Multi-Objective Casting Production Scheduling Problem by a Neighborhood Structure Enhanced Discrete NSGA-II: An Application from Real-World Workshop

Weihua Tan（✉ tanwh@hnu.edu.cn ）

Hunan University   https://orcid.org/0000-0003-0920-1522

Xiaofang Yuan

Hunan University   https://orcid.org/0000-0001-7280-7207

Yuhui Yang

Hunan University

Lianghong Wu

Hunan University of Science and Technology

Multi-objective casting production scheduling problem by a neighborhood structure enhanced discrete NSGA-II: an application from real-world workshop

Weihua Tan [1] · **Xiaofang Yuan** [1,*] · **Yuhui Yang** [1] · **Lianghong Wu**[2]

Abstract Casting production scheduling problem (CPSP) has attracted increasing research attention in recent years to facilitate the profits, efficiency, and environment issues of casting industry. Casting is often characterized by the properties of intensive energy consumption and complex process routes, which motivate the in-depth investigation on construction of practical multi-objective scheduling models and development of effective algorithms. In this paper, for the first time, the multi-objective casting production scheduling problem (MOCPSP) is constructed to simultaneously minimize objectives of defective rate, makespan, and total energy consumption. Moreover, a neighborhood structure enhanced discrete NSGA-II (N-NSGA-II) is designed to better cope with the proposed MOCPSP. In the N-NSGA-II, the advantage of selection mechanism of NSGA-II is fully utilized for selecting non-dominate solution, three neighborhood structures are elaborately designed to strengthen the ability of the local search, and a novel solution generating approach is proposed to increase the diversity of solutions for global search. Finally, a real-world case is illustrated to evaluate the performance of the N-NSGA-II. Computational results show that the proposed N-NSGA-II obtains a wider range of non-dominated solutions with better quality compared to other well-known multi-objective algorithms.

Keywords Multi-objective casting production scheduling problem · Defective rate · Variable processing speed · Non-dominated sorting genetic algorithm · Neighborhood structure

1 Introduction

Casting is one of machine manufacture industry foundation crafts, which is closely tied with the development of equipment manufacturing industries like automobile, machine tool, aviation, aerospace, and national defense. In the modern casting industry, multi-variety and small batch production mode is becoming popular and necessary [6] in order to meet the growing demand for customization products. Meanwhile, due to the energy-intensive [14] nature of casting industry, efficient and green casting manufacturing earns an increasing concern for the last decades [28]. As we known, efficient scheduling can serves as a powerful tool to improve the production cost, manufacturing cycle and environment issue [30]. Therefore, the casting production scheduling problem (CPSP) under the shop floor environment for practical application attracts increasing attention from both the industry and the research community.

Two main specific attributes of casting process need to be considered when dealing with CPSP. Firstly, casting processing is a typical energy-intensive and high-pollution manufacturing [36]. Some operations such as melting and thermal treatments will involve great amount of time, energy consumption and pollute emission. Therefore, multi objectives need to be simultaneously addressed in CPSP. Secondly, the casting process is extremely complex [30]. On the one hand, the casting products generally have long process routes and the routes are various for different casting products in the mixed production lines. On the other hand, large number of multi-functional machines are involved in the casting workshops. These characteristics lead to enormous feasible scheduling schemes of CPSP, necessitating the development of powerful optimization algorithms.

The CPSP can be viewed as the extension of flexible job-shop scheduling problem(FJSP) [30]. In pioneer works, objectives of makespan and total energy consumption are widely considered in FJSP [37]. Mokhtari et.al [25] develop joint scheduling of the production and the maintenance operation in the flexible job shop environment with the objectives to minimize the makespan, total energy consumption and maximize the total availability of system. Wu et.al [40] consider deterioration effect into FJSP, in which the actual processing time of the operation is variable under the influence of the deterioration effect, and the FJSP-DEEC model is proposed to optimize the makespan and the total energy consumption. Moreover, Dai et.al [8] address an energy-efficient flexible job shop scheduling problem with transportation constraint and formulate a multi-objective optimization model to minimize the objectives of makespan and the total energy consumption. As demonstrated above, the performance of the

✉Xiaofang Yuan
E-mail: yuanxiaofang@hnu.edu.cn

1 College of Electrical and Information Engineering, Hunan University, Changsha, Hunan, 410082, China
2 College of Information and Electrical Engineering, Hunan University of Science and Technology, Xiangtan 411201, China

shop floor has been significantly promoted by optimizing makespan and the total energy consumption. Besides, defective rate is another vital issue for casting production. In the practical workshop, the performance of machine tool deteriorates gradually with the production process [42], resulting in the increase in defective rate. Due to the attributes of casting process such as long operation-chain, low defective rate of each operation in the route is stringent required. Moreover, because of the energy-intensive property of casting, wide-employed maintenance strategies [35,31] which require frequent shutting down of machines is costly and insufficient for controlling the defective rate of casting products. Unfortunately, to the best of our knowledge, no existing research has addressed the objective of defective rate in CPSP. To this end, a multi-objective casting production scheduling problem (MOCPSP) is constructed considering the objectives of defective rate, makespan, and total energy consumption simultaneously in this paper.

The proposed MOCPSP is an NP-hard problem. With the nature of extreme complexity, exact approach is unable to find optimal solutions in a reasonable time when the problem size is large [11]. Various multi-objective heuristic techniques have been proposed in the last decades. Recent researches [4,12] employ different hybrid algorithms to solve multi-objective FJSP by combining some of those heuristic techniques and achieved remarkable results. Shen et.al [33] develop a modified multi-objective evolutionary algorithm to solve the MOFJSP. Zhang et.al [45] propose a hybrid optimization algorithm for MOFJSP combining a particle swarm optimization algorithm and a tabu search algorithm. Li et.al [18,19] develop two hybrid algorithms by integrating tabu search algorithm with variable neighborhood search strategy. Moslehi et.al [26] report a hybrid particle swarm optimization with a special designed local search strategy. Yuan et.al [43] propose new memetic algorithms by incorporating a novel local search algorithm into the NSGA-II. Shao et.al [32] develop a hybrid discrete particle swarm optimization combined with simulated annealing algorithm to solve the MOFJSP with objectives of minimizing makespan, maximal machine workload and total workload of machines. To better cope with the proposed MOCPSP, an efficient hybrid heuristic algorithm with enhanced performance is designed in this paper.

The main contributions of this paper can be outlined as follows:

(i) To narrow the gap between theoretical research and real-world casting production scheduling application, a novel MOCPSP model is developed considering defective rate, makespan and total energy consumption simultaneously, which is the first study in this area.

(ii) Facing the extreme complexity of the proposed MOCPSP, a neighborhood structure enhanced discrete NSGA-II (N-NSGA-II) is developed. In the N-NSGA-II, three neighborhood structures are elaborated designed to strengthen the ability of the local search and a novel solution generating approach is proposed to increase the diversity of solution for global search.

The remainder of this paper is organized as follows. Section 2 describes the MOCPSP in detail and gives the formulation of the MOCPSP. Section 3 illustrates the proposed N-NSGA-II algorithm for solving the MOCPSP. Section 4 presents the computational results and analysis of a real-world casting case by the N-NSGA-II and other well-known algorithms. The conclusion and future work are presented in Section 5.

2 Description and formulation of the MOCPSP

In this section, the overall description of the proposed MOCPSP is given firstly. After that, the models of the three objectives are formulated respectively. The formulation of the MOCPSP model is presented at last.

2.1 Problem description

In the MOCPSP, the objectives of defective rate, makespan, and total energy consumption are addressed as the indicators of production quality, efficiency, and environmental issues. In most practical workshop, the processing speed of machine is changeable [3]. For example, the processing speed of computer numerically controlled manufacturing machine for an operation is adjusted by controlling cutting speed or the feed rate [1]. Pioneer researches [15,21,38,23] have proved that considering variable processing speed into production scheduling is practical for resource-saving and time-efficient attempt. To address real-world casting scenarios, the impact of variable processing speed for each operation is non-negligible. Research on scheduling problem with variable processing speed can be classified into two categories [22]: (1) continuous processing speed varying within a limited range;(2) discrete processing speed selecting from a predefined speed set. The discrete processing speed is suitable when considering practical applications [22] and is adapted in this paper.

Thus, the MOCPSP consists of three sub-problems: routing problem, sequencing problem and speed determination problem. Former two are origin from classical FJSP [41]. Routing problem is to assign each operation to a machine out of a set of alternative machines. Sequencing problem is to sequence operation the assigned operations on all machines under the constraint. Speed determination problem is to select a processing speed of the assigned machine for each operation.

Therefore, the MOCPSP can be defined as follows:

There is a set of n jobs $\boldsymbol{J} = \{J_1, J_2, ..., J_n\}$ and a set of m machines $\boldsymbol{M} = \{M_1, M_2, ..., M_m\}$, and M_k ($k \in \{1, 2, ..., m\}$) has p_k alternative speed ratios $\boldsymbol{\alpha} = \left\{\alpha_1^k, \alpha_2^k, ..., \alpha_{p_k}^k\right\}$ (the speed ratios increase gradually with the growth of index). J_i ($i \in \{1, 2, ..., n\}$) contains the number of s_i operations and each operation of J_i follows the precedence relation. Operation O_{ij} needs to be processed on one machine out of an alternative machine set S_{ij} ($S_{ij} \subseteq \{1, 2, .., M\}$) containing at least one machine. And at least one operation can be processed on more than one machine.

Notations of the proposed MOCPSP are listed in Table 1.

To simplify and clarify the proposed problem, some assumptions are predefined.

Table 1 Notations of the proposed MOCPSP

Notation	Descriptions
n	the number of job
s_i	the number of the operations for job i
m	the number of machine
i, i'	the index for job, $i, i' = 1, 2, .., n$
j, j'	the index for operation, $j, j' = 1, 2, .., s_i$
k	the index for machine, $k = 1, 2, .., m$
O_{ij}	the ith operation of job i
α_z^k	the zth speed ratio of M_k, $z \in \{1, 2, ..., p_k\}$
α_1^k	the reference speed ratio
$t_{ij,k}^{refer}$	the reference processing time of O_{ij} processed on M_k corresponding to α_1^k
$t_{ij,k}$	the processing time of O_{ij} processed on M_k
DR_i	the defective rate of job i
dr_k	the reference defective rate of M_k corresponding to α_1^k
drO_{ij}^k	the defective rate of O_{ij} processed on M_k
PP_z^k	the processing power of M_k under the zth speed ratio
PI^k	the idle power of M_k
EP_{ij}	the processing energy consumption for O_{ij}
c_{ij}	the completion time of O_{ij}
$X_{ij,k}$	if the operation O_{ij} is processed on M_k, $X_{ij,k} = 1$; otherwise, $X_{ij,k}=0$
$Y_{ij,i'j'}$	if the operation O_{ij} is the predecessor of $O_{i'j'}$, $Y_{ij,i'j'}=1$; if the operation O_{ij} is the successor of $O_{i'j'}$, $Y_{ij,i'j'}=-1$
$Z_{ij,k}$	the speed ratio for O_{ij} on M_k, $Z_{ij,k} \in \{1, 2, ..., p_k\}$
$\overline{DR}$	the mean defective rate of n jobs
C_{max}	the completion time of all jobs
TEC	the total energy consumption

– All machines and raw materials are available at time 0;
– No setup operation for machine happens during production;
– Each operation is processed without interruption;
– The switch between processing mode and idle mode requires no time;
– Conversion time is negligible.

2.2 Models of the three objectives

The models of defective rate, makespan and total energy consumption are described as follows.

2.2.1 The model of defective rate

In general, the defective rate varies with the machine deterioration and production rate [27]. When the machine shifts to the 'out-of-control' state [16], some percentage of the items produced are defective and a faster production rate deteriorates the quality of the process [13,24]. Since casting production scheduling has minor production cycle, the degree of machine deterioration can be viewed as unchanged and the concept of production rate can be described as the processing speed of operation under the shop floor environment. In this regard, the defective rate is merely the function of processing speed in the addressed situation.

Defective rate of an operation is the possibility of this operation processed with defect. In the addressed casting shop floor scenario, a product is not defective only when all operations of it processed without defects. Therefore, the DR_i can be calculated by Eq.(1).

$$DR_i = 1 - \prod_{j=1}^{s_i} \left(1 - \sum_{k=1}^{m} X_{ij,k} drO_{ij}^k \right) \tag{1}$$

Defective rate increases with the acceleration of processing speed. The idea in [20] is adapted and the defective rate is assumed to be a linear function of processing speed. If O_{ij} is processed on M_k, then

$$\frac{drO_{ij}^k}{\alpha_{Z_{ij,k}}^k} = \frac{dr_k}{\alpha_1^k} \tag{2}$$

With Eqs.(1) and (2), the mean defective rate $\overline{DR}$ of n jobs is formulated as Eq.(3).

$$\overline{DR} = \frac{\sum_{i=1}^{n} \left[1 - \prod_{j=1}^{s_i} \left(1 - \sum_{k=1}^{m} X_{ij,k} dr_k \frac{\alpha_{Z_{ij,k}}^k}{\alpha_1^k} \right) \right]}{n} \tag{3}$$

2.2.2 The model of makespan

The objective of makespan(C_{max}) is modeled as the matter of routine [17], the formulation is shown as Eq.(4).

$$C_{max} = max\{C_{ij}\} \tag{4}$$

The processing time of O_{ij} processed on M_k is calculated by Eq.(5).

$$t_{ij,k} = \frac{t_{ij,k}^{refer}}{\alpha_{Z_{ij,k}}^k} \tag{5}$$

2.2.3 The total energy consumption model

For an operation, it is commonly assumed that higher processing speed will reduce the processing time while consuming more energy [5]. Machines are generally in two main modes once turning on: processing mode and idle mode. When the machine is processing, it is in active mode and the output power is referred as the processing power. When the machine is in idle time between two processing operations, low output power is needed to hold the machine which is referred as idle power. In this paper, processing power is increased with the acceleration of processing speed. The value of processing power is predefined corresponding to the discrete processing speed. If O_{ij} is processed on M_k, the processing energy consumption is calculated by Eq.(6).

$$EP_{ij} = PP_{Z_{ij,k}}^k \frac{t_{ij,k}^{refer}}{\alpha_{Z_{ij,k}}^k} \tag{6}$$

Idle power is much lower than processing power and is regarded to be constant for a machine. To simplify the problem, it is assumed that all the machines keep turning on before all jobs are finished.

Therefore, the total energy consumption(TEC) is formulated as Eq.(7).

$$TEC = \sum_{k=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{s_i} PP_{Z_{ij,k}}^k X_{ij,k} \frac{t_{ij,k}^{refer}}{\alpha_{Z_{ij,k}}^k} + \sum_{k=1}^{m}\left(C_{max} - \sum_{i=1}^{n}\sum_{j=1}^{s_i} X_{ij,k}\frac{t_{ij,k}^{refer}}{\alpha_{Z_{ij,k}}^k}\right)PI^k \tag{7}$$

2.3 The model of MOCPSP

In the MOCPSP, the defective rate, the makespan and the total energy consumption are optimized simultaneously, shown as Eqs.(8)~(11).

$$f = min(\overline{DR}, C_{max}, TEC) \tag{8}$$

s.t.

$$\sum_{k=1}^{m} X_{ij,k} = 1, \forall i, j, \ k \subseteq S_{ij} \tag{9}$$

$$C_{ij} - C_{i(j-1)} \geq \frac{t_{ij,k}^{refer}}{\alpha_{Z_{ij,k}}^k} X_{ij,k}, j = 1, 2, .., s_i \tag{10}$$

$$\left(C_{i'j'} - C_{ij} - \frac{t_{i'j',k}^{refer}}{\alpha_{Z_{i'j',k}}^k}\right)X_{ij,k}X_{i'j',k}\left(Y_{ij,i'j'} + 1\right) + \left(C_{ij} - C_{i'j'} - \frac{t_{ij,k}^{refer}}{\alpha_{Z_{ij,k}}^k}\right)X_{ij,k}X_{i'j',k}\left(1 - Y_{ij,i'j'}\right) \geq 0, \forall i, j, i', j', k \tag{11}$$

Eq.(9) ensures that one operation should only be processed once by one of the available machines. Eq.(10) guarantees the predefined precedence relation of operations. Eq.(11) represents that a machine can only process one operation at a time.

3 The proposed N-NSGA-II for the MOCPSP

NSGA-II [10] is one of the most popular multi-objective optimizers. The main features of NSGA-II are summed as follows:
(1) computational complexity is reduced by a faster non-dominated sorting algorithm.
(2) solution diversity is improved by utilizing crowding distant to evaluate the density of non-dominated solutions.
In the light of those advantages, NSGA-II has been successfully applied to tackle the continuous optimization problems [39, 2, 34]. However, the MOCPSP is a discrete combinatorial optimization problem with extreme complexity, whose solution space is discontinuous and enormous. Utterly random local search mechanism of NSGA-II is time-costly and has poor convergence. Therefore, in this paper the improvements of the proposed N-NSGA-II are mainly presented in the two aspects: enhanced local search by neighborhood structures and novel solution generating approach. The detail of proposed N-NSGA-II for MOCPSP is successively described in the following subsections.

Table 2 The speed ratio, power and reference defective rate information of the 4*4 instance

Machine	Discrete speed ratio/processing power (kW)					Idle power (kW)	Reference defective rate
M_1	1/2.16	1.1/2.68	1.2/3.43	1.3/4.11	1.4/4.56	0.18	0.013
M_2	1/1.68	1.2/2.32	1.4/3.09	1.6/3.73	-	0.19	0.002
M_3	1/2.54	1.15/3.18	1.3/4.21	1.45/5.03	1.6/6.61	0.22	0.018
M_4	1/1.97	1.3/2.65	1.6/3.52	-	-	0.23	0.030

Table 3 The reference processing time information of the 4*4 instance

Job	operation	Reference processing time			
		M1	M2	M3	M4
Job1	O_{11}	5	4	9	-
	O_{12}	11	20	-	14
	O_{13}	8	8	-	-
Job2	O_{21}	14	22	16	-
	O_{22}	-	17	17	19
	O_{23}	20	-	15	-
	O_{24}	10	-	-	10
Job3	O_{31}	8	12	-	10
	O_{32}	-	-	6	6
	O_{33}	19	16	-	16
Job4	O_{41}	-	18	-	16
	O_{42}	30	30	25	-

	O_{11}	O_{12}	O_{13}	O_{21}	O_{22}	O_{23}	O_{24}	O_{31}	O_{32}	O_{33}	O_{41}	O_{42}
X_{OS}	2	5	8	1	7	9	12	4	6	11	3	10
X_{MA}	2	4	1	3	2	3	1	4	4	1	2	3
X_{PSR}	0.3	0.7	0.1	0.1	0.4	0.6	0.7	0.4	0.6	0.5	0.4	0.8

Fig. 1 An illustration of the coding method

3.1 Coding and decoding methods

In N-NSGA-II, three vectors are included corresponding to three decision variables: operation sequencing vector, machine assignment vector and speed ratio seed vector. They are represented by X_{OS}, X_{MA} and X_{SRS}, the dimensions of which equal to the number of all operation to be processed.

The X_{OS} is a permutation of the number of all operations. Note that the operations belong to the same job should order in ascending to avoid infeasible solution [7]. The gene in X_{MA} is the machine number. The machine for the operation should be selected from the alternative set. The gene in X_{SRS} is a random number range from 0 to 1. The speed ratio seed of O_{ij} is noted as v_{ij}. The speed ratio for O_{ij} on M_k is calculated by Eq.(12).

$$Z_{ij,k} = v_{ij} \otimes p_k, \ v_{ij} \in (0,1] \tag{12}$$

where $\otimes$ represents the ceil operator.

The "active" decoding is applied in this paper, the procedure of which is described concretely in [7].

To better demonstrate the coding and decoding methods, a case with 4 jobs and 4 machines(4*4) is taken for instance. The speed ratio, power and reference defective rate information is shown in Table.2 and the reference processing time information is shown in Table.3. The three-vector coding is illustrated in Fig.1.

Take O_{22} as an example. The allocating sequence of O_{22} is 7th. The machine for O_{22} is M_2, which has 4 discrete speed ratios. So $Z_{22,2}$ is calculated by Eq.(12) and value is 2, which means the 2nd speed ratio of M_2 is selectd for O_{22}. And the defective rate, the processing time and processing power for O_{22} are obtained by Eq.(2), Eq.(5) and Eq.(6) respectively.

The Gantt chart of the illustrated coding is shown in Fig.2.

3.2 Initialization

For X_{OS}, the sequence of all genes is randomly initialized. For X_{MA}, Zhang et.al [44] developed an effective initialization method to assign operation to the suitable machine among which the two initial strategies named global selection (GS) and local selection (LS) are applied in this paper to initialize X_{MA}. The proportion of GS is referred as r_{GS} and the proportion of LS is equal to $1 - r_{GS}$. For X_{SRS}, the solution set is equally divided into three portions. The first portion is initialized with 1 for all genes, the second portion is initialized with 0 for all genes, the rest is randomly initialized for all genes.
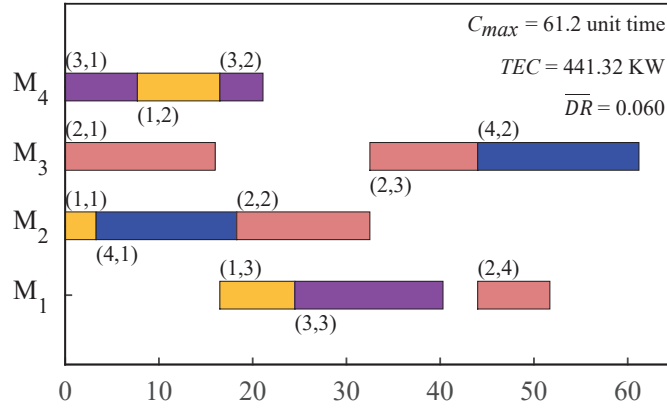
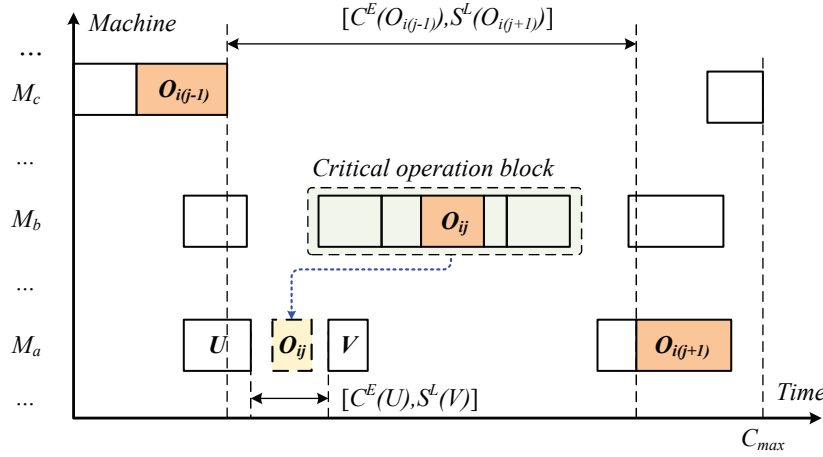**Fig. 2** The Gantt chart of the coding in Fig.1

Fig. 3 Schematic of the NS1

3.3 Enhanced local search by neighborhood structures

Neighborhood structure embedded with the knowledge of MOCPSP serves as a powerful tool for local search. Let $NS_k (k \in \{1, 2, ..., k_{max}\})$ to be one of the set of the k_{max} predefined neighborhood structures. $NS_k(s)$ is the set of neighbor solutions of s in the NS_k.

In this paper, three neighborhood structures are designed corresponding to the three decision variables, denoted as NS_1, NS_2 and NS_3. And the three decision variables can be divided into two categories: operation movements (decision variables X and Y) and speed ratio seed (decision variable Z) mutation.

Operation movements are based on the concept of critical path, critical operation and critical operation block [46]. Operation movements including two situations: cross-machine movement and same-machine movement, corresponding to NS_1 and NS_2. Illustrations of NS_1 and NS_2 are shown in Fig.3 and Fig.4 respectively. O_{ij} is a critical operation from a critical operation block containing at least two operations. $O_{i(j-1)}$ and $O_{i(j+1)}$ are the predecessor and successor operations of the same job adjacent to O_{ij} respectively. S^L represents the latest start time and C^E represents the earliest completion time of an operation. Movement of O_{ij} is limited by $O_{i(j-1)}$ and $O_{i(j+1)}$, whose range is $[C^E(O_{i(j-1)}), S^L(O_{i(j+1)})]$.

NS_1: cross-machine movement for operation is illustrated in Fig.3. In MOCPSP, the number of machine should be more than one. Assume U and V are two consecutive operations on machine Ma (U is processed before V) and O_{ij} is not processed on Ma. If Ma is one of the alternative machines of O_{ij} and Eq.(13) is satisfied, O_{ij} is moved to Ma. If no such movement exists, O_{ij} is moved out to another random machine in the set of alternative machines.

$$\begin{cases} [0, S^L(O_{i(j+1)})] \cap [C^E(U), S^L(V)] > \dfrac{t_{ij,Ma}^{refer}}{\alpha_{Z_{ij,Ma}}^{Ma}}, & j = 1 \\[2ex] [C^E(O_{i(j-1)}), C_{max}] \cap [C^E(U), S^L(V)] > \dfrac{t_{ij,Ma}^{refer}}{\alpha_{Z_{ij,Ma}}^{Ma}}, & j = s_i \\[2ex] [C^E(O_{i(j-1)}), S^L(O_{i(j+1)})] \cap [C^E(U), S^L(V)] > \dfrac{t_{ij,Ma}^{refer}}{\alpha_{Z_{ij,Ma}}^{Ma}}, & otherwise \end{cases} \tag{13}$$

NS_2: same-machine movement for operation is illustrated in Fig.4. The purpose of NS_2 is to move critical operation O_{ij} out the critical operation block on the same machine to shorten the length of critical operation block. There are three situations:

1) if O_{ij} is the internal operation of the critical operation block, O_{ij} would be moved before the first operation or after the last operation of the critical operation block within the range of $[C^E(O_{i(j-1)}), S^L(O_{i(j+1)})]$, as shown in Fig.4(a).
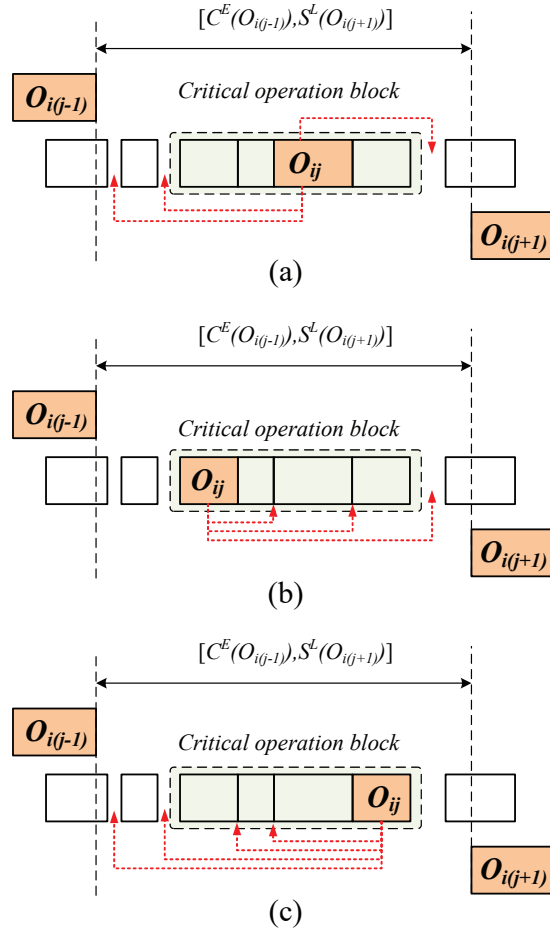
Fig. 4 Schematic of the NS2

2) if O_{ij} is the start operation of the critical operation block, O_{ij} would be moved backwards within the range of $\left[C^E(O_{i(j-1)}), S^L(O_{i(j+1)})\right]$, as shown in Fig.4(b).

3) if O_{ij} is the end operation of the critical operation block, O_{ij} would be moved forwards within the range of $\left[C^E(O_{i(j-1)}), S^L(O_{i(j+1)})\right]$, as shown in Fig.4(c).

Speed ratio seed (decision variable Z) is adjusted by mutation operator. The detail is given as follows.

NS_3: two parameters are defined: mutation probability (p_m) and step size (σ). p_m determines the number of operation to mutate in operation string. Assume that one of the solutions is to mutate. Select another solution S_2 from the solution set as reference solution and the speed ratio seed of O_{ij} in $\beta + 1$ iteration is updated by Eq.(14).

$$v_{ij}^{\beta+1} = \left(v_{ij}^{\beta,refer} - v_{ij}^{\beta}\right) \times \sigma \times \epsilon + v_{ij}^{\beta} \tag{14}$$

Where $v_{ij}^{\beta,refer}$ is the speed ratio seed of O_{ij} of the reference solution, ϵ is a random number range from 0 to 1.

3.4 Elitism strategy

To reserve more optimal solutions, the elitism strategy is adapted in the N-NSGA-II. The reservation takes place after local search. In the first iteration, the first Pareto front solutions are viewed as elites and reserved. In the following iterations, the first Pareto front solutions found in the current iteration are combined with the former elites, and solutions in the new first Pareto front are viewed as elites and reserved.

3.5 Novel solution generating approach

To improve the diversity of solution, a novel solution generating approach is designed in N-NSGA-II. The illustration of the proposed solution generating approach is shown in Fig.5.

Step1-Inherit the genes from solution set. For a new solution, the gene of each operation is inherited from the gene of the same operation of a solution randomly selected from solution set. As shown in Fig.5, O_{11}, O_{12}, O_{21}, O_{22} of the new solution are inherited from S_2, S_3, S_3, S_2 respectively. Repeat the procedure until generating $Npop_{SG}$ new solutions.
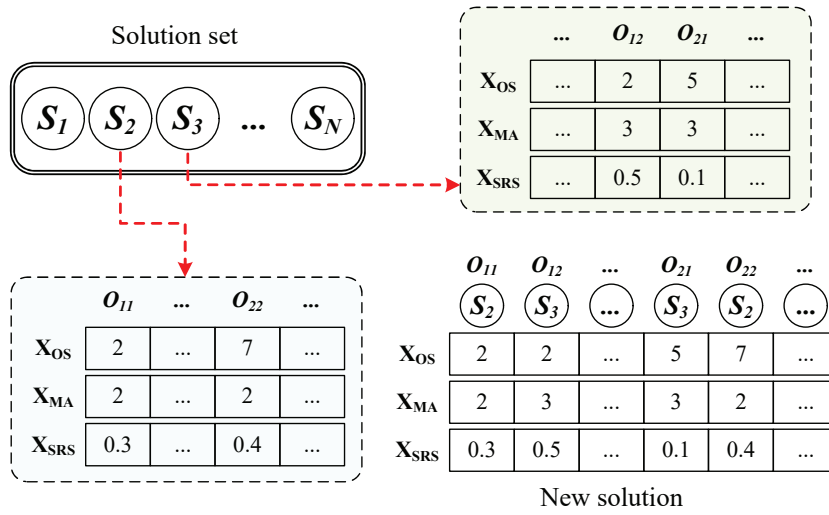
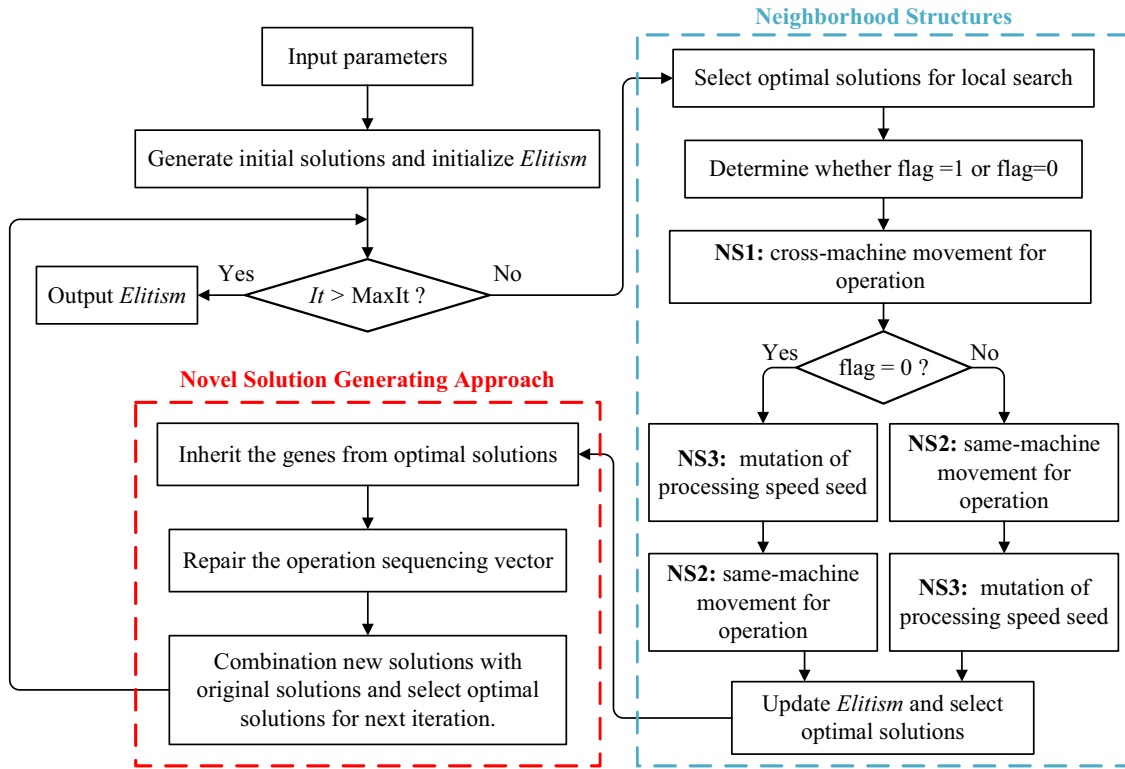**Fig. 5** Schematic of the novel solution generating approach

Fig. 6 Flowchart of the proposed N-NSGA-II

Step2-Repair the X_{OS} vector. When there is the repeated number in X_{OS} vector, as shown in Fig.5, the repair is needed to avoid infeasible solution. Repair follows the rules:(1) The operations with the same X_{OS} gene are random sorted;(2) The operation with larger X_{OS} gene comes after the smaller one. The procedure is described as follows:

(1) Sort X_{OS} of all operations with by ascending.

(2) For the operations with the same X_{OS} gene, switch their positions randomly.

(3) Number all operations successively with their positions, the first position is numbered 1.

(4) Update the X_{OS} with the corresponding operation number.

Step3-Combination and selection. Combine the $Npop_{SG}$ new solutions with the initial solution set to form a new solution set and utilize the selection mechanism of NSGA-II to select $Npop$ optimal solutions for next iteration.

3.6 Stopping criterion

The algorithm terminates after a certain number of iterations. This number is predefine and referred as MaxIt.
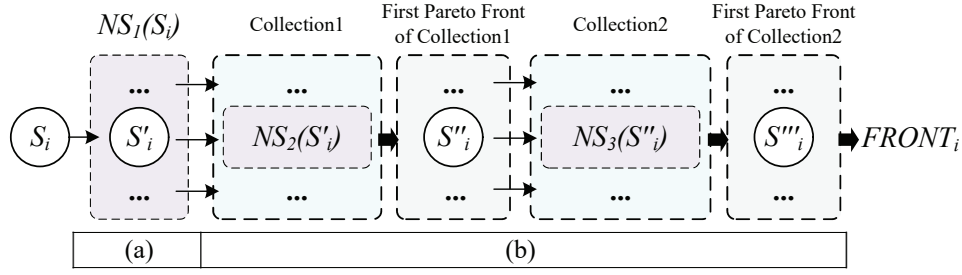
Fig. 7 Schematic of local search by the three neighborhood structures(flag=1)

3.7 The procedure of N-NSGA-II

The flowchart of the proposed N-NSGA-II is illustrated in Fig.5. The overall procedure is described as follows.

Input parameters: MaxIt, r_{GS}, $Npop$, $Npop_L$, $Npop_{SG}$, p_m, σ, τ

Step1. Initialization: Generate $Npop$ initial solutions by initialization approach in 3.2.

Step2. Stopping criterion: Determine whether the stopping criterion is met. If not, go to step3. If so, output the Elitism.

Step3. Local search by neighborhood structures: Select the $Npop_L(Npop_L < Npop)$ optimal solutions by the selection mechanism of NSGA-II. Perform the local search with the designed neighborhood structures for the optimal solutions. For solution S_i, the local search procedures are:

Step3-1. Generate a random number r ranging from 0 to 1. If r is greater than the search constant τ, flag=1; otherwise, flag=0. The $FRONT_i$ is used to temporally reserve first Pareto front solutions found for S_i.

Step3-2. Find all neighbor solutions of S_i in NS_1 and output the solution set $NS_1(S_i)$ to Step3-3, as shown in Fig.7(a).

Step3-3. If flag equals to 1, the procedure of local search by NS_2 and NS_3 successively is illustrated in Fig.7(b): (1)find all neighbor solutions of each solution from $NS_1(S_i)$ in NS_2, which make up Collection1, and reserve the first Pareto front solutions of Collection1, (2)find all neighbor solutions of each solution from Collection1 in NS_3, which make up the Collection2, and reserve the first Pareto front solutions of Collection2, (3)output all solutions in Collection2 to $FRONT_i$. For the circumstance that flag equals to 0, switch the sequence of NS_2 and NS_3 and follow the same procedure as the circumstance that flag equals to 1.

Step3-4. If S_i is not a member of $FRONT_i$, select one solution from $FRONT_i$ and name it as S_i, return to Step3-2. Otherwise, output $FRONT_i$.

Step4. Elitism strategy: Combine all $FRONT_i$s as well as the initial solution of this iteration with Elitism and perform the selection of NSGA-II [10]. Update the Elitism with the first Pareto front solutions of the combination and output the $Npop$ optimal solutions to step 5.

Step5. Novel solution generating approach: Perform novel solution generating approach in 3.5, output $Npop$ solutions and return to step 2.

4 Computational experiments and discussions

In this section, a real-world casting production scheduling case is investigated. Four performance metrics for measuring the Pareto front obtained by algorithms are employed. Then, the effectiveness of the neighborhood structures and the novel solution generating approach are proved. At last, the proposed N-NSGA-II is compared with two well-known multi-objective algorithms to evaluate the performance of N-NSGA-II on the MOCPSP. All algorithms are coded in Matlab R2018b and run on a computer with Intel Core i7(3.0GHz), 8 GB RAM under Windows 10 operation environment.

4.1 Case description

To consider MOCSPS in the real-world casting workshop, a case with 20 jobs and 15 machines is presented. In this case, 15 common operations are addressed and the machines for each operation are listed in Table.4. The process route of each job and the corresponding reference processing time information are shown in Table.5. The speed ratio, power and reference defective rate information of machines are listed in Table.6.

4.2 Performance metrics

To give quantitative evaluations on the performance of multi-objective optimization algorithms, four metrics are employed including generational distance (GD), contribution rate (CR), inverse generational distance (IGD) and set coverage(c-metric). The definitions of the four metrics are listed as follows.

Table 4 Operations and alternative machines of the real-world casting case

Operation				
O1	*O2*	*O3*	*O4*	*O5*
Modling	Melting	Clearing	Thermal treatment	Rough turning
$[M_1, M_2]$	M_3	$[M_4,M_5]$	M_6	$[M_7,M_8,M_9]$
O6	*O7*	*O8*	*O9*	*O10*
Semi-extractive turning	Extractive turning	Fine turning	Rough milling	Semi-extractive milling
$[M_8,M_9]$	$[M_8,M_9]$	$[M_9]$	$[M_8,M_9,M_{10},M_{11},M_{12}]$	$[M_{10},M_{11}]$
O11	*O12*	*O13*	*O14*	*O15*
Extractive milling	Weld repairing	Extractive grinding	Fine grinding	Flaw detection
M_{11}	M_{13}	$[M_8,M_9,M_{12},M_{14}]$	$[M_{12},M_{14}]$	M_{15}

(1) Convergence metrics: GD [47]

$$GD = \frac{\sqrt{\sum_{i=1}^{|\Omega|} D_i^2(\Omega,\Omega^*)}}{|\Omega|} \tag{15}$$

where Ω^* is the true first Pareto front(TFPF) and Ω is the first Pareto front found by an algorithm. $D_i(\Omega,\Omega^*)$ is the shortest Euclidean distance between the solution i in Ω and the solutions in Ω^*.

(2) Quality metric: CR [29]

$$CR = \frac{|\{x|x \in \Omega \ \& \ x \in \Omega^*\}|}{|\Omega^*|} \times 100\% \tag{16}$$

(3) Comprehensive metric: IGD [48]

$$IGD = \frac{\sum_{i=1}^{|\Omega^*|} D_i(\Omega^*,\Omega)}{|\Omega^*|} \tag{17}$$

where $D_i(\Omega^*,\Omega)$ is the shortest Euclidean distance between the solution i in Ω^* and the solutions in Ω.

(4) Set coverage metric: c-metric [48]

$$C(P,Q) = \frac{|\{q \in Q: \exists \ p \in P, \ p \succ q\}|}{Q} \tag{18}$$

where P and Q are the first Pareto fronts obtained by two algorithms, $p \succ q$ represents the q is dominated by p.

The smaller value of GD and IGD while the bigger value of CR represents good performance. And the bigger value of $C(P,Q)$ than $C(Q,P)$ represents the P is preferable. Since the TFPF of the case is unknown, the first Pareto front of the solutions found by all employed algorithms is viewed as the TFPF in this paper.

Table 5: Process route and the corresponding reference processing time information

Process route of each job and the reference processing time on the alternative machines							
Job1 O1	O2	O3	O4	O5	O9	O13	O6
[46,35]	6	[16,17]	33	[29,34,11]	[34,16,25,13,26]	[26,16,27,13]	[30,15]
O7	O12	O14	O10	O11	O8	O15	
[37,19]	14	[31,11]	[26,11]	11	20	17	
Job2 O1	O2	O3	O4	O13	O5	O6	O7
[55,27]	8	[26,18]	41	[27,13,27,13]	[33,37,15]	[37,15]	[39,15]
O14	O9	O10	O12	O8	O11	O15	
[33,11]	[28,12,33,11,26]	[28,17]	13	18	16	8	
Job3 O1	O2	O3	O4	O13	O9	O12	O10
[58,34]	5	[23,15]	36	[34,16,29,12]	[26,18,23,12,28]	13	[24,18]
O14	O5	O6	O7	O8	O11	O15	
[30,13]	[30,43,11]	[44,15]	[43,15]	16	15	7	
Job4 O1	O2	O3	O4	O9	O5	O10	O11
[46,26]	4	[18,12]	30	[30,16,27,18,35]	[32,34,14]	[27,15]	15
O6	O13	O7	O14	O8	O12	O15	
[31,11]	[38,12,31,12]	[43,13]	[30,11]	12	13	7	
Job5 O1	O2	O3	O4	O5	O13	O9	O6
[37,27]	8	[19,16]	30	[27,27,17]	[43,11,26,11]	[30,18,29,17,30]	[34,18]
O12	O7	O8	O10	O11	O14	O15	
15	[43,17]	20	[23,14]	12	[33,15]	8	
Job6 O1	O2	O3	O4	O9	O5	O10	O6
[39,28]	7	[27,13]	45	[26,16,25,14,34]	[30,30,11]	[38,11]	[41,15]
O11	O12	O7	O13	O14	O18	O15	

Process route of each job and the reference processing time on the alternative machines

	15	[15]	[38,13]	[32,20,34,12]	[31,15]	13	11
Job7	O1	O2	O3	O4	O9	O13	O10	O5
	[55,41]	4	[18,11]	31	[38,17,35,16,35]	[38,17,32,15]	[22,15]	[28,35,15]
	O6	O14	O11	O7	O12	O8	O15	
	[42,17]	[35,12]	18	[33,18]	15	16	12	
Job8	O1	O2	O3	O4	O5	O9	O6	O13
	[45,17]	7	[18,11]	37	[27,39,12]	[32,19,28,18,34]	[37,19]	[42,11,32,14]
	O10	O7	O12	O14	O11	O8	O15	
	[29,14]	[31,19]	14	[28,15]	17	19	10	
Job9	O1	O2	O3	O4	O5	O13	O9	O6
	[63,43]	4	[23,19]	32	[33,40,14]	[36,20,26,13]	[43,16,37,15,33]	[27,19]
	O14	O10	O7	O12	O11	O8	O15	
	[33,11]	[31,16]	[42,13]	13	16	20	13	
Job10	O1	O2	O3	O4	O13	O5	O6	O9
	[43,29]	4	[22,16]	32	[42,13,30,14]	[28,42,13]	[43,18]	[26,18,29,15,26]
	O7	O12	O10	O14	O8	O11	O15	
	[35,16]	13	[21,16]	[27,13]	14	12	18	
Job11	O1	O2	O3	O4	O13	O5	O12	O6
	[44,45]	4	[30,16]	44	[36,17,29,12]	[26,38,12]	11	[44,20]
	O7	O14	O8	O9	O10	O11	O15	
	[27,19]	[30,11]	16	[40,12,22,18,30]	[24,16]	15	16	
Job12	O1	O2	O3	O4	O9	O5	O6	O12
	[63,26]	7	[26,20]	43	[39,11,36,14,35]	[30,37,15]	[44,16]	15
	O13	O10	O7	O11	O8	O14	O15	
	[41,13,31,11]	[36,15]	[37,18]	16	17	[30,15]	7	
Job13	O1	O2	O3	O4	O12	O9	O5	O13
	[56,45]	7	[19,16]	37	11	[36,11,27,13,32]	[28,44,13]	[27,12,32,13]
	O6	O14	O10	O7	O11	O8	O15	
	[41,14]	[30,11]	[35,13]	[31,18]	12	11	9	
Job14	O1	O2	O3	O4	O9	O13	O5	O14
	[49,12]	6	[17,12]	44	[29,15,29,18,30]	[26,18,33,11]	[31,31,13]	[33,15]
	O10	O6	O11	O7	O8	O12	O15	
	[12,16]	[28,13]	11	[44,13]	13	15	16	
Job15	O1	O2	O3	O4	O9	O5	O13	O14
	[65,45]	6	[20,20]	39	[45,16,21,17,28]	[31,35,20]	[30,19,35,12]	[29,13]
	O10	O11	O6	O7	O12	O18	O15	
	[21,16]	18	[34,17]	[36,11]	14	19	18	
Job16	O1	O2	O3	O4	O9	O12	O10	O5
	[54,29]	8	[22,19]	29	[39,15,27,13,31]	13	[34,13]	[27,30,19]
	O13	O6	O7	O14	O8	O11	O15	
	[37,14,32,12]	[40,18]	[35,17]	[31,11]	19	17	19	
Job17	O1	O2	O3	O4	O13	O5	O6	O9
	[44,45]	8	[18,11]	31	[26,17,33,15]	[29,33,18]	[30,18]	[45,20,34,15,34]
	O14	O7	O8	O10	O11	O12	O15	
	[34,12]	[35,13]	20	[25,12]	14	14	19	
Job18	O1	O2	O3	O4	O5	O9	O13	O6
	[43,35]	6	[16,18]	35	[33,37,11]	[32,19,28,13,29]	[36,12,29,14]	[29,19]
	O14	O10	O7	O8	O11	O12	O15	
	[32,11]	[31,13]	[43,19]	12	11	11	12	
Job19	O1	O2	O3	O4	O13	O5	O6	O9
	[40,32]	5	[27,20]	40	[33,15,33,11]	[33,31,19]	[29,19]	[35,11,32,14,27]
	O10	O7	O14	O11	O8	O12	O15	
	[31,17]	[31,14]	[32,13]	13	19	13	9	
Job20	O1	O2	O3	O4	O9	O10	O11	O12
	[56,39]	8	[23,13]	35	[36,13,33,11,32]	[27,13]	18	14
	O5	O6	O7	O8	O13	O14	O15	
	[31,29,18]	[31,15]	[38,11]	18	[28,13,26,13]	[33,13]	11	

Table 6 The speed ratio, power and reference defective rate information of machines

	Discrete speed ratio/processing power (kW)						Idle power (kW)	Reference defective rate
M_1	1.00/2.34	1.17/2.98	1.33/3.75	1.50/4.49			0.23	0.020
M_2	1.00/2.21	1.23/2.81	1.47/3.98	1.70/4.85			0.21	0.009
M_3	1.00/2.14	1.20/2.59	1.40/3.46	1.60/3.97			0.22	0.007
M_4	1.00/2.48	1.30/3.35	1.6/4.51				0.19	0.019
M_5	1.00/1.66	1.10/2.11	1.20/2.48	1.30/2.71	1.40/3.11	1.50/3.74	0.21	0.010
M_6	1.00/2.03	1.20/2.73	1.40/3.57				0.22	0.018
M_7	1.00/2.51	1.13/3.29	1.27/4.19	1.40/4.79			0.24	0.002
M_8	1.00/1.88	1.08/2.07	1.16/2.66	1.24/3.37	1.32/3.80	1.40/4.39	0.19	0.005
M_9	1.00/1.90	1.17/2.50	1.33/3.01	1.50/3.97			0.19	0.001
M_{10}	1.00/1.76	1.10/2.17	1.20/2.77	1.30/3.03	1.40/3.67		0.20	0.005
M_{11}	1.00/1.89	1.13/2.21	1.27/2.49	1.40/2.76			0.20	0.017
M_{12}	1.00/2.17	1.08/2.76	1.16/3.55	1.24/3.97	1.32/4.91	1.40/5.38	0.19	0.018
M_{13}	1.00/2.39	1.13/2.95	1.27/3.36	1.40/3.83			0.20	0.016
M_{14}	1.00/1.72	1.25/2.36	1.50/3.05				0.22	0.014
M_{15}	1.00/2.02	1.20/2.74	1.40/3.74				0.21	0.014

Table 7 Parameter settings of the N-NSGA-II and its variants

Parameters	A N-NSGA-II	B NSGA-II+Novel SGA	C NSGA-II+NS	D NSGA-II
Population size($Npop$)	80	80	80	80
Maximum iterations(MaxIt)	50	500	50	500
Proportion of GS in initialization(r_{GS})	0.4	0.4	0.4	0.4
Crossover+mutation operators [43]	**None**	**None**	✓	✓
Crossover probability	-	-	0.7	0.7
Mutation probability	-	-	0.4	0.4
Neighborhood structures	✓	**None**	✓	**None**
Population size of NS($Npop_L$)	20	-	20	-
Mutation probability(p_m)	0.1	-	0.1	-
Step size(σ)	0.5	-	0.5	-
search constant(τ)	0.6	-	0.6	-
Novel SGA	✓	✓	**None**	**None**
Population size of new solution by ($Npop_{SG}$)	40	40	-	-

4.3 Effectiveness of the neighborhood structures and the novel solution generating approach

To verify the effectiveness of the proposed neighborhood structures(NS) and the novel solution generating approach(novel SGA), the N-NSGA-II is compared with its variants. The parameter settings of the algorithms are list in Table.7. All the algorithms run 20 times independently and the elapsed time is approximately 500s(CPU time) per time with the parameter setting. The computational results on metrics of GD, CR and IGD are shown in Table.8 and on c-metric are shown in Table.9. A, B, C and D denote the N-NSGA-II, NSGA-II+Novel SGA, NSGA-II+NS and NSGA-II respectively in Table.8 and Table.9.

4.3.1 Discussion on the effectiveness of the neighborhood structures

To discuss the effectiveness of the neighborhood structures, A and B can be used as a group of comparison while C and D can be used as the other. It can be seen from Table.8 and Table.9 that A is superior to B while C is superior to D on all the metrics. As shown in Table.8, both A and C have better value than their comparisons on the metrics of GD, CR and IGD. Moreover, as shown in Table.9, $c(A,B)>c(B,A)$ and $c(C,D)>c(D,C)$. The notable differences indicate that the enhanced local search by proposed neighborhood structures is effective for improving the performance of N-NSGA-II.

4.3.2 Discussion on the effectiveness of the novel solution generating approach

To discuss the effectiveness of the novel solution generating approach, A and C can be used as a group of comparison while B and D can be used as the other. It can be seen from Table.8 and Table9 that A is superior to D while B is superior to D on all the metrics. A similar analysis as 4.3.1 can be made and it is shown that the novel solution generating approach has positive impact on the performance of N-NSGA-II.

4.4 Comparisons of N-NSGA-II with other algorithms

To further demonstrate the overall performance of the N-NSGA-II, the SPEA-II [48] and NSGA-III [9] are employed to compare with the N-NSGA-II. The parameter settings of SPEA-II and NSGA-III are listed in Table.10 while the parameter setting of N-NSGA-II is the same as listed in Table.7. The computational process is the same as that in 4.3.

Table 8 Computational results of N-NSGA-II and its variants on GD, CR and IGD

	GD	CR	IGD
A	8.65e-3	52.1%	1.09e-1
B	1.23e-1	12.3%	1.48e-1
C	2.73e-2	30.8%	1.21e-1
D	1.86e-1	4.8%	1.90e-1

Table 9 Computational results of N-NSGA-II and its variants on *c*-metric

$c(A,B)$	$c(B,A)$	$c(C,D)$	$c(D,C)$	$c(A,C)$	$c(C,A)$	$c(B,D)$	$c(D,B)$
7.50e-1	0	**8.36e-1**	0	**4.36e-1**	5.00e-2	**4.13e-1**	1.50e-1
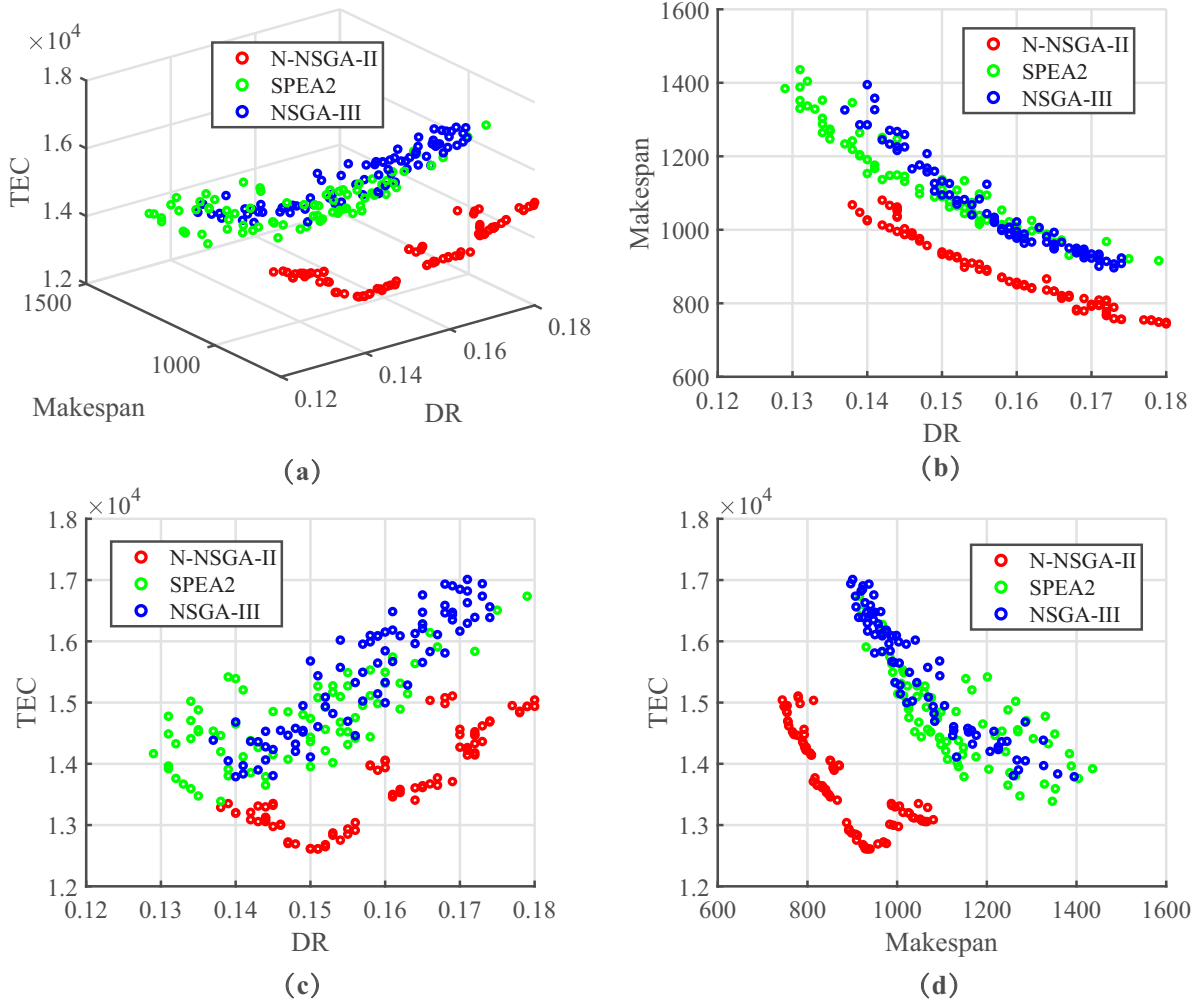
Fig. 8 The Pareto fronts obtained by N-NSGA-II, SPEA-II and NSGA-III under different perspectives

Table 10 Parameter settings of SPEA-II and NSGA-III

	SPEA-II	NSGA-III
Population size	80	80
Maximum iterations	500	500
Archive size [48]	80	-
Number of division [9]	-	5
Crossover+mutation operators [43]		
Crossover rate	0.7	0.7
Mutation rate	0.4	0.4

The Pareto fronts obtained by N-NSGA-II, SPEA-II and NSGA-III under different perspectives are shown in Fig.8. It is clear that the N-NSGA-II approaches closer to the TFPF than the other two algorithms, which proves the superiority of the N-NSGA-II. Fig. 8(a) shows a 3D view with three objectives represented by defective rate, makespan and total energy consumption respectively. It can be observed from Fig. 8(b) that the defective rate is conflicted to the makespan. Moreover, defective rate of

Table 11 Computational results of of N-NSGA-II, SPEA-II and NSGA-III on GD, CR and IGD

	GD	CR	IGD
N-NSGA-II	**0**	**82.5%**	**1.11e-1**
SPEA-II	2.10e-1	17.5%	2.83e-1
NSGA-III	3.09e-1	0	3.25e-1

Table 12 Computational results of N-NSGA-II, SPEA-II and NSGA-III on c-metric

$c(A,E)$	$c(E,A)$	$c(A,F)$	$c(F,A)$
7.76e-1	0	**9.86e-1**	0

Table 13 Computational results of N-NSGA-II, SPEA-II and NSGA-III on c-metric

No.	Defective rate	Makespan	Total energy consumption (kW)	Score	Obtained by
1	0.147	969.5	1.272e+4	0.7531	
2	0.148	957.1	1.267e+4	0.7529	
3	0.150	933.2	1.262e+4	0.7526	
4	0.147	976.2	1.270e+4	0.7519	
5	0.140	1024.6	1.320e+4	0.7517	**N-NSGA-II**
6	0.140	1027	1.320e+4	0.7509	
7	0.150	939.7	1.261e+4	0.7503	
8	0.151	928.8	1.261e+4	0.7469	
9	0.151	933.7	1.261e+4	0.7452	
10	0.138	1068	1.329e+4	0.7426	
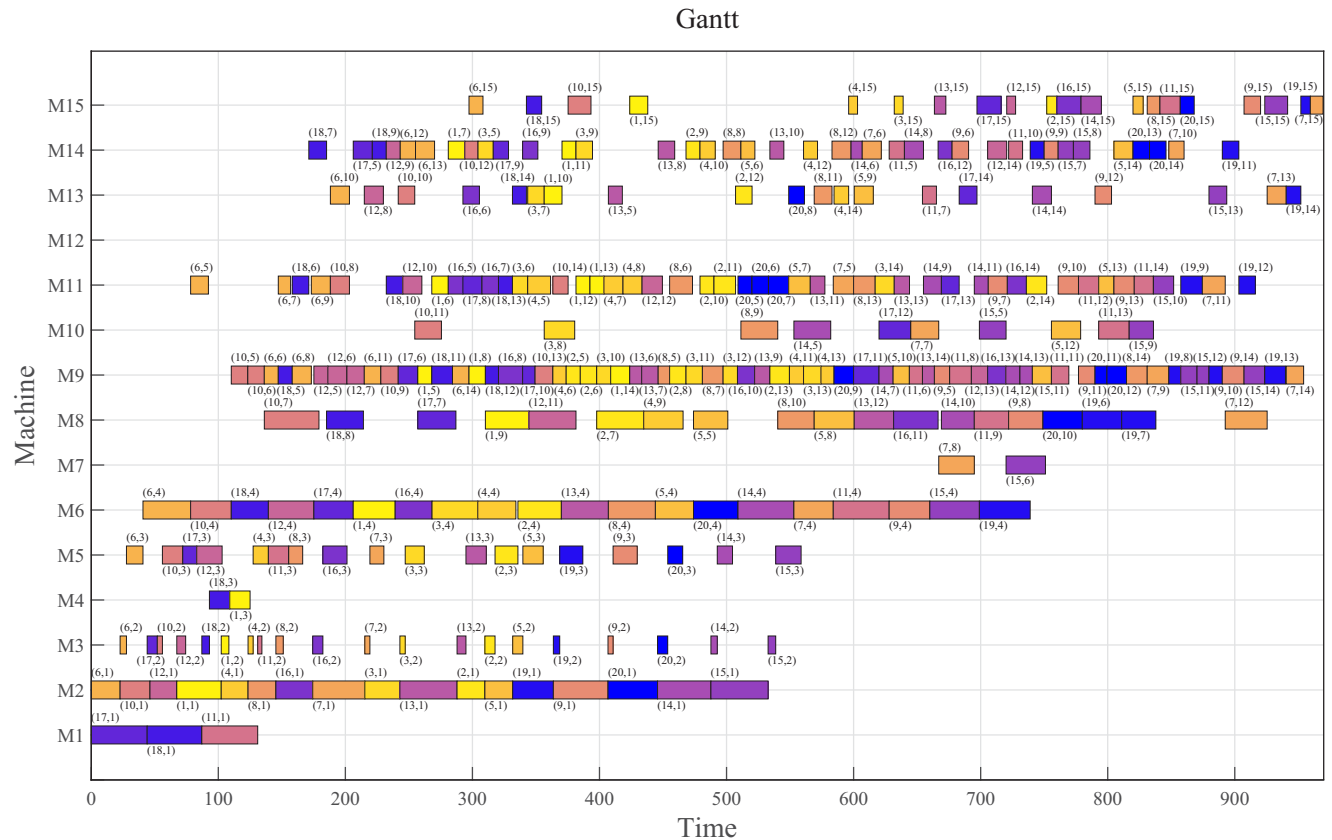
Fig. 9 The Gantt chart of the most satisfying solution for the case

the solutions obtained by N-NSGA-II is much lower than that by other algorithms under the condition of the same makespan and the minimum makespan is obtained by the proposed N-NSGA-II. In Fig. 8(c) gives a 2D view subjected to defective rate and total energy consumption. The minimum TEC is obtained by the N-NSGA-II. Fig. 8(d) presents the results only containing the makespan and TEC. It is clear that the solutions generated by N-NSGA-II are much better than those by other algorithms.

The computational results on metrics of GD, CR and IGD are shown in Table.11 and on c-metric are shown in Table.12. A, E and F donate the N-NSGA-II, SPEA-II and NSGA-III respectively in Table.12. As shown in Table.11, the N-NSGA-II

is significantly better than SPEA-II and NSGA-III on the metrics of GD, CR and IGD. Furthermore, as shown in Table.12, N-NSGA-II overwhelmingly dominates in pairwise comparison with SPEA-II and NSGA-III. Consequently, the superior of the proposed N-NSGA-II is validated.

Moreover, to obtain the most satisfying solution in the Pareto front, the approach in [8] is introduced to the paper. Assume that the weight vector W=(0.4 0.3 0.3)^T regarding to the objectives of defective rate, makespan and total energy consumption respectively. By normalizing the fitness and multiplying by the W, the scores of each solution are obtained. The higher score represents better quality. The top 10 scores among all solutions are listed in Table.13 and the Gantt chart of the solution with the highest score is illustrated in Fig.9. As shown in Table.13, the top 10 scoring solutions are all obtained by N-NSGA-II, which gives further proof of the superiority of the N-NSGA-II.

5 Conclusions and future work

In this paper, the casting production scheduling problem(CPSP) under the shop floor environment is investigated. A set of comprehensive and practical scheduling scheme for casting production is obtained by establishing a multi-objective CPSP to minimize three objectives of defective rate, makespan and total energy consumption simultaneously. Furthermore, the N-NSGA-II is proposed for solving the multi-objective CPSP. The effectiveness of the proposed neighborhood structures and novel solution generating approach of the N-NSGA-II is validated. Moreover, the overall performance of the N-NSGA-II is evaluated comprehensively via comparison experiments and the superiority of the N-NSGA-II is verified.

There are also some limitations of this work. On the aspect of problem formulation, the model accuracy can be improved by involving realistic constraints. For example, energy-intensive operations like melting and thermal treatment are often constrained to certain periods to avoid peak hours of energy consumption. Another example is the processing interval constraint which means the time interval between two certain operations is constrained according to the process requirements. Moreover, unexpected events such as immediate job insertion and machine breakdown may occur in the practical casting production workshops. On the aspect of algorithm, since reinforcement learning (RL) has made remarkable achievements in recent years, RL-based approaches will be a promising way to enhance the performance of existing heuristic algorithms. Besides, constructing the systematic rules by incorporating the knowledge of specific problems will greatly reduce the invalid search such that the performance of algorithms can be promoted.

References

1. Akturk, M.S., Ilhan, T.: Single cnc machine scheduling with controllable processing times to minimize total weighted tardiness. Computers & Operations Research **38**(4), 771–781 (2011)
2. Asefi, H., Jolai, F., Rabiee, M., Araghi, M.E.T.: A hybrid nsga-ii and vns for solving a bi-objective no-wait flexible flowshop scheduling problem. The International Journal of Advanced Manufacturing Technology **75**, 1017–1033 (2014)
3. Atan, M.O., Akturk, M.S.: Single cnc machine scheduling with controllable processing times and multiple due dates. International Journal of Production Research **46**(21), 6087–6111 (2008)
4. Chaudhry, I.A., Khan, A.A.: A research survey: review of flexible job shop scheduling techniques. International Transactions in Operational Research **23**(3), 551–591 (2016)
5. Che, A., Lv, K., Levner, E., Kats, V.: Energy consumption minimization for single machine scheduling with bounded maximum tardiness. In: 2015 IEEE 12th International Conference on Networking, Sensing and Control, pp. 146–150 (2015)
6. Chen, X., An, Y., Zhang, Z., Li, Y.: An approximate nondominated sorting genetic algorithm to integrate optimization of production scheduling and accurate maintenance based on reliability intervals. Journal of Manufacturing Systems **54**, 227–241 (2020)
7. Cheng, R., Gen, M., Tsujimura, Y.: A tutorial survey of job-shop scheduling problems using genetic algorithms-i: representation. Computers & Industrial Engineering **30**(4), 983–997 (1996)
8. Dai, M., Tang, D., Giret, A., Salido, M.A.: Multi-objective optimization for energy-efficient flexible job shop scheduling problem with transportation constraints. Robotics and Computer-integrated Manufacturing **59**, 143–157 (2019)
9. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. IEEE Transactions on Evolutionary Computation **18**(4), 577–601 (2014)
10. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Transactions on Evolutionary Computation **6**(2), 182–197 (2002)
11. Fattahi, P., Mehrabad, M.S., Jolai, F.: Mathematical modeling and heuristic approaches to flexible job shop scheduling problems. Journal of Intelligent Manufacturing **18**(3), 331–342 (2007)
12. Gen, M., Lin, L.: Multiobjective evolutionary algorithm for manufacturing scheduling problems: state-of-the-art survey. Journal of Intelligent Manufacturing **25**(5), 849–866 (2014)

13. Hajej, Z., Rezg, N., Gharbi, A.: Quality issue in forecasting problem of production and maintenance policy for production unit. International Journal of Production Research **56**, 6147–6163 (2018)

14. Jiang, S., Zheng, Z., Liu, M.: A preference-inspired multi-objective soft scheduling algorithm for the practical steelmaking-continuous casting production. Computers & Industrial Engineering **115**, 582–594 (2018)

15. Kayan, R.K., Akturk, M.S.: A new bounding mechanism for the cnc machine scheduling problems with controllable processing times. European Journal of Operational Research **167**(3), 624–643 (2005)

16. Khouja, M., Mehrez, A.: Economic production lot size model with variable production rate and imperfect quality. Journal of the Operational Research Society **45**, 1405–1417 (1994)

17. Lei, D., Li, M., Wang, L.: A two-phase meta-heuristic for multiobjective flexible job shop scheduling problem with total energy consumption threshold. IEEE Transactions on Systems, Man, and Cybernetics **49**(3), 1097–1109 (2019)

18. Li, J., Pan, Q., Liang, Y.: An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems. Computers & Industrial Engineering **59**(4), 647–662 (2010)

19. Li, J., Pan, Q., Suganthan, P.N., Chua, T.J.: A hybrid tabu search algorithm with an efficient neighborhood structure for the flexible job shop scheduling problem. The International Journal of Advanced Manufacturing Technology **52**(5), 683–697 (2011)

20. Liu, G., Yang, H., Cheng, M.: A three-stage decomposition approach for energy-aware scheduling with processing-time-dependent product quality. International Journal of Production Research **55**(11), 3073–3091 (2017)

21. Lu, C., Li, X., Gao, L., Liao, W., Yi, J.: An effective multi-objective discrete virus optimization algorithm for flexible job-shop scheduling problem with controllable processing times. Computers & Industrial Engineering **104**, 156–174 (2017)

22. Luo, S., Zhang, L., Fan, Y.: Energy-efficient scheduling for multi-objective flexible job shops with variable processing speeds by grey wolf optimization. Journal of Cleaner Production **234**, 1365–1384 (2019)

23. Mansouri, S.A., Aktas, E., Besikci, U.: Green scheduling of a two-machine flowshop: Trade-off between makespan and energy consumption. European Journal of Operational Research **248**(3), 772–788 (2016)

24. Marchi, B., Zanoni, S., Zavanella, L., Jaber, M.Y.: Supply chain models with greenhouse gases emissions, energy usage, imperfect process under different coordination decisions. International Journal of Production Economics **211**, 145–153 (2019)

25. Mokhtari, H., Hasani, A.: An energy-efficient multi-objective optimization for flexible job-shop scheduling problem. Computers & Chemical Engineering **104**, 339–352 (2017)

26. Moslehi, G., Mahnam, M.: A pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search. International Journal of Production Economics **129**(1), 14–22 (2011)

27. Ouaret, S., Kenne, J., Gharbi, A.: Production and replacement policies for a deteriorating manufacturing system under random demand and quality. European Journal of Operational Research **264**(2), 623–636 (2018)

28. Peng, T., Xu, X.: Energy-efficient machining systems: a critical review. The International Journal of Advanced Manufacturing Technology **72**(9), 1389–1406 (2014)

29. Piroozfard, H., Wong, K.Y., Wong, W.P.: Minimizing total carbon footprint and total late work criterion in flexible job shop scheduling by using an improved multi-objective genetic algorithm. Resources Conservation and Recycling **128**, 267–283 (2018)

30. Qin, H., Fan, P., Tang, H., Huang, P., Fang, B., Pan, S.: An effective hybrid discrete grey wolf optimizer for the casting production scheduling problem with multi-objective and multi-constraint. Computers & Industrial Engineering **128**, 458 – 476 (2019)

31. Rezaeimalek, M., Tavakkolimoghaddam, R., Siadat, A., Dantan, J.: A novel model for the integrated planning of part quality inspection and preventive maintenance in a linear-deteriorating serial multi-stage manufacturing system. The International Journal of Advanced Manufacturing Technology **96**, 3633–3650 (2018)

32. Shao, X., Liu, W., Liu, Q., Zhang, C.: Hybrid discrete particle swarm optimization for multi-objective flexible job-shop scheduling problem. International Journal of Advanced Manufacturing Technology **67**, 2885–2901 (2013)

33. Shen, X., Han, Y., Fu, J.: Robustness measures and robust scheduling for multi-objective stochastic flexible job shop scheduling problems. Soft Computing **21**, 6531–6554 (2017)

34. Soto, C., Dorronsoro, B., Fraire, H.H., Cruzreyes, L., Gomezsantillan, C., Rangel, N.: Solving the multi-objective flexible job shop scheduling problem with a novel parallel branch and bound algorithm. Swarm and evolutionary computation **53**, 100632 (2020)

35. Tambe, P.P., Mohite, S., Kulkarni, M.S.: Optimisation of opportunistic maintenance of a multi-component system considering the effect of failures on quality and production schedule: A case study. The International Journal of Advanced Manufacturing Technology **69**(5), 1743–1756 (2013)

36. Tang, H., Chen, R., Li, Y., Peng, Z., Guo, S., Du, Y.: Flexible job-shop scheduling with tolerated time interval and limited starting time interval based on hybrid discrete pso-sa: An application from a casting workshop. Applied Soft Computing **78**, 176 – 194 (2019)

37. Turkyilmaz, A., Şenvar, O., Unal, I., Bulkan, S.: A research survey: heuristic approaches for solving multi objective flexible job shop problems. Journal of Intelligent Manufacturing pp. 1–35 (2020)

38. Uruk, Z., Gultekin, H., Akturk, M.S.: Two-machine flowshop scheduling with flexible operations and controllable processing times. Computers & Operations Research **40**(2), 639–653 (2013)

39. Vilcot, G., Billaut, J.: A tabu search and a genetic algorithm for solving a bicriteria general job shop scheduling problem. European Journal of Operational Research **190**(2), 398–411 (2008)

40. Wu, X., Shen, X., Li, C.: The flexible job-shop scheduling problem considering deterioration effect and energy consumption simultaneously. Computers & Industrial Engineering **135**, 1004–1024 (2019)

41. Xia, W., Wu, Z.: An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems. Comput. Ind. Eng. **48**, 409–425 (2005)

42. Xu, W., Cao, L.: Optimal maintenance control of machine tools for energy efficient manufacturing. The International Journal of Advanced Manufacturing Technology **104**(9), 3303–3311 (2019)

43. Yuan, Y., Xu, H.: Multiobjective flexible job shop scheduling using memetic algorithms. IEEE Transactions on Automation Science and Engineering **12**(1), 336–353 (2015)

44. Zhang, G., Gao, L., Shi, Y.: An effective genetic algorithm for the flexible job-shop scheduling problem. Expert Systems With Applications **38**(4), 3563–3573 (2011)

45. Zhang, G., Shao, X., Li, P., Gao, L.: An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem. Computers & Industrial Engineering **56**(4), 1309–1318 (2009)

46. Zhang, G., Zhang, L., Song, X., Wang, Y., Zhou, C.: A variable neighborhood search based genetic algorithm for flexible job shop scheduling problem. Cluster Computing **22**(5), 11561–11572 (2019)

47. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation **8**(2), 173–195 (2000)

48. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Transactions on Evolutionary Computation **3**(4), 257–271 (1999)