Journal of Systems Engineering and Electronics
Vol. 32, No. 2, April 2021, pp.297 – 317

Multi-objective reconfigurable production line scheduling for smart home appliances

LI Shiyun, ZHONG Sheng, PEI Zhi*, YI Wenchao, CHEN Yong, WANG Cheng, and ZHANG Wenzhu

College of Mechanical Engineering, Zhejiang University of Technology, Hangzhou 310023, China

Abstract: In a typical discrete manufacturing process, a new type of reconfigurable production line is introduced, which aims to help small- and mid-size enterprises to improve machine utilization and reduce production cost. In order to effectively handle the production scheduling problem for the manufacturing system, an improved multi-objective particle swarm optimization algorithm based on Brownian motion (MOPSO-BM) is proposed. Since the existing MOPSO algorithms are easily stuck in the local optimum, the global search ability of the proposed method is enhanced based on the random motion mechanism of the BM. To further strengthen the global search capacity, a strategy of fitting the inertia weight with the piecewise Gaussian cumulative distribution function (GCDF) is included, which helps to maintain an excellent convergence rate of the algorithm. Based on the commonly used indicators generational distance (GD) and hypervolume (HV), we compare the MOPSO-BM with several other latest algorithms on the benchmark functions, and it shows a better overall performance. Furthermore, for a real reconfigurable production line of smart home appliances, three algorithms, namely non-dominated sorting genetic algorithm-II (NSGA-II), decomposition-based MOPSO (dMOPSO) and MOPSO-BM, are applied to tackle the scheduling problem. It is demonstrated that MOPSO-BM outperforms the others in terms of convergence rate and quality of solutions.

Keywords: reconfigurable production line, improved particle swarm optimization (PSO), multi-objective optimization, flexible flowshop scheduling, smart home appliances.

DOI: 10.23919/JSEE.2021.000026

1. Introduction

With the rapid development of information technology, the global manufacturing market has entered a digitalized age, and fierce competition of the manufacturing technology and managerial strategy increasingly affect the survival and development of the enterprises. In order to meet the growing market demand, the production paradigms of most companies have shifted from the original mass production to the flexible production with smaller batches [1]. However, for small- and mid-size enterprises (SMEs), to cope with the ever-changing customer requirements is challenging, given that the manufacturing is resource and equipment sensitive. In response to this challenge, the development of a reconfigurable production line could be a fast deployable solution. In the past several decades, many researchers have studied the reconfigurable production line scheduling problem, which are summarized in Table 1. By consulting the related literature, it is found that most of the existing papers study the production scheduling from a mathematical perspective, without considering the configurations of the production line [2−21]. A highly flexible production setting may bring greater benefits to the manufacturing system.

To echo that purpose, this paper proposes a new mode of reconfigurable manufacturing system, where the modules with different functions are assembled together to form a dedicated production line. The reconfigurable production line could be decomposed into separate production units, which preserves great flexibility and saves cost significantly in comparison with the traditional one-time installation of equipment.

Besides the production line itself, this reconfigurable setting poses great challenges on the scheduling algorithm. A good schedule could greatly reduce processing costs, shorten the processing completion cycle, improve the utilization rate of equipment, and directly increase the economic benefits of the enterprise. Currently, the scheduling strategy for such a reconfigurable production line mainly relies on the experiences of the planners, which leads to many problems such as low production efficiency, high equipment load imbalance, and serious production related profit loss. Therefore, the optimization of the task sequencing becomes the key issue for the flexible production line scheduling [22].

Manuscript received October 28, 2020.

*Corresponding author.

This work was supported by the National Natural Science Foundation of China (71871203; 52005447; L1924063), and Zhejiang Provincial Natural Science Foundation of China (LY18G010017; LQ21E050014).

Table 1 Closely related literature on the reconfigurable production line scheduling

Study	Multi-objective	Flexible production line	Recon-figurable	Flow shop	Methodology	Application in industry
KUO et al. 1999 [2]		√			Colored timed petri net+balanced budget work standard algorithm+dispatch rules	Hybrid assembly line in automotive industry
Guo et al. 2006 [3]	√	√			Bi-level GA	Multi- and mixed-model in an apparel assembly
Huang et al. 2009 [4]	√	√		√	ACO+parameter tuning	—
Davoudpour et al. 2009 [5]	√	√		√	Greedy randomized adaptive search procedure+a non-regular optimization criterion	Hybrid line with sequence-dependent setup times
Dudas et al. 2011 [6]	√	√		√	Simulation-based innovization	Automotive machining line
Liu et al. 2011 [7]		√		√	Improved PSO	—
Chaube et al. 2012 [8]	√		√		NSGA-II+Reconfigurable process plan	—
Dai et al. 2013 [9]	√	√		√	GA+simulated annealing algorithm (SA)	Metalworking workshop
Jolai et al. 2013 [10]	√	√		√	SA+ normalized weighted multi-objective decision making (MODM)	—
Sheikh et al. 2013 [11]	√	√		√	GA+ linear programming	—
Tran et al. 2013 [12]	√	√		√	Hybrid water flow algorithm+landscape analysis+optimal Pareto solution set	—
Naderi et al. 2014 [13]				√	Hybrid PSO	—
Ghaleb et al. 2015 [14]		√		√	PSO+Tabu search	—
Choi et al. 2015 [15]	√		√		Dispatching rule based algorithm	Production line in motorcycle field
Dou et al. 2016 [16]	√		√	√	NSGA-II+ mixed integer programming	Parts processing simulation for production line
Zhao et al. 2017 [17]	√		√		Object oriented timed colored petri net (OOTCPN)-GASA	Wood manufacturing system
Asghar et al. 2018 [18]	√		√		GA	Part family production line
Gong et al. 2020 [19]	√	√		√	Hybrid evolutionary algorithm (HEA)+variable neighborhood search	Energy-efficient line with worker flexibility
Han et al. 2020 [20]	√			√	Heuristic+self-adaptive evolution operators	—
Hasani et al. 2020 [21]	√	√		√	NSGA-II+overall nondominated vector generation (ONVG)	—
This work	√	√	√	√	MOPSO based on Brownian motion (MOPSO-BM)	Smart home appliance manufacturing

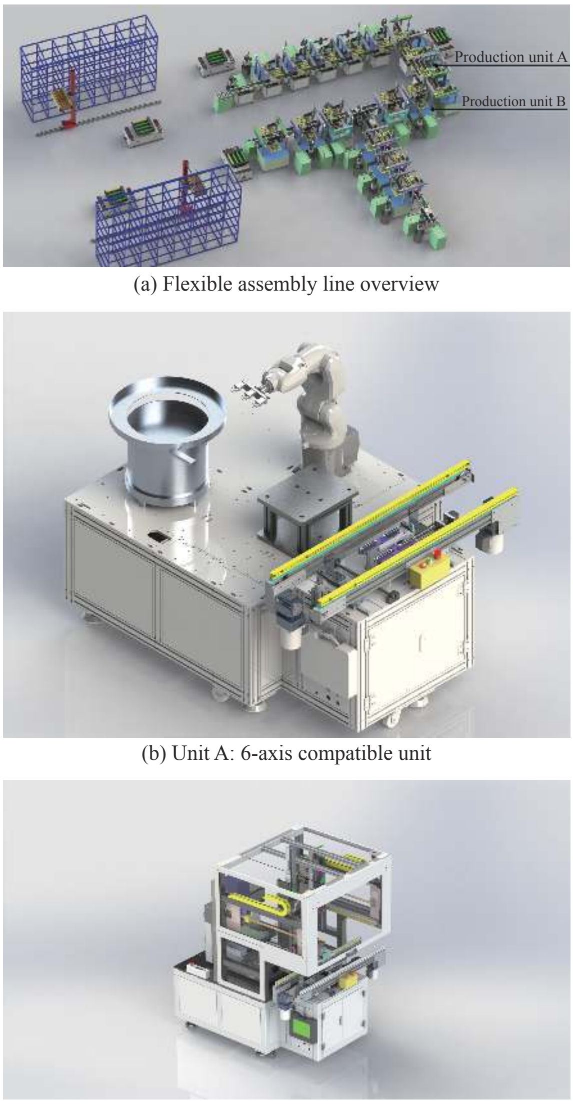
Intelligent algorithms are regarded as an effective remedy for complex scheduling problems, such as the genetic algorithm (GA), particle swarm optimization (PSO), and the ant colony (ACO) algorithm. Multi-objective PSO (MOPSO) is an intelligent algorithm first proposed by Moore and Chapman [23], which can effectively solve complex scheduling problems. However, the existing MOPSO has a weak global search capability, and is easy to fall into a local optimal state [24]. In order to avoid that deficiency and solve the flexible flowshop scheduling problem, we propose an improved MOPSO algorithm based on the Gaussian cumulative distribution function (GCDF) and BM for the reconfigurable production systems. GCDF is used to fit the inertia weight of the particle velocity update, and the random motion mechanism of BM helps the particles to jump out of the local optimum, so as to improve the global search ability of the particles.

The rest of this article is organized as follows. In Section 2, we propose a new type of reconfigurable production line. An optimization model is established with respect to the characteristics of the flexible production line in Section 3. In Section 4, the traditional MOPSO is presented, which is then improved by introducing the GCDF, BM, polynomial mutation, and double-archive mechanism. In Section 5, we obtain the indicator function values based on several Zitzler-Deb-Thiele (ZDT) test functions and flexible production benchmark tests, showing the effectiveness of the proposed algorithm. Then an actual production scheduling problem of the reconfigurable production line is tackled with the improved algorithm in Section 6. The last section concludes the paper with several future research directions.

2. The physical model

As the concept of smart home becomes a trend for every household worldwide, various home appliances are embedded with control chips to suit the use of commercial electronics. Since the companies are usually SMEs with limited start-up funds available, and the demand for smart home appliances fluctuate regularly, it is an excellent application setting for the reconfigurable production line.

The new type of the reconfigurable production line is realized by freely connecting detachable production units. Fig. 1(a) is a schematic diagram of a typical reconfigurable assembly line. The production unit A and production unit B could provide supports for 6-axis robotics and soldering machines, as shown in Fig. 1(b) and Fig. 1(c). Similarly, each production unit could be equipped with a standardized processing equipment to perform a specific processing task. Meanwhile, the same production unit could be used as a group of parallel machines to perform the same manufacturing task within the assembly line. When multiple production units are connected and coordinated with the central planning system, the workpiece could be automatically processed or assembled without manual intervention.



(a) Flexible assembly line overview

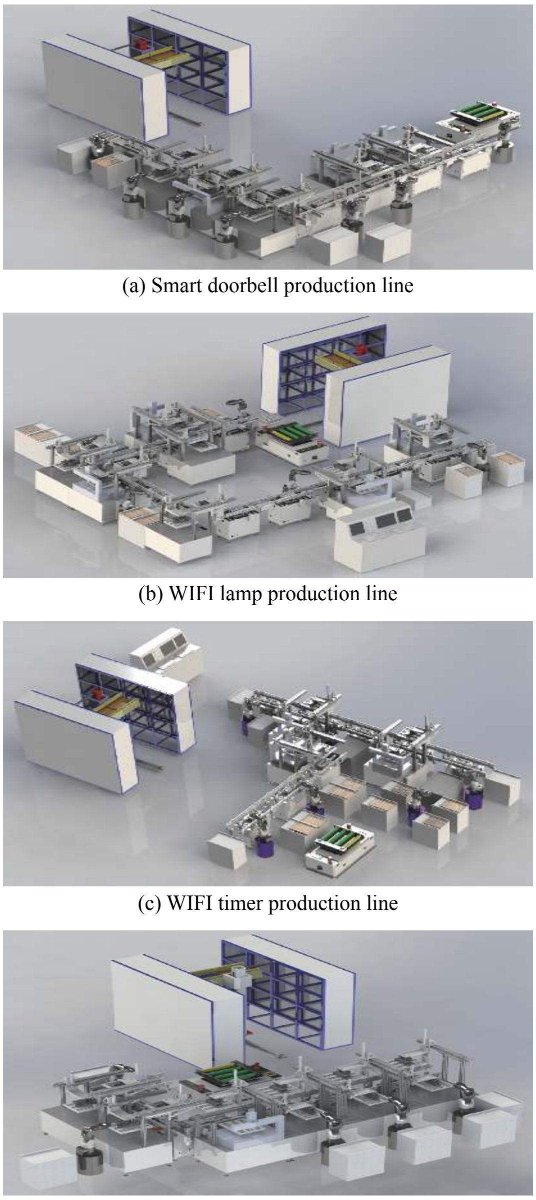
(b) Unit A: 6-axis compatible unit

(c) Unit B: Soldering compatible unit

Fig. 1 Various production units in reconfigurable assembly line

This reconfigurable production line demonstrates two essential features. First, the production line could be reorganized according to the actual working steps, avoiding the back and forth flow of work-in-process. Second, the reconfigurable production line could help adjust the configuration of production units, which leads to full use of each machine unit, thereby reducing the production cost.

The concept of the reconfigurable production line has now been fulfilled by a leading smart home appliance manufacturer named BoTai electronics. Its product line currently includes four categories, including smart WIFI sockets, smart LED lights, smart thermometers, and smart doorbells. At present, as the market expands, the company needs to respond to the highly volatile customer needs. To this end, the equipment and the facility layout are reconfigurable to better suit the timely production process. As shown in Fig. 2, the four mainstream smart home appliance products are manufactured in the reconfigurable production line according to their respective manufacturing processes.



(a) Smart doorbell production line

(b) WIFI lamp production line

(c) WIFI timer production line

(d) Smart night light production line

Fig. 2 Reconfigurable assembly line for different smart home appliances

3. The multi-objective optimization model

The physical model of the reconfigurable assembly line introduced in this paper could transform all sorts of production systems into a flexible flow shop. The scheduling of the flexible flow shop problem could be described as follows. There are N products to be processed or assembled on M different machine units. K working steps are needed to complete the processing. As shown in Fig. 3, each step consists of several identical machine units (Unit A or Unit B). After each step is completed, the unfinished product enters the next processing step, and each product completes all the processing steps in the same order [9]. Given the processing time of each workpiece on different machine units, the target is to determine the sequence of the jobs in the system.

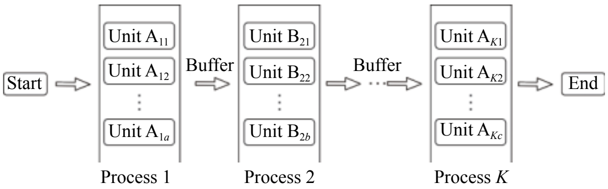


Fig. 3 Flexible flowshop scheduling problem

The basic assumptions are listed below.

(i) All machines are available at time t = 0, and all parts can be processed and produced at time 0.

(ii) The sequence of the processing steps of the same job type is determined.

(iii) Once the job starts processing, it cannot be interrupted until the processing is completed on the current machine.

(iv) The transportation time between two consecutive processing stages is ignored.

(v) There is no limit to the capacity of the buffer area between two consecutive processing stages.

In the actual scenario where multiple production indicators need to be considered simultaneously, a multi-objective optimization model is formulated. In order to increase the production efficiency and economic benefits, we set up four objective functions according to the actual improvement targets of the manufacturing system, namely the makespan minimization, the loss cost minimization, the total load of the machine minimization, and the number of completed orders maximization. The makespan refers to the time needed for producing one such product. The loss cost represents the sum of the processing cost, inventory cost and efficiency loss cost of all the machines. The total load of the machine denotes the total processing time of all the machines. Equations (1)−(5) are detailed mathematical expressions of the multi-objective functions. Equations (6)−(11) are the required constraints. To help the readers better capture the essence of the model, the parameters and decision variables are denoted in Table 2.

Table 2 Parameters and variables

Symbol	Description
i	The ith product in N products, $i = 1,2,\cdots, N$
j	The jth part in the J parts of the product, $j = 1,2,\cdots, J$
k	The kth step in the K processes, $k = 1,2,\cdots, K$
r	The rth machine in the M machines, $r = 1,2,\cdots, M$
T_{ijkr}	The processing time of the process k of the part j of the product i on the machine r
S_{ijkr}	The start time of the process k of the part j of the product i on the machine r
E_{ijkr}	The end time of the process k of the part j of the product i on the machine r
l_i	The order quantity of product i
$\boldsymbol{W}_h$	The collection of each order quantity
cm	Machine operating cost per unit time
gm	Inventory cost per unit time
bm	The cost of efficiency loss per unit time
δ	The cost adjustment coefficient of efficiency loss per unit time
d_{ij}	The inventory time of the part j of the product i

The multi-objective functions are listed as follows.

(i) To minimize the total load of the machines

$$\min F_1 = \min\left(l_i \cdot \sum_{i=1}^{N}\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{r=1}^{M} x_{ijkr} T_{ijkr} \right) \tag{1}$$

(ii) To minimize the loss cost

$$\min F_2 = \min\left\{ l_i \cdot \left[\sum_{i=1}^{N}\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{r=1}^{M} T_{ijkr} \cdot (cm + \delta \cdot bm) + gm \cdot \sum_{i=1}^{N}\sum_{j=1}^{J} d_{ij} \right] \right\} \tag{2}$$

(iii) To minimize the makespan

$$\min F_3 = \min_{1<i<N}\left(\max_{1<k<K} T_{ik} \right) \tag{3}$$

(iv) To maximize the number of completed orders

$$\max\ F_4 = \max \sum_{h} \boldsymbol{W}_h \prod_{i=1}^{N} U_i^h \tag{4}$$

When the makespan F_3 is no more than the delivery date $\boldsymbol{D}_h$, the order h is completed on time; otherwise, the delivery cannot be made on time.

$$U_i^h = \begin{cases} 1, & \forall F_3 \leqslant \boldsymbol{D}_h \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

The followings are the constraints of the multi-objective optimization model,

(i) Time constraints. The state of time should be non-negative.

$$T_{ijkr} \geqslant 0,\ d_{ij} \geqslant 0,\ \forall i;\ \forall j;\ \forall k;\ \forall r \tag{6}$$

(ii) Resource constraints. It is assumed that only one machine could be selected for a process.

$$\sum_{r=1}^{M} x_{ijkr} = 1,\ \ \forall i;\ \forall j;\ \forall k \tag{7}$$

(iii) Machine-job assignment constraint. If the process k of the jth part of the product i is processed on the machine r, $x_{ijkr} = 1$; otherwise, $x_{ijkr} = 0$.

$$X_{ijkr} = \{0,1\},\ \ \forall i;\ \forall j;\ \forall k;\ \forall r \tag{8}$$

(iv) The start time of the next process of the job should be no less than the completion time of its previous processing step.

$$\sum_{r=1}^{M} S_{ijkr} x_{ijkr} \geqslant \sum_{r=1}^{M} (S_{ij(k-1)r} + T_{ij(k-1)r}) x_{ij(k-1)r},\ \ \forall i;\ \forall j;\ \forall k \tag{9}$$

(v) The end time of any workpiece is no less than the sum of the start time, processing time and adjustment time of the workpiece.

$$E_{ijkr} \geqslant S_{ijkr} + T_{ijkr},\ \ \forall i;\ \forall j;\ \forall k;\ \forall r \tag{10}$$

4. Standard MOPSO

The standard PSO was proposed by Kennedy and Eberhart [25] in 1995. It is a swarm intelligent optimization algorithm, which mimics the social behavior of animal herds, such as the flock of birds and the school of fish in search for food. The PSO algorithm has been successfully applied in various scheduling problems [26]. It embraces advantages of simple principles, fewer parameters, and easy implementation in comparison with other swarm intelligence approaches. It has also been used to solve many complex optimization problems, which are often nonlinear [27], non-differentiable [28] and with multi-peak [29]. Moore and Chapman [23] extended the PSO approach to the multi-objective optimization problem in 1999. Coello introduced external archiving and special mutation operators [30], and then formally proposed the MOPSO. With the continuous development over the years [31−33], MOPSO has now become an active method in many engineering fields [34−36].

In a typical multi-objective optimization problem, there may be different optimization objectives for different subobjective functions. The general expression is as below:

$$\begin{cases} \min f(\boldsymbol{X}) = (f_1(\boldsymbol{X}), f_2(\boldsymbol{X}), \cdots, f_m(\boldsymbol{X})) \\ g_i(\boldsymbol{X}) \leqslant 0,\ \ i = 1,2,\cdots,k \\ h_j(\boldsymbol{X}) = 0,\ \ j = 1,2,\cdots,l \end{cases} . \tag{11}$$

In the above formulation, the decision variables are denoted as $\boldsymbol{X}$=($\boldsymbol{x}_1$, $\boldsymbol{x}_2$, $\cdots$, $\boldsymbol{x}_n$). In (11), m optimization goals are considered simultaneously. The goal of this programming model is to obtain $\boldsymbol{X}^* = (\boldsymbol{x}_1^*, \boldsymbol{x}_2^*, \cdots, \boldsymbol{x}_n^*)$ so that $f(\boldsymbol{X}^*)$ obeys the constraints and approximates the minimum objective value.

Suppose that the MOPSO algorithm searches in the $\boldsymbol{D}$ dimension space, and a population is composed of n particles, $\boldsymbol{x}_i=(\boldsymbol{x}_{i1}, \boldsymbol{x}_{i2}, \cdots, \boldsymbol{x}_{id})$, represents the current position of particle i, and $\boldsymbol{v}_i = (\boldsymbol{v}_{i1}, \boldsymbol{v}_{i2}, \cdots, \boldsymbol{v}_{id})$ stands for the current velocity of particle i. $\boldsymbol{p}_{i\text{best}} = (\boldsymbol{p}_{i\text{best}1}, \boldsymbol{p}_{i\text{best}2}, \cdots, \boldsymbol{p}_{i\text{best}d})$ is the current best local position searched by particle i, and $\boldsymbol{g}_{\text{best}} = (\boldsymbol{g}_{\text{best}1}, \boldsymbol{g}_{\text{best}2}, \cdots, \boldsymbol{g}_{\text{best}d})$ is the best global position obtained by the entire particle swarm. In addition, the particle velocity and position update formula are presented as follows.

$$\begin{cases} \boldsymbol{v}_{id}^{t+1} = \omega \boldsymbol{v}_{id}^{t} + c_1 r_1 (\boldsymbol{p}_{i\text{best}d} - \boldsymbol{x}_{id}^{t}) + c_2 r_2 (\boldsymbol{g}_{\text{best}d} - \boldsymbol{x}_{id}^{t}) \\ \boldsymbol{x}_{id}^{t+1} = \boldsymbol{x}_{id}^{t} + \boldsymbol{v}_{id}^{t+1} \end{cases} \tag{12}$$

Among them, t is the current number of iterations, ω is the inertial weight, c_1 and c_2 are the learning factors, r_1 and r_2 are the random numbers in [0,1], the combination of c_1 and r_1 restricts the particle to be affected by its own factors, and the combination of c_2 and r_2 restricts the particle to be affected by population factors.

Algorithm 1 The standard MOPSO procedure

Step 1 Initialize a particle population $\boldsymbol{X}_N$ so that each particle has a random position and a random velocity. Set the required basic operating parameters c_1, c_2, the maximum number of iterations MaxIter and inertia weight ω. Let iter = 0, obtain the objective value with respect to each particle, then add the non-inferior solutions to the external archive, and obtain the non-inferior solution archive A.

Step 2 Determine the initial local leaders and global leaders.

Step 3 Update the position and velocity of all particles based on (12), and update local leaders $\boldsymbol{X}_i^{p_{\text{best}}}$.

Step 4 Maintain the external archives based on the new non-inferior solutions to form external archives for the next iteration and select global leaders $\boldsymbol{X}_i^{g_{\text{best}}}$.

Step 5 iter = iter + 1, if the termination condition is satisfied or the maximum number of iterations is reached, the global optimal position and fitness values are output. Otherwise, go to Step 3.

4.1 Improved PSO with GCDF

In the standard MOPSO, the role of the inertial weights ω is to maintain the inertia of the particle motion, and to balance the global search and local search capabilities of the algorithm. If a large inertia weight is set, there is a strong global search capability and a fast convergence speed, but the local search capability is weakened, and the solution accuracy is compromised. Otherwise, if the inertia weight is small, it has a strong local search capability and good solution accuracy, but the global search capability decreases, and it tends to fall into the local optimum. This shows that the adjustment of the inertia weight could directly affect the performance of the algorithm. It is demonstrated that the overall decreased inertia weight will help the MOPSO obtain the approximate range of the solution in the early stage of the search, in the later stage it could improve the local search ability and accelerate the MOPSO convergence rate. Based on this fact, we propose a strategy for fitting ω via the GCDF.

Gaussian distribution, also known as normal distribution, is one widely applied continuous random distribution. The general probability density function of the Gaussian distribution is as follows:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{13}$$

where x is a random variable, μ is the position parameter of the Gaussian distribution, and σ describes the degree of dispersion in the Gaussian distribution. Gaussian mutation has been proven to be an effective way to improve the PSO [37]. Lee et al. [38] updated the velocity formula with Gaussian mutation, but only replaced the uniform random number in the velocity formula with a Gaussian random number. Higashi et al. [39] proposed a PSO with Gaussian mutation, which generates an ambiguity value from Gaussian mutation during particle iteration. In recent years, the GCDF gradually attracts the attention of researchers. The form of GCDF is shown as below:

$$F(x;\mu,\sigma)=\frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} \mathrm{e}^{\frac{-(t-\mu)^2}{2\sigma^2}} \mathrm{d}t. \tag{14}$$

Han et al. [40] applied the GCDF to map the probability model of the compact genetic algorithm, which improves the evaluation ability of the algorithm. The present study combines GCDF and MOPSO to propose a new MOPSO improvement strategy, referred to as GCDF-MOPSO. The core of GCDF-MOPSO is to construct a piecewise function to control the inertial weights based on GCDF. Extensive testing validates the effectiveness of this strategy by showing that the convergence rate and the overall performance are improved significantly. The test process is demonstrated in Section 4, and the improved formula of ω is as below:

$$\omega(x;\mu_1,\mu_2,\sigma_1,\sigma_2) = \begin{cases} \omega_{\max} - a\cdot \text{GCDF}(bx,\mu_1,\sigma_1), & 0 < x \leqslant 0.5 \\ \dfrac{\omega_{\max}+\omega_{\min}}{2} - a\cdot \text{GCDF}(bx,\mu_2,\sigma_2), & 0.5 < x < 1 \end{cases} \tag{15}$$

$$\omega(x;\mu_1,\mu_2,\sigma_1,\sigma_2)=$$
$$\begin{cases} \omega_{\max}-\dfrac{a}{\sigma_1\sqrt{2\pi}}\displaystyle\int_{-\infty}^{bx}\mathrm{e}^{\frac{-(t-\mu_1)^2}{2\sigma_1{}^2}}\mathrm{d}t, & 0<x\leqslant 0.5 \\ \dfrac{\omega_{\max}+\omega_{\min}}{2}-\dfrac{a}{\sigma_1\sqrt{2\pi}}\displaystyle\int_{-\infty}^{bx}\mathrm{e}^{\frac{-(t-\mu_2)^2}{2\sigma_2{}^2}}\mathrm{d}t, & 0.5<x<1 \end{cases}. \tag{16}$$

In order to find a suitable function curve which satisfies the degeneration mechanism, a large number of numerical experiments are conducted, and the results show that it is appropriate to take a=0.4, b=1, μ_1=0.2, μ_2=0.8, $\omega_{\max}$=0.9 and $\omega_{\min}$=0.1. Equations (15) and (16) are the improved formulae of ω in the proposed GCDF-MOPSO, where x = Iter/MaxIter. Fig. 4 shows the fitting curve based on the GCDF, and it is observed that the GCDF could be divided into two stages. When $0 < x < 0.5$, the algorithm is in the exploration stage, and it is still in the initial state of iterations. The entire solution space is large, and ω could perform a global search with large step sizes. At this stage, the global search ability is the strongest. When x=0.5, the global search ability and local search ability reach a relative balance. When $0.5 < x < 1$, the algorithm enters the "convergence" stage, the local search capability gradually increases, and the search step size decreases until ω degenerates to $\omega_{\min}$.

$$\omega(x)=\begin{cases} 0.9-\dfrac{2}{5\sigma_1\sqrt{2\pi}}\displaystyle\int_{-\infty}^{x}\mathrm{e}^{\frac{-(t-0.2)^2}{2\sigma_1{}^2}}\mathrm{d}t, & 0<x\leqslant 0.5 \\ \dfrac{0.9+0.1}{2}-\dfrac{2}{5\sigma_2\sqrt{2\pi}}\displaystyle\int_{-\infty}^{x}\mathrm{e}^{\frac{-(t-0.8)^2}{2\sigma_2{}^2}}\mathrm{d}t, & 0.5<x<1 \end{cases} \tag{17}$$

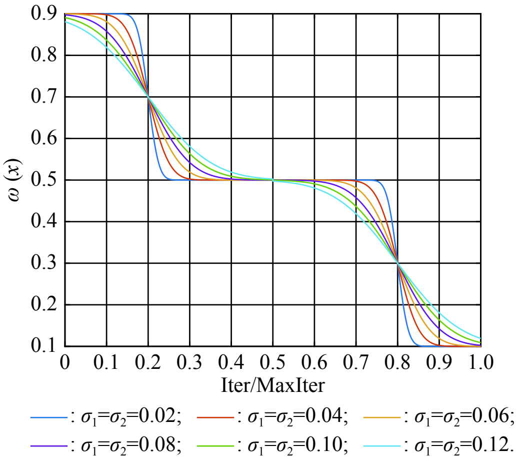


Fig. 4 Sample curves of the GCDF

4.2 Improved PSO with BM

The standard MOPSO algorithm bears the defect of low convergence rate while solving multi-objective problems. The particles falling into the local optimum easily is the primary cause for this problem. In order to improve this situation, the present study integrates the BM to help particles jump out of the current predicament of local optimum. Abdechiri et al. [41] proposed a Gases Brownian motion optimization (GBMO) algorithm, based on Gases Brownian motion and turbulent rotational motion. Aci et al. [42] used BM to improve the randomization stage of the dragonfly algorithm (DA). Typically, the BM is a Markovian stochastic process in continuous time and state space [43]. In the range of $t > 0$, it is with a continuous state function and a sample function. By assuming the transition process as $\{B(t), t\geqslant 0\}$, the standard BM satisfies the following conditions:

(i) $B(0) = 0$;

(ii) $\{B(t), t\geqslant 0\}$ is an homogeneous independent smooth incremental process;

(iii) For any $t > 0$, $B(t)$ is a random variable with Gaussian distribution, $B(t) \sim N(0, t)$;

(iv) For any $s < t$, $B(t) - B(s) \sim N(0, t-s)$, and $\mathrm{E}(B(t)) = 0$.

Fig. 5 shows the simulation trajectory after 500 steps of the BM in a 3-dimensional space with step size 1 and starting from the coordinate (0,0,0). A few studies focus on the combination of the BM and intelligent algorithms. Liu et al. [44] applied the BM to guide the search to solve the multidimensional knapsack problem (MKP). In the economic dispatch optimization problem, Han et al. [45] proposed a diffusion particle optimization (DPO) based on the diffusion mechanism of BM. In theory, a random walk can be defined as $\boldsymbol{X}_{N+1} = \boldsymbol{X}_N + \boldsymbol{B}_N$ [42], where $\boldsymbol{X}_N$ is the solution of step N, and $\boldsymbol{B}_N$ is a random vector. The random motion mechanism of BM can effectively help the algorithm to jump out of the local optimum, which could better improve the overall performance of the algorithm, and enhance the search ability in an uncertain environment.

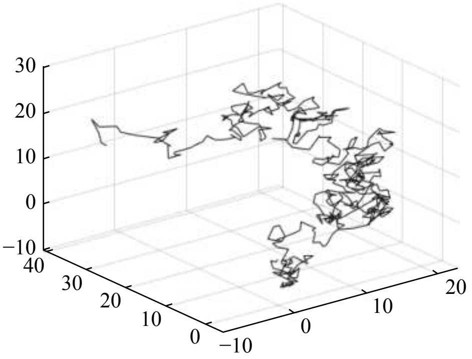


Fig. 5 Simulation of BM

$$\begin{cases} \boldsymbol{v}_{id}^{t+1} = \omega\boldsymbol{v}_{id}^{t}+c_1 r_1(\boldsymbol{p}_{i\mathrm{best}d}-\boldsymbol{x}_{id}^{t})+c_2 r_2(\boldsymbol{g}_{\mathrm{best}d}-\boldsymbol{x}_{id}^{t}) \\ \boldsymbol{x}_{id}^{t+1} = \boldsymbol{x}_{id}^{t}(1+\eta B(t))+\boldsymbol{v}_{id}^{t+1} \end{cases} \tag{18}$$

where η is the control coefficient of the random variable. For the reconfigurable production line scheduling problem, with an extensive set of numerical tests, we believe that $\eta = 0.5$ is recommended for the setting.

4.3 Mutation operation

The MOPSO-BM proposed in the present paper uses the special mutation operator proposed in [30]. This method ensures that every decision variable is explored in the global scope, and it can effectively improve the exploration ability of the algorithm, while avoiding premature convergence in some optimization problems.

4.4 Double-archive mechanism

To further enhance the search capability of the algorithm, we have also employed the double-archive mechanism proposed in the MOPSO-Lévy flight and double-archive (MOPSO-LFDA) [24]. In the double-archive mechanism, unlike the traditional archive method, where the particles with a small crowding distance are directly deleted when the main archive is full, the particles with a small crowding distance are put into the secondary archive. With the additional archive, more particles are retained, which prevents some particles from being deleted by mistake. It is shown that the mechanism could greatly improve the diversity of the solutions.

4.5 The MOPSO-BM algorithm

Algorithm 2 MOPSO-BM

Step 1 Initialize a particle population $\boldsymbol{X}_N$ so that each particle has a random vector position and a random vector velocity. Set the required basic operating parameters c_1 and c_2, the maximum number of iterations MaxIter, and the inertia weight $\omega_{\max}$, $\omega_{\min}$. Let iter = 1, calculate the objective value in accordance with each particle, then add the non-inferior solutions to the external archive, and obtain the non-inferior solution archive A.

Step 2 Determine the initial personal leaders and global leaders, based on the crowding distance of the particles in archive A.

Step 3 For iter to MaxIter

(i) Select the global leader $\boldsymbol{X}_i^{g_{\text{best}}}$ based on the crowding distance of the particles in archive A.

(ii) Update the velocity and position of particles according to (18).

(iii) Implement the adaptive polynomial mutation strategy.

(iv) Evaluate the particles according to the objective values.

(v) Update the external archive based on the double-archive mechanism.

(vi) If the ending criterion is met or the maximum number of iterations is reached, then output the global optimal position and fitness values.

The main computational cost of MOPSO-BM and MOPSO involves the iterative process of updating the position information and velocity information of the particles. Let us analyze the computational complexity of MOPSO-BM in detail. According to the MOPSO-BM algorithm flow, it can be seen that Step 3 has the largest calculation cost. Step 3(i) of MOPSO-BM needs to calculate the crowding distance of all particles in archive A and sort them, which requires $O(n^2)$ operations. Step 3(ii) and Step 3(iii) require $O(n)$ operations. In Step 3(iv), it is necessary to calculate the fitness according to the objective function. The calculation complexity depends on the complexity of the function $f(n)$, which is $O(f(n))$. In Step 3(v), the double-archive mechanism requires $O(n^2)$. Combining the number of iterations MaxIter and dimension D and comprehensively analyzing the above steps, the computational complexity of MOPSO-BM is $O(\text{MaxIter}\cdot D\cdot(n^2+f(n)))$.

5. Experiments on benchmarks

In order to verify the effectiveness of the algorithm proposed in this paper, benchmark test functions are employed in this section. In the process of running the algorithm, the number of population particles and the number of iterations are selected according to the complexity of the function. Furthermore, each function is tested multiple times to eliminate randomness. Here we use ZDT1, ZDT2, ZDT3, ZDT4 and ZDT6 introduced by Deb for testing (see Table 3) [46]. In the algorithm comparison tests, we also employ three groups of the testing data, from Kacem et al. [47,48] and Xia and Wu [49]. In the mentioned works, the production line is of partial flexibility, while the problems studied here are with full flexibility.

5.1 Test functions and evaluation index

The concept of generational distance (GD) is first introduced by Veldhuizen and Lamont [50], and it is used to measure the distance between the calculated Pareto frontier (PF) and the true Pareto frontier (True PF). GD is defined as in (19). $\boldsymbol{P}^*$ is a set of uniformly sampled solutions on True PF, and $\boldsymbol{S}$ represents the solution set PF of the multi-objective optimization algorithm. Dist($\boldsymbol{x}$, $\boldsymbol{P}^*$) stands for the Euclidean distance between the closest individuals from $\boldsymbol{x}\in\boldsymbol{S}$ to $\boldsymbol{P}^*$, and $|\boldsymbol{S}|$ is the cardinality of set $\boldsymbol{S}$. It is apparent that the smaller the GD value, the better the convergence of $\boldsymbol{S}$, and the closer it is to True PF.

$$\text{GD} = \frac{\sqrt{\sum_{\boldsymbol{x}\in\boldsymbol{S}} \text{Dist}(\boldsymbol{x},\boldsymbol{P}^*)^2}}{|\boldsymbol{S}|} \tag{19}$$

Table 3 Multi-objective benchmark function-ZDT series

Function name	Objective function	Variable bound	D	Property of the Pareto front
ZDT1	$\begin{cases} f_1(x) = x_1 \\ f_2(x) = g(x)\left[1 - \sqrt{x_1/g(x)}\right] \\ g(x) = 1 + \dfrac{9\left(\sum_{i=2}^{D} x_i\right)}{D-1} \end{cases}$	$x_i \in [0,1]$	30	Convex
ZDT2	$\begin{cases} f_1(x) = x_1 \\ f_2(x) = g(x)\left[1 - (x_1/g(x))^2\right] \\ g(x) = 1 + \dfrac{9\left(\sum_{i=2}^{D} x_i\right)}{D-1} \end{cases}$	$x_i \in [0,1]$	30	Nonconvex
ZDT3	$\begin{cases} f_1(x) = x_1 \\ f_2(x) = g(x)\left[1 - \sqrt{x_1/g(x)} - x_1 \sin(10\pi x_1)/g(x)\right] \\ g(x) = 1 + \dfrac{9\left(\sum_{i=2}^{D} x_i\right)}{D-1} \end{cases}$	$x_i \in [0,1]$	30	Convex disconnect
ZDT4	$\begin{cases} f_1(x) = x_1 \\ f_2(x) = g(x)\left[1 - \sqrt{x_1/g(x)}\right] \\ g(x) = 1 + 10(D-1) + \sum_{i=2}^{n}\left[x_i^2 - 10\cos(4\pi x_i)\right] \end{cases}$	$x_i \in [0,1]$ $x_i \in [-5,5]$ $i = 2,\cdots,D$	10	Nonconvex
ZDT6	$\begin{cases} f_1(x) = 1 - \mathrm{e}^{-4x_1}\sin^6(6\pi x_1) \\ f_2(x) = g(x)\left[1 - (f_1(x)/g(x))^2\right] \\ g(x) = 1 + 9\left[\dfrac{\left(\sum_{i=2}^{D} x_i\right)}{D-1}\right]^{0.25} \end{cases}$	$x_i \in [0,1]$	10	Nonconvex

Hypervolume (HV) [51] is proposed for comprehensive evaluation of the convergence and diversity. Given a set of preset reference points $\boldsymbol{r}^* = (r_1^*, r_2^*, \cdots, r_m^*)$ distributed in the target space, and a set of PF obtained by the algorithm, where $\boldsymbol{r}^*$ is dominated by all the solutions in $\boldsymbol{S}$. HV measures the volume of the target space dominated by $\boldsymbol{S}$ with $\boldsymbol{r}^*$ as the boundary, and its definition is shown in (20), where VOL($\cdot$) represents Lebesgue measure. It is observed that the larger the HV value, the closer $\boldsymbol{S}$ is to True PF.

$$\mathrm{HV}(\boldsymbol{S}) = \mathrm{VOL}\left(\bigcup_{\boldsymbol{x}\in \boldsymbol{S}} [f_1(\boldsymbol{x}), r_1^*] \times \cdots \times [f_m(\boldsymbol{x}), r_m^*]\right) \quad (20)$$

5.2 Parameter setting

As for the parameter setting of MOPSO-BM in Table 4, the overall preference is more conducive to solving multi-objective optimization problems. c_1 and c_2 are the acceleration coefficients of the particles, which are usually set to 2.0, but it has been found through testing that for the MOPSO-BM, a set of 1.5 can achieve a better overall performance. The flowchart of the MOPSO-BM algorithm is shown in Fig.6.

The parameters of MOPSO-BM include a, b, μ_1, μ_2, σ_1, σ_2 and η, among which μ_1, μ_2, σ_1, σ_2 and η are key parameters that can be adjusted and used, and their value ranges are $0.1 \leqslant \mu_1 \leqslant 0.4$, $0.6 \leqslant \mu_2 \leqslant 0.9$, $0.02 \leqslant \sigma_1 \leqslant 0.14$, $0.02 \leqslant \sigma_2 \leqslant 0.14$, $0 \leqslant \eta \leqslant 1$. In addition, a=0.4 and b=1 are determined, and the test shows that other values of a and b may cause the algorithm not to converge. Based on ZDT2, the key parameters of the MOPSO-BM are tested in detail on the PlatEMO. It can be seen from Fig. 7, Fig. 8 and Fig. 9 that when μ_1=0.2, μ_2=0.8, σ_1=σ_2=0.11, and η=0.5, a smaller GD and a larger HV are obtained. Therefore, these parameter settings are recommended, which can enable the algorithm to obtain a better convergence and overall performance.

Table 4 Parameters settings for different algorithms

Algorithm	Parameter
MOPSO-BM	$c_1 = c_2 = 1.5$, $a = 0.4$, $b = 1$, $\mu_1 = 0.2$, $\mu_2 = 0.8$, $\omega_{\max} = 0.9$, $\eta = 0.5$, $\omega_{\min} = 0.1$, $\sigma_1 = \sigma_2 = 0.11$, rep$A$size = 100, rep$B$size = 30
MOPSO-LFDA	$\omega \in [0.1, 0.5]$, $T \in [5, 30]$, rep$A$size = 100, rep$B$size = 30, $\theta = 5$
NSGA-Ⅱ	pro$C$ = pro$M$ = 1, dis$C$ = dis$M$ = 20
MOEA/D	pro$C$ = pro$M$ = 1, dis$C$ = dis$M$ = 20
SMPSO	$\omega = 0.4$, $c_1 = c_2 = 1.5$, $pm = 1/n$
dMOPSO	$\omega = 0.4$, $Ta = 2$

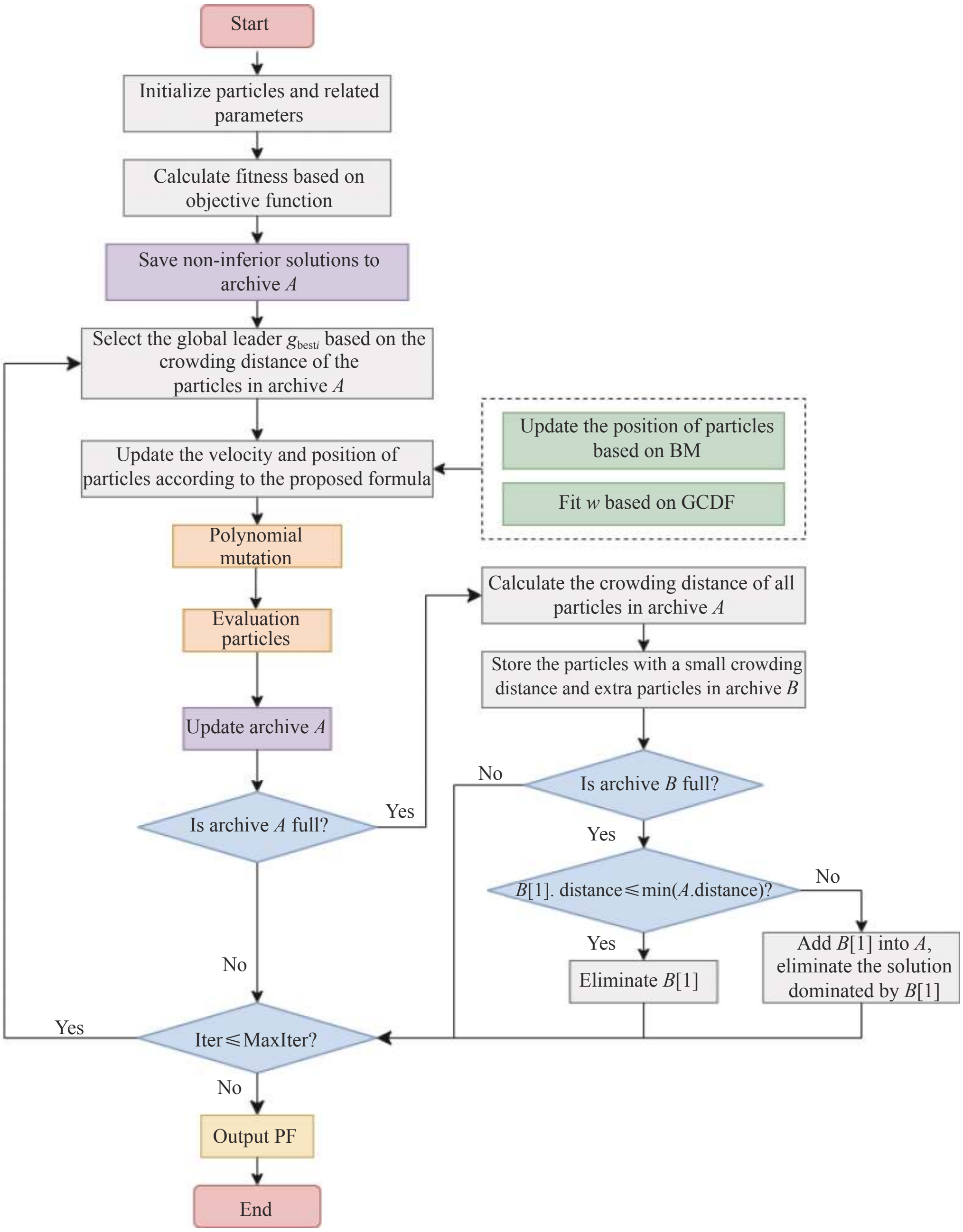

Start
Initialize particles and related parameters
Calculate fitness based on objective function
Save non-inferior solutions to archive A
Select the global leader $g_{\text{best}i}$ based on the crowding distance of the particles in archive A
Update the velocity and position of particles according to the proposed formula
Update the position of particles based on BM
Fit w based on GCDF
Polynomial mutation
Evaluation particles
Update archive A
Is archive A full?
Yes
Calculate the crowding distance of all particles in archive A
Store the particles with a small crowding distance and extra particles in archive B
Is archive B full?
No
Yes
B[1]. distance≤min(A.distance)?
No
Yes
Eliminate B[1]
Add B[1] into A, eliminate the solution dominated by B[1]
No
Yes
Iter≤MaxIter?
No
Output PF
End

Fig. 6 Flowchart of the MOPSO-BM algorithm

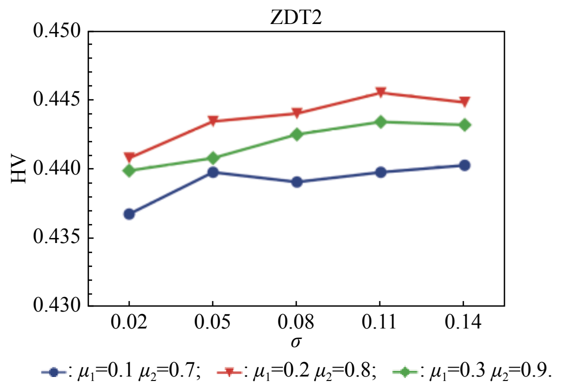


Fig. 7 Factor level trend (a)

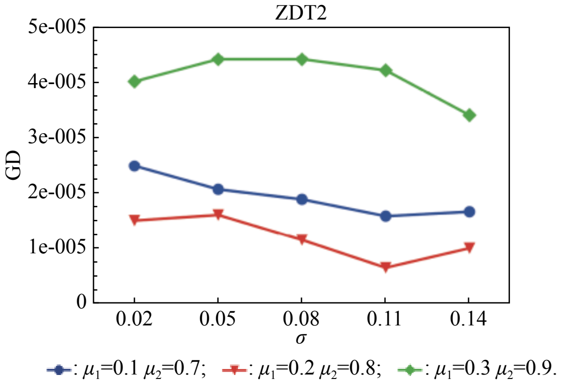


Fig. 8 Factor level trend (b)

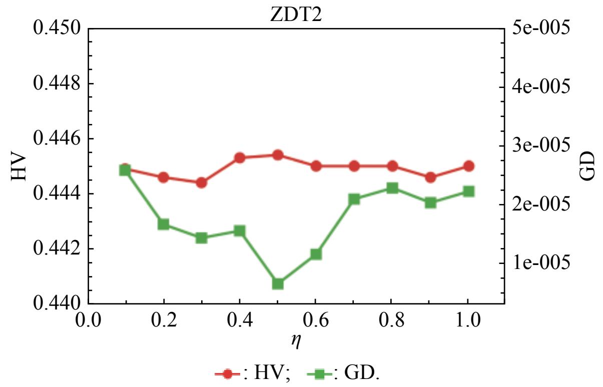


Fig. 9 Factor level trend (c)

5.3 Algorithm comparison

Based on the ZDT series of test functions, we test and compare several latest MOPSO algorithms with the one proposed here, including MOPSO-BM, MOPSO-LFDA [24], NSGA-Ⅱ [52], multi-objective evolutionary algorithm based on decomposition (MOEA/D) [53], speed-constrained multiobjective PSO (SMPSO) [54], decomposition-based multi-objective PSO (dMOPSO) [55]. MOPSO-LFDA is an improved algorithm proposed by Guan and Han, based on Lévy flight and double-archive mechanism. Lévy flight expands the search range of the particles and improves the probability of particles jumping out of the local area. The double-archive mechanism keeps as many particles as possible. The improved algorithm proposed in this paper also employs the double-archive mechanism, so it is necessary to compare the performance of MOPSO-BM and MOPSO-LFDA.

In the numerical experiments, in order to ensure the comparability of the data, uniform parameters are assumed. The number of particle populations is set to 100, and the maximum evaluation is set to 30 000. Table 4 describes the detailed parameters of all the listed algorithms, which are run on the PlatEMO [56]. The parameter setting of MOPSO-BM in Table 4 refers to Subsection 5.2, and the parameters of other algorithms refer to MOPSO-LFDA. These settings allow for a better comparison.

All the algorithms are independently run 30 times to take the mean and standard deviation for the comparative analysis. The computing platform is built with Matlab R2018b, Intel(R) Core (TM) i7-7700 CPU @3.60 GHz, 16 GB RAM.

Table 5 shows the GD obtained by solving the ZDT function using different algorithms. The smaller the GD value, the better the convergence performance of the algorithm. For ZDT1−ZDT4, the average and standard deviations of GD obtained by the proposed MOPSO-BM are the smallest. The results show that MOPSO-BM has excellent and stable convergence performance for ZDT1−ZDT4 functions. For ZDT6, NSGA-Ⅱ has the best performance.

Table 5 Test results of the GD based algorithms

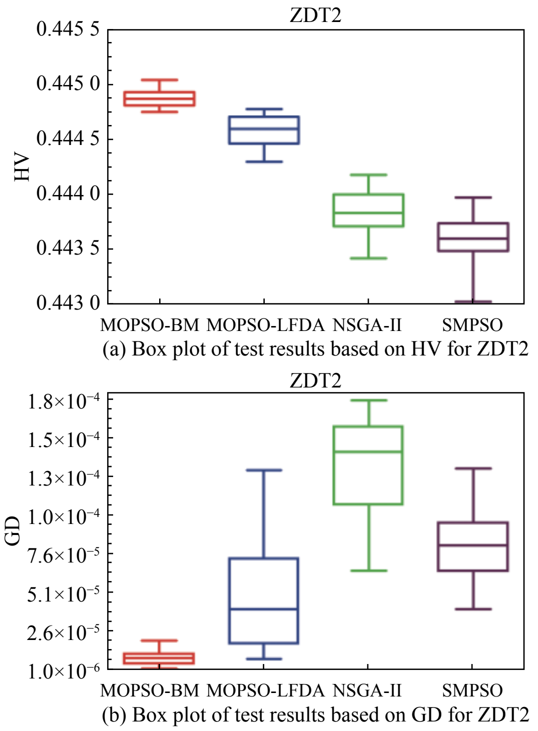
Function		MOPSO-BM	MOPSO-LFDA	NSGA-Ⅱ	MOEA/D	SMPSO	dMOPSO
ZDT1	Mean	**1.74E-05**	6.15E-05	1.58E-04	5.85E-04	9.72E-05	2.76E-03
	Standard	**1.23E-05**	4.02E-05	3.55E-05	1.83E-04	2.40E-05	7.11E-04
ZDT2	Mean	**5.88E-06**	4.95E-05	1.44E-04	1.34E-03	8.00E-05	3.90E-03
	Standard	**2.47E-06**	2.76E-05	3.55E-05	6.51E-04	2.53E-05	2.09E-03
ZDT3	Mean	**3.35E-05**	4.35E-05	7.61E-05	2.78E-03	1.05E-04	2.90E-03
	Standard	**3.97E-06**	1.34E-05	1.91E-05	2.97E-03	3.67E-05	9.57E-04
ZDT4	Mean	**6.69E-05**	1.42E-04	2.10E-04	1.66E-03	5.48E-01	6.36E-04
	Standard	**2.01E-05**	5.47E-05	9.73E-05	4.98E-04	4.10E-01	3.10E-04
ZDT6	Mean	2.41E-02	7.70E-02	**4.16E-05**	8.46E-04	9.97E-03	3.57E-04
	Standard	2.23E-02	4.04E-02	**2.84E-05**	2.86E-04	2.45E-02	9.35E-04

Table 6 shows the HV obtained by using different algorithms to solve the ZDT function. The larger the HV, the better the overall performance of the algorithm. Both MOPSO-BM and MOPSO-LFDA obtain a higher HV. However, through comparison, it is found that for ZDT1−ZDT4, the standard deviation of MOPSO-BM is smaller, indicating that the stability of MOPSO-BM is better.

Table 6 Test results of the HV based algorithms

Function		MOPSO-BM	MOPSO-LFDA	NSGA-II	MOEA/D	SMPSO	dMOPSO
ZDT1	Mean	**7.21E-01**	**7.21E-01**	7.19E-01	7.11E-01	7.19E-01	6.89E-01
	Standard	**4.51E-05**	7.26E-05	2.09E-04	6.90E-03	3.12E-04	7.88E-03
ZDT2	Mean	**4.45E-01**	**4.45E-01**	4.44E-01	4.14E-01	4.44E-01	3.90E-01
	Standard	**3.15E-05**	6.39E-05	1.62E-04	3.47E-02	3.20E-04	5.99E-02
ZDT3	Mean	**6.07E-01**	5.83E-01	6.05E-01	6.04E-01	6.01E-01	6.05E-01
	Standard	**7.20E-05**	1.52E-04	2.25E-02	3.51E-02	4.05E-03	1.10E-02
ZDT4	Mean	**7.21E-01**	**7.21E-01**	7.17E-01	6.98E-01	1.29E-02	7.13E-01
	Standard	**1.49E-04**	1.52E-04	1.26E-03	7.99E-03	4.59E-02	3.61E-03
ZDT6	Mean	**3.90E-01**	3.89E-01	3.88E-01	3.81E-01	3.88E-01	3.87E-01
	Standard	2.57E-04	**7.67E-05**	3.19E-04	2.48E-03	2.09E-04	4.98E-03

Fig. 10 shows that MOPSO-BM can obtain a better mean and smaller fluctuations compared with other algorithms.



(a) Box plot of test results based on HV for ZDT2

(b) Box plot of test results based on GD for ZDT2

Fig. 10 Box plots of test results based on HV and GD

Fig. 11 and Fig. 12 show the comparison between PF and True PF. Fig. 13 shows the convergence ability of different algorithms. It can be seen that the convergence ability of MOPSO-BM is outstanding. Table 7 shows the results of multiple algorithms on the flexible production scheduling problem. The Xia and Wu data set contains three test problems from Kacem et al. [47,48] and Xia and Wu [49].

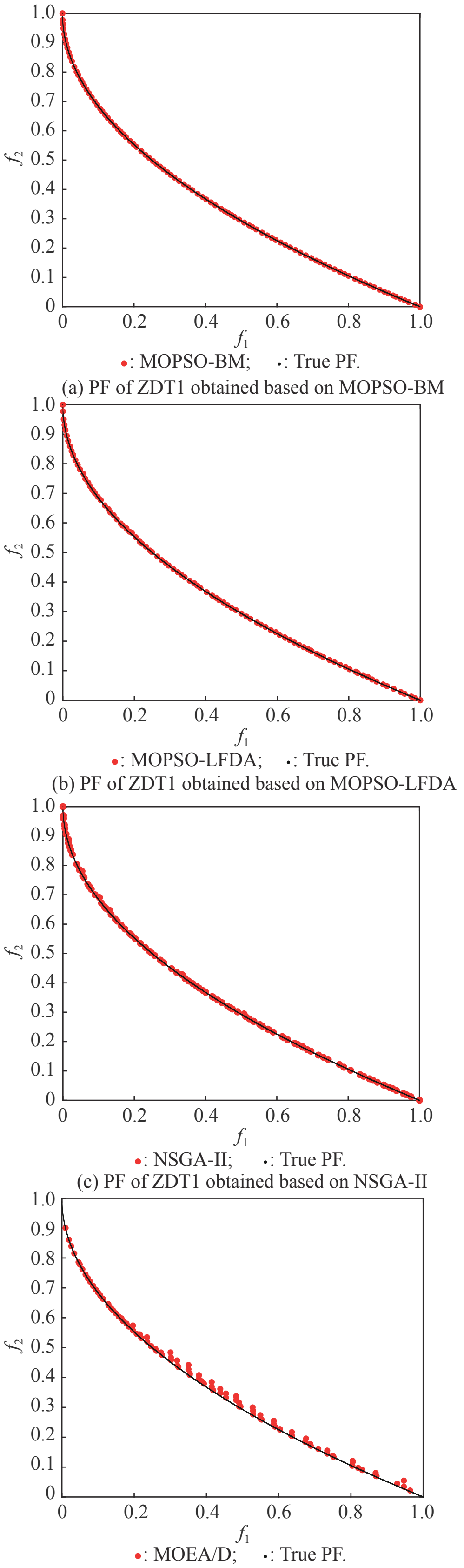
The three objective functions [48] are f_1, f_2, f_3, where f_3 represents the total processing load of all the machines. W_k stands for the load of a single machine among all the machines. f_2 demonstrates the maximum load of a single machine among all the machines.

$$f_1 = \min_{1<i<N}\left(\max_{1<k<K} T_{ik}\right),$$

$$f_2 = \min(\max_{1<k<K} W_k),$$

$$f_3 = \min \sum_k W_k,\ \ 1 \leqslant k \leqslant m.$$

The approach by localization and the controlled genetic algorithm (AL+CGA) algorithm was proposed by Kacem et al. [47,48], the PSO+SA algorithm was proposed by Xia and Wu [49], the multistage operation-based genetic algorithm (moGA) algorithm was proposed by Zhang and Gen [57], and the hybrid genetic algorithm (hGA) was proposed by Gao and Sun et al. [58]. Table 7 lists the optimal results of all the algorithms after five runs. It is observed that in searching for a better objective function value, the performance of MOPSO-BM is outstanding. Fig. 14 is a Gantt chart obtained by solving a 15×10 problem via PSO+SA. Fig. 15 is also a Gantt chart plotted via solving the same sized problem via MOPSO-BM. The vertical axis of the Gantt chart represents the equipment number, and the horizontal axis stands for the processing time. The number in the Gantt chart indicates the workpiece number and the process number. For example, “2,1” represents the first process of the second workpiece. Besides, the same color indicates the same workpiece.



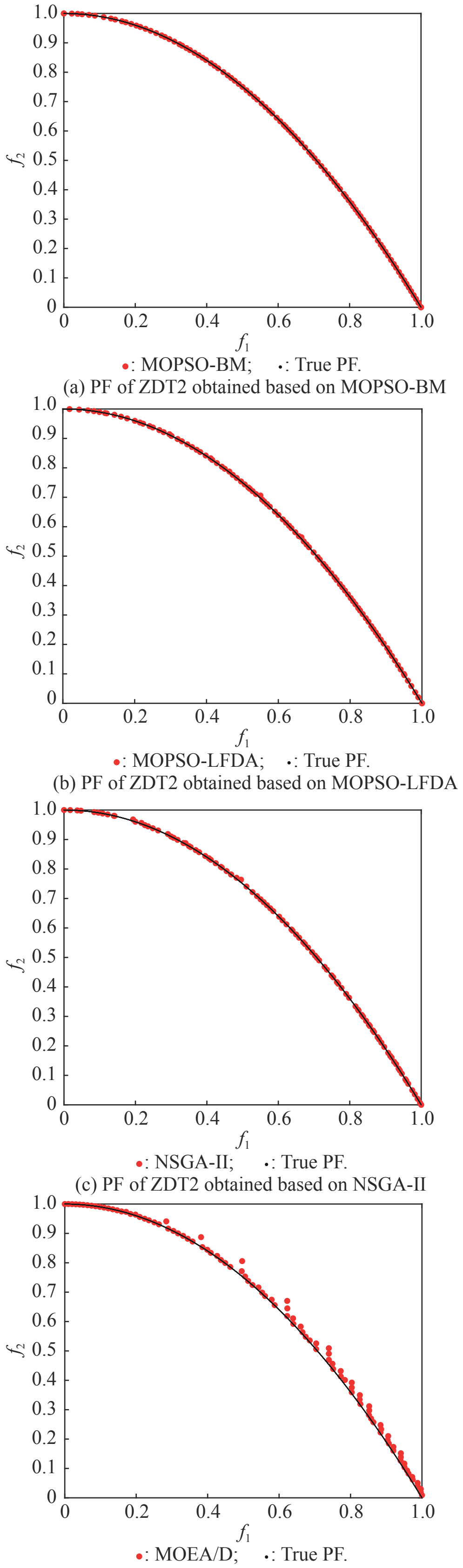
(a) PF of ZDT1 obtained based on MOPSO-BM

(b) PF of ZDT1 obtained based on MOPSO-LFDA

(c) PF of ZDT1 obtained based on NSGA-II

(d) PF of ZDT1 obtained based on MOEA/D

Fig. 11 PF and True PF of four algorithms for ZDT1



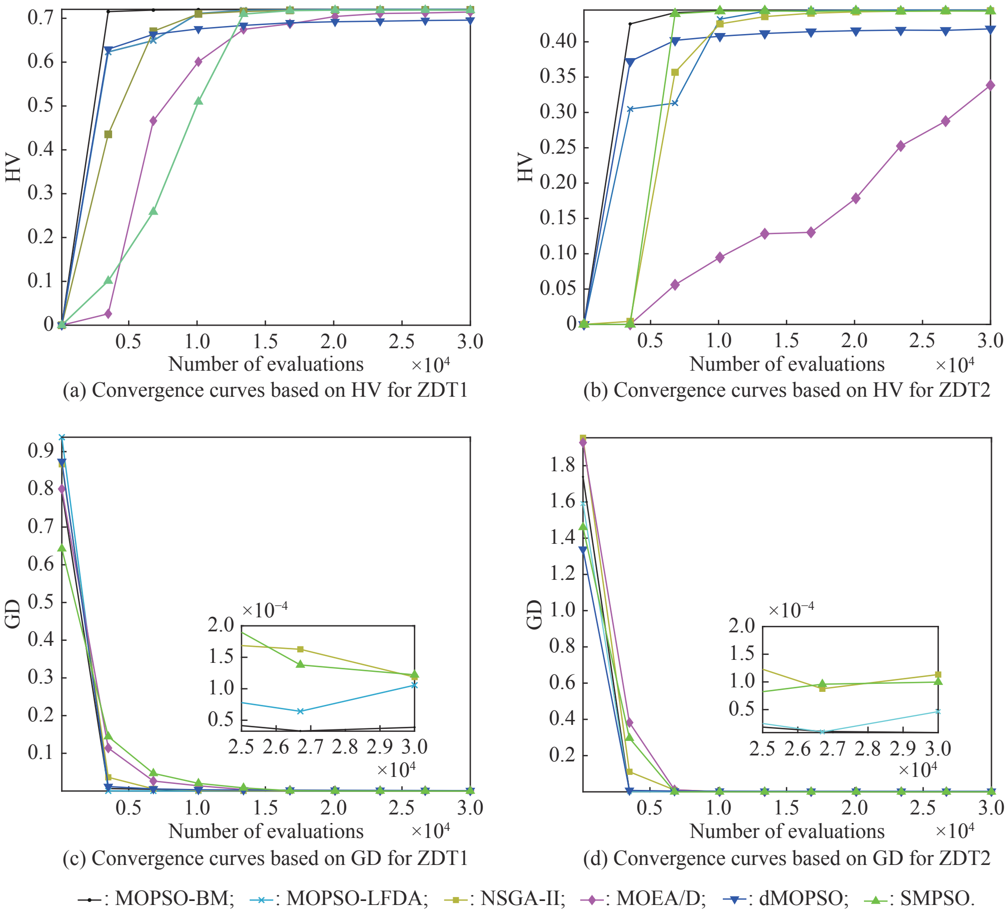
(a) PF of ZDT2 obtained based on MOPSO-BM

(b) PF of ZDT2 obtained based on MOPSO-LFDA

(c) PF of ZDT2 obtained based on NSGA-II

(d) PF of ZDT2 obtained based on MOEA/D

Fig. 12 PF and True PF of four algorithms for ZDT2



(a) Convergence curves based on HV for ZDT1

(b) Convergence curves based on HV for ZDT2

(c) Convergence curves based on GD for ZDT1

(d) Convergence curves based on GD for ZDT2

Fig. 13 Convergence curves of different algorithms

Table 7 Scheduling results for different algorithms on the Xia and Wu datasets

Problem (n×m)	Objective	AL+CGA		PSO+SA		moGA	hGA	MOPSO-BM	
								Result	Pop size
8×8	f_1	15	16	15	16	15	14	**15**	300
	f_2			12	13	14	12	**12**	
	f_3	79	75	75	73	73	77	**75**	
10×10	f_1	7		7		7	7	**7**	300
	f_2	5		6		5	5	**5**	
	f_3	45		44		43	43	**43**	
15×10	f_1	24		12			11	**11**	300
	f_2	11		11			11	**11**	
	f_3	91		91			91	**91**	

By comparing and analyzing Fig. 14 and Fig. 15, it is discovered that the makespan obtained by MOPSO-BM is smaller than that by PSO+SA, which validates that the MOPSO-BM scheduling scheme can provide the production schedule with a high efficiency. Second, it can be seen from Fig. 15 that M10, M7, and M2 will run at full load, effectively reducing the machine idle time. In comparison, the scheduling outcome in Fig. 14 suffers from more machine idle time and a lower resource utilization.

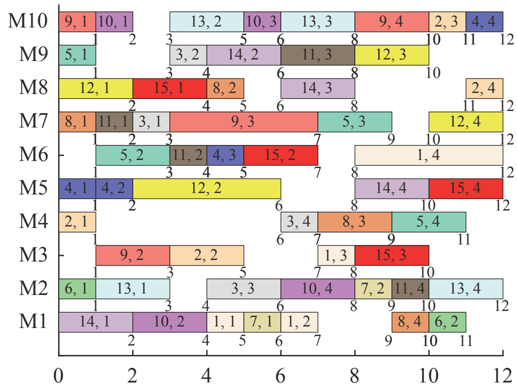


Fig. 14 Gantt chart for 15×10 problem solved via PSO+SA (f_1=12, f_2=11, f_3=91)

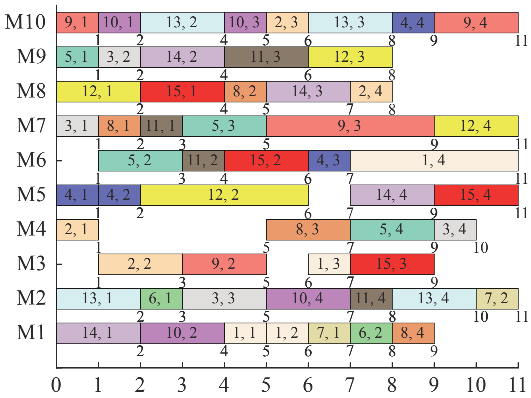


Fig. 15 Gantt chart for 15×10 problem solved via MOPSO-BM (f_1=11, f_2=11, f_3=91)

6. A case study for the smart home appliance production line

For an actual manufacturing plant responsible for smart home appliances, the reconfigurable assembly line in the mixed workshop processes three products at the same time, including parts processing and assembly. The models of the three smart sockets are SK539, SK504 and SK517, which are shown as detailed bill of materials in Fig. 16, Fig. 17, and Fig. 18, respectively. The detailed process and part processing information is listed in Table 8.

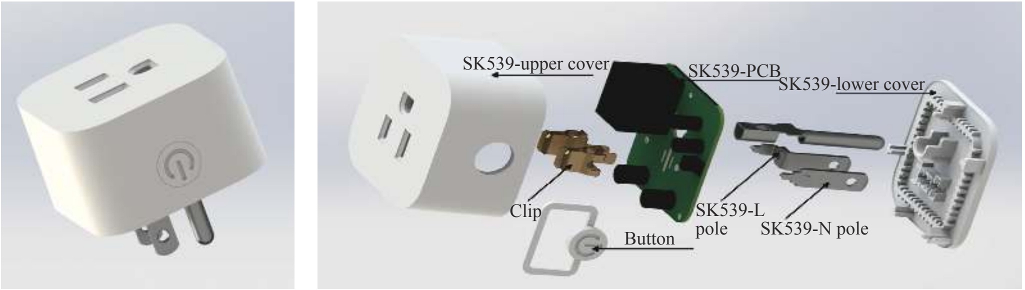


Fig. 16 Bill of materials for SK539

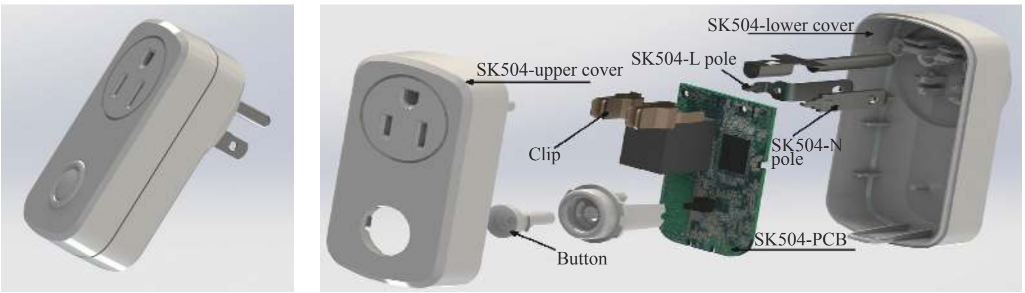


Fig. 17 Bill of materials for SK504

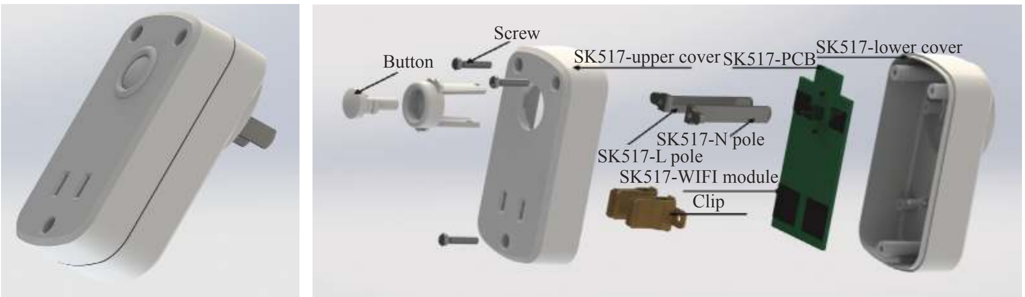


Fig. 18 Bill of materials for SK517

Table 8 Processing information for the jobs with respect to equipment

Job	Process	Processing information	Job	Process	Processing information
J1	O1,1	SK539-lower cover	J19	O19,1	Laser lettering lower cover
J2	O2,1	SK539-upper cover		O19,2	Insert L pole
J3	O3,1	SK504-lower cover		O19,3	Insert N pole
J4	O4,1	SK504-upper cover		O19,4	Install PCB
J5	O5,1	SK517-lower cover		O19,5	Welding L, N pole
J6	O6,1	SK517-upper cover		O19,6	Insert clip
J7	O7,1	Button		O19,7	Welding clip
J8	O8,1	SK539-L pole		O19,8	Test continuity
J9	O9,1	SK539-N pole		O19,9	Attach the cover
J10	O10,1	SK504-L pole		O19,10	Lock screw
J11	O11,1	SK504-N pole		O19,11	Test function
J12	O12,1	SK517-L pole	J20	O20,1	Laser lettering lower cover
J13	O13,1	SK517-N pole		O20,2	Welding WIFI module
J14	O14,1	Clip		O20,3	Insert L pole
J15	O15,1	SK539-PCB		O20,4	Insert N pole
J16	O16,1	SK504-PCB		O20,5	Install PCB
J17	O17,1	SK517-PCB		O20,6	Welding L, N pole
J18	O18,1	Laser lettering lower cover		O20,7	Insert clip
	O18,2	Insert L pole		O20,8	Welding clip
	O18,3	Insert N pole		O20,9	Test continuity
	O18,4	Install PCB		O20,10	Attach the upper cover
	O18,5	Welding L, N pole		O20,11	Lock screw
	O18,6	Insert clip		O20,12	Test function
	O18,7	Welding clip		Final Assembly	Counting and packing (Beyond the scope of this paper)
	O18,8	Test continuity			
	O18,9	Attach the upper cover			
	O18,10	Ultrasonic welding			
	O18,11	Test function			

In Table 8, the processing information of the three smart sockets SK539, SK504 and SK517 are demonstrated. Here we will use MOPSO-BM to solve the scheduling problem based on the three products on the reconfigurable flexible assembly line. “J1” means part 1, and “O1,1” stands for the first processing route of part 1. Table 9 presents the processing machine information located in the workshop. Fig. 19 reflects the sequence constraints of each processing procedure. Table 10 and Table 11 show the processing time required for all the processing steps on each equipment. “X” means that this step cannot be processed on this machine. It can be also seen from Table 10 and Table 11 that this production workshop adopts a partially flexible production method.

Table 9 Machine information in the flexible production line

Machine number	Type of equipment
M1, M2	Injection molding machine
M3, M4	Stamping machine
M5, M6	SMT placement machine
M7	Laser engraving machine
M8−M24	Six-axis industrial robot

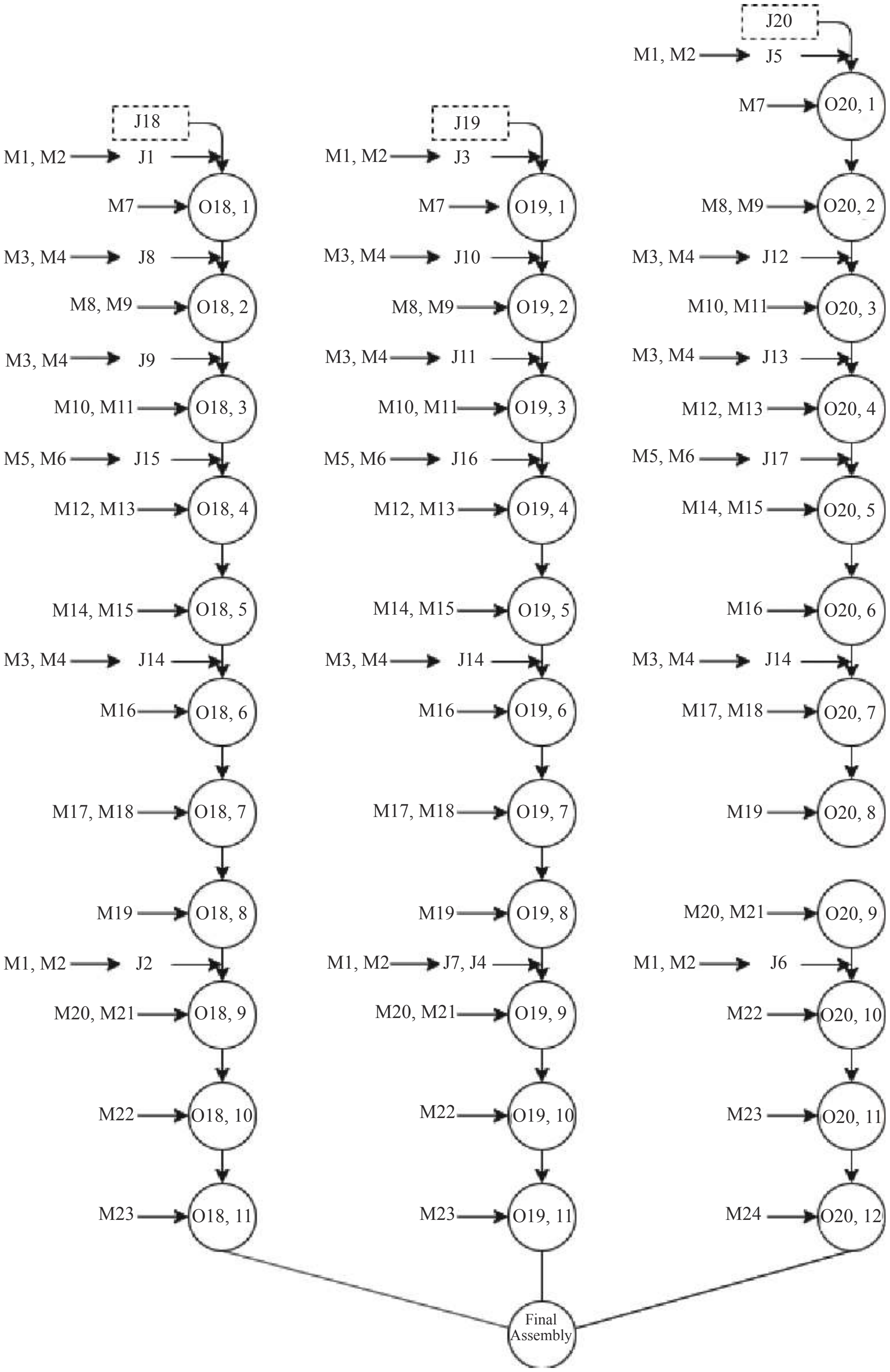


Fig. 19 Flow chart of the processing sequence for different jobs

To test and compare the three different algorithms, namely, the NSGA-Ⅱ algorithm, the dMOPSO algorithm and the MOPSO-BM algorithm, Table 12 lists the parameters used in all the algorithms. The solution results via the three algorithms are obtained in Table 13. The test results demonstrate that the MOPSO-BM algorithm can obtain more reasonable scheduling results under the same testing environment, and can achieve lower cost losses under the same order quantity and the minimum makespan.

Table 10 Processing time of each process (a)

s

Job	Process	M1	M2	M3	M4	M5	M6
J1	O1,1	9	8	X	X	X	X
J2	O2,1	9	7	X	X	X	X
J3	O3,1	6	5	X	X	X	X
J4	O4,1	6	7	X	X	X	X
J5	O5,1	7	7	X	X	X	X
J6	O6,1	7	7	X	X	X	X
J7	O7,1	9	9	X	X	X	X
J8	O8,1	X	X	6	7	X	X
J9	O9,1	X	X	5	6	X	X
J10	O10,1	X	X	6	9	X	X
J11	O11,1	X	X	6	6	X	X
J12	O12,1	X	X	6	9	X	X
J13	O13,1	X	X	5	8	X	X
J14	O14,1	X	X	6	6	X	X
J15	O15,1	X	X	X	X	9	5
J16	O16,1	X	X	X	X	8	9
J17	O17,1	X	X	X	X	8	9

Table 11 Processing time of each process (b)

s

Machine number	Process number	J18	J19	J20
M7	1	7	8	6
M8	2	9	9	10
M9	2	10	7	11
M10	3	11	14	12
M11	3	12	15	9
M12	4	9	11	10
M13	4	10	8	11
M14	5	11	10	9
M15	5	8	12	8
M16	6	3	4	4
M17	7	9	8	10
M18	7	7	11	8
M19	8	5	6	7
M20	9	12	11	12
M21	9	9	10	14
M22	10	7	8	5
M23	11	6	7	6
M24	12	X	X	6

Table 12 Parameters settings in different algorithms

Algorithm	Parameter	Setting
MOPSO-BM	$c_1 = c_2 = 1.5$, $a = 0.4$, $b = 1$, $\mu_1 = 0.2$, $\mu_2 = 0.8$, $\omega_{\max} = 0.9$, $\eta = 0.5$, $\omega_{\min} = 0.1$, $\sigma_1 = \sigma_2 = 0.11$, rep$A$size = 100, rep$B$size = 30	nPop : 100, MaxIt : 1 000, $cm = 10$, $gm = 2.0$, $bm = 1.0$, $\boldsymbol{W}_h = [6, 5, 10, 6, 15]$, $\boldsymbol{D}_h = [640, 550, 1\,050, 640, 1\,650]$
NSGA-Ⅱ	pro$C$ = pro$M$ = 1, dis$C$ = dis$M$ = 20	
dMOPSO	$\omega = 0.4$, $Ta = 2$	

Table 13 Comparative results of the scheduling problem with the three algorithms

Objective function	NSGA-Ⅱ	dMOPSO	MOPSO-BM
f_1	391	379	**378**
f_2	174 112	168 819	**168 378**
f_3	103	104	**103**
f_4	42	42	**42**

7. Conclusions

In the context of flexible intelligent manufacturing, the production units with various functions are integrated and reconfigured to cater the ever-changing customer demands. Starting from actual production, this paper proposes a new reconfigurable production line solution, which can help SMEs improve machine utilization and reduce production cost. For a typical reconfigurable assembly line, we establish a multi-objective optimization model to minimize the makespan, loss cost, and the total load of the machine, and in the meantime to maximize the number of completed orders. The conventional MOPSO is prone to be trapped in the local optimum due to its weak global search capability. To effectively tackle the multi-objective scheduling problem, an improved MOPSO algorithm is proposed based on GCDF and BM. The random motion mechanism of BM is combined with the MOPSO to effectively alleviate the weakness of the traditional algorithms. By using the piecewise GCDF to fit the inertia weight strategy, we could balance the global search ability and convergence rate of the algorithm during the iterations, and the solution quality could be improved as well.

Based on the evaluation indicators GD and HV, we test and compare the proposed MOPSO-BM algorithm with the other five latest multi-objective intelligent algorithms, for the benchmark functions from ZDT1 to ZDT6. The test results show that MOPSO-BM performs better in terms of convergence speed and solution quality. Besides, the proposed MOPSO-BM algorithm is tested on three benchmark problems of flexible flow shop scheduling, which validates the effectiveness of the algorithm. For a real case of the reconfigurable production line for smart home appliances, two latest algorithms, NSGA-Ⅱ and dMOPSO as well as MOPSO-BM, are employed to tackle the scheduling problem. The comparison results indicate that the MOPSO-BM algorithm outperforms the other two when dealing with such reconfigurable production line scheduling problems.

References

[1] BAI D Y, XUE H Y, et al. Effective algorithms for single-machine learning-effect scheduling to minimize completion-time-based criteria with release dates. Expert Systems with Applications, 2020, 156: 113445.

[2] KUO C H, HUANG H P, WEI K C. System modeling and real-time simulator for highly model-mixed assembly systems. Journal of Manufacturing Science and Engineering—Transactions of the ASME, 1999, 121(2): 282–289.

[3] GUO Z X, WONG W K, LEUNG S Y S, et al. A bi-level genetic algorithm for multi-objective scheduling of multi- and mixed-model apparel assembly lines. Multimedia Modeling, 2006, 4304: 934–941.

[4] HUANG R H, YANG C L. Solving a multi-objective overlapping flow-shop scheduling. International Journal of Advanced Manufacturing Technology, 2009, 42(9): 955–962.

[5] DAVOUDPOUR H, ASHRAFI M. Solving multi-objective SDST flexible flow shop using GRASP algorithm. International Journal of Advanced Manufacturing Technology, 2009, 44(7): 737–747.

[6] DUDAS C, FRANTZEN M, NG A H C. A synergy of multi-objective optimization and data mining for the analysis of a flexible flow shop. Robotics and Computer-Integrated Manufacturing, 2011, 27(4): 687–695.

[7] LIU Q M, DONG M. Improved particle swarm optimization algorithm application of the variable batch flow-shop problem. Advanced Science Letters, 2011, 4(6/7): 2369–2373.

[8] CHAUBE A, BENYOUCEF L, TIWARI M K. An adapted NSGA-2 algorithm based dynamic process plan generation for a reconfigurable manufacturing system. Journal of Intelligent Manufacturing, 2012, 23(4): 1141–1155.

[9] DAI M, TANG D B, GIRET A, et al. Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm. Robotics and Computer-Integrated Manufacturing, 2013, 29(5): 418–429.

[10] JOLAI F, ASEFI H, RABIEE M, et al. Bi-objective simulated annealing approaches for no-wait two-stage flexible flow shop scheduling problem. Scientia Iranica, 2013, 20(3): 861–872.

[11] SHEIKH S. Multi-objective flexible flow lines with due window, time lag, and job rejection. International Journal of Advanced Manufacturing Technology, 2013, 64(9−12): 1423–1433.

[12] TRAN T H, NG K M. A hybrid water flow algorithm for multi-objective flexible flow shop scheduling problems. Engineering Optimization, 2013, 45(4): 483–502.

[13] NADERI B, GOHARI S, YAZDANI M. Hybrid flexible flowshop problems: models and solution methods. Applied Mathematical Modelling, 2014, 38(24): 5767–5780.

[14] GHALEB M A, SURYAHATMAJA U S, ALHARKAN I M. Metaheuristics for two-stage no-wait flexible flow shop scheduling problem. Proc. of the International Conference on Industrial Engineering and Operations Management, 2015: 1–9.

[15] CHOI Y C. Dispatching rule-based scheduling algorithms in a single machine with sequence-dependent setup times and energy requirements. Procedia CIRP, 2016, 41: 135–140.

[16] DOU J P, LI J, SU C. Bi-objective optimization of integrating configuration generation and scheduling for reconfigurable flow lines using NSGA-II. International Journal of Advanced Manufacturing Technology, 2016, 86(5−8): 1945–1962.

[17] ZHAO Y R, FENG X Y, SHI L P, et al. The study on reconfigurable algorithm of the wood flexible manufacturing system based on OOTCPN-GASA. Proc. of the 2nd International Conference on Control and Robotics Engineering, 2017. DOI: 10.1109/ICCRE.2017.7935044.

[18] ASGHAR E, ZAMAN U K U, BAQAI A A, et al. Optimum machine capabilities for reconfigurable manufacturing systems. International Journal of Advanced Manufacturing Technology, 2018, 95(9−12): 4397–4417.

[19] GONG G L, CHIONG R, DENG Q W, et al. Energy-efficient flexible flow shop scheduling with worker flexibility. Expert Systems with Applications, 2020, 141: 112902.

[20] HAN Y Y, LI J Q, SANG H Y, et al. Discrete evolutionary multi-objective optimization for energy-efficient blocking flow shop scheduling with setup time. Applied Soft Computing, 2020, 93: 106343.

[21] HASANI A, HOSSEINI S M H. A bi-objective flexible flow shop scheduling problem with machine-dependent processing stages: trade-off between production costs and energy consumption. Applied Mathematics and Computation, 2020, 386: 125533.

[22] BOYSEN N, FLIEDNER M, SCHOLL A. Sequencing mixed-model assembly lines: survey, classification and model critique. European Journal of Operational Research, 2009, 192(2): 349–373.

[23] MOORE J, CHAPMAN R. Application of particle swarm to multi-objective optimization. Auburn, Alabama: Auburn University, 1999.

[24] GUAN T H, HAN F, HAN H. A modified multi-objective particle swarm optimization based on Levy flight and double-archive mechanism. IEEE Access, 2019, 7: 183444–183467.

[25] KENNEDY J, EBERHART R. Particle swarm optimization. Proc. of the IEEE International Conference on Neural Networks, 1995, 4: 1942–1948.

[26] COELHO L D S, AYALA V H, ALOTTO P. A multi-objective Gaussian particle swarm approach applied to electromagnetic optimization. IEEE Trans. on Magnetics, 2010, 46(8): 3289–3292.

[27] AL-DUWAISH H N. Identification of Hammerstein models with known nonlinearity structure using particle swarm optimization. Arabian Journal for Science and Engineering, 2006, 36(7): 1269–1276.

[28] JIANG S H, WANG Y, JI Z C. A new design method for adaptive IIR system identification using hybrid particle swarm optimization and gravitational search algorithm. Nonlinear Dynamics, 2015, 79(4): 2553–2576.

[29] ZHAO J, SUN J, LAI C H, et al. An improved quantum-behaved particle swarm optimization for multi-peak optimization problem. International Journal of Computer Mathematics, 2011, 88(3): 517–532.

[30] COELLO C A C, PULIDO G T, LECHUGA M S. Handling

multiple objectives with particle swarm optimization. IEEE Trans. on Evolutionary Computation, 2004, 8(3): 256–279.

[31] YANG J M, MU X W, CHE H J, et al. Improved multi-objective particle swarm optimization algorithm based on multiple strategies. Control and Decision, 2017, 32(3): 435–42. (in Chinese)

[32] KIAN S L, BUYAMIN S, AHMAD A, et al. An improved leader guidance in multi-objective particle swarm optimization. Proc. of the 6th Asia Modelling Symposium, 2012: 34–39.

[33] FAN J S. An improving multi-objective particle swarm optimization. Web Information Systems and Mining, 2010, 6318: 1–6.

[34] ZHOU A M, QU B Y, LI H, et al. Multi-objective evolutionary algorithms: a survey of the state of the art. Swarm and Evolutionary Computation, 2011, 1(1): 32–49.

[35] ZEINALZADEH A, MOHAMMADI Y, MORADI M H. Optimal multi-objective placement and sizing of multiple DGs and shunt capacitor banks simultaneously considering load uncertainty via MOPSO approach. International Journal of Electrical Power & Energy Systems, 2015, 67: 336–349.

[36] TAVANA M, LI Z J, MOBIN M, et al. Multi-objective control chart design optimization using NSGA-III and MOPSO enhanced with DEA and TOPSIS. Expert Systems with Applications, 2016, 50: 17–39.

[37] SUN Y, GAO Y L. A multi-objective particle swarm optimization algorithm based on Gaussian mutation and an improved learning strategy. Mathematics, 2019. DOI: 10.3390/math7020148.

[38] LEE C H, LEE Y C. Nonlinear systems design by a novel fuzzy neural system via hybridization of electromagnetism-like mechanism and particle swarm optimisation algorithms. Information Sciences, 2012, 186: 59–72.

[39] HIGASHI N, IBA H. Particle swarm optimization with Gaussian mutation. Proc. of the IEEE Swarm Intelligence Symposium, 2003: 72–79.

[40] HAN Z H, ZHANG Q. An improved compact genetic algorithm for scheduling problems in a flexible flow shop with a multi-queue buffer. Processes, 2019, 7(5): 302.

[41] ABDECHIRI M, MEYBODI M R, BAHRAMI H. Gases Brownian motion optimization: an algorithm for optimization (GBMO). Applied Soft Computing, 2013, 13(5): 2932–2946.

[42] ACI C I, GULCAN H. A modified dragonfly optimization algorithm for single- and multiobjective problems using Brownian motion. Computational Intelligence and Neuroscience, 2019, 2019(2): 1–17.

[43] HIDA T. Brownian motion. New York: Springer, 1980.

[44] LIU J J, WU C Z, CAO J, et al. A binary differential search algorithm for the 0-1 multidimensional knapsack problem. Applied Mathematical Modelling, 2016, 40(23/24): 9788–9805.

[45] HAN L, ROMERO C E, YAO Z. Economic dispatch optimization algorithm based on particle diffusion. Energy Conversion and Management, 2015, 105: 1251–1260.

[46] ZITZLER E, DEB K, THIELE L. Comparison of multiobjective evolutionary algorithms: empirical results. Evolutionary Computation, 2000, 8(2): 173–195.

[47] KACEM I, HAMMADI S, BORNE P. Approach by localization and multi-objective evolutionary optimization for flexible job-shop scheduling problems. IEEE Trans. on Systems, Man, and Cybernetics—Part C, 2002, 32(1): 1–13.

[48] KACEM I, HAMMADI S, BORNE P. Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic. Mathematics and Computers in Simulation, 2002, 60(3−5): 245–276.

[49] XIA W J, WU Z M. An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problem. Computers & Industrial Engineering, 2005, 48(2): 409–425.

[50] VELDHUIZEN D A V, LAMONT G B. Evolutionary computation and convergence to a Pareto front. Late Breaking Papers at the Genetic Programming Conference, 1998: 221−228.

[51] VELDHUIZEN D A V, LAMONT G B. Multi-objective evolutionary algorithm test suites. Proc. of the ACM symposium on Applied Computing, 1999: 351–357.

[52] DEB K, PRATAP A, AGARWAL S. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. on Evolutionary Computation, 2002, 6(2): 182–197.

[53] ZHANG Q F, LI H. MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. on Evolutionary Computation, 2007, 11(6): 712–731.

[54] NEBRO A J, DURILLO J J, GARCIA-NIETO J, et al. SMPSO: a new PSO-based metaheuristic for multi-objective optimization. Proc. of the IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making, 2009: 66–73.

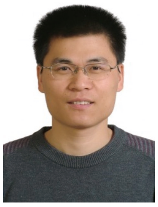
[55] MARTINEZ S Z, COELLO C A C. A multi-objective particle swarm optimizer based on decomposition. Proc. of the 13th Annual Genetic and Evolutionary Computation Conference, 2011: 69–76.

[56] TIAN Y, CHENG R, ZAHNG X Y, et al. PlatEMO: a MATLAB platform for evolutionary multi-objective optimization. IEEE Computational Intelligence Magazine, 2017, 12(4): 73–87.

[57] ZHANG H P, GEN M. Multistage-based genetic algorithm for flexible job-shop scheduling problem. Journal of Complexity International, 2005, 11: 223–232.

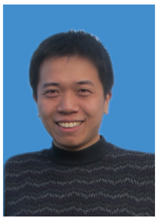
[58] GAO J, SUN L Y, GEN M S. A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems. Computers & Operations Research, 2008, 35(9): 2892–2907.

Biographies

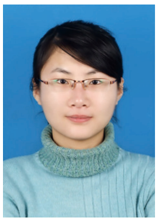


LI Shiyun was born in 1978. He received his B.E. degree in mechanical manufacturing and automation from Beijing Institute of Technology, China, in 2001, and his Ph.D. degree in management information technology in digital design and manufacture from Beijing Institute of Technology, China, in 2006. He is currently working as a lecturer at the College of Mechanical Engineering, Zhejiang University of Technology, China. His research interests include mathematical modeling and optimization as well as their applications in design and manufacturing management.
E-mail: lishiyun@zjut.edu.cn

ZHONG Sheng was born in 1995. He is currently working toward his master degree in the College of Mechanical Engineering, Zhejiang University of Technology. His research interests are intelligent algorithm design and application.
E-mail: zsheng2811@qq.com



PEI Zhi was born in 1982. He received his B.S. and Ph.D. degrees in industrial engineering from Tsinghua University, Beijing, China, in 2005 and 2011, respectively. He was a visiting professor with North Carolina State University, Raleigh, USA, in 2015. He is currently an associate professor with the College of Mechanical Engineering, Zhejiang University of Technology, China. His current research interests include manufacturing system modeling, machine scheduling, nonlinear optimization, and queuing theory.
E-mail: peizhi@zjut.edu.cn



YI Wenchao was born in 1989. She received her B.S. and Ph.D. degrees in industrial engineering from Huazhong University of Science and Technology, China, in 2011 and 2016, respectively. She is currently an associate professor with the College of Mechanical Engineering, Zhejiang University of Technology, China. Her current research interests include evolutionary algorithms and scheduling.
E-mail: yiwenchao@zjut.edu.cn

CHEN Yong was born in 1973. He received his B.S., M.S. and Ph.D. degrees in mechanical engineering from Zhejiang University, in 1995, 1997 and 2000 respectively. Since 2009, he has been a professor with the College of Mechanical Engineering, Zhejiang University of Technology. He is the author of more than 50 academic articles and holds five patents and 40 software copyrights. His research interests include intelligent system planning and intelligent algorithms.
E-mail: cy@zjut.edu.cn

WANG Cheng was born in 1982. He received his B.S. degree in industrial engineering from Zhejiang University of Technology in 2005 and Ph.D. degree in systems engineering from Tianjin University, China in 2010. Since 2016, he has been an associate professor with the College of Mechanical Engineering, Zhejiang University of Technology. His research interests include data driven decision making, complex systems, game theory and mechanism design.
E-mail: cwang@zjut.edu.cn

ZHANG Wenzhu was born in 1990. She received her B.S. degree in management science and engineering from Huazhong University of Science and Technology in 2013, and Ph.D. degree in management science and engineering from Fudan University in 2020. She was a visiting scholar with Washington University in Saint Louis, USA, in 2016. She is currently an assistant professor with the College of Mechanical Engineering, Zhejiang University of Technology, China. Her current research interests include game theory, operations management, and optimization.
E-mail: wzzhang@zjut.edu.cn