



# Low carbon flexible job shop scheduling problem considering worker learning using a memetic algorithm

Huan Zhu<sup>1</sup> · Qianwang Deng<sup>1</sup> · Like Zhang<sup>1</sup> · Xiang Hu<sup>1</sup> · Wenhui Lin<sup>1</sup>

Received: 13 July 2019 / Revised: 20 February 2020 / Accepted: 20 February 2020 /  
Published online: 27 February 2020  
© Springer Science+Business Media, LLC, part of Springer Nature 2020

## Abstract

Green low carbon flexible job shop problems have been extensively studied in recent decades, while most of them ignore the influence of workers. In this paper, we take workers into account and consider the effects of their learning abilities on the processing time and energy consumption. And then a new low carbon flexible job shop scheduling problem considering worker learning (LFJSP-WL) is investigated. To reduce carbon emission (CE), a novel CE assessment of machines is presented which combines the production scheduling strategies based on worker learning. A memetic algorithm (MA) is tailored to solve the LFJSP-WL with objectives of minimizing the makespan, total CE and total cost of workers. In LFJSP-WL, a three-layer chromosome encoding method is adopted and several approaches considering the problem characteristics are designed in population initialization, crossover and mutation. Besides, four effective neighborhood structures are developed to enhance the exploitation and exploration capacities, and the elite pool strategy is presented to reserve elite solutions along each iteration. The Taguchi method of DOE is used to obtain the best combination of the key parameters used in MA. Computational experiments conducted show that the MA is able to easily obtain better solutions for most of the tested 22 challenging problem instances compared to two other well-known algorithms, demonstrating its superior performance for the proposed LFJSP-WL.

**Keywords** Carbon emission · Flexible job shop scheduling problem · Worker learning · Memetic algorithm

## List of symbols

$n$	Total number of jobs
$m$	Total number of machines
$l$	Total number of workers
$p_i$	Total number of operations of job $i$

✉ Qianwang Deng  
deng\_arbeit@hnu.edu.cn

<sup>1</sup> State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, Hunan University, Changsha 410082, China

$i, h$	Index of jobs, $i, h = 1, 2, \dots, n$
$j, g$	Index of operations
$k, q$	Index of machines, $k, q = 1, 2, \dots, m$
$r$	Index of workers, $r = 1, 2, \dots, l$
$O_{ij}(O_{hg})$	The $j$ th( $g$ th) operation of job $i$ ( $h$ )
$t_{ij}(t_{hg})$	The basic processing time of operation $O_{ij}(O_{hg})$ (s)
$a_{ij}(a_{hg})$	The actual processing time of operation $O_{ij}(O_{hg})$ (s)
$e_{kr}$	The basic efficiency of worker $r$ when operating machine $k$
$e_{ir}$	The basic efficiency of worker $r$ when processing job $i$
$b_r$	The learning coefficient of worker $r$
$d_k(d_q)$	The limiting efficiency of machine $k$ ( $q$ )
$L$	A large enough number
$W_{ijkr}$	The cost of $O_{ij}$ on machine $k$ operated by worker $r$
$S_{ij}$	The starting time of operation $O_{ij}$ (s)
$C_{ij}(C_{hg})$	The completion time of operation $O_{ij}(O_{hg})$ (s)
$C_k$	The completion time of machine $k$ (s)
$CE_i$	The CE of job $i$ (kg CO <sub>2</sub> —e)
$e_k^s$	The CE of machine $k$ during one unit time when in the standby state (kg CO <sub>2</sub> —e)
$CE_k^s$	The CE of machine $k$ when in the stand-by state (kg CO <sub>2</sub> —e)
$CE_i^p$	The CE of job $i$ when in the processing state (kg CO <sub>2</sub> —e)
$CE_i^t$	The CE of job $i$ related to tool wear (kg CO <sub>2</sub> —e)
$CE_i^a$	The CE of job $i$ related to coolant and lubricant oil (kg CO <sub>2</sub> —e)
$V_{ij}$	The removal volume of materials
$Tck$	The mean interval time of the renewal of coolant on machine $k$ (s)
$Ttk$	The life of tool used on machine $k$ before regrinding (s)
$Tlk$	The mean interval time of the renewal of lubricant oil on machine $k$ (s)
$TQ_k^c(TQ_k^l)$	The initial coolant(lubricant oil) quantity of machine $k$ before updating (L)
$TL_k^t$	The energy consumption for regrinding tool once (kJ)
$EF_{ijk}^c(EF_{ijk}^l)$	The emission factor of coolant(lubricant oil) on machine $k$ (kg CO <sub>2</sub> —e/kg)
$EF^e$	The emission factor of electric energy (kg CO <sub>2</sub> —e/kg)
$\mu_{ijk}^a$	The carbon emission coefficient of auxiliary materials on machine $k$ (kg CO <sub>2</sub> —e/s)
$\mu_{ijk}^t$	The carbon emission coefficient of tool on machine $k$ (kg CO <sub>2</sub> —e/s)
$C_{max}$	The max completion time of all machines $k$ ( $q$ ) (s)
$A_{ijkr}$	The cumulative time of worker $r$ operating machine $k$ before $O_{ij}$ (s)
$B_{ijr}$	The cumulative time of worker $r$ processing job $i$ before $O_{ij}$ (s)
$X_{ijkr}$	$X_{ijkr} = 1$ If $O_{ij}$ is processed on machine $k$ operated by worker $r$ ; otherwise, $X_{ijkr} = 0$
$X_{ijk}$	$X_{ijk} = 1$ If $O_{ij}$ is processed on machine $k$ ; otherwise $X_{ijk} = 0$
$X_{igr}$	$X_{igr} = 1$ If $O_{ig}$ is operated by worker $r$ ; otherwise $X_{igr} = 0$

$X_{ij-hg}$	$X_{ij-hg} = 1$ If the completion time of $O_{ij}$ is greater than the starting time of $O_{hg}$ ; otherwise $X_{ij-hg} = 0$
$X_{ig-ij}$	$X_{ig-ij} = 1$ If $O_{ig}$ is processed before $O_{ij}$ ; otherwise $X_{ig-ij} = 0$
$X_{hgk-ijk}$	$X_{hgk-ijk} = 1$ If operation $O_{hg}$ is prior to $O_{ij}$ adjacently on machine $k$ ; otherwise $X_{hgk-ijk} = 0$

## 1 Introduction

The manufacturing industry consumes an enormous amount of energy to generate economic benefits while the increasing demand of energy has imposed tremendous pressure on the environment (Guemri et al. 2014; Sinoquet et al. 2011). For instance, the endothermic greenhouse gases (e.g., CO<sub>2</sub> and N<sub>2</sub>O) have made a great contribution to the global warming (China Energy and CO<sub>2</sub> Emission Research Group 2009). According to the International Energy Outlook 2016 (EIA International Energy Outlook 2016), the industrial sector worldwide would account for more than half of the total energy use through 2040. Furthermore, the energy-related CO<sub>2</sub> emissions around the world rise from 32.2 billion metric tons in 2012 to 35.6 billion metric tons in 2020 and to 43.2 billion metric tons in 2040. Thus, it is a great challenge from the viewpoint of enterprises and countries to decrease the ultimate achievement of emissions targets, which will depend on how to balance environmental goals with economic growth.

The CE assessment of machines and the advanced production scheduling play a significant role in balancing the economic benefits with the environmental impact (Ballestin et al. 2012; Kelly and Zyngier 2017; May et al. 2016; Piroozfard et al. 2018; Schinko et al. 2014). An empirical model based on power measurements was developed by Li and Kara (2011) who optimized the energy efficiency of processes under different cutting conditions by an effective method. Laurent et al. (2010) and Kara et al. (2010) studied the energy efficiency and the CE respectively in a manufacturing system by using life cycle assessment (LCA). Zheng and Wang (2015) investigated the low-carbon production scheduling problem with a Pareto-based estimation of distribution algorithm, concerning the objectives of makespan and CE. Liu and Huang (2014) introduced two scheduling problems involving two objectives (i.e. reduction of power consumption and carbon footprints) on batch-processing machines and a hybrid flow shop respectively. NSGA-II was used to solve their problems and an adaptive multi-objective GA is proposed to validate the results.

Flexible job shop problem (FJSP) is an extension of the job shop problem (JSP) that considers the flexibility of machines, and has been extensively existed in modern manufacture. With the development of customized manufacturing, workers with specific capacity are needed on various machines. To make the scheduling model closer to realistic production, it is essential to take worker resources into consideration. Fryer (2010) presented a dual-constraint JSP problem by considering the constraints of machines and workers simultaneously, where each worker was assumed with equal flexibility. Felan and Fry (2001) introduced the flexibility of workers for the same problem, and concluded that workers with different flexibility perform

better than all workers with equal flexibility. A variable neighborhood search was proposed by Lei and Guo (2014) in order to minimize the makespan of FJSP with worker flexibility. Xu et al. (2011) pointed out that workers supposed to be trained in several skills or departments such that they can be assigned to a variety of jobs as the need arises. Zheng and Wang (2016) investigated a FJSP that workers have different skills to operate different machines, and developed a knowledge-guided fruit fly optimization algorithm to solve their problem. A FJSP considering different labor wages under dynamic electricity pricing is solved by the non-dominated sorting genetic algorithm-III (NSGA-III) (Gong et al. 2019b). Ferjani et al. (2017) studied the dynamic multi-skilled workers assignment problem and considered the impact of fatigue.

Analyzing the literatures above, most of them studying the constraints imposed by workers assume that all workers have constant efficiency throughout the whole production process. From a realistic perspective, workers ought to have learning abilities that the efficiency of a worker is supposed to be meliorated with the accumulation operation time towards corresponding jobs or machines. This learning effect has been proved to exist in many empirical studies. The effects of learning in scheduling problems were considered for the first time by Biskup (1999). Ostermeier (2019) pointed out that workers are prone to learning and forgetting even in the short-term when they are working in a highly-manual production environment where several complex products are processed alternately, frequent new product introductions occur and a high degree of product variation and customisation is presented. Koulamas and Kyparisis (2007) presented a new learning effect for scheduling environments and derived some results on special cases for the two-machine flow-shop problem with position-based learning effect. Biskup (2008) gave a review on scheduling with learning effects and discussed the existing literature on scheduling with position-based learning effect and sum-of-processing-time based learning effect, respectively. More recently, Wu et al. (2018) investigated the FJSP with consideration of worker's learning ability and put forward a hybrid genetic algorithm to solve their problem, whereas they only optimized the maximum completion time (economic benefits), and ignored the environmental impact. This paper aims to further narrow the gap between the scheduling models and the realistic production while balancing the economic benefits and the environmental impact.

In this paper, we proposed a low carbon flexible job shop scheduling problem considering worker learning (LFJSP-WL) with reference to the CE reduction, the makespan and the total cost of workers. Since the JSP has been proved to be strong NP-hard combinatorial optimization problems due to its high computational complexity (Brucker and Schlie 1990; Garey et al. 1976; Zhou et al. 2009), the FJSP with worker learning must be more complex than JSP. Thus it is quite difficult for traditional approaches to obtain an optimal solution within a certain time.

For the multi-objective FJSP problem, the memetic algorithm (MA) hybridizing evolutionary algorithms with local search has shown promising performance (Kannan and Ramaraj 2010). In addition, MA takes advantage of both global and local search. Thereby, it is able to strike a better balance between exploration and intensification (Hertz and Kobler 2000) and it has been proved to be an effective method for FJSP (Frutos et al. 2010; Gong et al. 2019a; Ma et al. 2014a, 2014b; Phu-Ang

and Thammano 2017; Yi et al. 2016; Yuan and Xu 2013). In order to solve the new mathematical model (LFJSP-WL) efficiently, we put forward a memetic algorithm based on NSGA-II with a useful variable neighborhood search (VNS), in which a three-layer chromosome representation encoding method and an active schedule decoding method are proposed. Besides, we present a mixed initialize method which consists of three initialize strategies, and apply several useful mutation operators and crossover operators for the three subproblems respectively. The elite pool strategy is adopted to reserve elite solutions generated along with all iterations.

The rest of this paper is organized as follows. The mathematical model of the proposed LDFJSP-LE is introduced in Sect. 2. The memetic algorithm (MA) is elaborated in Sect. 3. The case study is presented in Sect. 4. Conclusions and future studies are described in Sect. 5.

## 2 The mathematical model of the proposed LDFJSP-LE

### 2.1 Problem description

The problem of LFJSP-WL is described as follows. It considers processing  $n$  independent jobs  $J=(J_1, J_2, \dots, J_n)$  on  $m$  machines  $M=(M_1, M_2, \dots, M_m)$  with  $l$  workers  $W=(W_1, W_2, \dots, W_l)$ . A job  $J_i$  consists of a predefined sequence of  $r_i$  operations  $(O_{i1}, O_{i2}, \dots, O_{ir})$ . Each  $O_{ij}$  indicates that the  $j$ th operation of  $J_i$  can be processed on a machine out of from the given machine set  $M$ . Each machine can be performed by at least one worker out of from the given worker set  $W$ . As regards worker, there are two different types (ElMaraghy et al. 1999), one supposed that every worker owns the same capacity, the other considered different workers have diverse skills and operate machines with varying efficiencies. The latter type is adopted in LFJSP-WL, furthermore, the skill level and learning ability of workers are taken into account, a worker with higher-level should have preferable learning parameters and higher-cost. Owing to workers' learning capacity, the efficiency of workers will be enhanced since the accumulation of processing time when performing different machines and jobs, i.e. worker's learning ability for machines and jobs. Nonetheless, efficiency will reach the limitation in spite of the study time increasing continuously. Three objectives, productivity (i.e. makespan), environmental impact (i.e. total CE) and economic influence (i.e. total cost of workers), are optimized simultaneously, which can provide guidance to production managers when they make scheduling decisions considering profit and environment protection. The model consists of three subproblems: (1) operation sequence; (2) machine assignment; (3) worker assignment.

To make it easier to understand, an instance of the proposed LFJSP-WL is shown in Table 1. There are two jobs, four machines and five workers. Job 1, 2 have two and three operations respectively. Machine 1, 2, 3, 4 can be operated by two, two, three and three corresponding workers chosen from the worker set with three different skill levels, the level classification and some learning parameters are organized in Table 2. For each column in Table 1. Column 1 indicates jobs. Column 2 shows the operations belonging to a certain job. Column 3 states the machines which can process the corresponding operations. Column 4 represents the workers which can

**Table 1** An instance of LFJSP-WL

Job	Operation	Machine	Worker	Cost of worker	Basic time	SEC	$\mu^l$	$\mu^a$	$P_s$	$M_e$
Job1	$O_{11}$	$M_1$	$W_1$	110	15	5.49	0.082	0.092	3.77	0.56
			$W_2$	95	12					
		$M_2$	$W_2$	100	5	4.26	0.016	0.063	2.26	0.60
	$O_{12}$	$M_3$	$W_4$	80	6	3.94	0.075	0.070	4.23	0.58
			$W_2$	98	20					
			$W_3$	90	16					
		$M_4$	$W_4$	85	18	4.80	0.031	0.059	4.76	0.63
			$W_1$	115	15					
			$W_3$	95	14					
			$W_5$	79	9					
Job2	$O_{21}$	$M_2$	$W_2$	93	10	5.45	0.078	0.063	2.26	0.60
			$W_4$	86	8					
		$M_3$	$W_2$	89	5	3.46	0.056	0.060	4.23	0.58
			$W_3$	97	10					
			$W_4$	75	14					
		$M_4$	$W_1$	105	10	2.96	0.046	0.089	4.76	0.63
			$W_3$	90	14					
	$W_5$		86	18						
	$W_4$		75	14						
	$O_{22}$	$M_1$	$W_1$	116	15	4.26	0.070	0.092	3.77	0.56
			$W_2$	96	12					
		$M_4$	$W_1$	109	20	5.07	0.039	0.059	4.76	0.63
			$W_3$	90	15					
			$W_5$	78	18					
	$O_{23}$	$M_1$	$W_1$	105	10	3.25	0.048	0.092	3.77	0.56
			$W_2$	95	9					
		$M_2$	$W_2$	98	15	5.24	0.056	0.063	2.26	0.60
$W_4$			85	14						
$W_2$			90	5	4.35					
$W_3$		89	6							
$W_4$		80	8							

**Table 2** Parameters of the workers with different levels

Worker	$W_1$	$W_2$	$W_3$	$W_4$	$W_5$
Level	1	2	2	3	3
Basic efficiency ( $B_e$ )	0.85	0.95	0.90	1.05	1.10
Learning coefficient ( $L_c$ )	-0.25	-0.18	-0.20	-0.12	-0.10

operate the corresponding machines. Column 5 lists the homologous cost of workers. Column 6 expounds the basic processing time before considering the improvement of worker’s efficiency. Especially, the actual time will be changed compared to the basic time with the accumulation processing time of the worker. Column

7–11 addresses the special energy consumption, the CE coefficient of cutting tools, the CE coefficient of auxiliary materials, the CE of machine tool at the stand-by state and the limiting efficiency, those mentioned parameters are related to corresponding machines respectively. For each row in Table 2. The level classification, basic efficiency and learning coefficient of diverse workers are listed where Level 1, 2, 3 represents the best-skilled worker set, the average-skilled worker set and the worst-skilled worker set. Take the first operation of job1 for example. There are two machines ( $M_1, M_2$ ) for this operation,  $M_1$  can be operated by  $W_1, W_2$  while  $M_2$  can be operated by  $W_2, W_4$ . The cost of the worker on  $M_1$  is 110 and 95 respectively. Similarly, the cost of the worker on  $M_2$  considering the skill level is 100 and 80 respectively, and corresponding basic time is 15, 12, 5 and 6 time units, the other variable values can be obtained in the same way.

Some assumptions are made to simplify our model. based on the assumptions which have been widely proposed in many other papers on FJSP (Demir and İşleyen 2014; Lei and Guo 2014; Li et al. 2012; Zhang et al. 2016), we further consider the worker resources and their learning ability, which indicates that our model is more close to reality, especially in a highly-manual production environment. The assumptions made for LDFJSP-LE are listed as follows:

All machines and workers are ready at time 0;

- (1) All raw materials are ready at time 0;
- (2) Jobs are mutually independent and have the same priority;
- (3) The efficiency of various workers only concerned with the accumulation of processing time;
- (4) Each machine can be operated by one and only one worker at a time;
- (5) Each operation can be processed by one and only one machine at a time;
- (6) An operation cannot be performed until its preceding operation completed;
- (7) Interruption is not allowed once the operation has started;
- (8) Workers with higher skill level rank supposed to be owned better parameters related to learning effect and cause the higher cost;
- (9) Setting up time of machines, transportation time among different positions and released time of jobs are negligible.

### 2.2 Problem formulation

Refer to Wu et al. (2018) and Zhang et al. (2015), the mathematical model can be established as follows.

$$A_{ijk} = \sum_{h=1}^n \sum_{g=1}^{p_h} a_{hg} \cdot X_{hg-ij} \cdot X_{hgkr} \tag{1}$$

$$B_{ijr} = \sum_{g=1}^{p_i} a_{ig} \cdot X_{ig-ij} \cdot X_{igr} \tag{2}$$

$$a_{ij} = t_{ij} \cdot \max(e_{kr} \cdot (1 + A_{ijk})^{b_r}, e_{ir} \cdot (1 + B_{ijr})^{b_r}, d_k) \tag{3}$$

Through Eqs. (1)–(3), the actual processing time of  $O_{ij}$  processed by worker  $r$  on machine  $k$  can be obtained. Owing to the differences between the accumulation time of worker  $r$  operating machine  $k$  ( $A_{ijk}$ ) and the accumulation working time of worker  $r$  processing job  $i$  ( $B_{ijr}$ ) before performing  $O_{ij}$ , the degree of familiarity with the job and the machine may be different for worker  $r$ , indicating that a worker’s efficiency is decided by the maximum of the two parts. According to the learning effect model (Kuo and Yang 2006), we build the efficiency model of each worker. The  $e_{kr} \cdot (1 + A_{ijk})^{b_r}$  indicates the efficiency of worker  $r$  towards machine  $k$  when considering learning effect, while the  $e_{ir} \cdot (1 + B_{ijr})^{b_r}$  indicates the efficiency of worker  $r$  for processing job  $i$  when considering the learning effect. It is worth to note that the bigger the value of a worker’s efficiency is, the longer time needs for the worker to perform a processing activity. Furthermore, each machine has its ultimate efficiency  $d_k$ , thus we calculate the final efficiency by  $\max(e_{kr} \cdot (1 + A_{ijk})^{b_r}, e_{ir} \cdot (1 + B_{ijr})^{b_r}, d_k)$ . Then the final actual processing time can be obtained via multiplying final efficiency by basic processing time  $t_{ij}$ .

$$CE_i = CE_i^p + CE_i^t + CE_i^a \tag{4}$$

Equation (4) shows the total carbon emission of job  $i$  which involves machining processes, tool wear and auxiliary materials.

$$SEC = C_0 + \frac{C_1}{MRR} \tag{5}$$

$$CE_i^p = \left( \sum_{j=1}^{p_i} \sum_{k=1}^m V_{ij} \cdot SEC_{ijk} \cdot X_{ijk} \right) \cdot EF^e \tag{6}$$

Equation (5) is the calculation of the special energy consumption (SEC) for removing 1 cm<sup>3</sup> material during processing (Kara and Li 2011), where MRR denotes the material removal rate, and  $C_0$  and  $C_1$  are special coefficients. Based on Eq. (5), the material removal energy can be calculated via multiplying SEC by the material removal volume ( $V_{ij}$ ) (Zhang et al. (2016)). Then the total CE of job  $i$  in the processing state can be calculated by Eq. (6) (Zhang et al. 2015), where  $X_{ijk}$  ensures that operation  $O_{ij}$  is performed on machine  $k$ , and  $EF^e$  represents the emission factor of electric energy (kg CO<sub>2</sub> – e/kg).

$$\begin{aligned} CE_i^t &= \sum_{j=1}^{p_i} \sum_{k=1}^m \frac{a_{ij}}{T_k^t} \cdot TL_k^t \cdot EF^e \cdot X_{ijk} \\ &= \sum_{j=1}^{p_i} \sum_{k=1}^m a_{ij} \left( \frac{TL_k^t \cdot EF^e \cdot X_{ijk}}{T_k^t} \right) \\ &= \sum_{j=1}^{p_i} \sum_{k=1}^m a_{ij} \cdot \mu'_{ijk} \end{aligned} \tag{7}$$



Tool wear is a crucial factor which usually appears during the processing in workshops. By regrinding, a cutting tool can get recovery after reaching the life limitation, but the CE (carbon emission) of regrinding cutting tool should be considered. According to Zhang et al. (2015), the CE of job  $i$  related to the cutting tool can be calculated by Eq. (7).

$$\begin{aligned}
 CE_i^a &= \sum_{j=1}^{p_i} \sum_{k=1}^m \left( \frac{a_{ij}}{T_k^c} \cdot TQ_k^c \cdot EF_{ijk}^c \cdot X_{ijk} + \frac{a_{ij}}{T_k^l} \cdot TQ_k^l \cdot EF_{ijk}^l \cdot X_{ijk} \right) \\
 &= \sum_{j=1}^{p_i} \sum_{k=1}^m a_{ij} \cdot X_{ijk} \left( \frac{TQ_k^c \cdot EF_{ijk}^c}{T_k^c} + \frac{TQ_k^l \cdot EF_{ijk}^l}{T_k^l} \right) \tag{8} \\
 &= \sum_{j=1}^{p_i} \sum_{k=1}^m a_{ij} \cdot \mu_{ijk}^a
 \end{aligned}$$

In addition to the resources mentioned above, the machine tool consumes a certain amount of auxiliary materials, particularly the coolant and lubricant oil. Therefore, coolant and lubricant oil are considered in this model. Since they will decrease bit by bit with the machine tool processing, the supplement actions should be taken after a fixed interval time. According to Zhang et al. (2015), the mathematical derivation of Eq. (8) is similar to Eq. (7). Besides, two types of CE are included: CE of coolant and CE of lubricant oil.

$$CE_k^s = \left( C_k - \sum_{i=1}^n \sum_{j=1}^{p_i} a_{ij} \cdot X_{ijk} \right) \cdot e_k^s \tag{9}$$

Equation (9) obtains the CE of machine tool at standby state, which is calculated by multiplying total standby time with CE per unit time during the standby state.

In this model, we optimize three multiple conflicting objectives simultaneously. After the analysis of Eqs. (1)-(9), the objectives are presented as below:

$$\text{Minimize } C_{max} = \max_{1 \leq k \leq m} \{ C_k \} \tag{10}$$

$$\text{Minimize } W = \sum_{i=1}^n \sum_{j=1}^{p_i} \sum_{k=1}^m \sum_{r=1}^l W_{ijk r} X_{ijk r} \tag{11}$$

$$\text{Minimize } CE_{total} = \sum_{i=1}^n CE_i + \sum_{k=1}^m CE_k^s \tag{12}$$

Subject to:

$$C_{ij} = S_{ij} + a_{ij} \tag{13}$$

$$C_{i(j+1)} - C_{ij} \geq a_{i(j+1)} \quad (14)$$

$$S_{ij} + (1 - X_{hgk-ijk}) \cdot L \geq C_{hg} \quad (15)$$

$$(C_{hg} - C_{ij} - a_{hg}) \cdot (C_{ij} - C_{hg} - a_{ij}) \cdot X_{ijk} \cdot X_{hgk} \leq 0 \quad (16)$$

$$(C_{hg} - C_{ij} - a_{hg}) \cdot (C_{ij} - C_{hg} - a_{ij}) \cdot X_{ijkr} \cdot X_{hgqr} \leq 0 \quad (17)$$

$$\sum_{k=1}^m X_{ijk} = 1 \quad (18)$$

$$\sum_{r=1}^l X_{ijkr} = 1 \quad (19)$$

Equation (10) represents the makespan objective. Equation (11) represents the total cost of workers objective. Equation (12) represents the total carbon emission objective. Constraint (13) explains the relation between completion time, starting time, and actual processing time of operation  $O_{ij}$ . Constraint (14) ensures the precedence constraint between operations of a job. Constraint (15) ensures that each operation cannot start until the immediate precedence operation completed on the same machine. Constraint (16) ensures that each machine can process at most one operation at a time. Constraint (17) ensures that each worker can process exactly one operation at a time. Constraint (18) ensures that each operation can be processed on exactly one machine. Constraint (19) ensures that each operation on a machine can be performed by one and only one worker.

### 3 Proposed MA for LDFJSP-LE

#### 3.1 The framework

In this section, we propose a memetic algorithm with variable neighborhood structures (MA) to solve LFJSP-WL. The genetic operators in MA include crossover operator, mutation operator and the local search which can improve the quality of solutions effectively. Furthermore, to retain the good solutions produced in each iteration, the elite pool strategy is adopted during evolution. The pseudo-code of MA is presented as below.

The pseudo-code of MA is as follows.

```

Set parameters in MA, record the current iteration as  $i$ 
initialize population  $R(i)$  with  $\alpha$  individuals, update the elite pool( $e_p$ )
while  $i < L$  do
  if  $p_c > \text{rand}$ 
    crossover,  $R(i)=[R(i), R'(i)]$ , if  $R'(i)$  consist of  $\beta(\beta < \alpha)$  individuals,  $(\alpha - \beta)$  new individuals are generated
  end if
  if  $p_m > \text{rand}$ 
    mutation,  $R(i)=[R(i), R''(i)]$ , if  $R''(i)$  consist of  $\beta(\beta < \alpha)$  individuals,  $(\alpha - \beta)$  new individuals are generated
  end if
  select  $p\%$  individuals randomly to do neighborhood search
  execute the non-dominated sorting, get the rank 1 Pareto solutions ( $pI$ )
  calculate the crowding distance
  get a mixture of  $e_p$  and  $pI$ , execute non-dominated sorting then update the elite pool( $e_p$ )
  obtain the offspring population  $R(i)$ 
end while
    
```

$p_c$ : the probability of crossover;  $p_m$ : the probability of mutation;  $L$ : the total iterations;

### 3.2 Encoding and decoding

A three-layer chromosome representation that contains operation sequence (OS) vector, machine assignment (MA) vector and worker assignment (WA) vector is adopted for this research (Gong et al. (2018a)). The length of the chromosome equals to  $3 * p$  where  $p$  stands for the total number of operations of all jobs. The OS vector consists of  $p$  integer values in the range of 1 to  $n$ , where  $n$  represents the total number of jobs, and different integer values which appear  $p_i$  times indicate the corresponding job-number. By scanning the OS vector from left to right, the  $j$ th occurrence of an integer value indicates the  $j$ th operation of the corresponding jobs. An example is presented in Fig. 1a, the first position shows the first operation of job 2, namely,  $O_{21}$ , and the fifth position shows  $O_{22}$ . An example of the MA vector is presented in Fig. 1b, the MA vector is as long as the OS vector which consists of three

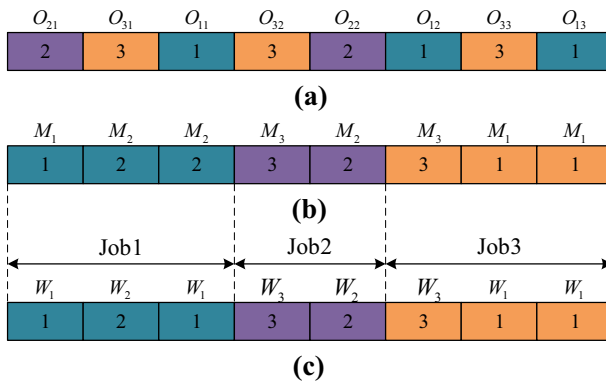


Fig. 1 Individual encoding example

parts owing to three jobs. The number in each part represents the machine-number selected for the corresponding operation. Number 1 in the first position of the first part represents machine 1 is selected for operation  $O_{11}$ . Number 2 in the next position of the first part represents machine 2 is selected for operation  $O_{22}$ . Number 2 in the last position of the first part represents machine 2 is selected for operation  $O_{13}$ . An example of the WA vector is shown in Fig. 1c, the length of WA vector also equals to OS vector and consists of three parts. The number in each part represents the worker-number selected for the corresponding operation. Number 3 in the first position of the second part represents worker 3 is selected for operation  $O_{21}$ . Number 2 in the next position of the second part represents worker 2 is selected for operation  $O_{22}$ . The example shown in Fig. 1 can be interpreted as follows:  $(O_{21}, M_3, W_3)$ ,  $(O_{31}, M_3, W_3)$ ,  $(O_{11}, M_1, W_1)$ ,  $(O_{32}, M_1, W_1)$ ,  $(O_{22}, M_2, W_2)$ ,  $(O_{12}, M_2, W_2)$ ,  $(O_{33}, M_1, W_1)$ ,  $(O_{13}, M_2, W_1)$ .

For the above chromosome representation, an active decoding method referring to Gong et al. (2019b) which considers the constraints of machines and workers simultaneously is applied for this research. The details of the decoding method are as follows:

*Step 1* Extract operation ( $O_{ij}$ ) from OS vector in sequence from left to right, then get the corresponding machine ( $M_k$ ) and worker ( $W_r$ );

*Step 2* Get the worker's ( $W_r$ ) accumulation processing time of  $M_k$  and job  $i$  before processing  $O_{ij}$  respectively. Next, calculate the actual processing time of  $O_{ij}$  by combining worker's learning parameters;

*Step 3* Find the blank spaces, namely, the idle time between adjacent operations which processed prior to  $O_{ij}$ , from left to right. Then obtain the blank space  $[T_{ks}, T_{ke}]$  existing on  $M_k$  and  $[T_{rs}, T_{re}]$  existing on  $W_r$ . Additionally, getting the completion time of  $O_{i(j-1)}$ , namely,  $C_{i(j-1)}$ ;

*Step 4* Get all the overlapping parts  $[T_{is}, T_{ie}]$  between  $[T_{ks}, T_{ke}]$  and  $[T_{rs}, T_{re}]$ . If  $T_{ie} - \max(T_{is}, C_{i(j-1)}) \geq a_{ij}$ , and the earliest processing time  $O_{ij}$  can be obtained by  $t_s = \max(T_{is}, C_{i(j-1)})$ . The completion time of  $O_{ij}$  can be calculated by  $t_e = t_s + a_{ij}$ . Otherwise, go to step 5;

*Step 5* If  $O_{ij}$  cannot insert into the idle period, the earliest processing time  $O_{ij}$  can be calculated by  $t_s = \max(M_{kc}, W_{rc})$ , where  $M_{kc}$  represents the completion time of the last operation processed on machine  $k$ ,  $W_{rc}$  represents the completion time of the last operation performed by worker  $r$ . The completion time of  $O_{ij}$  can be obtained by  $t_e = t_s + a_{ij}$ ;

*Step 6* If all operations in OS vector are scheduled, go to the end. Otherwise, go to step 1.

### 3.3 Initialization of population

The population initialization plays a vital role in improving the quality and diversity of the population. Since the problem consists of three subproblems, we employ three initial strategies for the three corresponding subproblems respectively. The OS vectors are generated by using three strategies: (1) Sequence the operations randomly; (2) A job with the longest remaining processing time has

the priority to be selected; (3) Choose a job with the maximum rest operations. The MA vectors also produced by three methods: (1) Select a machine out of alternative machines randomly; (2) Choose a machine with the lowest global processing time; (3) Give preference to the machine with the smallest *SEC*. Similarly, we initial the WA vectors with three strategies: (1) Select a worker out of alternative workers randomly; (2) Choose a worker with the lowest global processing time; (3) A worker with the lowest cost has the prior to be chosen.

### 3.4 Crossover operator

Crossover operator is an indispensable element during the reproduction phase which generates new individuals (offspring) by exchanging good genes between two different parental chromosomes. Two different crossover operators are employed respectively for this problem. The job-based crossover (JBX) operator is proposed for OS. First, we divide the *Jobset J* into two subsets *Jobset1* and *Jobset2* randomly; Next copy any genes in parent 1 belongs to *Jobset1* into offspring 1 in the same position and copy any genes in parent 2 belongs to *Jobset2* into offspring 2 in the same position; Finally, add any genes in parent 2 belongs to *Jobset2* into offspring 1 in the remain position one by one, and add any genes in parent 1 belongs to *Jobset1* into offspring 2 in the remain position one by one. An example is shown in Fig. 2. For MA and WA, the random probability crossover (RPX) operator is proposed for this problem. The theory is similar to JBX and the main difference is that a binary vector unit is used instead of subsets, where the 0–1 expresses the two different subsets. An example of MA is shown in Fig. 3.

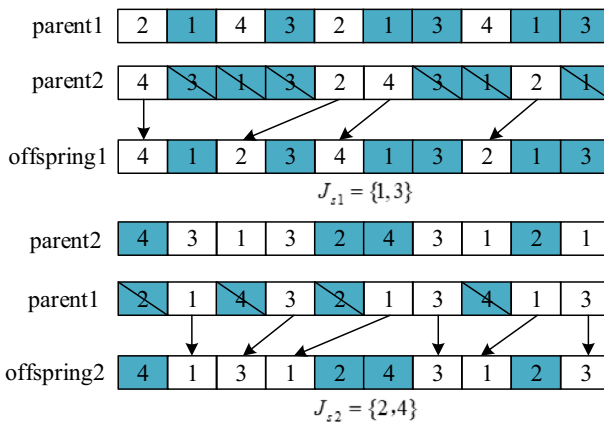


Fig. 2 OS job-based crossover operator

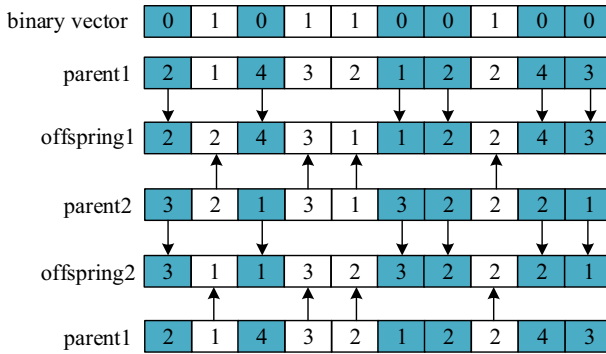


Fig. 3 MA random probability crossover operator

### 3.5 Mutation operator

Mutation operator can provide extra variation into the population so that enhances the diversity of the population. Three mutation operators are constructed for this problem respectively. The procedures for the mutation are depicted as follows and an example of OS is shown in Fig. 4.

*Step 1* Generate a random number  $R_m$ . If  $R_m < M_r$ , where  $M_r$  states the mutation rate, go to Step 2, else go to the end;

*Step 2* Get two different positions *point1*, *point2* in the OS vector randomly, and generate two random position numbers  $m_r$ ,  $w_r$  in the corresponding MA vector, WA vector respectively, then go to Step 3;

*Step 3* For OS vector, insert the gene in *point1* into the position before *point2*, as shown in Fig. 4, then go to step 4;

*Step 4* For MA vector, replace the machine in position  $m_r$  with another machine randomly amongst the alternative machines and choose an available worker randomly, then go to step 5;

*Step 5* For WA vector, replace the worker in position  $w_r$  with another worker randomly amongst the alternative workers.

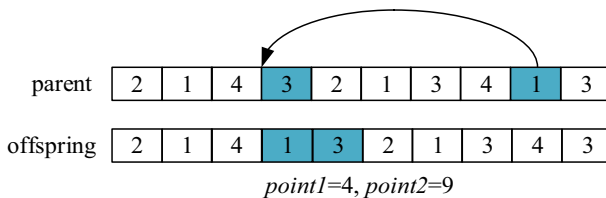


Fig. 4 OS mutation operator

### 3.6 Local search with four neighborhood structures

The neighborhood structure is a mechanism that can produce neighbor solutions by utilizing a small change in the original solutions. Inspired by Lei and Guo (2014), we construct four neighborhood structures as follows:

- Neighborhood structure 1( $N_1$ ):  $N_1$  is designed for the OS vector. It first gets a critical block with more than two operations, then obtains the first operation in this critical block, finally moves an internal operation which differs from the first operation to the beginning of the block.
- Neighborhood structure 2( $N_2$ ):  $N_2$  is also designed for the OS vector. It first obtains a critical block with more than two operations, next get the last operation in this critical block. Then moves an internal operation which differs from the last operation to the rear of the block.
- Neighborhood structure 3( $N_3$ ):  $N_3$  is designed for the MA vector. It first selects a gene in MA vector randomly, then selects a new machine with the smallest *SEC* from the alternative machines, and lastly changes the worker randomly for the corresponding new machine.
- Neighborhood structure 4( $N_4$ ):  $N_4$  is designed for the WA vector. Similarly, it obtains a worker gene  $w_k$  in WA vector randomly, then replaces  $w_k$  with the lowest cost out of from alternative workers.

The pseudo-code of the variable neighborhood search is as follows.

---

```

Set parameters in neighborhood search, get an individual  $x$  randomly
get the corresponding chromosome  $x$ , initialize  $a=0$ ,  $b=0$ ,  $c=0$ ,  $d=0$ ,  $e=0$ ;
if  $a=0$  or ( $a=1$  and  $b=1$  and  $c=1$  and  $d=1$ )
    produce a new chromosome  $y$  by applying  $N_1$  and evaluate the new chromosome; if  $y$  is better than  $x$ , let
         $x=y$ ,  $a=0$ ,  $e=0$ ; else,  $a=1-a$ ,  $e=e+1$ ;
if  $b=0$ 
    use  $N_2$  to update the chromosome  $y$  and evaluate it; if  $y$  is better than  $x$ , let  $x=y$ ,  $b=0$ ,  $e=0$ ; else,  $b=1-b$ ,
         $e=e+1$ ;
if  $c=0$ 
    generate new chromosome  $y$  by using  $N_3$  and evaluate the new chromosome; if  $y$  is better than  $x$ , let  $x=y$ ,
         $c=0$ ,  $e=0$ ; else,  $c=1-c$ ,  $e=e+1$ ;
if  $d=0$ 
    produce a new chromosome  $y$  based on the given chromosome by applying  $N_4$  and evaluate it; if  $y$  is
        better than  $x$ , let  $x=y$ ,  $d=0$ ,  $e=0$ ; else,  $d=1-d$ ,  $e=e+1$ ;
if  $e > t$ , terminated;

```

**End For**

Return the new chromosome

---

$a, b, c, d, e, t$ : the parameters of neighborhood search

### 3.7 Elite pool

To reserve elite solutions along with each iteration, the notion of elite pool is used in the proposed MA. The elite pool ( $e_p$ ) initially filled with a number of solutions generated by initialization. Then at each iteration obtains the non-dominated solutions ( $n_s$ ) produced by non-dominated sorting, and updates the elite pool finally through doing non-dominated sorting amongst the mixture of  $n_s$  and  $e_p$ .

The pseudo-code of updating the elite pool is as follows.

---

Get the  $n_s$  and previous  $e_p$

let the mixture of  $n_s$  and  $e_p$  do the non-dominated sorting, get the number of solutions ( $Sl_i$ ) in rank  $i$

**For**  $i = 1 \rightarrow r_n$

if  $Sl_i \geq se$  and  $Sl_{i-1} < se$

put all solutions in rank 1 into the elite pool

fill the elite pool using solutions in rank  $i$  according to the crowding distances

end if

**End For**

Return the elite pool

---

$r_n$  : the number of rank

$Sl_i$  : the number of solutions in rank  $i$

$se$  : the size of the elite pool

## 4 Case study

### 4.1 Performance evaluate metrics

In order to evaluate the effectiveness of MA, three commonly used metrics are adopted. They are the Set Coverage ( $SC$ ), the inverted generational distance ( $IGD$ ) and the ratio of non-dominated solutions ( $R_{nd}$ ) (Yuan and Hua 2014; Wu and Che 2019), which can be calculated as follows.

1.  $C(A, B)$  measures the fractions of members of  $B$  which are dominated by members of  $A$ , it can be obtained by Eq. (20).

$$C(A, B) = \frac{|\{b \in B | \exists a \in A : a \text{ dominates } b\}|}{|B|} \quad (20)$$

where  $A$  and  $B$  denote the approximation to Pareto front (PF) respectively,  $|B|$  represents the number of solutions in  $B$ .  $C(A, B) = 1$  indicates that all solutions in  $B$  are dominated by at least one solution in  $A$ , and  $C(A, B) = 0$  shows that all solutions in  $B$  are not dominated by any solution in  $A$ . If  $C(A, B)$  is larger than  $C(B, A)$ , that means the solutions in  $A$  is better than solutions in  $B$  in a sense.

2. Let  $P^*$  denotes the set of non-dominated solutions amongst all solutions obtained by all runs of all algorithms, the  $IGD$  of set  $A$  can be calculated by Eq. (21).



$$IGD(A, P^*) = \frac{1}{|P^*|} \sum_{x \in P^*} \min_{y \in A} d(x, y) \tag{21}$$

where  $d(x, y)$  shows the Euclidean distance between point  $x$  and  $y$ ,  $|P^*|$  represents the size of  $P^*$ . The smaller value of  $IGD(A, P^*)$  states that set  $A$  is closer to  $P^*$ . In addition, the objectives of every non-dominated solutions should be normalized by Eq. (22) before calculating  $IGD$ .

$$\tilde{f}_i(x) = (f_i(x) - f_i^{\min}) / (f_i^{\max} - f_i^{\min}), i = 1, 2, 3 \tag{22}$$

where  $f_i^{\min}$  is the minimal value of the  $i$ th objective amongst all obtained solutions, besides,  $f_i^{\max}$  states the maximum.

3. Let  $N$  denotes the non-dominated solutions obtained by one algorithm, and the  $N_{nd}$  of the algorithm shows the number of solutions in  $N$  that remains in  $P^*$ .

$$R_{nd} = N_{nd} / |P^*| \tag{23}$$

where the definition of  $P^*$  is identical with Eq. (21).  $R_{nd}=0$  expresses that solutions obtained by the algorithm all dominated by those solutions obtained by other algorithms.  $R_{nd}=1$  represents that any solution obtained by the algorithm cannot be dominated by any other solutions obtained by other algorithms.

## 4.2 Design of experiments

### 4.2.1 Experiment setting

All the algorithms are coded in MATLAB R2016 and implemented on a desktop computer with an IntelCorei3 CPU with 3.60 GHz frequency and 4 GB RAM.

### 4.2.2 Data source

The flexible job-shop scheduling problem (FJSP) is commonly seen in the real manufacturing industry. A reasonable FJSP solution can effectively help manufacturers to improve production efficiency and reduce production costs. Some benchmarks have been proposed by researchers based on realistic production conditions, where consist of many actual production problems that plague the enterprises (e.g. maximal completion time of machines, workload of the most loaded machine, total workload of all machines etc.). Although the benchmarks contain many practical problems, the benchmarks can also be expanded easily to use if some new factors or actual production targets that have not been considered before appear, (e.g. Demir and İşleyen 2014; Kacem et al. 2014; Zheng and Wang 2016; Gong et al. 2018b; Gong et al. 2019a).

In our problem, we consider the objectives of minimizing the makespan, total CE and total cost of workers, which are related to the interests of the companies, especially when they make scheduling decisions considering profit and environment protection. We construct our benchmarks for testing LFJSP-WL extended from

Dauzèreperès and Paulli (1997) and Brandimarte (1993), both of them are widely used for FJSP. For the sake of simplicity, the original names of these benchmarks are used in this paper. Some supplementary data for extension are shown in Table 3. The intervals in rows 1–4 may be different in different levels (worker set), and the value of variables in rows 5–8 are related to diverse machines. Besides, the removal volume of materials and CE coefficient of the tool are given in the last two rows respectively, and the values are related to various operations. All the parameters in column 1 mentioned in Table 3 are distributed according to the given intervals and distribution mode (normal distribution and uniform distribution). The total number of workers is generated by  $[m \times 0.6]$  while the partial worker flexible is considered in this paper, namely, some machines cannot be operated by an apart of workers.

The emission factors used in this paper can be obtained according to 2050 China energy and CO<sub>2</sub> emission report (China Energy and CO<sub>2</sub> Emission Research Group 2009) and China energy conservation and emission reduction development report (China Energy Saving and Investment Company 2009). For example, the emission factors of coolant ( $EF^c$ ), lubricant oil ( $EF^l$ ) and electric energy ( $EF^e$ ) are 3.782, 0.469 and 2.41.

### 4.2.3 Parameters setting

In MA, there are four critical parameters, i.e., population size ( $PS$ ), crossover rate ( $CR$ ), mutation rate ( $MR$ ) and neighborhood search rate ( $NR$ ). To obtain the best combination of these parameters in our algorithm, The Taguchi method of DOE (Montgomery 2008) is used with instance mk04. Different levels for each parameter are listed in Table 4 and the combinations are shown in Table 5. For each parameter combination, the iteration index is 100 and the proposed MA runs 10 times independently. One of the proposed metrics, i.e.,  $R_{nd}$ , is adopted to measure each combination. We first obtain the exact Pareto front and individuals in the elite pool of all experiments ( $D_T$ ), and the Pareto front ( $D_E$ ) amongst  $D_T$  can be calculated with  $D_T$  as the reference front. Finally, the average percentage of Pareto front ( $\bar{R}$ ) in  $D_E$  for each experiment is used to measure the effectiveness of various combinations.

**Table 3** Parameters for extension

Parameter	Level 1	Level 2	Level 3
Basic efficiency ( $e_{kr}$ )	U[0.8,1.2]	U[0.9,1.2]	U[1.0,1.2]
Basic efficiency ( $e_{ir}$ )	U[0.8,1.2]	U[0.9,1.2]	U[1.0,1.2]
Learning coefficient ( $b_r$ )	U[-0.2,-0.3]	U[-0.15,-0.25]	U[-0.1,-0.2]
Cost of worker ( $W$ )	N(120,10 <sup>2</sup> )	N(100,10 <sup>2</sup> )	N(80,10 <sup>2</sup> )
Limiting efficiency ( $d_k$ )	U[0.5,0.6]	U[0.5,0.6]	U[0.5,0.6]
CE coefficient of auxiliary materials ( $\mu^a$ )	U[0.06,0.1]	U[0.06,0.1]	U[0.06,0.1]
CE in the idle state ( $es_k$ )	U[1,4]	U[1,4]	U[1,4]
Special energy consumption ( $SEC$ )	N(5,1 <sup>2</sup> )	N(5,1 <sup>2</sup> )	N(5,1 <sup>2</sup> )
Removal volume of materials ( $V$ )	N(320,80 <sup>2</sup> )	N(320,80 <sup>2</sup> )	N(320,80 <sup>2</sup> )
CE coefficient of tool ( $\mu^t$ )	U[0.01,0.2]	U[0.01,0.2]	U[0.01,0.2]

**Table 4** Parameter level

Parameters	Factor levels			
	1	2	3	4
<i>PS</i>	60	120	180	240
<i>CR</i>	0.2	0.5	0.7	0.9
<i>MR</i>	0.05	0.1	0.15	0.2
<i>NR</i>	0.1	0.15	0.25	0.3

**Table 5** Results of parameter combination

Index	Four parameters				$R_{nd}$ (%)
	<i>PS</i>	<i>CR</i>	<i>MR</i>	<i>NR</i>	
1	1	1	1	1	0.00000
2	1	2	2	2	2.73038
3	1	3	3	3	11.2628
4	1	4	4	4	5.29010
5	2	1	2	3	3.92491
6	2	2	1	4	7.33788
7	2	3	4	1	2.04778
8	2	4	3	2	0.68259
9	3	1	3	4	8.53242
10	3	2	4	3	5.29010
11	3	3	1	2	4.26621
12	3	4	2	1	2.38908
13	4	1	4	2	13.1399
14	4	2	3	1	2.21843
15	4	3	2	4	17.5768
16	4	4	1	3	13.3106

Based on the results in Table 5, the trend of four critical parameters is illustrated in Fig. 5. It can be seen that different combinations of the four parameters have various influences on the final results. The level with the maximum  $R_{nd}$  is the best one for every factor. According to Fig. 5, we choose  $PS=180$ ,  $CR=0.7$ ,  $MR=0.1$ , and  $NR=0.3$  for the following experiment parameters of the LFJSP-WL benchmarks.

### 4.3 Effectiveness of the elite pool

A comparison experiment is executed to test the effectiveness of the elite pool proposed in our paper. To ensure fairness, all parameters and operators of MA are consistent, the only difference is that one algorithm with the elite pool, i.e. MA, while another modified memetic algorithm without the elite pool, i.e. MMA. The parameter settings are following the results of Taguchi method and set the iteration index as 100. After 10 independent runs, MA gets the corresponding non-dominated

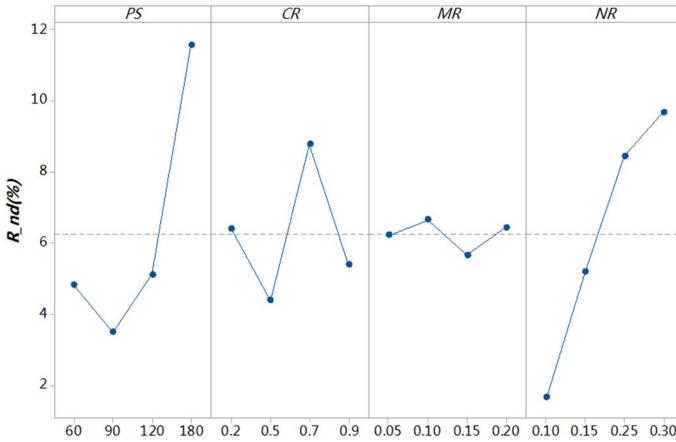


Fig. 5 The factor level trend of PS, CR, MR, NR

solutions and the solutions in the elite pool respectively, while MMA obtains the corresponding non-dominated solutions. Then both algorithms integrated the obtained solutions as a comparison population, one metric (i.e. SC) is adopted to measure the results.

According to the results shown in Table 6. The C(B,A) are all equal to 0 in all problems which states that all solutions obtained by MA cannot be dominated by any solution obtained by MMA. In addition, the value of C(A,B) in each problem is greater than 0 expressing that some solutions obtained by MMA are dominated by at least one solution obtained by MA. What is more, the C(A,B) in the problem la14 is approximately equals to 0.5 which indicates that nearly half of the solutions obtained by MMA are dominated. Based on the above discussion, the effectiveness of the elite pool can be verified.

Table 6 Effectiveness test of elite pool

Problem	Size $n \times m \times w$	A: MA C(A,B)	B: MMA C(B,A)	Problem	Size $n \times m \times w$	A: MA C(A,B)	B: MMA C(B,A)
mk01	10×6×4	0.084211	0.000000	la02	10×5×3	0.028571	0.000000
mk02	10×6×4	0.024590	0.000000	la14	20×5×3	0.483333	0.000000
mk03	15×8×5	0.228261	0.000000	la15	20×5×3	0.147239	0.000000
mk04	15×8×5	0.102273	0.000000	la21	15×10×6	0.100000	0.000000
mk05	15×4×3	0.483333	0.000000	la22	15×10×6	0.144330	0.000000
mk06	10×15×9	0.094737	0.000000	la29	20×10×6	0.000000	0.000000
mk07	20×5×3	0.475862	0.000000	la30	20×10×6	0.095892	0.000000
mk08	20×10×6	0.090047	0.000000	la31	30×10×6	0.157360	0.000000
mk09	20×10×6	0.097561	0.000000	la32	30×10×6	0.057971	0.000000
mk10	20×15×9	0.062016	0.000000	la39	15×15×9	0.081395	0.000000
la01	10×5×3	0.317071	0.000000	la40	15×15×9	0.158730	0.000000

### 4.4 Comparison of two initialization methods

Two initialization methods are constructed in this paper. One introduced in Sect. 3 (method1), another method modified from Zhang et al. (2011) which includes global selection, local selection and random selection (method2). To obtain the best effectiveness of MA, a comparison experiment between method1 and method2 is carried out, the algorithm settings except the initialization method are identical.

From the comparison results listed in Table 7, method1 outperforms method2 in 15 problems out of 22 problems. Especially,  $C(B,A)=0$  in la21, la22, la29 and la31 problems, which shows that all solutions obtained by method1 cannot be dominated by solutions obtained by method2 in these problems. Overall, method 1 behaves better than method 2.

### 4.5 Comparison of NSGA-II, NNIA, NNIALS and MA

To make a more detailed performance analysis with respect to MA towards LFJSP-WL, we compare MA with three other commonly used multi-objective algorithms, namely, NSGA-II (Deb et al. 2002), NNIA (Gong et al. 2008) and NNIALS (NNIA combines our proposed local search).  $IGD$ ,  $SC$  and  $R_{nd}$  are used to evaluate the performance of different algorithms. All algorithms have the same parameters and set the iteration index as 100. After each algorithm runs 10 times independently, the comparison results are listed in Tables 8 and 9.

In Table 8, the value of  $C(A,B)$  listed in the third column outperforms the value of  $C(B,A)$  listed in the fourth column significantly for all test problems, which means that MA derives better solutions than NSGA-II. With regard to the comparison of MA and NNIA, MA also shows great superiority to NNIA in the light of the values listed in the fifth column and sixth column for all test problems. For the last two columns expresses the comparison of MA and NNIALS, we can observe that MA obtains better solutions in 21 test problems out of 22 test problems.

**Table 7** Comparison of initialization methods

Problem	Size $n \times m \times w$	A: method1 C(A,B)	B: method2 C(B,A)	Problem	Size $n \times m \times w$	A: method1 C(A,B)	B: method2 C(B,A)
mk01	10×6×4	0.671875	0.151515	la02	10×5×3	0.867347	0.045455
mk02	10×6×4	0.704918	0.113208	la14	20×5×3	0.821429	0.044444
mk03	15×8×5	0.142857	0.761905	la15	20×5×3	0.288889	0.400000
mk04	15×8×5	0.120690	0.850746	la21	15×10×6	0.936170	0.000000
mk05	15×4×3	0.136842	0.520000	la22	15×10×6	1.000000	0.000000
mk06	10×15×9	0.523810	0.044444	la29	20×10×6	0.990909	0.000000
mk07	20×5×3	0.448598	0.373333	la30	20×10×6	0.671875	0.120690
mk08	20×10×6	0.386364	0.287671	la31	30×10×6	0.888889	0.000000
mk09	20×10×6	0.118644	0.630137	la32	30×10×6	0.990000	0.016129
mk10	20×15×9	0.190476	0.276923	la39	15×15×9	0.250000	0.323529
la01	10×5×3	0.876712	0.098361	la40	15×15×9	0.773333	0.051724

**Table 8** Set Coverage comparison of MA, NSGA-II, NNIA and NNIALS

Problem	Size $n \times m \times w$	A: MA	B: NSGA-II	A: MA	B: NNIA	A: MA	B: NNIALS
		C(A,B)	C(B,A)	C(A,B)	C(B,A)	C(A,B)	C(B,A)
mk01	10×6×4	1.000000	0.000000	0.941177	0.000000	0.757009	0.000000
mk02	10×6×4	1.000000	0.000000	0.935484	0.000000	0.831325	0.014599
mk03	15×8×5	1.000000	0.000000	1.000000	0.000000	0.871795	0.000000
mk04	15×8×5	1.000000	0.000000	0.902985	0.000000	0.6111111	0.000000
mk05	15×4×3	1.000000	0.000000	0.988304	0.000000	0.828221	0.010870
mk06	10×15×9	1.000000	0.000000	1.000000	0.000000	1.000000	0.000000
mk07	20×5×3	0.966667	0.006579	1.000000	0.000000	0.803922	0.052632
mk08	20×10×6	1.000000	0.000000	1.000000	0.000000	0.912500	0.000000
mk09	20×10×6	1.000000	0.000000	1.000000	0.000000	0.904762	0.000000
mk10	20×15×9	1.000000	0.000000	1.000000	0.000000	1.000000	0.000000
la01	10×5×3	1.000000	0.000000	0.765101	0.038462	0.366972	0.384615
la02	10×5×3	1.000000	0.000000	0.990991	0.000000	0.551724	0.054795
la14	20×5×3	1.000000	0.000000	1.000000	0.000000	0.566929	0.095238
la15	20×5×3	1.000000	0.000000	0.894309	0.000000	0.625000	0.052941
la21	15×10×6	1.000000	0.000000	1.000000	0.000000	0.772727	0.000000
la22	15×10×6	1.000000	0.000000	1.000000	0.000000	0.605634	0.000000
la29	20×10×6	1.000000	0.000000	1.000000	0.000000	1.000000	0.000000
la30	20×10×6	1.000000	0.000000	1.000000	0.000000	0.618421	0.011834
la31	30×10×6	1.000000	0.000000	0.906250	0.000000	0.934066	0.004975
la32	30×10×6	1.000000	0.000000	1.000000	0.000000	0.905882	0.012903
la39	15×15×9	1.000000	0.000000	1.000000	0.000000	1.000000	0.000000
la40	15×15×9	1.000000	0.000000	1.000000	0.000000	1.000000	0.000000

In Table 9, the  $R_{nd}$  and  $IGD$  comparison of the four algorithms are presented. In terms of  $R_{nd}$ , MA outperforms the other three algorithms by a large margin for all 22 test problems. Besides, MA obtains all better values of  $IGD$  compared to NSGA-II and NNIA. It can be found that 21 test problems out of 22 test problems MA is superior to NNIALS. In summary, MA shows its effectiveness for the proposed LFJSP-WL, while the values of NNIALS also indicates the validity of our proposed local search in a sense.

## 5 Conclusions and future studies

This paper put forward a novel LFJSP-WL model that addresses productivity (i.e. makespan), environmental impact (i.e. total CE) and economic influence (i.e. total cost of workers). The CE of the model consists of four categories, namely, the energy consumption of processing, the consumption of auxiliary materials, the tool wear and the energy consumption in the stand-by stage of the machine tool. An effective memetic algorithm is tailored to solve this problem, some useful

**Table 9**  $R_{nd}$  and  $IGD$  comparison of MA, NSGA-II, NNIA and NNIALS

Problem	Size $n \times m \times w$	MA		NSGA-II		NNIA		NNIALS	
		$R_{nd}$	$IGD$	$R_{nd}$	$IGD$	$R_{nd}$	$IGD$	$R_{nd}$	$IGD$
mk01	10×6×4	0.9000	0.0360	0.0000	0.2558	0.0000	0.2066	0.1000	0.1326
mk02	10×6×4	0.9581	0.0064	0.0000	0.2992	0.0000	0.2637	0.0419	0.1390
mk03	15×8×5	0.9802	0.0098	0.0000	0.4290	0.0000	0.5036	0.0198	0.2344
mk04	15×8×5	0.9325	0.0425	0.0000	0.4313	0.0000	0.3198	0.0675	0.2129
mk05	15×4×3	0.9358	0.0074	0.0000	0.2014	0.0000	0.1781	0.0642	0.0950
mk06	10×15×9	1.0000	0.0000	0.0000	0.5626	0.0000	0.6018	0.0000	0.2817
mk07	20×5×3	0.9375	0.0136	0.0057	0.2177	0.0000	0.2156	0.0568	0.1223
mk08	20×10×6	0.9849	0.0044	0.0000	0.4536	0.0000	0.3832	0.0151	0.3383
mk09	20×10×6	0.9771	0.0046	0.0000	0.2931	0.0000	0.2759	0.0229	0.2127
mk10	20×15×9	1.0000	0.0000	0.0000	0.5174	0.0000	0.5070	0.0000	0.3159
la01	10×5×3	0.6975	0.0664	0.0000	0.2258	0.0126	0.1629	0.2899	0.0636
la02	10×5×3	0.8259	0.0538	0.0000	0.3828	0.0000	0.3371	0.1741	0.0737
la14	20×5×3	0.8401	0.0236	0.0000	0.3749	0.0000	0.1965	0.1599	0.0794
la15	20×5×3	0.9177	0.0223	0.0000	0.4384	0.0000	0.1927	0.0823	0.1440
la21	15×10×6	0.9660	0.0200	0.0000	0.5975	0.0000	0.3956	0.0340	0.2552
la22	15×10×6	0.9097	0.0524	0.0000	0.5159	0.0000	0.4172	0.0903	0.1634
la29	20×10×6	1.0000	0.0000	0.0000	0.5138	0.0000	0.4526	0.0000	0.2992
la30	20×10×6	0.9305	0.0224	0.0000	0.4165	0.0000	0.3333	0.0695	0.1803
la31	30×10×6	0.9596	0.0096	0.0000	0.2828	0.0270	0.1504	0.0135	0.1630
la32	30×10×6	0.9780	0.1630	0.0000	0.4636	0.0000	0.3987	0.0220	0.2471
la39	15×15×9	1.0000	0.0000	0.0000	0.9602	0.0000	0.7799	0.0000	0.5055
la40	15×15×9	1.0000	0.0000	0.0000	0.7417	0.0000	0.6644	0.0000	0.3951

initialization methods, crossover operators and mutation operators are presented, and the variable neighborhood search with four neighborhood structures is proposed to strengthen the local search capacity. Besides, the elite pool is adopted to reserve elite solutions along with evolution. To evaluate the effectiveness of MA, we constructed 22 benchmarks based on the classic FJSP benchmarks from Dauzèrepères and Paulli (1997) and (Brandimarte (1993)). The experimental results show that the proposed LFJSP-WL model can be solved effectively by MA.

In the future, some limitations in this paper can be further studied. First, consider the CE of inventory, the time of transportation and the workpiece clamping time. Second, address the uncertain events, for example, the machine breakdown, the arrival of new jobs and the cancellation of the order. Finally, the combination of learning and forgetting effects would be an interesting topic.

**Acknowledgements** The authors are grateful to the editor and anonymous referees for their valuable comments and suggestions. This work is financially supported by the National Key R&D Program of China (2018YFB1701400), the National Natural Science Foundation of China (71473077), the State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body (71775004).

## References

- Ballestin F, Mallor F, Mateo PM (2012) Production scheduling in a market-driven foundry: a mathematical programming approach versus a project scheduling metaheuristic algorithm. *Optim Eng* 13:663–687. <https://doi.org/10.1007/s11081-011-9157-z>
- Biskup D (1999) Single-machine scheduling with learning considerations. *Eur J Oper Res* 115:173–178. [https://doi.org/10.1016/s0377-2217\(98\)00246-x](https://doi.org/10.1016/s0377-2217(98)00246-x)
- Biskup D (2008) A state-of-the-art review on scheduling with learning effects. *Eur J Oper Res* 188:315–329. <https://doi.org/10.1016/j.ejor.2007.05.040>
- Brandimarte P (1993) Routing and scheduling in a flexible job shop by tabu search. *Ann Oper Res* 41:157–183. <https://doi.org/10.1007/BF02023073>
- Brucker P, Schlie R (1990) Job-shop scheduling with multi-purpose machines. *Computing* 45:369–375. <https://doi.org/10.1007/BF02238804>
- China Energy and CO2 Emission Research Group (2009) 2050 China energy and CO2 emission report. Science Press, Beijing, China
- China Energy Saving and Investment Company (2009) China energy conservation and emission reduction development report. China Water Power Press, Beijing, China
- Dauzèrepères S, Paulli J (1997) An integrated approach for modeling and solving the general multi-processor job-shop scheduling problem using tabu search. *Ann Oper Res* 70:281–306. <https://doi.org/10.1023/a:1018930406487>
- Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6:182–197. <https://doi.org/10.1109/4235.996017>
- Demir Y, İşleyen SK (2014) An effective genetic algorithm for flexible job-shop scheduling with overlapping in operations. *Int J Prod Res* 52:3905–3921. <https://doi.org/10.1080/00207543.2014.889328>
- International Energy Outlook 2016 (Online) (2016) EIA. <https://www.eia.gov/outlooks/archive/ieo16/>. Accessed May 2016
- ElMaraghy H, Patel V, Abdallah IB (1999) A genetic algorithm based approach for scheduling of dual-resource constrained manufacturing systems. *CIRP Ann-Manuf Technol* 48:369–372. [https://doi.org/10.1016/S0007-8506\(07\)63204-1](https://doi.org/10.1016/S0007-8506(07)63204-1)
- Felan J, Fry T (2001) Multi-level heterogeneous worker flexibility in a dual resource constrained (DRC) job-shop. *Int J Prod Res* 39:3041–3059. <https://doi.org/10.1080/00207540110047702>
- Ferjani A, Ammar A, Pierreval H, Elkosantini S (2017) A simulation-optimization based heuristic for the online assignment of multi-skilled workers subjected to fatigue in manufacturing systems. *Comput Ind Eng* 112:663–674. <https://doi.org/10.1016/j.cie.2017.02.008>
- Frutos M, Olivera AC, Tohmé F (2010) A memetic algorithm based on a NSGAI scheme for the flexible job-shop scheduling problem. *Ann Oper Res* 181:745–765. <https://doi.org/10.1007/s10479-010-0751-9>
- Fryer JS (2010) Organizational structure of dual-constraint job shops. *Decis Sci* 5:45–57. <https://doi.org/10.1111/j.1540-5915.1974.tb00594.x>
- Garey MR, Johnson DS, Sethi R (1976) The Complexity of flowshop and jobshop scheduling. *Math Oper Res* 1:117–129. <https://doi.org/10.2307/3689278>
- Gong M, Jiao L, Du H, Bo L (2008) Multiobjective immune algorithm with nondominated neighbor-based selection. *Evol Comput* 16:225–255. <https://doi.org/10.1162/evco.2008.16.2.225>
- Gong G, Deng Q, Gong X, Liu W, Ren Q (2018a) A new double flexible job-shop scheduling problem integrating processing time, green production, and human factor indicators. *J Clean Prod* 174:560–576. <https://doi.org/10.1016/j.jclepro.2017.10.188>
- Gong X, Deng Q, Gong G, Liu W, Ren Q (2018b) A memetic algorithm for multi-objective flexible job-shop problem with worker flexibility. *Int J Prod Res* 56:2506–2522. <https://doi.org/10.1080/00207543.2017.1388933>
- Gong G, Deng Q, Chiong R, Gong X, Huang H (2019a) An effective memetic algorithm for multi-objective job-shop scheduling. *Knowl-Based Syst*. <https://doi.org/10.1016/j.knosys.2019.07.011>
- Gong X, De Pessemer T, Martens L, Joseph W (2019b) Energy- and labor-aware flexible job shop scheduling under dynamic electricity pricing: A many-objective optimization investigation. *J Clean Prod* 209:1078–1094. <https://doi.org/10.1016/j.jclepro.2018.10.289>
- Guemri M, Neffati A, Caux S, Ngueveu SU (2014) Management of distributed power in hybrid vehicles based on DP or Fuzzy Logic. *Optim Eng* 15:993–1012. <https://doi.org/10.1007/s11081-013-9235-5>



- Hertz A, Kobler D (2000) A framework for the description of evolutionary algorithms. *Eur J Oper Res* 126:1–12. [https://doi.org/10.1016/S0377-2217\(99\)00435-X](https://doi.org/10.1016/S0377-2217(99)00435-X)
- Kacem I, Hammadi S, Borne P (2014) Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic. *Mat Comput Simul* 60:245–276. [https://doi.org/10.1016/s0378-4754\(02\)00019-8](https://doi.org/10.1016/s0378-4754(02)00019-8)
- Kannan SS, Ramaraj N (2010) A novel hybrid feature selection via symmetrical uncertainty ranking based local memetic search algorithm. *Knowl-Based Syst* 23:580–585. <https://doi.org/10.1016/j.knosys.2010.03.016>
- Kara S, Li W (2011) Unit process energy consumption models for material removal processes. *CIRP Ann-Manuf Technol* 60:37–40. <https://doi.org/10.1016/j.cirp.2011.03.018>
- Kara S, Manmek S, Herrmann C (2010) Global manufacturing and the embodied energy of products. *CIRP Ann-Manuf Technol* 59:29–32. <https://doi.org/10.1016/j.cirp.2010.03.004>
- Kelly JD, Zyngier D (2017) Unit-operation nonlinear modeling for planning and scheduling applications. *Optim Eng* 18:133–154. <https://doi.org/10.1007/s11081-016-9312-7>
- Koulamas C, Kyriaris GJ (2007) Single-machine and two-machine flowshop scheduling with general learning functions. *Eur J Oper Res* 178:402–407. <https://doi.org/10.1016/j.ejor.2006.01.030>
- Kuo WH, Yang D-L (2006) Minimizing the total completion time in a single-machine scheduling problem with a time-dependent learning effect. *Eur J Oper Res* 174:1184–1190. <https://doi.org/10.1016/j.ejor.2005.03.020>
- Laurent A, Olsen SI, Hauschild MZ (2010) Carbon footprint as environmental performance indicator for the manufacturing industry. *CIRP Ann-Manuf Technol* 59:37–40. <https://doi.org/10.1016/j.cirp.2010.03.008>
- Lei D, Guo X (2014) Variable neighbourhood search for dual-resource constrained flexible job shop scheduling. *Int J Prod Res* 52:2519–2529. <https://doi.org/10.1080/00207543.2013.849822>
- Li W, Kara S (2011) An empirical model for predicting energy consumption of manufacturing processes: a case of turning process. *Proc Inst Mech Eng Part B-J Eng Manuf* 225:1636–1646. <https://doi.org/10.1177/2041297511398541>
- Li J, Pan Q, Xie S (2012) An effective shuffled frog-leaping algorithm for multi-objective flexible job shop scheduling problems. *Appl Math Comput* 218:9353–9371. <https://doi.org/10.1016/j.amc.2012.03.018>
- Liu CH, Huang DH (2014) Reduction of power consumption and carbon footprints by applying multi-objective optimisation via genetic algorithms. *Int J Prod Res* 52:337–352. <https://doi.org/10.1080/00207543.2013.825740>
- Ma J, Lei Y, Wang Z, Jiao L, Liu R (2014a) A Memetic algorithm based on immune multi-objective optimization for flexible job-shop scheduling problems. In: *Evolutionary Computation (CEC), 2014 IEEE Congress on, 2014. IEEE*, pp 58–65. <https://doi.org/10.1109/CEC.2014.6900331>
- Ma W, Yi Z, Zeng J, Liang S, Jiao L (2014b) A memetic algorithm for solving flexible job-shop scheduling problems. In: *Evolutionary computation*. <https://doi.org/10.1109/CEC.2014.6900332>
- May G, Stahl B, Taisch M (2016) Energy management in manufacturing: toward eco-factories of the future—a focus group study. *Appl Energy* 164:628–638. <https://doi.org/10.1016/j.apene.2015.11.044>
- Montgomery DC (2008) *Design & analysis of experiments*. Wiley, New York
- Ostermeier FF (2019) The impact of human consideration, schedule types and product mix on scheduling objectives for unpaced mixed-model assembly lines. *Int J Prod Res*. <https://doi.org/10.1080/00207543.2019.1652780>
- Phu-Ang A, Thammano A (2017) Memetic algorithm based on marriage in honey bees optimization for flexible job shop scheduling problem. *Memetic Comput* 9:1–15. <https://doi.org/10.1007/s12293-017-0230-9>
- Piroozfard H, Wong KY, Wong WP (2018) Minimizing total carbon footprint and total late work criterion in flexible job shop scheduling by using an improved multi-objective genetic algorithm. *Resour Conserv Recycl* 128:267–283. <https://doi.org/10.1016/j.resconrec.2016.12.001>
- Schinko T, Bednar-Friedl B, Steininger KW, Grossmann WD (2014) Switching to carbon-free production processes: Implications for carbon leakage and border carbon adjustment. *Energy Policy* 67:818–831. <https://doi.org/10.1016/j.enpol.2013.11.077>
- Sinoquet D, Rousseau G, Milhau Y (2011) Design optimization and optimal control for hybrid vehicles. *Optim Eng* 12:199–213. <https://doi.org/10.1007/s11081-009-9100-8>

- Wu R, Li Y, Guo S, Xu W (2018) Solving the dual-resource constrained flexible job shop scheduling problem with learning effect by a hybrid genetic algorithm. *Adv Mech Eng*. <https://doi.org/10.1177/1687814018804096>
- Wu X, Che A (2019) A memetic differential evolution algorithm for energy-efficient parallel machine scheduling. *Omega* 82:155–165. <https://doi.org/10.1016/j.omega.2018.01.001>
- Xu J, Xu X, Xie SQ (2011) Recent developments in dual resource constrained (DRC) system research. *Eur J Oper Res* 215:309–318. <https://doi.org/10.1016/j.ejor.2011.03.004>
- Yi W, Pan B, Pan B (2016) Solving flexible job shop scheduling using an effective memetic algorithm. *Int J Comput Appl Technol*. <https://doi.org/10.1504/IJCAT.2016.074454>
- Yuan Y, Hua X (2014) Multiobjective flexible job shop scheduling using memetic algorithms. *IEEE Trans Autom Sci Eng* 12:336–353. <https://doi.org/10.1109/TASE.2013.2274517>
- Yuan Y, Xu H (2013) A memetic algorithm for the multi-objective flexible job shop scheduling problem. *Gecco'13: proceedings of the 2013 genetic and evolutionary computation conference*. <https://doi.org/10.1145/2463372.2463431>
- Zhang G, Gao L, Shi Y (2011) An effective genetic algorithm for the flexible job-shop scheduling problem. *Expert Syst Appl* 38:3563–3573. <https://doi.org/10.1016/j.eswa.2010.08.145>
- Zhang C, Gu P, Jiang P (2015) Low-carbon scheduling and estimating for a flexible job shop based on carbon footprint and carbon efficiency of multi-job processing. *Proc Inst Mech Eng Part B-J Eng Manuf* 229:328–342. <https://doi.org/10.1177/0954405414527959>
- Zhang Z, Tang R, Tao P, Tao L, Jia S (2016) A method for minimizing the energy consumption of machining system: integration of process planning and scheduling. *J Clean Prod* 137:1647–1662. <https://doi.org/10.1016/j.jclepro.2016.03.101>
- Zheng HY, Wang L (2015) Reduction of carbon emissions and project makespan by a Pareto-based estimation of distribution algorithm. *Int J Prod Econ* 164:421–432. <https://doi.org/10.1016/j.ijpe.2014.12.010>
- Zheng X-l, Wang L (2016) A knowledge-guided fruit fly optimization algorithm for dual resource constrained flexible job-shop scheduling problem. *Int J Prod Res* 54:5554–5566. <https://doi.org/10.1080/00207543.2016.1170226>
- Zhou R, Nee AYC, Lee HP (2009) Performance of an ant colony optimisation algorithm in dynamic job shop scheduling problems. *Int J Prod Res* 47:2903–2920. <https://doi.org/10.1080/00207540701644219>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.