 | 0.03 | 0.56 | 0.27 | 0.39 | 0.08 |
| 14a | 0.99 | 0.38 | 0.69 | 0.13 | 0.98 | 0.83 | 0.90 | 0.04 | 0.79 | 0.38 | 0.59 | 0.13 |
| 15a | 1.01 | 0.42 | 0.80 | 0.16 | 1.00 | 0.76 | 0.87 | 0.07 | 1.00 | 0.33 | 0.72 | 0.20 |
| 16a | 0.71 | 0.40 | 0.54 | 0.08 | 0.97 | 0.87 | 0.93 | 0.03 | 0.64 | 0.27 | 0.43 | 0.11 |
| 17a | 1.08 | 0.45 | 0.67 | 0.14 | 0.95 | 0.84 | 0.89 | 0.03 | 0.81 | 0.33 | 0.55 | 0.16 |
| 18a | 0.98 | 0.32 | 0.78 | 0.17 | 1.03 | 0.77 | 0.86 | 0.07 | 1.22 | 0.45 | 0.76 | 0.19 |
| mt10c1 | 0.68 | 0.29 | 0.53 | 0.10 | 1.08 | 0.69 | 0.83 | 0.09 | 0.81 | 0.29 | 0.48 | 0.14 |
| mt10cc | 0.71 | 0.34 | 0.51 | 0.09 | 0.89 | 0.67 | 0.81 | 0.06 | 0.63 | 0.27 | 0.46 | 0.10 |
| mt10x | 0.83 | 0.43 | 0.60 | 0.09 | 0.94 | 0.61 | 0.80 | 0.09 | 0.76 | 0.45 | 0.58 | 0.10 |
| mt10xx | 0.76 | 0.42 | 0.58 | 0.10 | 0.96 | 0.62 | 0.83 | 0.09 | 0.76 | 0.36 | 0.52 | 0.11 |
| mt10xxx | 0.83 | 0.83 | 0.55 | 0.12 | 0.98 | 0.98 | 0.82 | 0.09 | 0.79 | 0.79 | 0.49 | 0.11 |
| mt10xy | 0.61 | 0.38 | 0.53 | 0.06 | 0.92 | 0.67 | 0.82 | 0.07 | 0.67 | 0.31 | 0.50 | 0.10 |
| mt10xyz | 0.86 | 0.31 | 0.57 | 0.12 | 0.95 | 0.69 | 0.82 | 0.07 | 0.64 | 0.40 | 0.51 | 0.07 |
| setb4c9 | 0.62 | 0.30 | 0.45 | 0.09 | 0.91 | 0.74 | 0.84 | 0.04 | 0.65 | 0.33 | 0.47 | 0.10 |
| setb4cc | 0.73 | 0.34 | 0.50 | 0.13 | 0.96 | 0.71 | 0.86 | 0.06 | 0.64 | 0.29 | 0.48 | 0.10 |
| setb4x | 0.65 | 0.33 | 0.48 | 0.08 | 0.97 | 0.77 | 0.86 | 0.06 | 0.61 | 0.32 | 0.45 | 0.09 |
| setb4xx | 0.66 | 0.31 | 0.46 | 0.10 | 1.09 | 0.83 | 0.89 | 0.06 | 0.62 | 0.24 | 0.50 | 0.10 |
| setb4xxx | 0.73 | 0.31 | 0.47 | 0.10 | 0.94 | 0.78 | 0.87 | 0.05 | 0.75 | 0.33 | 0.52 | 0.10 |
| setb4xy | 0.79 | 0.27 | 0.50 | 0.12 | 0.95 | 0.77 | 0.86 | 0.05 | 0.79 | 0.33 | 0.50 | 0.14 |
| setb4xyz | 0.65 | 0.38 | 0.50 | 0.08 | 1.01 | 0.77 | 0.89 | 0.06 | 0.67 | 0.29 | 0.49 | 0.10 |
| seti5c12 | 0.60 | 0.27 | 0.43 | 0.08 | 0.95 | 0.79 | 0.87 | 0.05 | 0.63 | 0.31 | 0.45 | 0.09 |
| seti5cc | 0.62 | 0.32 | 0.47 | 0.08 | 0.99 | 0.80 | 0.88 | 0.05 | 0.59 | 0.30 | 0.40 | 0.08 |
| seti5x | 0.67 | 0.33 | 0.44 | 0.08 | 0.98 | 0.79 | 0.89 | 0.06 | 0.81 | 0.29 | 0.44 | 0.11 |
| seti5xx | 0.75 | 0.28 | 0.46 | 0.11 | 0.96 | 0.82 | 0.88 | 0.04 | 0.61 | 0.35 | 0.46 | 0.08 |
| seti5xxx | 0.68 | 0.29 | 0.45 | 0.09 | 0.96 | 0.73 | 0.88 | 0.06 | 0.60 | 0.26 | 0.43 | 0.08 |
| seti5xy | 0.67 | 0.29 | 0.47 | 0.11 | 1.01 | 0.80 | 0.89 | 0.06 | 0.57 | 0.31 | 0.44 | 0.08 |
| seti5xyz | 0.57 | 0.32 | 0.46 | 0.07 | 1.00 | 0.80 | 0.87 | 0.04 | 0.67 | 0.26 | 0.46 | 0.10 |
| Average | 0.77 | 0.36 | 0.55 | 0.11 | 0.98 | 0.78 | 0.88 | 0.05 | 0.73 | 0.33 | 0.50 | 0.11 |
| NB | 12 | 16 | 10 | – | 3 | 0 | 0 | – | 27 | 28 | 34 | – |

for γ and Δ , it shows that ENSHA can obtain more competitive nondominated solutions than NSGAII and ENSHA_V. As for Ω , it is clear that ENSHA can achieve good performance and the obtained nondominated solutions are closer to APF than NSGAII and ENSHA_V. Therefore, the proposed EDA-based machine learning mechanism and the cooperation-based refinement strategy are helpful in enhancing the overall search ability of ENSHA.

To make a reliable conclusion on the performance of ENSHA, the single factor analysis of variance (ANOVA) [65,66] is used to analyze the significant differences among NSGAII, ENSHA_V and ENSHA at 95% CI. In doing so, we are interesting in the significant differences among algorithms in terms of three performance metrics. The results of ANOVA associated with γ , Δ and Ω are given in Table 13.

As shown in Table 13, the p -values for γ , Δ and Ω are all less than 0.05, so there are significant differences among NSGAII, ENSHA_V and ENSHA at 95% CI. Since there are three algorithms in the comparisons, a post hoc analysis using Tukey's multiple comparison tests [65,66] is conducted to further show the significant difference of the pairwise comparisons of algorithms. Note here that we perform Tukey's tests based on the mean values of performance metrics for all instances at 95% CI. The results of Tukey's multiple cooperation tests are given in Table 14.

From Table 14, the p -values for three pairwise comparisons are less than 0.05 in terms of Δ and Ω , implying the significant differences between ENSHA and other algorithms for such two

Table 12Comparisons of NSGAII, ENSHA_V and ENSHA (Ω).

| Ins. | NSGAII | | | | ENSHA_V | | | | ENSHA (this work) | | | |
|----------|-------------|-------------|-------------|------|---------|-------------|------|------|-------------------|-------------|-------------|------|
| | Worst | Best | Mean | SD | Worst | Best | Mean | SD | Worst | Best | Mean | SD |
| 01a | 0.00 | 1.00 | 0.47 | 0.33 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.69 | 0.31 |
| 02a | 0.00 | 1.00 | 0.45 | 0.34 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.71 | 0.33 |
| 03a | 0.00 | 1.00 | 0.39 | 0.40 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.70 | 0.37 |
| 04a | 0.00 | 1.00 | 0.48 | 0.37 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.66 | 0.39 |
| 05a | 0.00 | 1.00 | 0.47 | 0.39 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.64 | 0.38 |
| 06a | 0.00 | 1.00 | 0.60 | 0.41 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.60 | 0.40 |
| 07a | 0.00 | 1.00 | 0.44 | 0.43 | 0.00 | 0.00 | 0.00 | 0.00 | 0.07 | 1.00 | 0.65 | 0.38 |
| 08a | 0.00 | 1.00 | 0.24 | 0.31 | 0.00 | 0.00 | 0.00 | 0.00 | 0.14 | 1.00 | 0.87 | 0.25 |
| 09a | 0.00 | 1.00 | 0.52 | 0.44 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.71 | 0.36 |
| 10a | 0.00 | 1.00 | 0.44 | 0.42 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.68 | 0.34 |
| 11a | 0.00 | 1.00 | 0.40 | 0.41 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.66 | 0.38 |
| 12a | 0.00 | 1.00 | 0.52 | 0.45 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.61 | 0.42 |
| 13a | 0.00 | 1.00 | 0.20 | 0.28 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.87 | 0.24 |
| 14a | 0.00 | 1.00 | 0.36 | 0.40 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.76 | 0.37 |
| 15a | 0.00 | 1.00 | 0.26 | 0.37 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.85 | 0.33 |
| 16a | 0.00 | 1.00 | 0.43 | 0.34 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.73 | 0.31 |
| 17a | 0.00 | 1.00 | 0.29 | 0.40 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.82 | 0.31 |
| 18a | 0.00 | 1.00 | 0.49 | 0.43 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.79 | 0.35 |
| mt10c1 | 0.00 | 1.00 | 0.46 | 0.38 | 0.00 | 0.44 | 0.03 | 0.10 | 0.00 | 1.00 | 0.63 | 0.37 |
| mt10cc | 0.00 | 1.00 | 0.44 | 0.34 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.64 | 0.37 |
| mt10x | 0.00 | 1.00 | 0.58 | 0.39 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.54 | 0.37 |
| mt10xx | 0.00 | 1.00 | 0.39 | 0.34 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.71 | 0.30 |
| mt10xxx | 0.00 | 1.00 | 0.52 | 0.29 | 0.00 | 0.75 | 0.06 | 0.19 | 0.05 | 1.00 | 0.56 | 0.26 |
| mt10xy | 0.06 | 1.00 | 0.51 | 0.31 | 0.00 | 0.00 | 0.00 | 0.05 | 0.00 | 1.00 | 0.62 | 0.31 |
| mt10xyz | 0.00 | 1.00 | 0.46 | 0.36 | 0.00 | 1.00 | 0.11 | 0.28 | 0.00 | 1.00 | 0.60 | 0.37 |
| setb4c9 | 0.00 | 1.00 | 0.55 | 0.34 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.54 | 0.32 |
| setb4cc | 0.00 | 1.00 | 0.53 | 0.41 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.58 | 0.40 |
| setb4x | 0.00 | 1.00 | 0.53 | 0.33 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.59 | 0.32 |
| setb4xx | 0.00 | 1.00 | 0.58 | 0.32 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.55 | 0.36 |
| setb4xxx | 0.00 | 1.00 | 0.54 | 0.38 | 0.00 | 0.00 | | | | | | |

Table 14

Results of Tukey's tests (NSGAII, ENSHA_V and ENSHA).

| Indicator | ENSHA_V vs. NSGAII | | | ENSHA_V vs. ENSHA | | | NSGAII vs. ENSHA | | |
|---------------------|---------------------|-------------------|--------------------|---------------------|-------------------|--------------------|---------------------|-------------------|--------------------|
| | γ | Δ | Ω | γ | Δ | Ω | γ | Δ | Ω |
| Difference of means | 537.32 | 0.32 | -0.46 | 571.13 | 0.37 | -0.66 | 33.81 | 0.05 | -0.20 |
| SE of difference | 51.53 | 0.02 | 0.02 | 51.53 | 0.02 | 0.02 | 51.53 | 0.02 | 0.02 |
| 95% CI | 659.68 ^a | 0.37 ^a | -0.49 ^b | 693.49 ^a | 0.41 ^a | -0.69 ^b | 156.18 ^a | 0.09 ^a | -0.24 ^b |
| p-value | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.79 | 0.04 | 0.00 |

^aUpper bound of 95% CI.^bLower bound of 95% CI.**Table 15**Comparisons of MOPSO, MOEA/D, MOGLS, SPEAII and ENSHA (γ).

| Ins. | MOPSO | | | | MOEA/D | | | | MOGLS | | | | SPEAII | | | | ENSHA (this work) | | | |
|----------|---------|--------|---------|--------|---------------|-------------|--------------|--------|---------|---------|---------|--------|---------------|-------------|--------|--------|-------------------|-------------|--------------|--------|
| | Worst | Best | Mean | SD | Worst | Best | Mean | SD | Worst | Best | Mean | SD | Worst | Best | Mean | SD | Worst | Best | Mean | SD |
| 01a | 1215.18 | 355.93 | 640.05 | 208.09 | 282.51 | 0.00 | 86.13 | 84.60 | 1986.19 | 1499.92 | 1744.41 | 128.98 | 344.38 | 16.99 | 134.37 | 87.29 | 39.39 | 0.00 | 9.80 | 12.22 |
| 02a | 1128.02 | 375.79 | 695.52 | 225.46 | 195.12 | 0.00 | 81.81 | 63.49 | 1824.19 | 1429.00 | 1650.19 | 108.77 | 566.92 | 0.00 | 145.04 | 141.15 | 112.67 | 0.00 | 17.08 | 28.26 |
| 03a | 764.48 | 309.87 | 544.13 | 105.70 | 233.08 | 0.00 | 76.09 | 74.98 | 1455.57 | 959.06 | 1166.90 | 127.94 | 416.41 | 0.00 | 90.87 | 100.40 | 125.93 | 0.00 | 24.07 | 35.85 |
| 04a | 983.44 | 339.49 | 751.08 | 155.63 | 295.03 | 0.00 | 86.80 | 84.33 | 2093.38 | 1713.28 | 1868.50 | 107.62 | 389.72 | 51.79 | 152.47 | 89.19 | 111.07 | 0.00 | 21.39 | 36.71 |
| 05a | 885.14 | 198.39 | 575.16 | 152.66 | 233.51 | 0.00 | 108.39 | 71.25 | 1692.82 | 1324.04 | 1523.52 | 111.64 | 268.85 | 2.12 | 127.54 | 80.74 | 72.09 | 0.00 | 13.62 | 25.55 |
| 06a | 990.41 | 332.00 | 630.19 | 163.04 | 309.59 | 0.00 | 99.05 | 91.10 | 1691.20 | 1165.29 | 1384.20 | 118.48 | 518.58 | 0.00 | 137.59 | 113.28 | 88.53 | 0.00 | 10.96 | 20.30 |
| 07a | 1411.69 | 656.15 | 1015.01 | 197.37 | 245.46 | 0.18 | 86.21 | 71.33 | 2893.25 | 1924.00 | 2408.96 | 239.17 | 511.10 | 0.00 | 166.53 | 132.67 | 221.38 | 0.00 | 29.21 | 56.71 |
| 08a | 1552.63 | 398.75 | 857.83 | 281.51 | 256.14 | 0.00 | 60.97 | 85.92 | 1983.44 | 1490.46 | 1726.58 | 130.92 | 224.12 | 32.89 | 116.48 | 61.19 | 109.52 | 0.00 | 33.24 | 33.75 |
| 09a | 1085.91 | 506.18 | 857.74 | 158.80 | 285.42 | 0.00 | 79.26 | 80.26 | 1433.37 | 1033.21 | 1295.68 | 101.43 | 379.66 | 0.00 | 78.29 | 94.97 | 241.44 | 0.00 | 64.14 | 77.67 |
| 10a | 1485.36 | 514.43 | 932.18 | 269.00 | 181.83 | 0.00 | 56.30 | 58.91 | 2707.30 | 1779.98 | 2330.09 | 271.12 | 364.28 | 14.89 | 187.54 | 110.84 | 126.34 | 0.00 | 27.61 | 38.93 |
| 11a | 1282.14 | 417.47 | 898.24 | 233.40 | 160.93 | 0.00 | 42.68 | 53.86 | 2087.28 | 1167.14 | 1777.03 | 210.60 | 450.27 | 2.57 | 110.83 | 118.73 | 252.03 | 0.00 | 51.78 | 64.04 |
| 12a | 1182.01 | 485.37 | 805.73 | 225.55 | 351.65 | 0.00 | 73.96 | 83.04 | 1505.63 | 1001.01 | 1336.49 | 114.73 | 358.37 | 0.00 | 103.04 | 95.41 | 257.14 | 0.00 | 43.82 | 73.46 |
| 13a | 1789.20 | 667.80 | 1137.11 | 296.47 | 258.20 | 0.00 | 75.84 | 86.25 | 3380.41 | 2198.07 | 2748.55 | 310.59 | 445.59 | 0.00 | 189.46 | 132.77 | 384.69 | 0.00 | 66.56 | 95.77 |
| 14a | 2491.91 | 660.10 | 1420.98 | 531.98 | 671.78 | 0.00 | 82.56 | 157.90 | 2631.58 | 1705.74 | 2095.81 | 219.04 | 574.99 | 0.00 | 175.96 | 156.06 | 447.14 | 0.00 | 68.17 | 113.53 |
| 15a | 1309.52 | 383.34 | 854.25 | 264.68 | 438.24 | 0.00 | 89.31 | 117.21 | 1461.31 | 839.38 | 1141.01 | 186.27 | 567.87 | 0.00 | 127.48 | 173.11 | 451.94 | 0.00 | 64.12 | 118.24 |
| 16a | 3007.55 | 721.21 | 1277.19 | 527.90 | 211.34 | 0.00 | 26.24 | 55.29 | 3247.12 | 2016.85 | 2683.13 | 264.40 | 558.09 | 0.00 | 188.48 | 140.11 | 198.67 | 0.00 | 61.98 | 55.68 |
| 17a | 1767.22 | 549.56 | 1221.99 | 317.48 | 286.72 | 0.00 | 100.56 | 97.51 | 2389.09 | 1429.52 | 1955.51 | 288.00 | 505.68 | 0.00 | 135.83 | 136.98 | 256.37 | 0.00 | 66.21 | 89.46 |
| 18a | 1367.84 | 579.91 | 921.38 | 217.94 | 449.97 | 0.00 | 129.87 | 156.19 | 1563.61 | 903.42 | 1264.36 | 178.63 | 213.17 | 0.00 | 56.28 | 71.89 | 627.70 | 0.00 | 64.60 | 169.66 |
| mt10c1 | 262.01 | 28.47 | 165.91 | 69.30 | 136.85 | 3.55 | 64.74 | 44.02 | 743.59 | 545.81 | 642.88 | 55.60 | 201.20 | 0.32 | 97.08 | 44.89 | 42.00 | 0.00 | 8.93 | 11.77 |
| mt10cc | 313.45 | 59.44 | 182.30 | 57.90 | 108.48 | 0.00 | 39.66 | 32.65 | 731.42 | 441.33 | 600.27 | 81.70 | 259.32 | 0.00 | 110.19 | 77.33 | 101.28 | 0.00 | 32.02 | 31.92 |
| mt10x | 306.47 | 85.81 | 187.27 | 65.81 | 204.45 | 0.00 | 42.25 | 51.72 | 768.52 | 478.84 | 608.59 | 71.25 | 241.36 | 15.19 | 72.15 | 56.90 | 72.57 | 0.00 | 34.03 | 18.50 |
| mt10xx | 320.71 | 45.96 | 179.62 | 65.12 | 148.57 | 0.00 | 51.19 | 45.82 | 788.00 | 499.51 | 636.61 | 77.65 | 165.63 | 2.71 | 72.84 | 50.63 | 177.92 | 0.00 | 20.35 | 40.23 |
| mt10xxx | 322.59 | 81.05 | 181.11 | 66.40 | 79.32 | 0.00 | 24.24 | 23.50 | 691.27 | 426.00 | 571.26 | 61.19 | 168.04 | 0.00 | 71.81 | 50.27 | 79.79 | 0.00 | 13.42 | 19.69 |
| mt10xy | 279.20 | 41.96 | 151.20 | 64.57 | 148.47 | 0.00 | 54.00 | 43.03 | 753.16 | 423.20 | 572.42 | 83.00 | 213.18 | 1.20 | 83.25 | 50.87 | 62.87 | 0.00 | 22.36 | 18.27 |
| mt10xyz | 502.93 | 53.19 | 207.46 | 99.79 | 254.96 | 0.00 | 59.18 | 63.35 | 827.30 | 455.27 | 609.46 | 103.10 | 207.11 | 1.10 | 95.28 | 60.72 | 231.20 | 0.00 | 31.50 | 61.78 |
| setb4c9 | 461.24 | 105.68 | 260.40 | 86.03 | 104.84 | 0.00 | 32.60 | 27.01 | 1016.56 | 641.28 | 873.03 | 94.76 | 185.44 | 23.86 | 87.40 | 49.79 | 99.45 | 0.00 | 18.31 | 26.49 |
| setb4c4 | 410.97 | 111.35 | 273.40 | 83.38 | 123.64 | 0.00 | 43.90 | 38.60 | 1173.79 | 603.22 | 802.23 | 128.99 | 180.86 | 33.11 | 98.86 | 47.00 | 72.41 | 0.00 | 18.65 | 25.45 |
| setb4x4 | 525.96 | 125.35 | 256.54 | 93.08 | 123.82 | 0.00 | 26.79 | 35.48 | 1078.10 | 559.51 | 806.35 | 124.76 | 359.83 | 0.00 | 118.36 | 77.13 | 72.19 | 0.00 | 25.77 | 23.20 |
| setb4xx | 476.03 | 95.26 | 274.28 | 94.32 | 160.58 | 0.57 | 46.91 | 46.56 | 1077.88 | 676.19 | 860.82 | 105.82 | 214.13 | 7.21 | 80.17 | 58.92 | 55.65 | 0.00 | 17.54 | 18.25 |
| setb4xxx | 608.61 | 135.64 | 268.39 | 115.50 | 171.95 | 1.63 | 61.70 | 53.04 | 1267.06 | 648.89 | 827.84 | 134.08 | 387.95 | 1.44 | 104.64 | 88.62 | 30.25 | 0.00 | 66.90 | 10.42 |
| setb4xy | 473.92 | 151.68 | 284.05 | 81.44 | 226.49 | 0.00 | 39.14 | 52.94 | 955.14 | 654.27 | 825.40 | 84.89 | 393.30 | 2.84 | 118.87 | 90.56 | 113.64 | 0.00 | 22.40 | 27.50 |
| setb4xyz | 592.02 | 88.46 | 277.97 | 123.79 | 137.25 | 0.79 | 34.29 | 33.76 | 1039.78 | 659.48 | 846.31 | 91.06 | 158.49 | 2.35 | 70.82 | 44.55 | 71.64 | 0.00 | 12.03 | 16.20 |
| seti5c12 | 739.53 | 250.95 | 435.38 | 113.06 | 210.18 | 0.00 | 63.98 | 71.63 | 1555.43 | 1011.93 | 1282.30 | 163.48 | 453.15 | 3.79 | 132.04 | 115.61 | 99.50 | 0.00 | 22.75 | 27.63 |
| seti5cc | 812.49 | 162.22 | 433.34 | 184.07 | 171.56 | 0.00 | 42.71 | 50.89 | 1444.79 | 867.84 | 1149.27 | 152.66 | 398.26 | 21.50 | 153.66 | 85.42 | 136.78 | 0.00 | 28.77 | 36.68 |
| seti5x | 531.15 | 228.48 | 347.02 | 87.27 | 104.62 | 0.00 | 33.17 | 38.46 | 1454.73 | 897.97 | 1114.93 | 140.66 | 330.39 | 0.00 | 132.04 | 91.42 | 161.95 | 0.00 | 46.65 | 51.58 |
| seti5xx | 531.97 | 186.52 | 381.50 | 111.78 | 122.01 | 0.00 | 26.93 | 28.23 | 1434.48 | 999.42 | 1238.66 | 105.42 | 216.63 | 0.00 | 84.65 | 58.57 | 65.45 | 0.00 | 27.59 | 23.01 |
| seti5xxx | 826.12 | 298.83 | 430.43 | 143.34 | 245.98 | 0.00 | 52.96 | 73.69 | 1514.22 | 984.34 | 1228.79 | 137.79 | 253.62 | 17.25 | 110.00 | 71.49 | 88.43 | 0.00 | 24.50 | 29.74 |
| seti5xy | 716.63 | 258.43 | 427.35 | 113.24 | 153.01 | 0.00 | 43.23 | 49.01 | 1488.41 | 979.54 | 1252.37 | 117.22 | 321.91 | 49.34 | 129.17 | 66.74 | 152.34 | 0.00 | 25.09 | 38.32 |
| seti5xyz | 668.92 | 212.03 | 375.45 | 117.40 | | | | | | | | | | | | | | | | |

Table 16Comparisons of MOPSO, MOEA/D, MOGLS, SPEAII and NSHA (Δ).

| Ins. | MOPSO | | | | MOEA/D | | | | MOGLS | | | | SPEAII | | | | ENSHA (this work) | | | |
|----------|-------------|------|------|------|--------|------|------|------|-------------|------|------|------|--------|------|------|------|-------------------|-------------|-------------|------|
| | Worst | Best | Mean | SD | Worst | Best | Mean | SD | Worst | Best | Mean | SD | Worst | Best | Mean | SD | Worst | Best | Mean | SD |
| 01a | 0.93 | 0.79 | 0.87 | 0.04 | 1.00 | 0.73 | 0.87 | 0.06 | 0.93 | 0.79 | 0.88 | 0.04 | 1.19 | 0.93 | 1.05 | 0.07 | 0.56 | 0.32 | 0.46 | 0.08 |
| 02a | 0.96 | 0.81 | 0.89 | 0.04 | 0.98 | 0.70 | 0.87 | 0.07 | 0.94 | 0.75 | 0.87 | 0.05 | 1.45 | 0.89 | 1.16 | 0.14 | 0.79 | 0.29 | 0.51 | 0.13 |
| 03a | 0.99 | 0.79 | 0.89 | 0.05 | 1.01 | 0.63 | 0.82 | 0.10 | 0.95 | 0.76 | 0.87 | 0.05 | 1.72 | 1.04 | 1.32 | 0.14 | 0.84 | 0.45 | 0.63 | 0.13 |
| 04a | 0.95 | 0.81 | 0.87 | 0.04 | 0.97 | 0.66 | 0.84 | 0.08 | 0.97 | 0.80 | 0.88 | 0.04 | 1.30 | 0.82 | 1.09 | 0.11 | 0.56 | 0.31 | 0.43 | 0.06 |
| 05a | 1.02 | 0.79 | 0.88 | 0.05 | 0.95 | 0.73 | 0.83 | 0.06 | 0.97 | 0.78 | 0.86 | 0.05 | 1.42 | 0.77 | 1.15 | 0.15 | 0.79 | 0.32 | 0.52 | 0.12 |
| 06a | 0.98 | 0.80 | 0.88 | 0.05 | 0.97 | 0.62 | 0.79 | 0.09 | 0.94 | 0.76 | 0.87 | 0.05 | 1.61 | 1.17 | 1.27 | 0.10 | 0.94 | 0.43 | 0.64 | 0.14 |
| 07a | 1.00 | 0.86 | 0.92 | 0.04 | 1.00 | 0.81 | 0.88 | 0.05 | 0.97 | 0.85 | 0.90 | 0.04 | 1.32 | 0.80 | 1.12 | 0.12 | 0.70 | 0.32 | 0.50 | 0.11 |
| 08a | 1.03 | 0.81 | 0.88 | 0.06 | 0.98 | 0.66 | 0.84 | 0.08 | 0.96 | 0.80 | 0.87 | 0.04 | 1.43 | 0.81 | 1.23 | 0.15 | 1.00 | 0.35 | 0.62 | 0.12 |
| 09a | 0.98 | 0.74 | 0.88 | 0.06 | 1.00 | 0.58 | 0.84 | 0.10 | 0.92 | 0.78 | 0.86 | 0.04 | 1.66 | 1.15 | 1.31 | 0.14 | 0.93 | 0.34 | 0.68 | 0.14 |
| 10a | 0.96 | 0.83 | 0.90 | 0.04 | 0.97 | 0.77 | 0.87 | 0.05 | 0.96 | 0.83 | 0.89 | 0.03 | 1.38 | 0.76 | 1.05 | 0.16 | 0.77 | 0.31 | 0.54 | 0.13 |
| 11a | 1.05 | 0.84 | 0.90 | 0.04 | 0.98 | 0.78 | 0.87 | 0.05 | 1.00 | 0.81 | 0.89 | 0.05 | 1.50 | 0.96 | 1.26 | 0.12 | 0.80 | 0.35 | 0.59 | 0.14 |
| 12a | 0.94 | 0.72 | 0.89 | 0.05 | 0.96 | 0.65 | 0.85 | 0.08 | 0.95 | 0.73 | 0.86 | 0.05 | 1.64 | 1.11 | 1.34 | 0.14 | 0.92 | 0.38 | 0.71 | 0.13 |
| 13a | 1.02 | 0.87 | 0.93 | 0.04 | 0.96 | 0.80 | 0.90 | 0.04 | 0.95 | 0.86 | 0.90 | 0.02 | 1.24 | 0.73 | 1.06 | 0.12 | 0.71 | 0.43 | 0.61 | 0.08 |
| 14a | 0.94 | 0.86 | 0.89 | 0.02 | 0.94 | 0.65 | 0.86 | 0.08 | 0.94 | 0.79 | 0.88 | 0.04 | 1.42 | 0.91 | 1.20 | 0.14 | 0.83 | 0.37 | 0.70 | 0.12 |
| 15a | 1.02 | 0.54 | 0.84 | 0.10 | 1.16 | 0.79 | 0.92 | 0.08 | 1.02 | 0.76 | 0.87 | 0.07 | 1.71 | 1.21 | 1.38 | 0.15 | 1.30 | 0.50 | 0.79 | 0.18 |
| 16a | 1.04 | 0.85 | 0.93 | 0.04 | 0.98 | 0.87 | 0.92 | 0.03 | 0.95 | 0.84 | 0.91 | 0.03 | 1.21 | 0.63 | 1.00 | 0.15 | 0.73 | 0.36 | 0.57 | 0.11 |
| 17a | 0.95 | 0.80 | 0.89 | 0.04 | 0.99 | 0.70 | 0.89 | 0.08 | 0.95 | 0.82 | 0.88 | 0.03 | 1.57 | 1.15 | 1.33 | 0.12 | 0.99 | 0.42 | 0.73 | 0.13 |
| 18a | 0.98 | 0.71 | 0.84 | 0.07 | 1.04 | 0.54 | 0.88 | 0.12 | 0.93 | 0.77 | 0.85 | 0.04 | 1.66 | 1.16 | 1.38 | 0.15 | 0.93 | 0.43 | 0.79 | 0.12 |
| mt10c1 | 0.92 | 0.68 | 0.81 | 0.07 | 0.88 | 0.61 | 0.77 | 0.06 | 0.98 | 0.73 | 0.84 | 0.06 | 1.36 | 0.93 | 1.14 | 0.13 | 0.75 | 0.34 | 0.51 | 0.13 |
| mt10cc | 0.95 | 0.63 | 0.83 | 0.08 | 0.93 | 0.67 | 0.78 | 0.07 | 0.88 | 0.74 | 0.81 | 0.03 | 1.37 | 0.88 | 1.19 | 0.12 | 0.67 | 0.33 | 0.47 | 0.09 |
| mt10x | 1.04 | 0.49 | 0.81 | 0.11 | 0.92 | 0.71 | 0.81 | 0.07 | 0.92 | 0.69 | 0.81 | 0.06 | 1.39 | 0.92 | 1.19 | 0.14 | 0.82 | 0.33 | 0.55 | 0.14 |
| mt10xx | 0.97 | 0.64 | 0.80 | 0.08 | 0.87 | 0.55 | 0.76 | 0.08 | 0.90 | 0.74 | 0.83 | 0.04 | 1.39 | 0.90 | 1.18 | 0.12 | 0.78 | 0.35 | 0.54 | 0.10 |
| mt10xxx | 0.97 | 0.64 | 0.83 | 0.08 | 0.90 | 0.70 | 0.82 | 0.06 | 0.99 | 0.64 | 0.82 | 0.08 | 1.44 | 0.92 | 1.16 | 0.14 | 0.65 | 0.36 | 0.50 | 0.10 |
| mt10xy | 0.94 | 0.74 | 0.84 | 0.05 | 0.90 | 0.71 | 0.81 | 0.06 | 0.94 | 0.71 | 0.80 | 0.06 | 1.38 | 0.94 | 1.19 | 0.13 | 0.83 | 0.34 | 0.54 | 0.14 |
| mt10xyz | 0.98 | 0.66 | 0.81 | 0.09 | 0.99 | 0.64 | 0.78 | 0.08 | 0.98 | 0.75 | 0.84 | 0.05 | 1.42 | 0.84 | 1.16 | 0.14 | 0.77 | 0.35 | 0.55 | 0.11 |
| setb4c9 | 0.95 | 0.71 | 0.87 | 0.05 | 0.94 | 0.70 | 0.84 | 0.07 | 0.93 | 0.76 | 0.87 | 0.04 | 1.40 | 0.75 | 1.13 | 0.15 | 0.84 | 0.32 | 0.49 | 0.13 |
| setb4cc | 1.05 | 0.70 | 0.85 | 0.07 | 0.88 | 0.62 | 0.80 | 0.07 | 0.94 | 0.77 | 0.85 | 0.06 | 1.36 | 0.86 | 1.12 | 0.14 | 0.66 | 0.35 | 0.49 | 0.09 |
| setb4x4 | 0.93 | 0.76 | 0.84 | 0.04 | 0.97 | 0.67 | 0.85 | 0.07 | 0.95 | 0.75 | 0.85 | 0.05 | 1.38 | 0.86 | 1.10 | 0.14 | 0.67 | 0.25 | 0.47 | 0.10 |
| setb4xx | 0.98 | 0.79 | 0.88 | 0.05 | 0.91 | 0.58 | 0.80 | 0.08 | 0.92 | 0.78 | 0.85 | 0.05 | 1.50 | 0.96 | 1.20 | 0.12 | 0.68 | 0.32 | 0.51 | 0.11 |
| setb4xxx | 0.95 | 0.76 | 0.87 | 0.04 | 0.91 | 0.71 | 0.82 | 0.06 | 0.94 | 0.81 | 0.86 | 0.03 | 1.32 | 0.91 | 1.13 | 0.11 | 0.72 | 0.30 | 0.49 | 0.14 |
| setb4xy | 0.91 | 0.72 | 0.84 | 0.05 | 0.91 | 0.68 | 0.82 | 0.06 | 0.91 | 0.72 | 0.84 | 0.04 | 1.37 | 0.93 | 1.11 | 0.12 | 0.70 | 0.33 | 0.52 | 0.11 |
| setb4xyz | 0.93 | 0.77 | 0.85 | 0.05 | 0.99 | 0.58 | 0.81 | 0.09 | 1.01 | 0.79 | 0.86 | 0.05 | 1.55 | 0.71 | 1.12 | 0.20 | 0.73 | 0.34 | 0.53 | 0.11 |
| seti5c12 | 1.00 | 0.81 | 0.88 | 0.05 | 0.93 | 0.71 | 0.86 | 0.05 | 0.94 | 0.80 | 0.87 | 0.04 | 1.27 | 0.88 | 1.08 | 0.10 | 0.76 | 0.34 | 0.51 | 0.11 |
| seti5cc | 0.97 | 0.77 | 0.86 | 0.04 | 0.93 | 0.73 | 0.86 | 0.06 | 0.94 | 0.70 | 0.85 | 0.06 | 1.26 | 0.71 | 0.99 | 0.15 | 0.57 | 0.30 | 0.43 | 0.09 |
| seti5x | 1.00 | 0.77 | 0.87 | 0.06 | 0.95 | 0.75 | 0.84 | 0.06 | 1.00 | 0.83 | 0.88 | 0.05 | 1.28 | 0.76 | 1.06 | 0.13 | 0.71 | 0.34 | 0.48 | 0.10 |
| seti5xx | 0.96 | 0.77 | 0.88 | 0.05 | 0.93 | 0.69 | 0.84 | 0.06 | 0.98 | 0.82 | 0.88 | 0.04 | 1.19 | 0.84 | 1.06 | 0.11 | 0.65 | 0.33 | 0.48 | 0.10 |
| seti5xxx | 0.98 | 0.83 | 0.88 | 0.04 | 0.97 | 0.69 | 0.83 | 0.08 | 1.01 | 0.78 | 0.87 | 0.05 | 1.21 | 0.77 | 1.02 | 0.12 | 0.73 | 0.37 | 0.50 | 0.09 |
| seti5xy | 1.07 | 0.71 | 0.87 | 0.07 | 0.93 | 0.65 | 0.83 | 0.07 | 0.95 | 0.78 | 0.88 | 0.04 | 1.40 | 0.81 | 1.12 | 0.14 | 0.65 | 0.29 | 0.45 | 0.11 |
| seti5xyz | 1.06 | 0.81 | 0.88 | 0.06 | 0.92 | 0.68 | 0.83 | 0.05 | 0.97 | 0.81 | 0.87 | 0.04 | 1.21 | 0.57 | 1.06 | 0.15 | 0.68 | 0.28 | 0.53 | 0.11 |
| Average | 0.98 | 0.75 | 0.87 | 0.06 | 0.96 | 0.68 | 0.84 | 0.07 | 0.95 | 0.77 | 0.86 | 0.05 | 1.41 | 0.89 | 1.16 | 0.13 | 0.77 | 0.35 | 0.55 | 0.11 |
| NB | 2 | 0 | 0 | – | 0 | 0 | 0 | – | 6 | 0 | 0 | – | 0 | 0 | 0 | – | 35 | 39 | 39 | – |

by using the above formula. In the comparisons, all algorithms are independently executed 21 times for all instances. To be fair, these compared algorithms adopt the same termination condition with $MaxFES = 500,000$. The average CPU time of algorithms are given in Fig. 15, where the trend of ENSHA is highlighted in shadow. The comparison results associated with γ , Δ and Ω are given in Tables 15 to 17. From Fig. 15, we can see that the average CPU time of ENSHA is similar to that of MOEA/D and SPEAII but is less than that of MOPSO and MOGLS for almost all the instances. In this sense, the overall efficiency of ENSHA is very acceptable compared with the state-of-the-art algorithms. It also can be seen from Tables 15 to 17 that almost all the performance metrics of ENSHA are much better than those of MOPSO, MOEA/D, MOGLS, and SPEAII. These results show that ENSHA is a more effective algorithm than the state-of-the-art algorithms and is significant for obtaining high-quality solutions of MHPMSP_MOSST/C. Moreover, to make a reliable assessment on ENSHA's performance, we also carry out the single factor ANOVA (at 95% CI) to evaluate the significant differences between ENSHA and the other algorithms. The results of ANOVA are presented in Table 18 where the p -values for γ , Δ and Ω are all less than 0.05. Therefore, there are significant differences among the above five compared algorithms at 95% CI. Furthermore, as shown in Table 19, we carry out a post hoc analysis based on the Tukey's multiple comparison tests (at 95% CI) to evaluate the significant differences of four pairwise algorithms. It should also be noted that the Tukey's tests

are performed based on the mean values of three performance metrics for 39 instances.

As shown in Table 19, in terms of Δ and Ω , the p -values for all pairwise comparisons are less than 0.05, which implies the significant differences between ENSHA and the state-of-the-art algorithms. As for γ , the p -value for "ENSHA vs. MOGLS" is less than 0.05, whereas the p -values for "ENSHA vs. MOEA/D" and "ENSHA vs. SPEAII" are larger than 0.05. Thus, we need to go further to analyze the convergence performance of ENSHA. The box plots considering the mean values of γ for 39 instances are given in Fig. 16. We can see that ENSHA, MOEA/D and SPEAII can achieve good performance with respect to γ , and these algorithms can obtain the preponderant median values trending to 0. Therefore, regarding γ , the p -values for "ENSHA vs. MOEA/D" and "ENSHA vs. SPEAII" are larger than 0.05, which indicates the good convergence performance of ENSHA. Besides, the p -values of Ω are less than 0.05 for "ENSHA vs. MOEA/D" and "ENSHA vs. SPEAII", so that ENSHA can obtain more efficient nondominated solutions than MOEA/D and SPEAII. Hence, ENSHA has better performance than the other four algorithms. Furthermore, we draw the box plots of MOPSO, MOEA/D, MOGLS, SPEAII and ENSHA in terms of instances 01a, mt10c1, setb4c9 and seti5cc, as shown in Fig. 17. These plots verify the robustness and effectiveness of our ENSHA.

Table 17Comparisons of MOPSO, MOEA/D, MOGLS, SPEAII and NSHA (Ω).

| Ins. | MOPSO | | | | MOEA/D | | | | MOGLS | | | | SPEAII | | | | ENSHA (this work) | | | |
|----------|-------|------|------|------|--------|-------------|-------------|------|-------|------|------|------|-------------|-------------|------|-------------|-------------------|-------------|-------------|------|
| | Worst | Best | Mean | SD | Worst | Best | Mean | SD | Worst | Best | Mean | SD | Worst | Best | Mean | SD | Worst | Best | Mean | SD |
| 01a | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.28 | 0.38 | 0.00 | 0.00 | 0.00 | 0.00 | 0.73 | 0.25 | 0.19 | 0.55 | 1.00 | 0.84 | 0.16 | |
| 02a | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.26 | 0.42 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.26 | 0.29 | 0.34 | 1.00 | 0.82 | 0.22 | |
| 03a | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.30 | 0.40 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.29 | 0.38 | 0.00 | 1.00 | 0.69 | 0.33 | |
| 04a | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.31 | 0.41 | 0.00 | 0.00 | 0.00 | 0.00 | 0.49 | 0.16 | 0.15 | 0.12 | 1.00 | 0.77 | 0.31 | |
| 05a | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.17 | 0.35 | 0.00 | 0.00 | 0.00 | 0.00 | 0.88 | 0.19 | 0.22 | 0.36 | 1.00 | 0.86 | 0.23 | |
| 06a | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.30 | 0.41 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.13 | 0.25 | 0.00 | 1.00 | 0.83 | 0.26 | |
| 07a | 0.00 | 0.00 | 0.00 | 0.00 | 0.96 | 0.26 | 0.37 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.32 | 0.30 | 0.00 | 1.00 | 0.77 | 0.30 | |
| 08a | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.47 | 0.40 | 0.00 | 0.00 | 0.00 | 0.00 | 0.68 | 0.27 | 0.19 | 0.07 | 1.00 | 0.70 | 0.30 | |
| 09a | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.37 | 0.43 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.48 | 0.40 | 0.00 | 1.00 | 0.52 | 0.42 | |
| 10a | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.43 | 0.38 | 0.00 | 0.00 | 0.00 | 0.00 | 0.71 | 0.28 | 0.24 | 0.24 | 1.00 | 0.75 | 0.26 | |
| 11a | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.53 | 0.41 | 0.00 | 0.00 | 0.00 | 0.00 | 0.03 | 0.93 | 0.40 | 0.29 | 0.00 | 1.00 | 0.54 | 0.36 |
| 12a | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.33 | 0.40 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.33 | 0.38 | 0.00 | 1.00 | 0.68 | 0.44 | |
| 13a | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.45 | 0.46 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.38 | 0.28 | 0.00 | 1.00 | 0.63 | 0.35 | |
| 14a | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.57 | 0.44 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.33 | 0.33 | 0.00 | 1.00 | 0.58 | 0.41 | |
| 15a | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.38 | 0.42 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.49 | 0.42 | 0.00 | 1.00 | 0.49 | 0.43 | |
| 16a | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.76 | 0.41 | 0.00 | 0.00 | 0.00 | 0.00 | 0.12 | 1.00 | 0.40 | 0.23 | 0.00 | 1.00 | 0.52 | 0.28 |
| 17a | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.28 | 0.42 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.42 | 0.32 | 0.00 | 1.00 | 0.62 | 0.46 | |
| 18a | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.38 | 0.44 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.61 | 0.37 | 0.00 | 1.00 | 0.53 | 0.46 | |
| mt10c1 | 0.00 | 0.35 | 0.03 | 0.09 | 0.00 | 0.89 | 0.22 | 0.32 | 0.00 | 0.00 | 0.00 | 0.00 | 0.97 | 0.14 | 0.23 | 0.25 | 1.00 | 0.79 | 0.24 | |
| mt10cc | 0.00 | 0.06 | 0.01 | 0.02 | 0.00 | 1.00 | 0.35 | 0.40 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.22 | 0.29 | 0.00 | 1.00 | 0.69 | 0.34 | |
| mt10x | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.31 | 0.38 | 0.00 | 0.00 | 0.00 | 0.00 | 0.83 | 0.25 | 0.27 | 0.15 | 1.00 | 0.71 | 0.30 | |
| mt10xx | 0.00 | 0.35 | 0.02 | 0.08 | 0.00 | 1.00 | 0.26 | 0.38 | 0.00 | 0.00 | 0.00 | 0.00 | 0.76 | 0.28 | 0.23 | 0.00 | 1.00 | 0.72 | 0.36 | |
| mt10xxx | 0.00 | 0.05 | 0.00 | 0.01 | 0.00 | 1.00 | 0.42 | 0.38 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.29 | 0.28 | 0.02 | 1.00 | 0.72 | 0.29 | |
| mt10xy | 0.00 | 0.36 | 0.02 | 0.08 | 0.00 | 1.00 | 0.26 | 0.37 | 0.00 | 0.00 | 0.00 | 0.00 | 0.89 | 0.18 | 0.23 | 0.23 | 1.00 | 0.80 | 0.24 | |
| mt10xyz | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.30 | 0.38 | 0.00 | 0.00 | 0.00 | 0.00 | 0.86 | 0.23 | 0.22 | 0.00 | 1.00 | 0.71 | 0.36 | |
| setb4c9 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.40 | 0.41 | 0.00 | 0.00 | 0.00 | 0.00 | 0.63 | 0.25 | 0.21 | 0.07 | 1.00 | 0.71 | 0.25 | |
| setb4cc | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.37 | 0.41 | 0.00 | 0.00 | 0.00 | 0.00 | 0.61 | 0.19 | 0.18 | 0.03 | 1.00 | 0.72 | 0.31 | |
| setb4x | 0.00 | 0.15 | 0.01 | 0.03 | 0.00 | 1.00 | 0.56 | 0.40 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.23 | 0.26 | 0.23 | 1.00 | 0.63 | 0.27 | |
| setb4xx | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.94 | 0.36 | 0.40 | 0.00 | 0.00 | 0.00 | 0.00 | 0.82 | 0.35 | 0.27 | 0.14 | 1.00 | 0.69 | 0.27 | |
| setb4xxx | 0.00 | 0.06 | 0.00 | 0.01 | 0.00 | 0.96 | 0.26 | 0.33 | 0.00 | 0.00 | 0.00 | 0.00 | 0.92 | 0.22 | 0.22 | 0.41 | 1.00 | 0.83 | 0.20 | |
| setb4xy | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.46 | 0.38 | 0.00 | 0.00 | 0.00 | 0.00 | 0.87 | 0.26 | 0.25 | 0.05 | 1.00 | 0.65 | 0.25 | |
| setb4xyz | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.94 | 0.38 | 0.35 | 0.00 | 0.00 | 0.00 | 0.00 | 0.89 | 0.27 | 0.25 | 0.28 | 1.00 | 0.73 | 0.24 | |
| seti5c12 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.43 | 0.45 | 0.00 | 0.00 | 0.00 | 0.00 | 0.88 | 0.32 | 0.28 | 0.25 | 1.00 | 0.70 | 0.30 | |
| seti5cc | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.51 | 0.42 | 0.00 | 0.00 | 0.00 | 0.00 | 0.50 | 0.18 | 0.17 | 0.21 | 1.00 | 0.71 | 0.25 | |
| seti5x | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.53 | 0.43 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.30 | 0.28 | 0.11 | 1.00 | 0.58 | 0.32 | |
| seti5xx | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.56 | 0.33 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.34 | 0.27 | 0.10 | 1.00 | 0.61 | 0.22 | |
| seti5xxx | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.43 | 0.38 | 0.00 | 0.00 | 0.00 | 0.00 | 0.61 | 0.24 | 0.18 | 0.13 | 1.00 | 0.69 | 0.30 | |
| seti5xy | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.46 | 0.41 | 0.00 | 0.00 | 0.00 | 0.00 | 0.57 | 0.19 | 0.19 | 0.08 | 1.00 | 0.71 | 0.29 | |
| seti5xyz | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.38 | 0.41 | 0.00 | 0.00 | 0.00 | 0.00 | 0.65 | 0.31 | 0.19 | 0.00 | 1.00 | 0.75 | 0.30 | |
| Average | 0.00 | 0.04 | 0.00 | 0.01 | 0.00 | 0.99 | 0.39 | 0.40 | 0.00 | 0.00 | 0.00 | 0.00 | 0.86 | 0.29 | 0.26 | 0.11 | 1.00 | 0.69 | 0.30 | |
| NB | 0 | 0 | 0 | – | 0 | 34 | 1 | – | 0 | 0 | 0 | – | 2 | 17 | 0 | – | 23 | 39 | 38 | – |

5.7. Discussions

In this subsection, we will conduct discussions on two issues: (1) The approximations of APF and PF; (2) The trends of the values of NB obtained by ENSHA and other compared algorithms. The aim of the former is to show the reasonability of constructing the APF from the union of K sets of nondominated solutions $S_1 \cup \dots \cup S_k$. And the latter is to systematically

assess the solution quality of the proposed ENSHA. Specially, all algorithms in the following discussions are classified into two groups: (1) Group 1: NSGAIUENSHA_VUENSHA; (2) Group 2: MOPSOUMOEA/DUMOGLSUSPEAIIUENSHA.

5.7.1. Approximations of APF and PF

The enumeration method is first executed to obtain the true PFs of two randomly generated small-sized instances, i.e., SI_1 and SI_2.⁴ And then we compare the PFs of SI_1 and SI_2 with the APFs identified from the unions of nondominated solutions of the corresponding algorithms. Noted that the size of SI_1 is $n \times m = 6 \times 3$ and $[opr_1, \dots, opr_6] = [2, 1, 2, 2, 1, 2]$, and the size of SI_2 is $n \times m = 5 \times 2$ and $[opr_1, \dots, opr_5] = [2, 2, 2, 2, 2]$. The solution

⁴ The small-sized instances SI_1 and SI_2 can be downloaded at <http://dx.doi.org/10.17632/7xvtjn3df.1>.

Table 19

Results of Tukey's multiple tests (state-of-the-art algorithms and ENSHA).

| Indicator | ENSHA vs. MOPSO | | | ENSHA vs. MOEA/D | | | ENSHA vs. MOGLS | | | ENSHA vs. SPEAII | | |
|---------------------|----------------------|--------------------|-------------------|---------------------|--------------------|-------------------|-----------------------|--------------------|-------------------|---------------------|--------------------|-------------------|
| | γ | Δ | Ω | γ | Δ | Ω | γ | Δ | Ω | γ | Δ | Ω |
| Difference of means | –531.81 | –0.31 | 0.69 | –28.18 | –0.29 | 0.31 | –1263.61 | –0.31 | 0.69 | –83.85 | –0.61 | 0.40 |
| SE of difference | 70.47 | 0.02 | 0.02 | 70.47 | 0.02 | 0.02 | 70.47 | 0.02 | 0.02 | 70.47 | 0.02 | 0.02 |
| 95% CI | –337.67 ^a | –0.27 ^a | 0.64 ^b | 165.96 ^a | –0.24 ^a | 0.26 ^b | –1069.47 ^a | –0.27 ^a | 0.64 ^b | 110.29 ^a | –0.56 ^a | 0.35 ^b |
| p-value | 0.00 | 0.00 | 0.00 | 0.99 | 0.00 | 0.000 | 0.00 | 0.00 | 0.76 | 0.00 | 0.00 | 0.00 |

^aUpper bound of 95% CI.

^bLower bound of 95% CI.

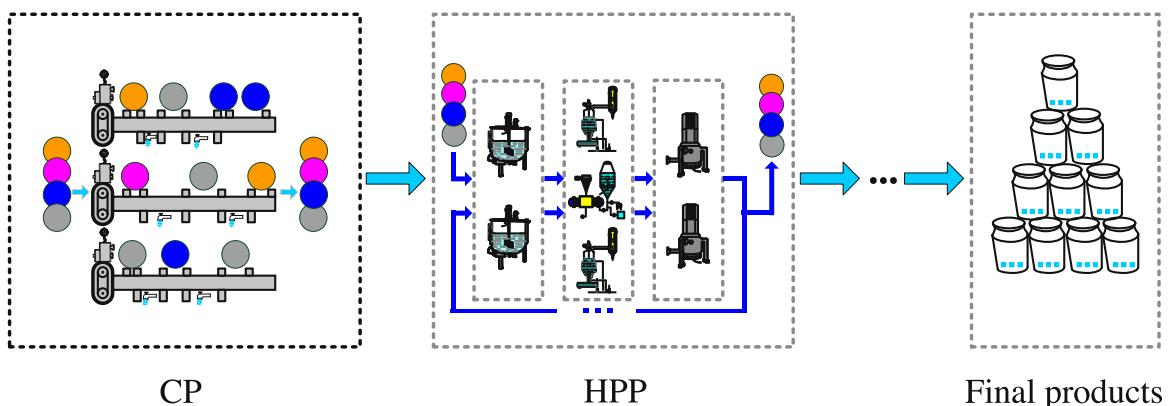


Fig. 21. Production process of the 6-Ethylchenodeoxycholic acid (C₂₆H₄₄O₄).

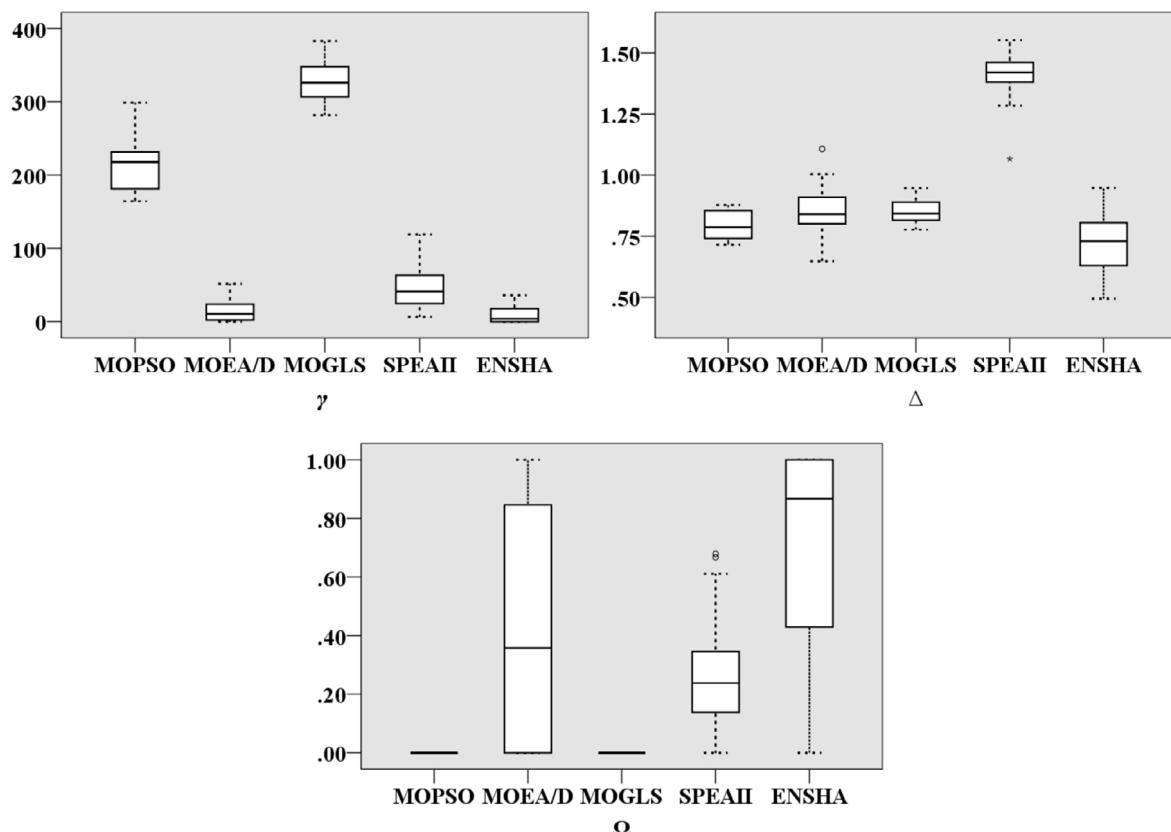


Fig. 22. Box plots for MOPSO, MOEA/D, MOGLS, SPEAII and ENSHA (case study).

spaces of SI_1 and SI_2 contain 226,800 and 113,400 solutions, respectively. For SI_1 and SI_2, we calculate the objectives for each solution and thereby obtain the true PFs⁵ of them. The numbers of solutions in SI_1'PF and SI_2'PF are 4 and 11, respectively.

Using the same stopping conditions and parameters in Sections 5.5 and 5.6, the APFs of SI_1 and SI_2 associated with Groups 1 and 2 can be obtained. The comparisons of APFs and PFs are shown in Fig. 18, which demonstrate that the APF of each group for each instance is a good approximation of the true PF for all instances. Therefore, it is reasonable to use APF for calculating the performance metrics of adopted algorithms.

5.7.2. Trends of NB regarding Group 1 and Group 2

Based on the comparison results in Sections 5.5 and 5.6, we summarize the trends of the values of NB with regard to Group 1 and Group 2 in Figs. 19 and 20, respectively. We can see that, in terms of *Worst*, *Best* and *Mean*, the values of NB obtained by ENSHA are much better than those of the other algorithms. To sum up, our ENSHA is capable of good performance for solving MOFJSSP_SDST/C.

5.8. Real-life case study

A real-life case study is provided to verify the application values of ENSHA. In this case study, we focus on the production process of the 6-Ethylchenodeoxycholic acid (C₂₆H₄₄O₄) for a cooperative pharmaceutical enterprise in Kunming, Yunnan,

⁵ The PDFs of SI_1 and SI_2 can be downloaded at <https://data.mendeley.com/datasets/c74ymmgvkp/draft?a=488168aa-20ee-460e-ad68-489c5b2f21ef>.

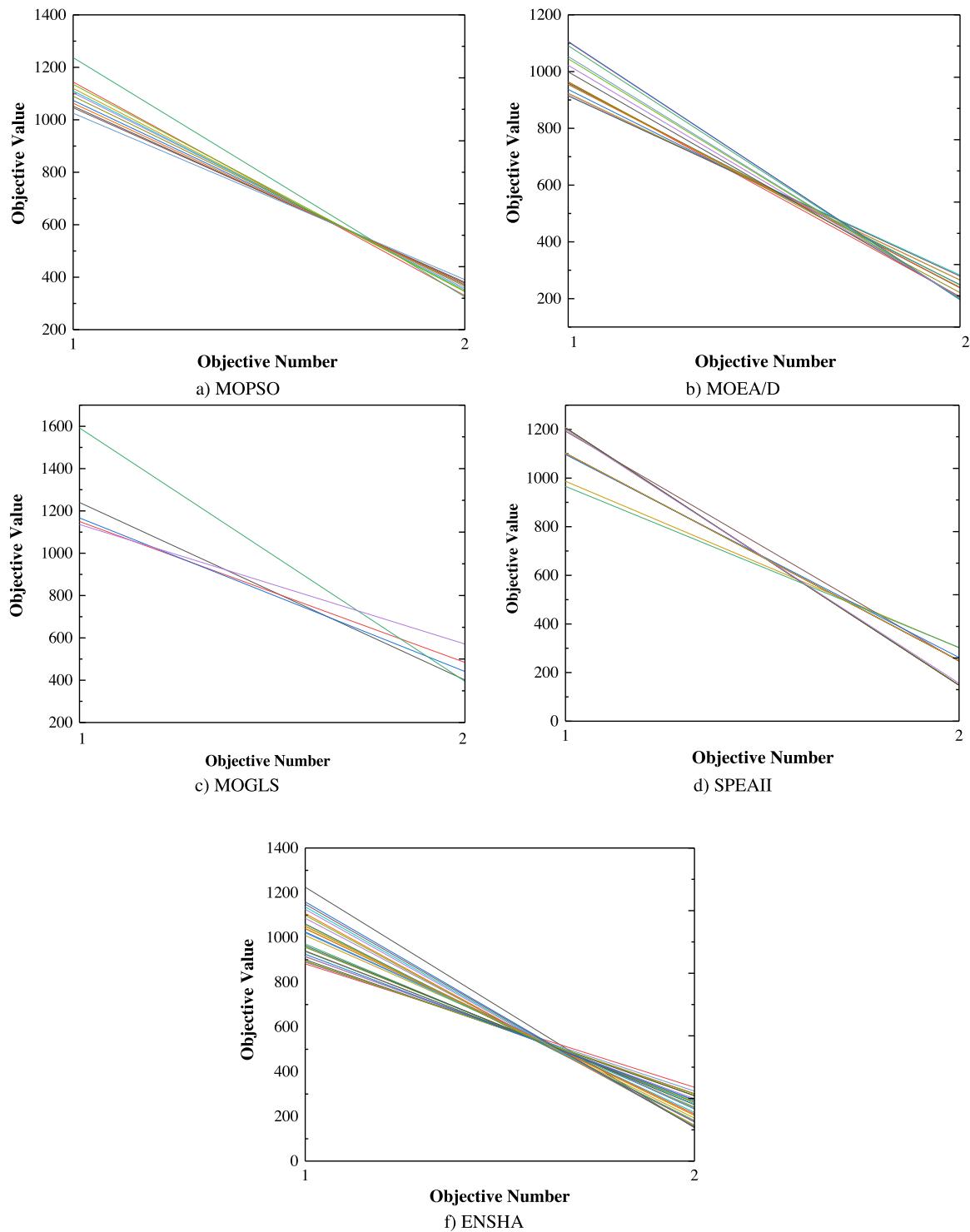


Fig. 23. Value paths for MOPSO, MOEA/D, MOGLS, SPEAII and ENSHA (case study).

Table 20

Comparison results based on case study.

| Metric | MOPSO | | | | MOEA/D | | | | MOGLS | | | | SPEAII | | | | ENSHA (this work) | | | |
|----------|--------|--------|--------|-------|--------|------|-------|-------|--------|--------|--------|-------|--------|------|-------|-------|-------------------|------|-------|-------|
| | Worst | Best | Mean | SD | Worst | Best | Mean | SD | Worst | Best | Mean | SD | Worst | Best | Mean | SD | Worst | Best | Mean | SD |
| γ | 298.64 | 164.20 | 214.20 | 38.39 | 51.68 | 0.00 | 15.37 | 15.28 | 382.90 | 281.83 | 327.64 | 28.68 | 119.09 | 6.37 | 44.24 | 27.40 | 35.97 | 0.00 | 11.13 | 15.45 |
| Δ | 0.88 | 0.72 | 0.79 | 0.06 | 1.11 | 0.65 | 0.85 | 0.11 | 0.95 | 0.78 | 0.85 | 0.05 | 1.55 | 1.07 | 1.41 | 0.11 | 0.95 | 0.49 | 0.72 | 0.12 |
| Ω | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.47 | 0.40 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.68 | 0.27 | 0.21 | 0.00 | 1.00 | 0.69 | 0.34 |
| CPU (s) | 135.65 | | | | 106.58 | | | | 146.06 | | | | 139.31 | | | | 131.48 | | | |

China. In the production process of C26H44O4 (see Fig. 21), the CP, which is utilized for purifying the crude materials, can provide the materials with the satisfied purity for its downstream process, i.e., the high-purity preparation (HPP). Based on our survey, the CP is a bottleneck for improving the production efficiency and the quality of final products. The CP for C26H44O4 is a typical kind of MOFJSSP_SDST/C, since the optimization of the makespan and TSC is significant for the overall benefits.

However, in this cooperative enterprise, the decision making of CP is mainly conducted by manual operations since its launch in 2016. Indeed, it is an intractable work for schedulers due to the expensive labor consume. For example, for a practical case with $n \times m = 30 \times 5$, it will take about one hour for making the final scheduling scheme. In addition, manual operations may encounter difficulties for multi-objective optimization problem, because of the conflicting features of multiple objectives. Hence, it is both necessary and practical to introduce effective MOEAs in the decision making.

The factory data⁶ of CP from the cooperative enterprise are collected, where there are 30 batches (i.e., products) of crude materials for the C26H44O4 and 5 crystallizers (i.e., machines) with different capacities and working modes. Using the same termination condition and parameters mentioned in Section 5.6, we independently run ENSHA as well as MOPSO, MOEA/D, MOGLS and SPEAII 21 times, and the comparison results are reported in Table 20. Moreover, we draw the box plots of ENSHA and the other four algorithms in Fig. 22. Comparison results and box plots demonstrate that ENSHA is an effective and robust solution method for practical applications. Note that the average CPU time of ENSHA is 131.48 s which is acceptable in real-life environments.

In addition, to show the distributions of obtained nondominated solutions, we draw the multiple objectives on the value paths for MOPSO, MOEA/D, MOGLS, SPEAII and ENSHA in Fig. 23. It is clear from these paths that ENSHA is able to obtain better diversity and convergence of nondominated solutions than the other four algorithms.

Moreover, we have invited three experienced schedulers to independently make the schemes for this case in thirty minutes.⁷ The results of ENSHA and three schedulers are illustrated in Fig. 24. From Fig. 24, it is clear that ENSHA can obtain a number of nondominated solutions with higher diversity and convergence in terms of both the makespan and TSC. Hence, the overall solution quality of ENSHA is obviously better than those manual results. In practical applications, the managers can flexibly select a desirable nondominated solution to serve as their final scheduling scheme. For example, to improve production efficiency, the managers can select a nondominated solution having relative smaller makespan (corresponding to relative larger TSC). On the contrary, to pursue more profit, they can select a nondominated solution with smaller TSC.

From the test results of Sections 5.5 to 5.8, it can be drawn that MCEDA has a powerful search engine for tackling the multi-objective scheduling problems. Unlike the traditional MOEAs (e.g., the above compared algorithms), ENSHA utilizes an EDA-based machine learning mechanism to reserve promising patterns of excellent individuals, which is helpful for effectively guiding the search to promising regions and avoiding the promising patterns being destroyed or inappropriately fused [68,69]. Hence, ENSHA's novel bi-population-oriented evolutionary framework combining both the genetic operators and an EDA-based machine learning mechanism can effectively perform the search in different promising regions of solution space. This is the main reason why ENSHA outperforms those famous MOEAs.

⁶ The utilized factory data can be downloaded at <http://dx.doi.org/10.17632/ynvz3vxttj.1>.

⁷ The result of three schedulers can be downloaded at <https://data.mendeley.com/datasets/x8nvf3tfnn/draft?a=e2ec3ad8-b293-42fa-96fd-9dd29c3cd796>.

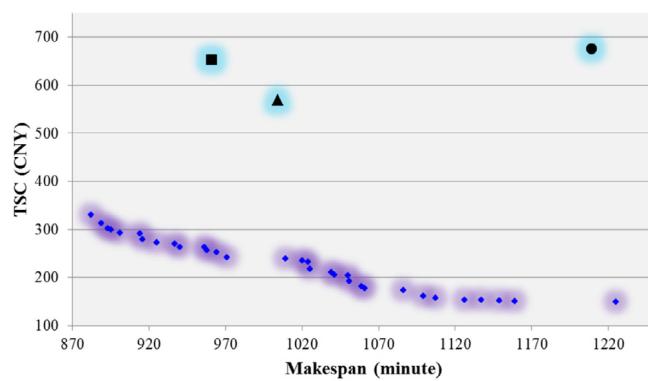


Fig. 24. Results of ENSHA and three schedulers.

6. Conclusions and future work

This paper presented an elitist nondominated sorting hybrid algorithm (ENSHA) for solving the multi-objective flexible job-shop scheduling problem with sequence-dependent setup times/costs (MOFJSSP_SDST/C). The objectives of the problem are the minimization of the maximal completion time (i.e., makespan) and the total setup costs (TSC). This is the first report on the application of multi-objective evolutionary algorithm (MOEA) for MOFJSSP with both efficiency-focused and economic focused objectives. Firstly, on the basis of the problem-dependent features of MOFJSSP_SDST/C, the solution representation and three specific job assignment rules were designed. Secondly, a novel and effective bi-population framework was developed to find promising regions in the solution space, in which the main population (MP) was generated by means of the genetic operations with the elitist nondominated sorting scheme. The auxiliary population (AP) was obtained through using a specific EDA-based machine learning mechanism and a PMX-based method. The next generation MP containing excellent individuals in both MP and AP was then achieved via a well-designed cooperation-based refinement strategy. Thirdly, the simulation results and comparisons demonstrated that ENSHA outperforms the state-of-the-art algorithms including NSGAII, MOEA/D, MOGLS and SPEAII. Finally, a real-life case study verified the application values of ENSHA.

In the future, our work is to develop general scheduling software based on ENSHA for pharmaceutical enterprises and investigate the applications of the presented method to the multi-objective scheduling problems with energy consumption.

Acknowledgments

This research was partially supported by National Science Foundation of China (No. 51665025, No. 71601180, and No. 60904081), Applied Basic Research Key Project of Yunnan, China, and National Key Research and Development Program of China (No. 2017YFB120700).

References

- [1] J. Adams, E. Balas, D. Zawack, The shifting bottleneck procedure for job shop scheduling, *Manage. Sci.* 34 (3) (1988) 391–401.
- [2] H. Karimi, S.H.A. Rahmati, M. Zandieh, An efficient knowledge-based algorithm for the flexible job shop scheduling problem, *Knowl.-Based Syst.* 36 (6) (2012) 236–244.
- [3] J. Lin, A hybrid biogeography-based optimization for the fuzzy flexible job shop scheduling problem, *Knowl.-Based Syst.* 78 (1) (2015) 59–74.
- [4] C. Gutiérrez, I. García-Magariño, Modular design of a hybrid genetic algorithm for a flexible job-shop scheduling problem, *Knowl.-Based Syst.* 24 (1) (2011) 102–112.

- [5] S. Gangan, S.K. Khator, A.J.G. Babu, Multiobjective decision making approach for determining alternate routing in a flexible manufacturing system, *Comput. Ind. Eng.* 13 (1–4) (1987) 112–117.
- [6] S.M. Lee, H.J. Jung, A multi-objective production planning model in a flexible manufacturing environment, *Int. J. Prod. Res.* 27 (11) (1989) 1981–1992.
- [7] N. Shahsavari-Pour, B. Ghasemishabankareh, A novel hybrid meta-heuristic algorithm for solving multi objective flexible job shop scheduling, *J. Manuf. Syst.* 32 (4) (2013) 771–780.
- [8] L. Wang, G. Zhou, Y. Xu, et al., An enhanced Pareto-based artificial bee colony algorithm for the multi-objective flexible job-shop scheduling, *Int. J. Adv. Manuf. Technol.* 60 (9–12) (2012) 1111–1123.
- [9] L. Wang, S.Y. Wang, M. Liu, A Pareto-based estimation of distribution algorithm for the multi-objective flexible job-shop scheduling problem, *Int. J. Prod. Res.* 51 (12) (2013) 3574–3592.
- [10] K.Z. Gao, P.N. Suganthan, Q.K. Pan, et al., Discrete harmony search algorithm for flexible job shop scheduling problem with multiple objectives, *J. Intell. Manuf.* 27 (2) (2016) 363–374.
- [11] W. Xia, Z. Wu, An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems, *Comput. Ind. Eng.* 48 (2005) 112–117.
- [12] G. Zhang, X. Shao, P. Li, et al., An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem, *Comput. Ind. Eng.* 56 (4) (2009) 1309–1318.
- [13] M.E.T. Araghi, F. Jalai, M. Rabiee, Incorporating learning effect and deterioration for solving a SDST flexible job-shop scheduling problem with a hybrid meta-heuristic approach, *Int. J. Comput. Integr. Manuf.* 27 (8) (2014) 733–746.
- [14] A. Bagheri, M. Zandieh, Bi-criteria flexible job-shop scheduling with sequence-dependent setup times-Variable neighborhood search approach, *J. Manuf. Syst.* 30 (1) (2011) 8–15.
- [15] L. Shen, S. Dauzère-Pérès, J.S. Neufeld, Solving the flexible job shop scheduling problem with sequence-dependent setup times, *European J. Oper. Res.* 265 (2018) 503–516.
- [16] M. Mousakhani, Sequence-dependent setup time flexible job shop scheduling problem to minimise total tardiness, *Int. J. Prod. Res.* 51 (12) (2013) 3476–3487.
- [17] A. Rossi, Flexible job shop scheduling with sequence-dependent setup and transportation times by ant colony with reinforced pheromone relationships, *Int. J. Prod. Econ.* 153 (4) (2014) 253–267.
- [18] C. Özgüven, Y. Yavuz, L. Özbaşır, Mixed integer goal programming models for the flexible job-shop scheduling problems with separable and non-separable sequence dependent setup times, *Appl. Math. Model.* 36 (2) (2012) 846–858.
- [19] M. Laumanns, SPEA2: Improving the strength pareto evolutionary algorithm. Technical Report, Gloriastrasse 35, 2001.
- [20] K. Deb, A. Pratap, S. Agarwal, et al., A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Trans. Evol. Comput.* 6 (2) (2002) 182–197.
- [21] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, *IEEE Trans. Evol. Comput.* 11 (6) (2007) 712–731.
- [22] A. Jaszkiewicz, On the performance of multiple-objective genetic local search on the 0/1 knapsack problem - a comparative experiment, *IEEE Trans. Evol. Comput.* 6 (4) (2002) 402–412.
- [23] X. Cai, Y. Li, Z. Fan, et al., An external archive guided multiobjective evolutionary algorithm based on decomposition for combinatorial optimization, *IEEE Trans. Evol. Comput.* 19 (4) (2015) 508–523.
- [24] E. Balas, Machine sequencing via disjunctive graphs: an implicit enumeration algorithm, *Oper. Res.* 17 (6) (1969) 941–957.
- [25] R. Zhang, C. Wu, A simulated annealing algorithm based on block properties for the job shop scheduling problem with total weighted tardiness objective, *Comput. Oper. Res.* 38 (5) (2011) 854–867.
- [26] J.Q. Li, Q.K. Pan, Y.C. Liang, An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems, *Comput. Ind. Eng.* 59 (4) (2010) 647–662.
- [27] B. Qian, L. Wang, D.X. Huang, et al., Scheduling multi-objective job shops using a memetic algorithm based on differential evolution, *Int. J. Adv. Manuf. Technol.* 35 (9–10) (2008) 1014–1027.
- [28] I. Kacem, S. Hammadi, P. Borne, Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems, *IEEE Trans. Syst. Man Cybern. C* 32 (1) (2002) 1–13.
- [29] M.A. Fernández Pérez, F.M.P. Raupp, A Newton-based heuristic algorithm for multi-objective flexible job-shop scheduling problem, *J. Intell. Manuf.* 27 (2) (2016) 409–416.
- [30] W.J. Xia, Z.M. Wu, An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems, *Comput. Ind. Eng.* 48 (2005) 409–425.
- [31] X. Shao, W. Liu, Q. Liu, et al., Hybrid discrete particle swarm optimization for multi-objective flexible job-shop scheduling problem, *Int. J. Adv. Manuf. Technol.* 67 (2013) 2885–2901.
- [32] G. Mosleh, M. Mahnam, A pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search, *Int. J. Prod. Econ.* 129 (1) (2011) 14–22.
- [33] J.C.H. Pan, J.S. Chen, A comparative study of schedule-generation procedures for the reentrant shops scheduling problem, *Int. J. Ind. Eng. Theory Appl. Pract.* 11 (4) (2004) 313–321.
- [34] Y. He, C.W. Hui, Automatic rule combination approach for single-stage process scheduling problems, *AIChE J.* 53 (8) (2007) 2026–2047.
- [35] Y. He, C.W. Hui, A rule-based genetic algorithm for the scheduling of single-stage multi-product batch plants with parallel units, *Comput. Chem. Eng.* 32 (12) (2008) 3067–3083.
- [36] Q.K. Pan, L. Wang, B. Qian, A novel differential evolution algorithm for bi-criteria no-wait flow shop scheduling problems, *Comput. Oper. Res.* 36 (8) (2009) 2498–2511.
- [37] B. Qian, L. Wang, D.X. Huang, et al., An effective hybrid DE-based algorithm for flow shop scheduling with limited buffers, *Comput. Oper. Res.* 35 (9) (2009) 2791–2806.
- [38] H. Aytug, M. Koufa, F.E. Vergara, Use of genetic algorithms to solve production and operations management problems: A review, *Int. J. Prod. Res.* 41 (17) (2003) 3955–4009.
- [39] C. Srinivas, B.R. Reddy, K. Ramji, et al., Sensitivity analysis to determine the parameters of genetic algorithm for machine layout, *Procedia Mater. Sci.* 6 (2014) 866–876.
- [40] R. Murugeswari, S. Radhakrishnan, D. Devaraj, A multi-objective evolutionary algorithm based QoS routing in wireless mesh networks, *Appl. Soft Comput.* 40 (2016) 517–525.
- [41] W. Yong, Z. Jie, A. Kevin, et al., Collaboration and transportation resource sharing in multiple centers vehicle routing optimization with delivery and pickup, *Knowl.-Based Syst.* 160 (2018) 296–310.
- [42] P.W. Poon, J.N. Carter, Genetic algorithm crossover operators for ordering applications, *Comput. Oper. Res.* 22 (1) (1995) 135–147.
- [43] R. Abu-Zitar, A.M. Al-Fahednuseirat, Performance evaluation of genetic algorithms and evolutionary programming in optimization and machine learning, *J. Cybern.* 33 (3) (2002) 203–223.
- [44] D.E. Goldberg, Holl, J. H., Genetic algorithms and machine learning, *Mach. Learn.* 3 (2–3) (1988) 95–99.
- [45] J. Shapiro, Genetic algorithms in machine learning, *Lecture Notes in Comput. Sci.* 2049 (2001) 146–168.
- [46] C.Y. Lee, S. Piramuthu, Y.K. Tsai, Job shop scheduling with a genetic algorithm and machine learning, *Int. J. Prod. Res.* 35 (4) (1997) 1171–1191.
- [47] L. Wang, S. Wang, Y. Xu, et al., A bi-population based estimation of distribution algorithm for the flexible job-shop scheduling problem, *Comput. Ind. Eng.* 62 (4) (2012) 917–926.
- [48] J.N. Shen, L. Wang, S.Y. Wang, A bi-population EDA for solving the no-idle permutation flow-shop scheduling problem with the total tardiness criterion, *Knowl.-Based Syst.* 74 (2015) 167–175.
- [49] L. Wang, S. Wang, X. Zheng, A hybrid estimation of distribution algorithm for unrelated parallel machine scheduling with sequence-dependent setup times, *IEEE/CAA J. Autom. Sin.* 3 (3) (2016) 235–246.
- [50] S. Dauzère-Pérès, J. Paulli, An integrated approach for modeling and solving the general multiprocessor job-shop scheduling problem using tabu search, *Ann. Oper. Res.* 70 (1) (1997) 281–306.
- [51] J.B. Chambers, J.W. Barnes, Flexible job shop scheduling by Tabu search. Technical report series ORP 96-09. Department of Mechanical Engineering, University of Texas at Austin, 1996.
- [52] B. Qian, L. Wang, D.X. Huang, et al., Multi-objective no-wait flow-shop scheduling with a memetic algorithm based on differential evolution, *Soft Comput.* 13 (8–9) (2009) 847–869.
- [53] B. Qian, L. Wang, D.X. Huang, et al., An effective hybrid DE-based algorithm for flow shop scheduling with limited buffers, *Comput. Oper. Res.* 36 (2009) 209–233.
- [54] J. Shen, L. Wang, H. Zheng, A modified teaching-learning-based optimisation algorithm for bi-objective re-entrant hybrid flowshop scheduling, *Int. J. Prod. Res.* 54 (12) (2017) 3622–3639.
- [55] H.M. Cho, S.J. Bae, J. Kim, et al., Bi-objective scheduling for reentrant hybrid flow shop using Pareto genetic algorithm, *Comput. Ind. Eng.* 61 (3) (2011) 529–541.
- [56] D.A.V. Veldhuizen, G.B. Lamont, On measuring multiobjective evolutionary algorithm performance, in: Congress on Evolutionary Computation, Vol. 1, IEEE Service Center, Piscataway, New Jersey, 2000, pp. 204–211, (7).
- [57] C.A.C. Coello, Cortés Nareli Cruz, Solving multiobjective optimization problems using an artificial immune system, *Genet. Program. Evolvable Mach.* 6 (2) (2005) 163–190.
- [58] D.A.V. Veldhuizen, G.B. Lamont, Multiobjective evolutionary algorithm research: A history and analysis. Dept. Elec. Comput. Eng. Graduate School of Eng. Air Force Inst. Technol. Wright-Patterson AFB, OH, Tech. Rep. TR-98-03, 1998.
- [59] J.R. Schott, Fault Tolerant Design using Single and Multicriteria Genetic Algorithm Optimization (Master's thesis), Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, 1995.

- [60] J. Sadeghi, S.T.A. Niaki, Two parameter tuned multi-objective evolutionary algorithms for a bi-objective vendor managed inventory model with trapezoidal fuzzy demand, *Appl. Soft Comput.* **J.** 30 (C) (2015) 567–576.
- [61] J. Sadeghi, S. Sadeghi, S.T.A. Niaki, A hybrid vendor managed inventory and redundancy allocation optimization problem in supply chain management: An NSGA-II with tuned parameters, *Comput. Oper. Res.* **41** (1) (2014) 53–64.
- [62] S. Khalilpourazari, S. Khalilpourazary, A Robust Stochastic Fractal Search approach for optimization of the surface grinding process, *Swarm Evol. Comput.* **38** (2017) 173–186.
- [63] S. Khalilpourazari, S.H.R. Pasandideh, A. Ghodratnama, Robust possibilistic programming for multi-item EOQ model with defective supply batches: Whale optimization and water cycle algorithms, *Neural Comput. Appl.* (2018) (Publish on line).
- [64] D.C. Montgomery, *Design and Analysis of Experiments*, Wiley, Hoboken, NJ, 2005.
- [65] S. Khalilpourazari, S.H.R. Pasandideh, Multi-objective optimization of multi-item EOQ model with partial backordering and defective batches and stochastic constraints using MOWCA and MOGWO, *Oper. Res.* (2018) (Publish on line).
- [66] S. Khalilpourazari, S.H. Reza Pasandide, S.T. Akhavan Niaki, Optimization of multi-product economic production quantity model with partial back-ordering and physical constraints: SQP, SFS, SA, and WCA, *Appl. Soft Comput.* **49** (12) (2016) 770–791.
- [67] C.A.C. Coello, G.T. Pulido, M.S. Lechuga, Handling multiple objectives with particle swarm optimization, *IEEE Trans. Evol. Comput.* **8** (3) (2004) 256–279.
- [68] P. Larrañaga, J.A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, Springer Science & Business Media, 2001.
- [69] J. Ceberio, E. Irurozki, A. Mendiburu, J.A. Lozano, A review on estimation of distribution algorithms in permutation-based combinatorial optimization problems, *Prog. Artif. Intell.* **1** (1) (2012) 103–117.