



Research article

Single-objective flexible job-shop scheduling problem based on improved dung beetle optimization

Shuangji Yao^{1,2}, Yunfei Guo^{1,2,*}, Botao Yang^{1,2}, You Lv³, Marco Ceccarelli⁴, Xiaoshuang Dai³ and Giuseppe Carbone⁵

¹ Department School of Vehicle and Energy, Yanshan University, Qinhuangdao 066004, China

² Hebei Key Laboratory of specialized Transportation Equipment, Qinhuangdao 066004, China

³ Shenyang Xinbaolu Aviation Technology Co., Ltd, Shenyang, 110000 China

⁴ Department of Industrial Engineering, University of Rome Tor Vergata, Rome 0018511, Italy

⁵ Department of Mechanical, Energy and Management Engineering, University of Calabria, Rende, Italy

* **Correspondence:** Email: 15610139752@163.com; Tel: +86-15610139752.

Academic Editor: Mingfeng Wang

Abstract: Given the characteristics of the flexible job-shop scheduling problem and the practical production of a given enterprise, a flexible job-shop scheduling model was proposed to minimize the maximum completion time. A novel algorithm was proposed to solve the model by integrating the dung beetle optimization algorithm and the simulated annealing algorithm. Algorithmic improvements include the design of a single-layer process encoding scheme with machine selection during decoding to ensure a higher level of the initial population. During population update, the dung beetle optimization algorithm was applied for optimization, followed by simulated annealing operations to enhance the convergence speed and solution quality of the algorithm. Through simulation experiments and comparisons with other algorithms, the effectiveness and superiority of the proposed algorithm were validated. In addition, the feasibility of the algorithm was tested through a real-world factory production case. In conclusion, the improvements made in this paper to the algorithms and scheduling models offer valuable insights into the educational aspects of job-shop scheduling. For instance, the single-layer encoding proposed herein simplifies the coding process, making it more accessible for beginners. Additionally, the accompanying decoding strategy yields relatively higher-quality initial solutions, facilitating subsequent optimization processes by accelerating convergence without compromising solution quality. Students were able to gain a better understanding of real workshop conditions through this project, going beyond the sole goal of minimizing completion time. They began to consider more complex situations in the machining process, such as machine breakdowns, changes in machining schedules, and the load on bottleneck

machines and total machine load. This allowed students to have a holistic view of a complex production workshop. In terms of education, the project improved students' ability to consider practical aspects when solving problems and provided them with a way to solve problems.

Keywords: flexible job-shop scheduling problems, maximum completion time, dung beetle optimizer, simulated annealing, single layer encoding

1. Introduction

The flexible job-shop scheduling problem (FJSP) is an extension of the traditional job-shop scheduling problem [1,2] and has been proven to be an NP-hard problem. Due to limitations of production equipment and costs, the majority of production workshops in China still rely on manual scheduling, resulting in low scheduling efficiency and difficulty in handling complex situations. On the other hand, companies that utilize intelligent scheduling methods [3–5] for production processing demonstrate superior efficiency and performance in highly complex production scenarios. Therefore, intelligent scheduling becomes particularly important, and the quality of algorithms plays a crucial role. In recent years, several metaheuristic algorithms have been proposed, such as particle swarm optimization (PSO) [6], grey wolf optimization (GWO) [7], and whale optimization algorithm (WOA) [8]. Scholars have achieved significant progress in solving shop scheduling problems by employing these emerging metaheuristic algorithms [9–11]. Shen et al. [12] improved the global and local optimization capabilities and diversity of a bird swarm by modifying the structure of the bird swarm, by introducing crossover and mutation operators, and by incorporating a variable neighborhood search algorithm in different bird swarms, thereby enhancing the algorithm's ability to escape local optima and improving convergence. Chen et al. [13] proposed a self-learning genetic algorithm based on reinforcement learning (RL) to intelligently adjust key parameters of the genetic algorithm (GA), leading to outstanding performance in solving flexible job-shop scheduling problems. Wang et al. [14] introduced a population improvement strategy to the genetic algorithm, significantly enhancing its effectiveness in flexible job-shop scheduling problems. Lin et al. [15] proposed a discrete PSO algorithm with adaptive inertia weight and verified its effectiveness and implementation through comparative experiments. N.I. Anuar et al. [16] presented an improved PSO algorithm with enhanced search capability through particle reinitialization, systematic switching of the best solution, and taboo search-based mutation. Zhao et al. [17] proposed an improved ant colony algorithm by enhancing the ant colony's pheromone updating method, thereby improving the algorithm's efficiency and optimization results. Luan et al. [18] proposed a novel metaheuristic algorithm, whale swarm algorithm (WSA), for flexible job-shop scheduling problems and verified its effectiveness. Yuan et al. [19] proposed a new hybrid harmony search algorithm (HHS), which enhances the local search capability of the harmony search (HS) algorithm. Li et al. [20] introduced a hybrid artificial bee colony algorithm based on taboo search, which improves the effectiveness of the algorithm by enhancing the quality of the population. In summary, each algorithm has its own advantages and disadvantages. Therefore, several improved algorithms that are proposed for the flexible job-shop scheduling problem have focused on addressing the shortcomings of the algorithms themselves, aiming to achieve more desirable results. However, there have been few attempts to improve the connection between the algorithm and the problem, specifically the encoding and decoding schemes. This situation may lead to a scenario where algorithm improvements reach their limits but there is no improvement in the flexible job-shop scheduling problem itself. This paper

deals with that by proposing simultaneous improvements to the encoding and decoding approaches as well as the algorithm itself.

In this paper, a flexible job-shop scheduling model with the objective of minimizing the maximum completion time is established, and a novel algorithm that combines the dung beetle optimization algorithm and simulated annealing algorithm is used to solve this model. To address the issues of low convergence accuracy and susceptibility to local optima in flexible job-shop scheduling algorithms, a series of improvements are made to the dung beetle optimization. Considering the high complexity of dual-layer encoding for operations and machines, which increases the difficulty of optimization, this paper adopts a single-layer approach for process encoding and incorporates machine selection during decoding, significantly improving the quality of the initial solutions. During population update, the dung beetle optimization algorithm is applied for optimization, followed by simulated annealing operations on the population to enhance the algorithm's convergence speed and solution quality. Finally, through simulation experiments and comparisons with other algorithms, the effectiveness and superiority of the proposed algorithm are validated. Additionally, the feasibility of the algorithm is demonstrated through a case study of factory production.

2. Problem description

The description of the flexible job-shop scheduling problem (FJSP) [21–23] is as follows: There are n jobs $\{J_1, J_2, \dots, J_n\}$ that need to be processed on m machines $\{M_1, M_2, \dots, M_m\}$. Each job J_i consists of one or more operations, and the sequence of operations is predetermined. O_{ij} represents the j -th operation of job J_i , which can be processed on one of the different machines, denoted as M_{ij} . The processing time for each operation varies depending on the machine used. The scheduling objective of FJSP is to select the most suitable machine for each operation O_{ij} , to determine the optimal processing sequence of operations for each machine, and to determine the start time for each operation, in order to optimize the performance metrics of the entire system. Therefore, the flexible job-shop scheduling problem consists of two sub-problems: determining the machine assignments for each job and determining the processing sequence on each machine.

Some flexible job-shop scheduling problem mathematical models are shown in Table 1.

Table 1. Flexible job-shop scheduling problem mathematical models.

Constraint	Formulation
Objective function	$f = \min(\max(C_i)) \quad 1 \leq i \leq n$ (1)
Machine constraints	$\sum_{k=1}^m X_{ijk} = 1, 1 \leq i \leq n$ (2)
Processing sequence constraints	$S_{ijk} + t_{ijk} \leq S_{i(j+1)k}, 1 \leq i \leq n, 1 \leq k \leq n$ (3)
Processing cannot be interrupted	$S_{ijk} + t_{ijk} = E_{ijk}, 1 \leq i \leq n, 1 \leq k \leq n$ (4)
Processing time constraints	$S_{ijk} \geq 0, 1 \leq i \leq n, 1 \leq k \leq m$ (5)

In Eq (1), C_i represents the completion time of job J_i . Equation (2) indicates that the operation j of job J_i is processed on machine k , that $X_{ijk} = 1$, otherwise $X_{ijk} = 0$. This equation ensures that each operation of a job is processed on exactly one machine. Equation (3) introduces the variables S_{ijk} ,

which represents the start time of the operation j of job J_i on machine k , and t_{ijk} , which represents the processing time of the operation j of job J_i on machine k . $S_{i(j+1)k}$ represents the start time of the operation $j+1$ of job J_i on machine k . Equation (3) enforces the constraint that the processing sequence of different operations of the same job must be respected. In Eq (4), E_{ijk} represents the completion time of the operation j of job J_i on machine k . This equation indicates that once an operation starts, it cannot be interrupted, and it also implies that a job can be processed at most on one machine at any given time. Equation (5) states that all jobs can start processing at time zero, meaning there are no restrictions on the initial start times of the jobs.

3. Basic algorithm description

3.1. Dung beetle optimizer

The dung beetle optimizer (DBO) [24], proposed by Professor Shen Bo's team at Donghua University in November 2022, is a novel swarm intelligence optimization algorithm. It follows their previous work on the sparrow search algorithm (SSA) [25]. The DBO algorithm is inspired by the behaviors of dung beetles, including rolling ball, dancing, breeding, foraging, and stealing. It was specifically designed for global optimization problems and first introduced in the 79th issue of the Journal of Supercomputing in the United States.

Five behavior patterns [24] and corresponding position update formulations are shown in Table 2.

Table 2. Dung beetle behavior patterns and position update formulations.

Behavior patterns	Position update formulations
Rolling ball	$x_i(t+1) = x_i(t) + \alpha \times k \times x_i(t-1) + b \times \Delta x$ $\Delta x = x_i(t) - X^w $ (6)
Dancing	$x_i(t+1) = x_i(t) + \tan(\theta) x_i(t) - x_i(t-1) $ (7)
Breeding	$Lb^* = \max(X^* \times (1 - R), Lb)$ $Ub^* = \max(X^* \times (1 + R), Ub)$ $B_i(t+1) = X^* + b_1 \times (B_i(t) - Lb^*) + b_2 \times (B_i(t) - Ub^*)$ (8)
Foraging	$Lb^b = \max(X^b \times (1 - R), Lb)$ $Ub^b = \max(X^b \times (1 + R), Ub)$ $x_i(t+1) = x_i(t) + C_1 \times (x_i(t) - Lb^b) + C_2 \times (x_i(t) - Ub^b)$ (9)
Stealing	$x_i(t+1) = X^b + S \times g \times x_i(t) - X^* + x_i(t) - X^b $ (10)
	$x_i(t+1) = x_i(t) + C_1 \times (x_i(t) - Lb^b) + C_2 \times (x_i(t) - Ub^b)$ (11)
	$x_i(t+1) = X^b + S \times g \times x_i(t) - X^* + x_i(t) - X^b $ (12)

In Eq (6), the symbol t represents the current iteration number, and $x_i(t)$ represents the position of the dung beetle i in the population at the iteration t . The constant value $k \in (0, 0.2]$ denotes the divergence factor, b is a constant value between 0 and 1, and α is a natural coefficient set to -1 or 1, where 1 represents no bias and -1 represents a deviation from the original direction. X^w represents the worst position in the current population, and Δx is used to simulate the variation of light intensity.

Equation (7) introduces the variable θ , which is an angle between 0 and π that represents the divergence angle. $|x_i(t) - x_i(t-1)|$ represents the difference between the position of the dung beetle i at the iteration t and its position at the iteration $t-1$. It is important to note that if θ equals 0, $\pi/2$, or π ,

the position of the dung beetle is not updated.

In Eq (8), X^* represents the current local best position, and Lb^* and Ub^* represent the lower and upper bounds of the breeding region, respectively. $R=I-t/T_{max}$, where T_{max} represents the maximum number of iterations, and Lb and Ub are the upper and lower bounds of the optimization problem.

Equation (9) introduces $B_i(t)$, which represents the position information of the dung beetle i at the iteration t . b_1 and b_2 are two independent random vectors of size $I \times D$, where D represents the dimension of the optimization problem.

In Eq (10), X^b represents the global best position, while Lb^b and Ub^b represent the lower and upper limits of the optimal foraging region.

Equation (11) involves the variable C_1 , which is a random number following a normal distribution, and C_2 is a random vector belonging to the interval (0, 1).

Equation (12) introduces the variable g , which is a random vector of size $I \times D$ following a normal distribution, and S represents a constant value.

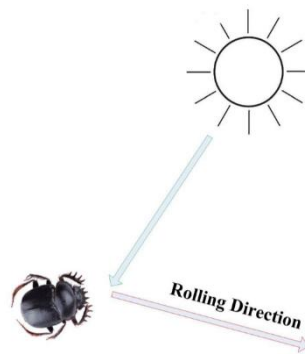


Figure 1. Conceptual model of dung beetle rolling ball trajectory (Xue and Shen 2023).

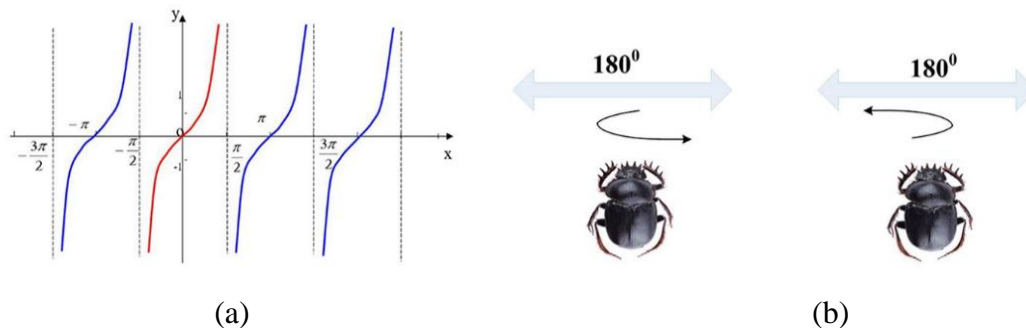


Figure 2. The relationship between dance behavior and tangent function (Xue and Shen 2023). (a) Tangent function curve; (b) a conceptual model of dung beetle dance behavior.

The algorithm partitions the population into five segments and updates their positions based on the five behavioral patterns by using corresponding formulations presented in Table 2. Finally, the global population is updated to determine the optimal position. Figure 1 depicts the direction of dung beetle rolling, guided by sunlight navigation. Figure 2 showcases the dance behaviors that are performed by dung beetles when encountering obstacles, contrasting it with the tangent function. Figure 2(a) is the tangent function curve and 2(b) is a conceptual model of dung beetle dance behavior.

3.2. Simulated annealing

The simulated annealing (SA) algorithm (Vecchi and Kirkpatrick 1983), that was proposed in 1983, is a well-established algorithm that has been widely used in the field of combinatorial optimization. This algorithm was inspired by the annealing process of solid materials. When the temperature of a solid is high, its particles move rapidly, and as the temperature decreases, the particles gradually reach a stable state. Due to the similarities between the SA algorithm and combinatorial optimization problems, it has been increasingly applied in the engineering domain. For solving minimization optimization problems, the main steps of SA are as follows:

Step 1: Set the initial parameters: a sufficiently large temperature T_0 , cooling rate α , and the number of iterations k . Let $T_0 = T$.

Step 2: Initialize the iteration counter $i = 1$. Generate an initial solution X_0 randomly. Set $X_{best} = X_0$ and calculate the objective function value $E(X_{best})$.

Step 3: Generate a neighboring solution X_{new} based on X_{best} . Calculate $E(X_{new})$ and the objective function increment $E = E(X_{new}) - E(X_{best})$.

Step 4: If $\Delta E < 0$, set $X_{best} = X_{new}$ and accept the new solution as the current solution.

Step 5: If $\Delta E > 0$, calculate the acceptance probability $p = e^{-\Delta E/T}$. Generate a random number r between 0 and 1. If $r < p$, accept the new solution by setting $X_{best} = X_{new}$. Otherwise, reject the new solution, and the original solution remains the best. Increment i by 1.

Step 6: Set $T = \alpha \times T$.

Step 7: Check the relationship between i and k . If i is less than k , repeat steps 3 to 5. Otherwise, exit the loop.

4. Improved dung beetle optimization algorithm based on simulated annealing algorithm

Due to the limited performance of the original DBO algorithm in the context of flexible job-shop scheduling models, particularly in terms of low convergence accuracy and susceptibility to local optima, the simulated annealing (SA) algorithm is considered to be integrated with the DBO algorithm. SA can avoid local optimization to a large extent in the early stage and has high convergence accuracy in the later stage. An enhanced algorithm called dung beetle optimization with simulated annealing (DBO + SA) is proposed for the flexible job-shop scheduling problem. In the SA component, perturbed solutions are generated by applying different crossover methods to the initial solutions. Furthermore, the algorithm incorporates simulated annealing operations into each iteration, allowing for a higher probability of accepting suboptimal solutions during the early stages of iteration. This approach helps prevent the algorithm from getting stuck in local optima. As the number of iterations increases, the probability of accepting suboptimal solutions gradually decreases, leading to improved convergence accuracy in the final solution.

4.1. Encoding and decoding

In current research, most scholars employ a two-stage encoding approach for flexible job-shop scheduling problems, involving work order sequencing and machine assignment selection [26–28]. Two-stage encoding requires longer lengths, which gives twice the total number of work orders when compared to single-layer encoding. This exponentially increases the number of possible

permutations, posing challenges for optimization. Additionally, the initial solutions that are generated using two-stage encoding are based on completely random machine assignments, which results in poor initial solutions and slower convergence rates. To address these issues, this paper proposes a single-layer encoding approach that only encodes work orders without considering machine assignments. A unique decoding scheme is developed to correspond to the single-layer encoding. Integer encoding [29–31] is employed during the encoding process, while the decoded solutions are obtained by rounding real numbers to integers and then performing the decoding process.

Different from the majority of two-stage double-layer encoding approaches, this proposal introduces a change in the encoding scheme by using a single-layer encoding. Only the work orders of the jobs are encoded, and the length of the encoding is the total number of work orders across all jobs. Random initialization is employed to enhance the global search capability of the initial population, and machine assignments are determined during the decoding process. By adopting the single-layer encoding, the encoding workload can be significantly reduced, making the encoding process more concise. Besides, single-layer encoding has better pedagogical results for beginners. The specific steps of the encoding process are formulated as follows:

Step 1: L represents the total number of work orders, L_i represents the total number of work orders for job i , $1 \leq i \leq \text{total number of jobs}$, and i is an integer. The numbers 1 to L_1 represent the number of work order for job 1, the numbers L_1+1 to L_2 represent the number of work order for job 2, and so on, until all work orders for all jobs have corresponding numbers.

Step 2: Generate a zero vector, x , of length L . Fill the numbers 1 through L randomly into L positions in the vector. At this stage, x represents a complete individual of a dung beetle.

Step 3: Repeat Step 2 multiple times until the predetermined population size is achieved.

Since only the work orders are encoded without encoding the machines, decoding requires selecting the machines corresponding to the work orders before proceeding with subsequent operations. Machine selection involves considering various factors, including the completion time of the previous work order, whether there are other work orders to be processed on the selected machine, the earliest available start time if the machine is chosen with other work orders, and whether it affects the subsequent work orders. Additionally, attention should be given to situations where multiple machines are available, such as how to make the selection.

To address the aforementioned issue, the solution proposed in this paper involves recording the involvement of each machine in the processing. Each machine has its dedicated timetable, which records the time slots when it is engaged in processing. Similarly, each work order has a progress table that keeps track of the current operation number for that work order. Whenever a work order completes an operation, the corresponding entry in the progress table is incremented. Additionally, a decoding table is created, consisting of five columns: work order number, machine number, operation number, start time, and end time. The number of rows in the decoding table is equal to the total number of operations. Following these steps, when decoding an encoded element, the work order number and operation number can be obtained. By referencing the decoding table entries corresponding to that work order, the completion time of the previous operation can be determined. Combining this with the timetables of all available machines, the earliest start time for the current operation can be identified. The machines are then selected in chronological order until a suitable machine that can accommodate the operation is found. The accompanying decoding strategy yields

relatively higher-quality initial solutions, facilitating subsequent optimization processes by accelerating convergence without compromising solution quality. The specific decoding rules are as follows:

Step 1: Identify the work order number and operation number corresponding to the encoded element.

Step 2: Utilizing the machine timetable, work-order operation progress table, and decoding table, retrieve the available machine codes for the current operation and their respective scheduling schemes based on the machines' availability.

Step 3: Based on the scheduled schemes for each machine, determine the available time slots for the current operation on each machine and arrange them in ascending order.

Step 4: Iterate through the available time slots and check if the current operation can fit within each interval. If a suitable slot is found, insert this operation. Then, proceed to Step 5.

Step 5: Update the machine timetable, work-order operation progress table, and decoding table accordingly. Then, return to Step 1 and repeat the process until all the encoding elements have been decoded. The process is shown in Figure 3.

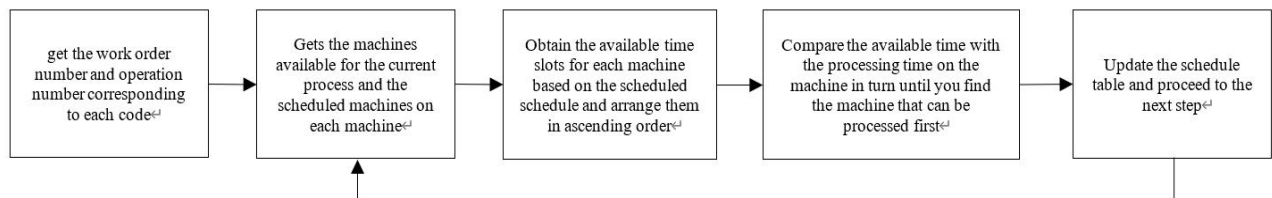


Figure 3. Decoding flowchart.

4.2. Algorithm and implementation

The specific steps of the DBO + SA algorithm are summarized in the flowchart in Figure 3.

Step 1: Set parameters and initialize the population: Set $t = 1$, define population size, maximum iteration count T_{max} , proportion of each behavior in the total population, deflection coefficient k , natural coefficient α_1 , constant b and S , initial temperature T_0 , and cooling coefficient α_2 . Create initial solutions and the initial population based on the encoding method described in Section 3.1.1.

Step 2: Calculate the objective function value: Decode the solutions using the decoding method described in Section 3.1.2 and retrieve the maximum value from the fifth column of the decoding table, which represents the objective function value.

Step 3: Position update: Update the positions of the individuals in the population according to their respective behavior patterns and the corresponding formulations. Note that the result of the certificate code calculated by the formula may be decimal. Before calculating the objective function value, the decimal values should be rounded to the nearest integers and then decoded.

Step 4: Simulated annealing operation on the population: Each individual in the population has a one-third probability of generating a perturbed solution using one of the three methods that are illustrated in Figures 4(a)–4(c). Assuming the original encoding is 1–5, random positions X1 and X2 are selected for subsequent operations. The inversion operation reverses the elements between X1 and X2, the exchange operation swaps the elements at positions X1 and X2, and the insertion

operation inserts the element at X2 after the element at position X1. Each individual in the population generates a perturbed solution through these operations. If the perturbed solution is better than the original solution, it is accepted directly. If the perturbed solution is worse, it is accepted with a probability of $p = e^{-\Delta E/T}$, where ΔE is the difference in objective function value between the perturbed solution and the original solution. At the beginning of the iteration, when T has a large value, p is close to 1, resulting in a higher probability of accepting worse solutions. As the iteration progresses and T decreases, p approaches 0, reducing the probability of accepting worse solutions.

Step 5: Update the population, replace the global best individual and its fitness value, and set $t = t + 1$.

Step 6: If $t > T$ or the stopping criteria are met, end the algorithm and output the current best solution. Otherwise, return to Step 3.

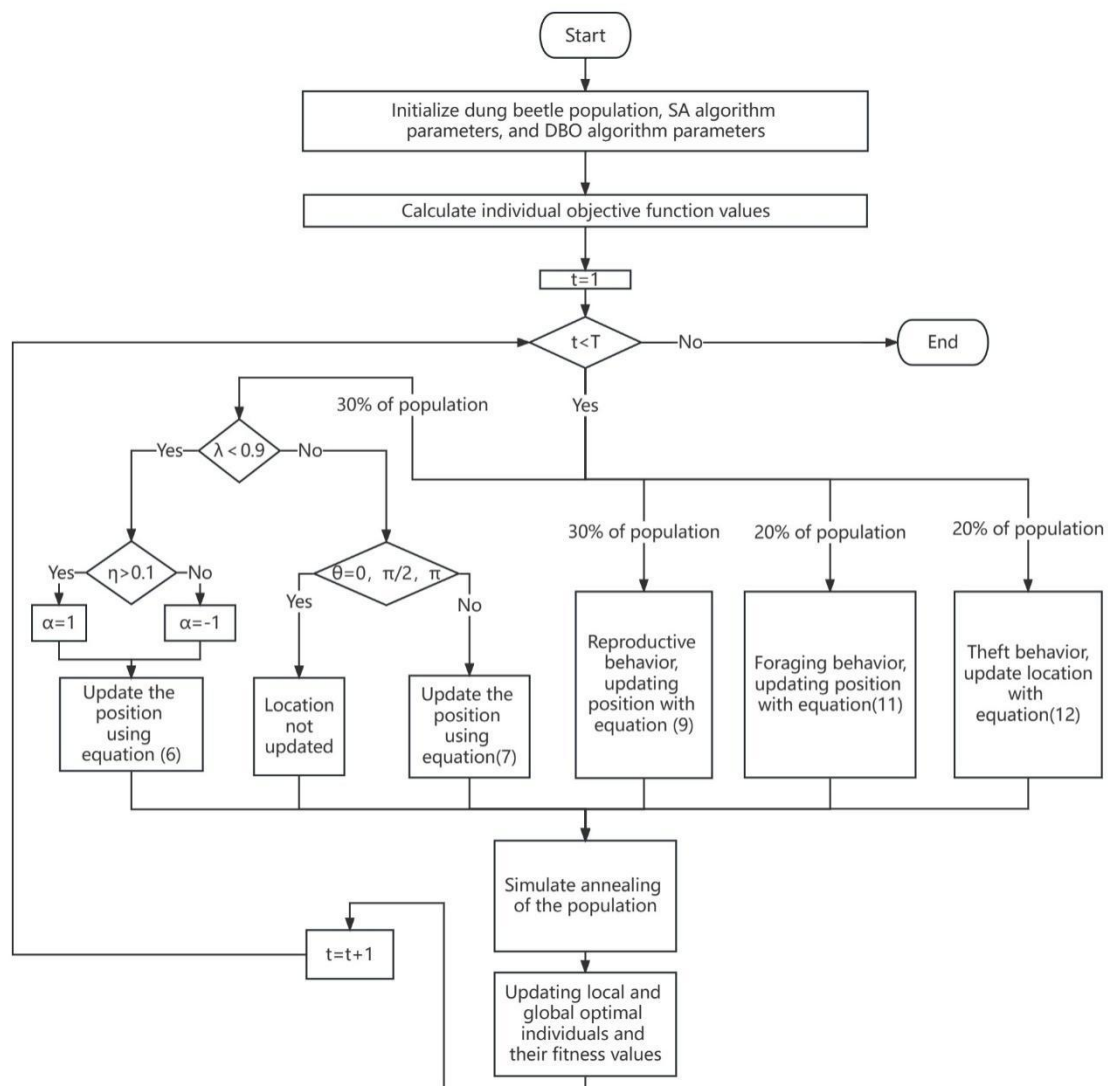


Figure 4. A flowchart of the proposed DBO+SA algorithm.

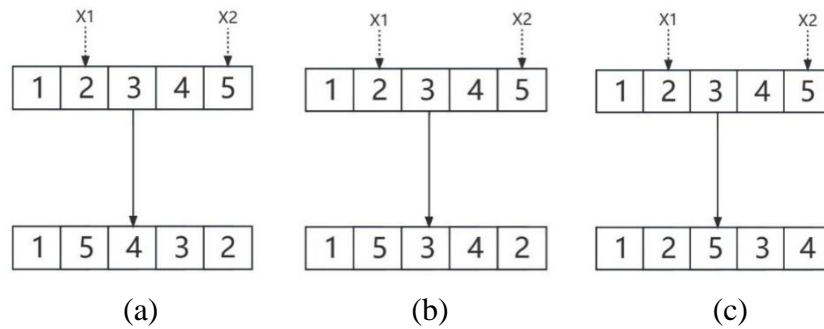


Figure 5. Generation method of perturbed solution. (a) Reverse, (b) exchange, and (c) insert.

After multiple tests, it was found that the proportion of each behavior can determine the emphasis of the algorithm. Rolling behavior helps increase the algorithm's global search capability, while dancing behavior accelerates the algorithm's convergence speed, enhancing its efficiency in the local search space. Breeding, foraging, and stealing behaviors contribute to refining and improving the already found solutions, thus enhancing the algorithm's convergence precision. After multiple comparisons, the proportions of rolling, dancing, breeding, foraging, and stealing behavioral patterns were determined to be 27%, 3%, 30%, 20%, and 20%, respectively. Additionally, x was set to 0.9, y to 0.1, initial temperature to 3000, and cooling coefficient to 0.97.

5. Comparative analysis of experiments

5.1. Comparison of standard examples

To validate the effectiveness of the integrated algorithm proposed in this study, we selected the widely-used Brandimarte benchmark instance (MK01-MK10) [32] from the FJSP standard scheduling problem. For FJSP-related algorithms, there are numerous test cases available, with the most commonly used ones categorized into three types: Brandimarte cases, Kacem cases, and custom-built cases. Brandimarte cases are semi-flexible, Kacem cases are fully flexible, and custom-built cases are not convenient for comparing multiple algorithms. As semi-flexible scenarios better reflect the current real-world workshop conditions, this paper opts for Brandimarte cases as the testbed for algorithm evaluation.

The DBO+SA algorithm was implemented using Matlab2017b and executed on a system running Windows 10, with an Intel(R) Core(TM) i5-6300HQ CPU @ 2.30GHz and 16 GB of memory. Set the population size to 100 and the maximum iteration count to 100.

To validate the effectiveness of the improvements made by incorporating the simulated annealing approach in the DBO+SA algorithm, a comparative experiment was conducted between the basic DBO and DBO+SA. In order to mitigate the influence of randomness, the algorithm was executed 10 times. The term "Best" represents the best result obtained among the 10 simulation experiments, while "Avg" denotes the average optimization outcome over the 10 simulations. The experimental results are summarized in Table 3, where the problem instances are defined by their dimensions, with "m" representing the number of jobs and "n" representing the number of machines. The results indicate the total processing time required to complete all jobs. The best results are highlighted in bold red in Table 3.

Table 3. Comparison between algorithm DBO and DBO+SA.

Example	Scale	DBO		DBO+SA		Increased amount		Increased percentage	
		Best	Avg	Best	Avg	Best	Avg	Best	Avg
MK01	10*6	42	43.2	40	41.3	2	1.9	4.8%	4.4%
MK02	10*6	35	36.1	31	31.6	4	4.5	11.4%	12.5%
MK03	15*8	204	206.8	204	205.1	0	1.7	0%	0.8%
MK04	15*8	70	71.3	60	63.2	10	8.1	14.3%	11.4%
MK05	15*4	185	187.1	175	177.6	10	9.5	5.4%	5.1%
MK06	10*10	86	87.8	72	78.6	9	9.2	16.3%	10.5%
MK07	20*5	175	176.0	152	154.3	23	21.7	13.1%	12.3%
MK08	20*10	525	528.4	523	526.4	2	2	0.4%	0.4%
MK09	20*10	330	331.2	315	316.1	15	15.1	4.5%	4.6%
MK10	20*15	264	266.7	231	234.8	33	31.9	12.5%	12.0%

Taking the MK03 instance from the aforementioned cases as an example, in Figure 6(a), it can be observed that the optimal value converges around the 20th generation and achieves a favorable convergence value. Moreover, Figure 6(b) illustrates that the average objective function value of the initial population is around 223. This demonstrates the effectiveness of the proposed encoding scheme in significantly improving the quality of the initial solutions. This further validates the superior convergence speed of the DBO+SA algorithm.

Figure 7 presents the Gantt chart generated based on the scheduling results for the MK03 instance. The chart uses the start time of the first operation as the reference zero time point, with the horizontal axis representing the total processing time in hours and the vertical axis denoting the machine number. Each rectangular block represents an operation, and the text within the block indicates the job number, operation number, and processing duration. For instance, "P(5,6)" denotes that operation 6 of job 5. The vertical position of the block indicates the machine number on which the operation is scheduled, while the left and right coordinates of the block represent the start and end times of the operation, respectively.

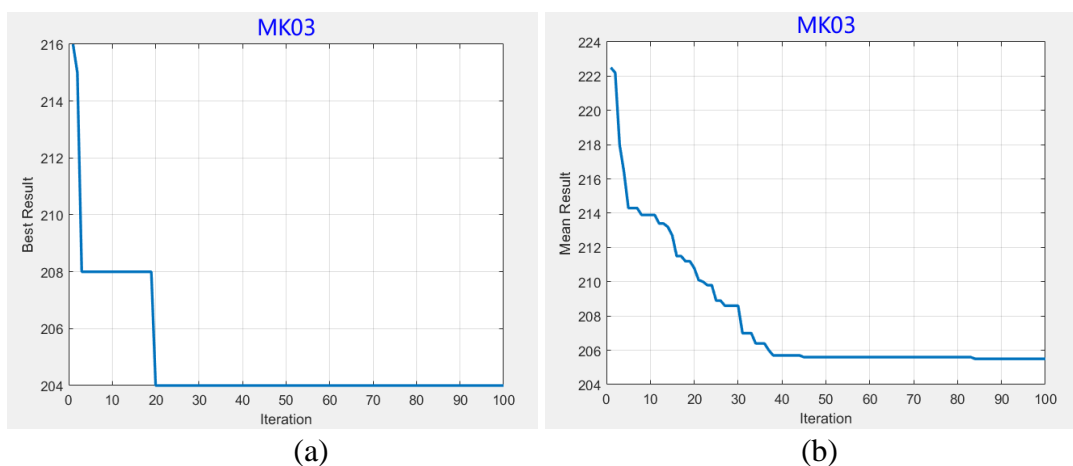


Figure 6. Convergence curve of the example MK03 results under DBO + SA algorithm. (a) Best value and (b) average value.

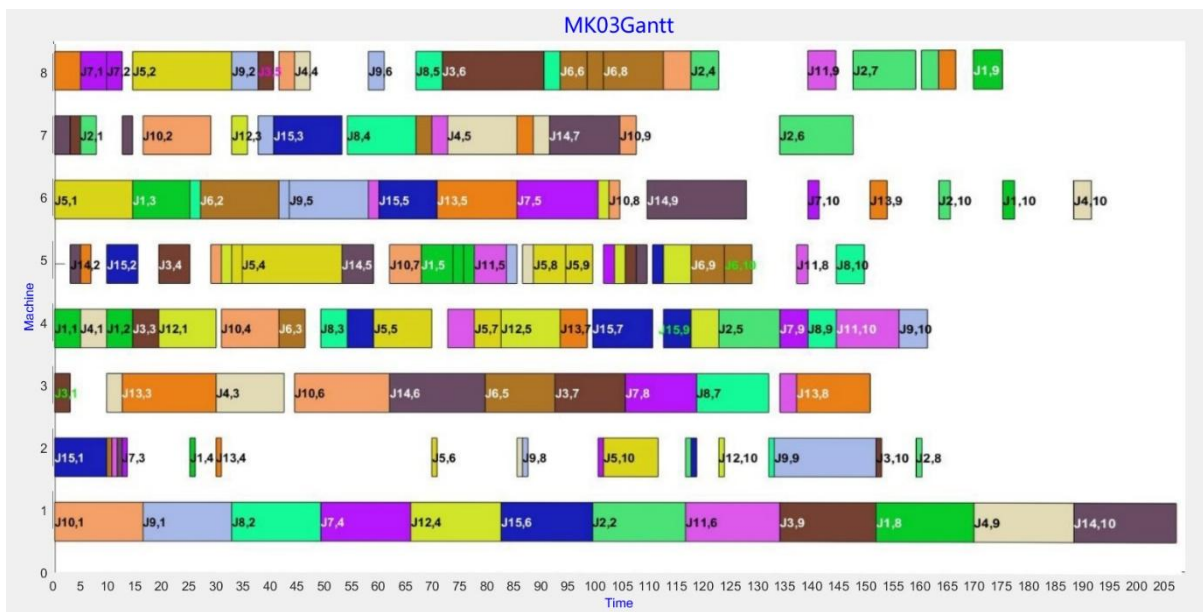


Figure 7. Gantt chart of MK03 scheduling results.

Simultaneously, the DBO+SA algorithm was compared with other metaheuristic algorithms, including the basic genetic algorithm, the quantum whale optimization algorithm [33], the hybrid grey wolf optimization algorithm [34], and the hybrid genetic optimization algorithm proposed in [35]. The optimal results are highlighted in bold red in Table 4.

From Table 3, it can be observed that the fused DBO+SA algorithm outperforms the initial DBO algorithm in terms of both the optimal and average values. This confirms the effectiveness of the DBO+SA algorithm. In Table 4, when comparing the DBO+SA algorithm with other algorithms, it is found that the DBO+SA algorithm outperforms the algorithms mentioned in the literature in 6 out of 10 standard cases. This further demonstrates the superiority of the DBO+SA algorithm.

Table 4. Comparison of DBO+SA algorithm with other algorithms.

Example	Literature [33]	Literature [34]	Literature [35]	DBO+SA
MK01	41	40	40	40
MK02	29	30	27	29
MK03	204	204	204	204
MK04	67	60	60	60
MK05	171	175	178	175
MK06	72	73	78	68
MK07	149	149	147	152
MK08	523	523	523	523
MK09	317	330	341	315
MK10	241	252	279	238

Finally, the convergence curves of each algorithm on instances MK01, MK02, MK06, and MK10 are depicted in Figure 8. Additionally, the Gantt charts of the scheduling solutions for DBO+SA on instances MK01, MK02, MK06, and MK10 are presented in Figures 9–12.

From Figure 8, it can be observed that the algorithm proposed in this paper exhibits a superior initial population than other algorithms. Consequently, it significantly increases the convergence

speed without compromising convergence precision, thanks to the encoding and decoding strategies proposed in this paper. This finding provides a valuable insight for researchers, suggesting the potential for improvement through enhancements to encoding and decoding methods.

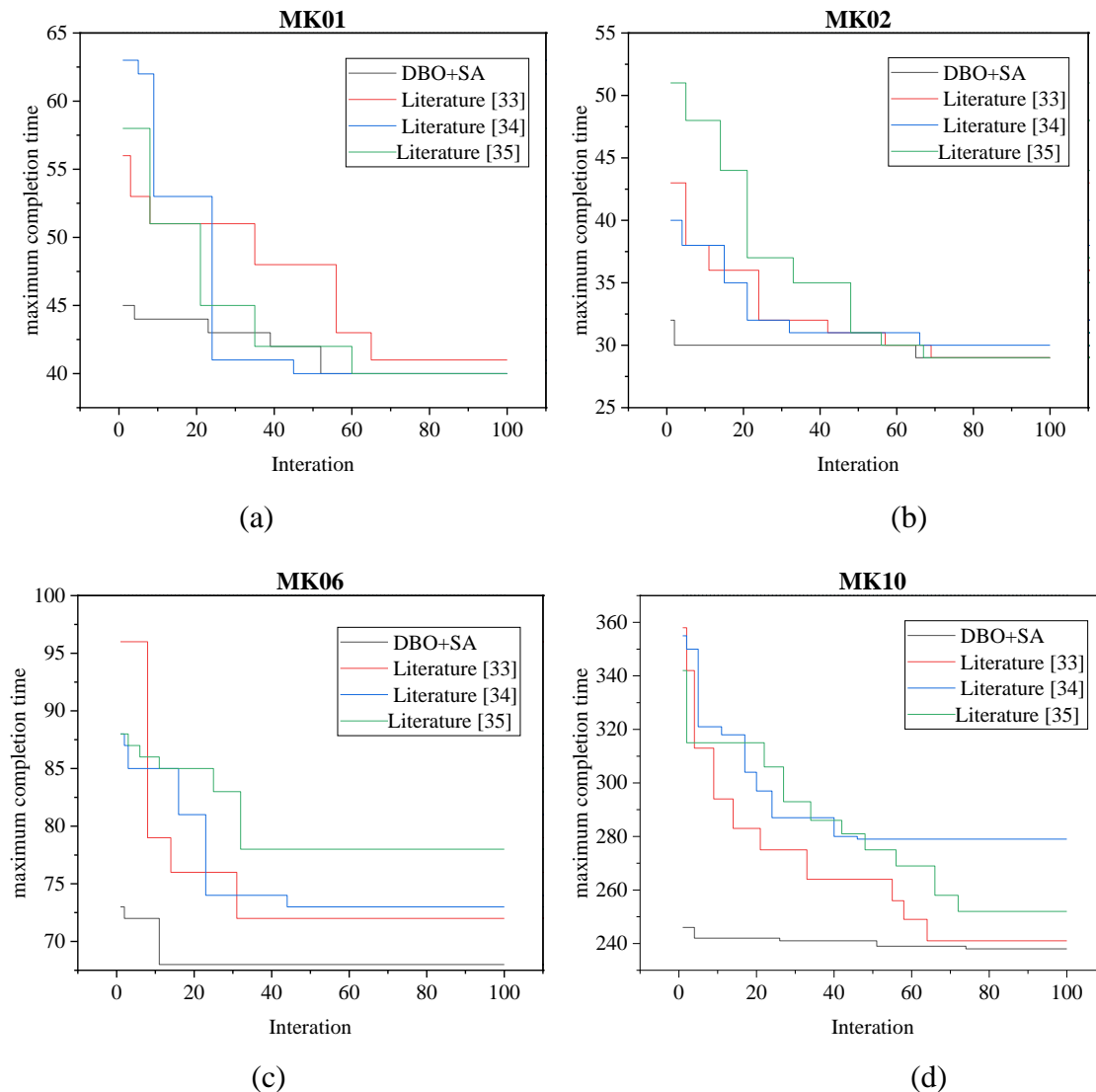


Figure 8. Convergence curves of instances MK01, MK02, MK06, and MK10 across different algorithms. (a) MK01, (b) MK02, (c) MK06, and (d) MK10.

As can be seen from Figures 9–12, although the overall scheduling scheme has achieved the goal of minimizing the maximum completion time, the machine load varies greatly. Some machines have almost no downtime events, while others have a lot of idle time, which will cause machines with high loads to malfunction more frequently and increase costs. Therefore, it is necessary for relevant scholars to work on multi-objective FJSP to make FJSP more relevant to the real situation in the workshop.

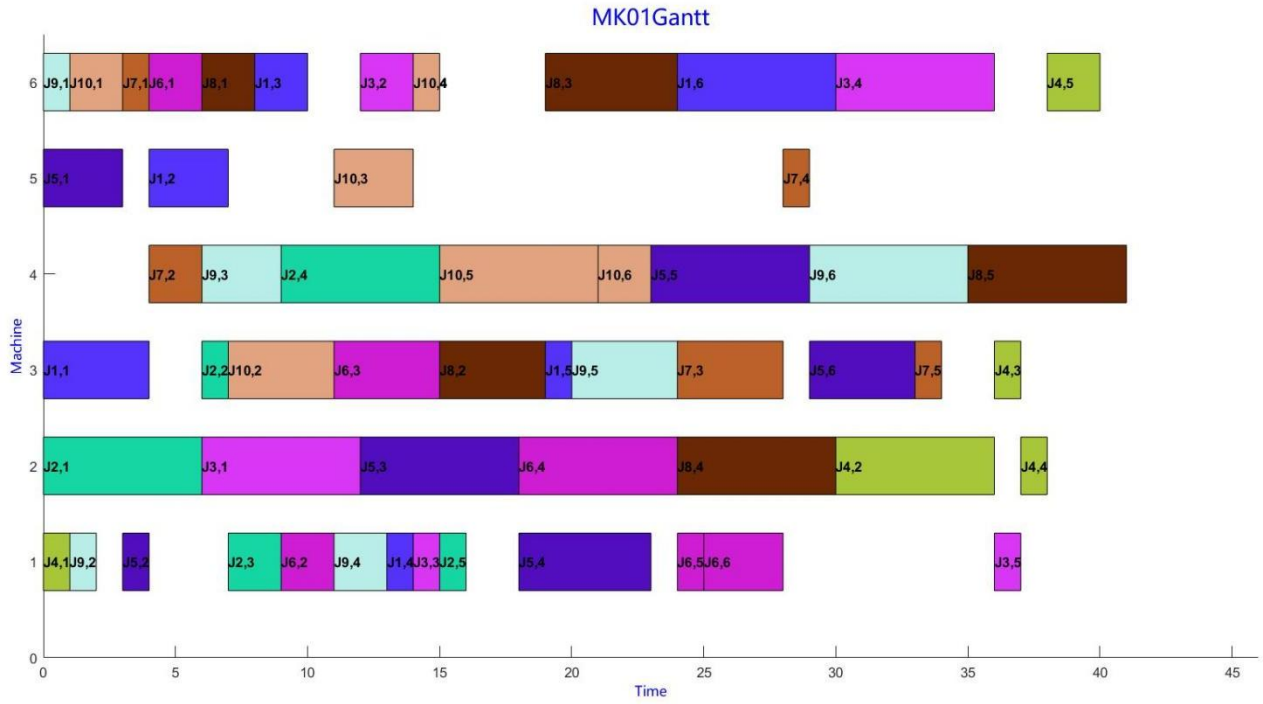


Figure 9. Gantt chart of MK01 scheduling results.

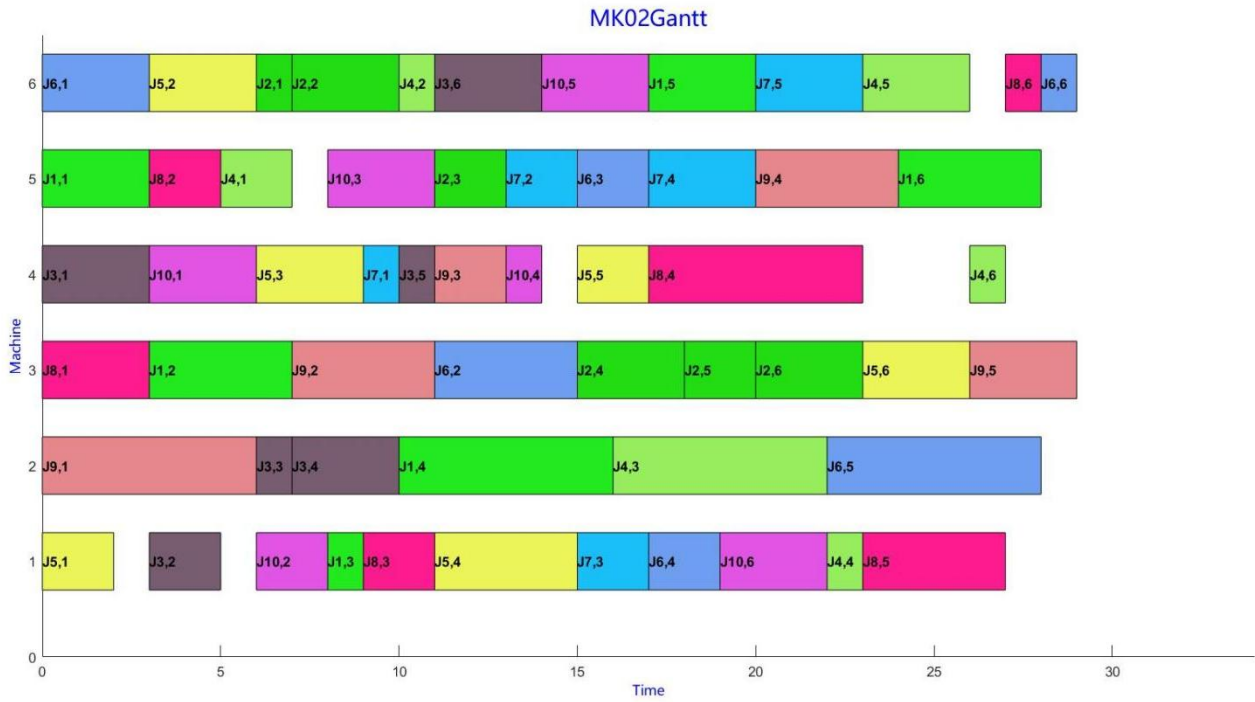


Figure 10. Gantt chart of MK02 scheduling results.

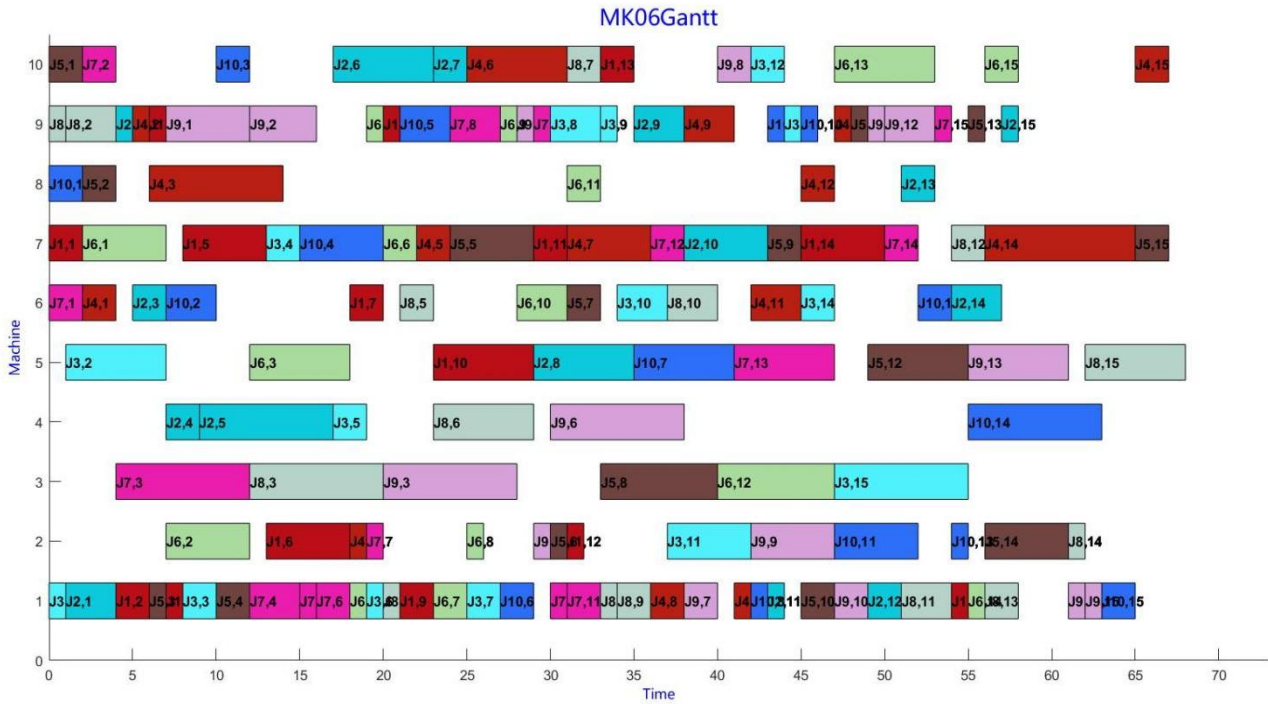


Figure 11. Gantt chart of MK06 scheduling results.

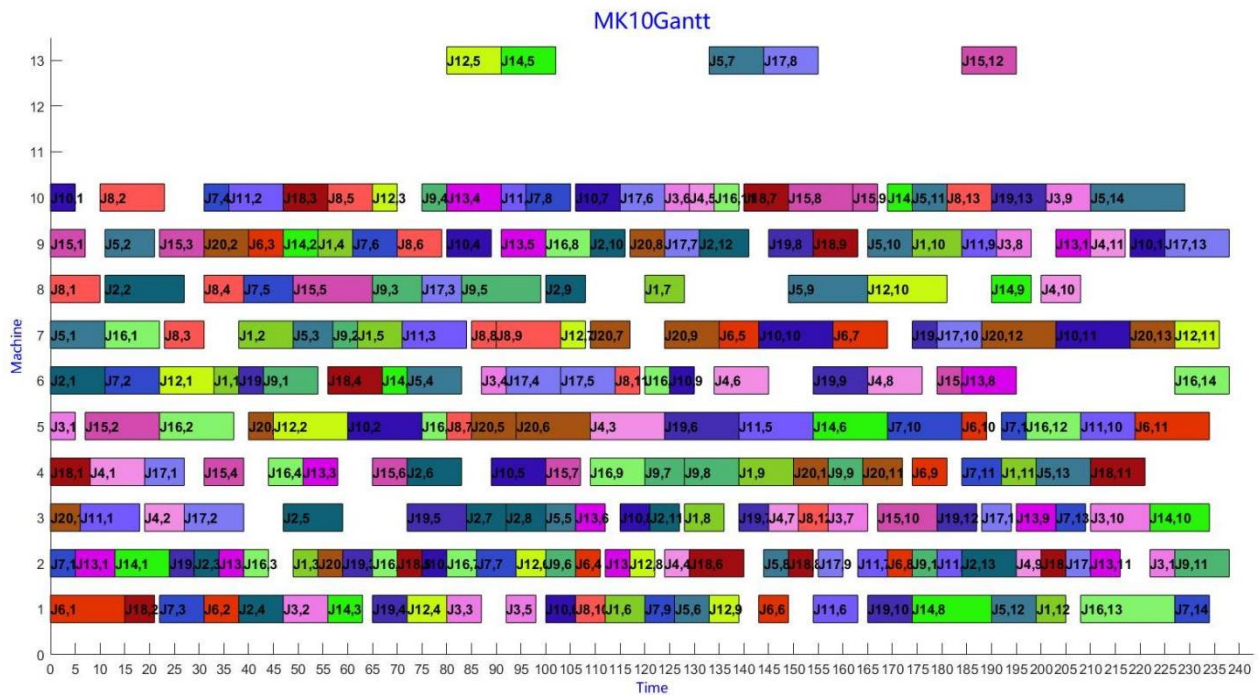


Figure 12. Gantt chart of MK10 scheduling results.

5.2. Simulation of a workshop study case

Although the algorithm presented in this paper performs well on standard test cases, validating its feasibility requires real-world workshop scheduling cases. To this end, we collected instance

information from company X's precision workshop, which is a partially flexible job shop. The workshop is represented by a scale of 28×24 , indicating 28 jobs, 24 machines, and a total of 130 operations. Due to space limitations, the instance information is divided into two parts and presented in the appendix. It has also been uploaded to the cloud for reference by readers [36]. Table 5 displays the job numbers, operation numbers, machine numbers available for use, and the processing times for the first 11 machines for each corresponding operation. Table 6 shows the processing times for the remaining 13 machines for each corresponding operation.

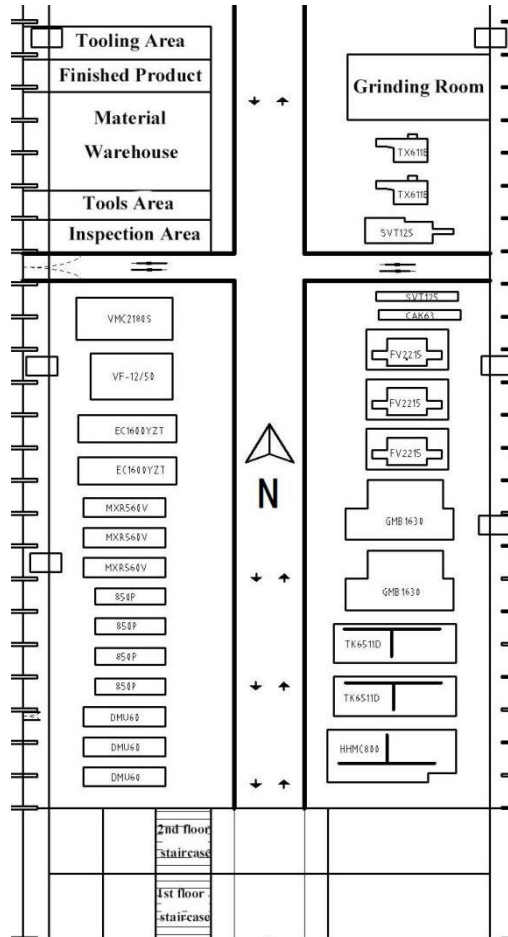


Figure 13. Workshop layout of company X.

Figure 13 illustrates the layout of company X's precision workshop, which is divided into three main sections. The upper end in Figure 13 represents the office area, while the middle section corresponds to the processing area, housing a total of 24 machining tools. The letters and numbers in Figure 13 denote the machine models, such as "DMU60" referring to a five-axis CNC machine tool from the German company DMG Mori. The left lower corner represents Machine 1, with subsequent machines numbered sequentially from bottom to top. The above side is dedicated to inspection, storage, and polishing of products. The green cross-shaped area in the middle represents the pedestrian logistics pathway within the workshop. Additionally, the workshop is equipped with three overhead cranes with a lifting capacity of 5 tons each, responsible for handling heavier workpieces. Figures 14(a) and 14(b) present the actual view of the workshop's interior.



Figure 14. Realistic view of X company's workshop interior.

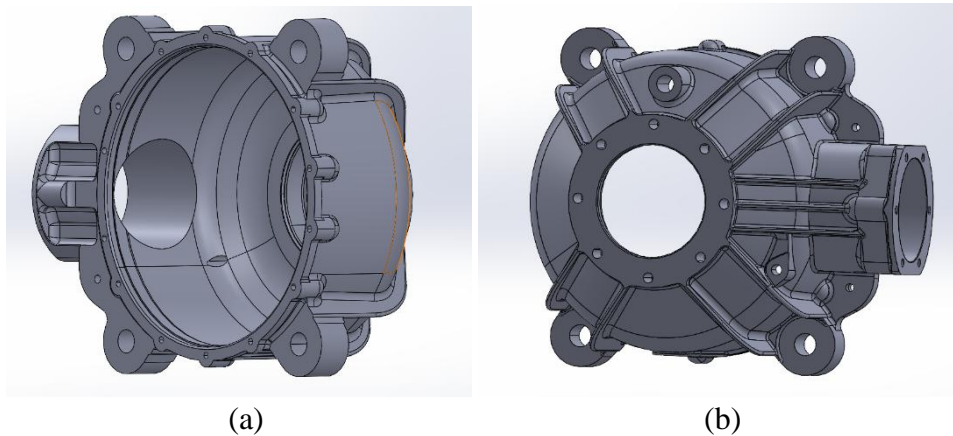


Figure 15. Three-dimensional model of 222 products. (a) Bottom of the 222 products; (b) top of the 222 products.



Figure 16. X company produces 222 products. (a) Machine tool 14 processing workpiece 8; (b) machine tool 3 processing workpiece 4.

The 222 products, developed and prototyped recently by company X, are processed using a CNC 5-axis lathe. Each product typically undergoes more than 20 machining operations, 4–6 positioning steps, and 3 inspection procedures, resulting in a complex and intricate production process. Figure 15(a) and 15(b) show two three-dimensional models of a 222 product. Figure 16(a) and 16(b) depict a real scene where Machine 14 and Machine 3 are processing workpiece 8 and workpiece 4, respectively. Machine 3 is a 5-axis machining center DMU60. It is responsible for processing the

side surface of workpiece 4. Machine 14 is a vertical lathe. It is responsible for processing the side surface of workpiece 8. Currently, the workshop adopts a manual scheduling approach, where human intervention is required to determine the next machining tool after completing each operation. This method is inefficient and labor-intensive. Taking the collected instance of 222 products mentioned in this paper as an example, excluding scheduling and handling time, the processing time for this instance is approximately 300 labor hours.

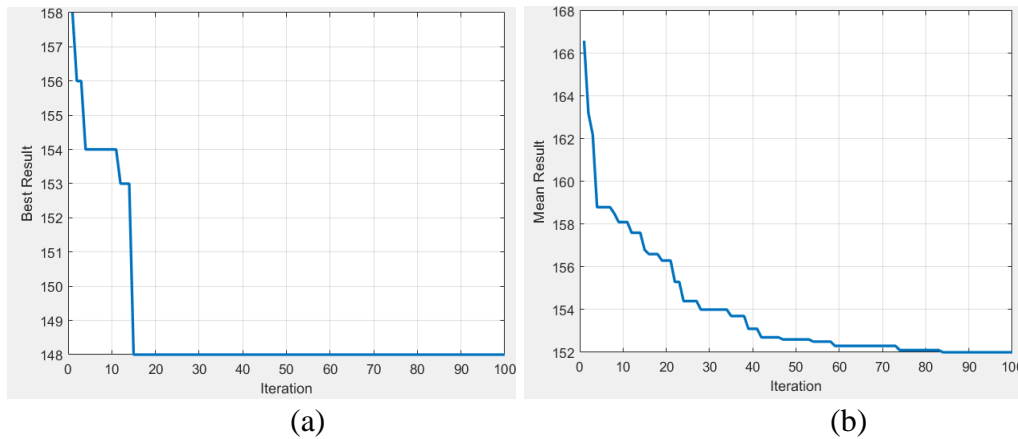


Figure 17. Convergence curve. (a) Best value; (b) average value.

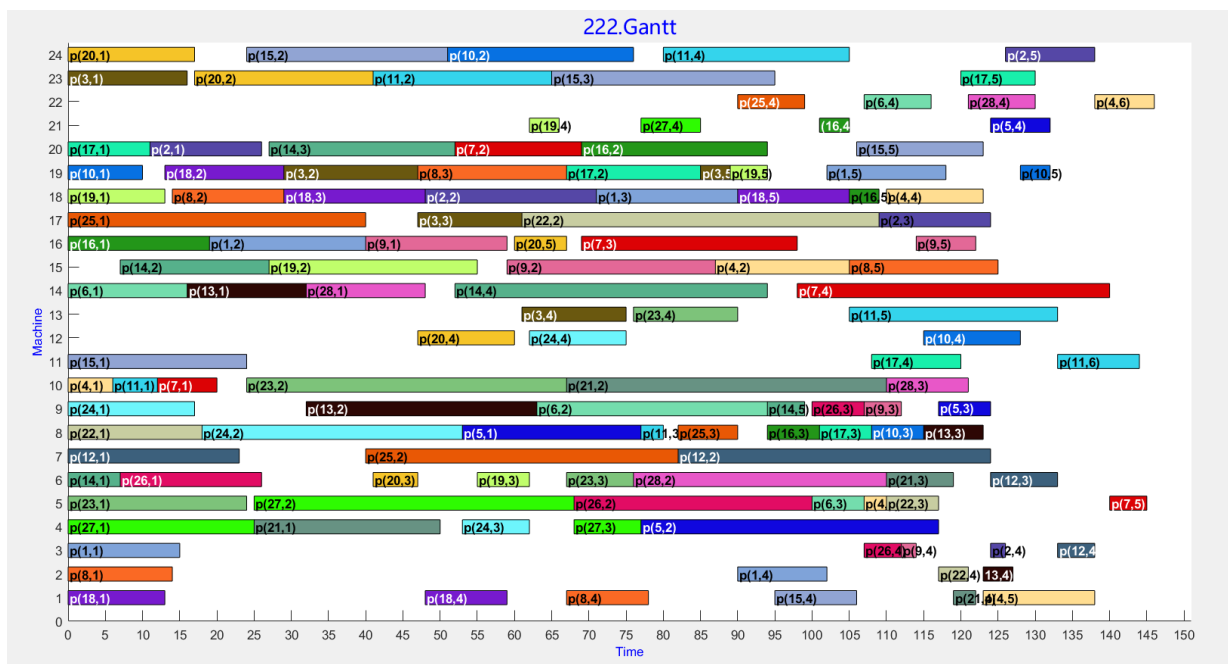


Figure 18. Scheduling plan Gantt chart of 222 product.

By applying the proposed single-layer encoding and DBO+SA algorithm optimization method presented in this paper, the flexible scheduling problem in the workshop can significantly improve production efficiency. With a population size of 200 and a maximum iteration count of 100 generations, the final result yields a maximum processing time of 146 labor hours for the given production tasks. Compared to the manual scheduling approach with a processing time of 300 labor

hours, the production efficiency has increased by over 50%. Figure 17(a) illustrates the convergence curve of the global optimal solution's objective, and Figure 17(b) illustrates the convergence curve of the population's average objective function. In Figure 17(a), it can be observed that the optimal value converges after 15 iterations, demonstrating a relatively rapid convergence rate. In Figure 17(b), it can be seen that the average value also enters a convergence phase after around 80 generations. Figure 18 displays the Gantt chart of the scheduling plan for the 222 products. From the chart, it is evident that the last operation to be completed is the 6th operation of Job 4, processed by Machine 22. Therefore, the maximum completion time for the 222 jobs is determined by the completion time of the 6th operation of Job 4, which is 146 labor hours. The successful scheduling of this instance demonstrates the feasibility and superiority of the DBO+SA algorithm in flexible job-shop scheduling.

5.3. Discussion

The students training details include:

- (1) Algorithm improvement: using CEC2021 test functions, the performance of a variety of algorithms were all tested, and the possibility of fusion of different algorithms was analyzed. Finally, DBO, which has greater development potential, and SA, which is suitable for use in fusing various algorithms, were selected from the multiple algorithms.
- (2) Coding and decoding: We tried the two-layer coding and decoding strategy based on machine allocation and process ordering and the unified coding strategy and found that they have their own advantages and disadvantages. Finally, we tried to combine the two coding strategies to form a single-layer coding and decoding strategy; after comparison, we found that the effect is good, and we finally determined the coding and decoding strategy in this paper.
- (3) Simulation experiments: Students were sent to Company X for simulation data collection. They recorded the basic details of the parts in an order and followed each part through the entire workshop scheduling production process. Collected data was then organized for simulation experiments.

The students were able to gain a better understanding of real workshop conditions through this project, going beyond the sole goal of minimizing completion time. They began to consider more complex situations in the machining process, such as machine breakdowns, changes in machining schedules, and the load of the bottleneck machine and the total machine load, which allowed them to have a holistic view of a complex production shop. This further enhanced the students' view of a complex production plant. In terms of education, the project improved students' ability to consider practical aspects when solving problems and provided them with a way to solve problems.

6. Conclusions

In this paper, the flexible job-shop scheduling problem (FJSP) was addressed with the objective of minimizing the maximum completion time in a manufacturing process. To tackle the issues of complex encoding and susceptibility to local optima in existing algorithms, an algorithm was proposed as DBO+SA by integrating the dung beetle optimization (DBO) and simulated annealing (SA) algorithms. The approach was achieved as follows:

First, a one-dimensional encoding scheme was employed to discretely initialize the population for the scheduling of operations in the flexible job-shop scheduling problem. This allows for the

application of the algorithm to the flexible job-shop scheduling problem. Second, an adaptive decoding method was applied to decode the encoding. This involved selecting the processing machine for each operation and calculating the objective function value. Next, the dung beetle optimization and simulated annealing algorithms were fused together, aiming to enhance both global and local search capabilities while ensuring effective exploration of the search space. Test results were analyzed with comparisons using the improved algorithms from other literature on the MK series instances proposed by Brandimarte. The results demonstrated the effectiveness of the algorithm. Furthermore, real-world production instances were collected from Company X's precision workshop. By applying the scheduling algorithm proposed in this paper, the production efficiency improved 50% when compared with the existing manual scheduling method employed in the workshop. It should be noted that the algorithm and model proposed in this paper are only applicable to the single-objective FJSP with the objective function of minimizing the maximum completion time. While it can accommodate various scales of production scenarios, the considerations and objective function are relatively singular, and thus may not address complex production scenarios comprehensively.

This further validates the feasibility of the proposed encoding method and algorithm. Future work could involve the application of this algorithm to more complex workshop scheduling problems to expand its range of applications. In conclusion, the improvements made to the algorithms and scheduling models in this paper offer valuable insights into educational aspects of job-shop scheduling. For instance, the single-layer encoding proposed herein simplifies the coding process, making it more accessible for beginners. Additionally, the accompanying decoding strategy yields relatively higher-quality initial solutions, facilitating subsequent optimization processes by accelerating convergence without compromising solution quality.

The students were able to gain a better understanding of real workshop conditions through the exercise of this project and were not limited to the pursuit of completion time as a goal. Instead, they began to consider more complex situations in the machining process, such as machine breakdowns, changes in machining schedules, and the load of the bottleneck machine and the total machine load, providing them with a holistic view of a complex production shop. The students' view of the big picture in a complex production plant was further enhanced. In terms of education, the project improves students' ability to consider practical aspects when solving problems and provides them with a way to solve them.

Author contributions

Shuangji Yao: Conceptualization, Methodology creation, Investigation, Resources; Yunfei Guo: Writing - Original Draft, Data curation, Result Analysis; Botao Yang: Methodology, Writing – review & editing; You Lv: Investigation, Result Analysis; Marco Ceccarelli: Conceptualization; Xiaoshuang Dai and Giuseppe Carbone: Investigation, Supervision. All authors have read and approved the final version of the manuscript for publication.

Acknowledgments

Project supported by the Foundation for Innovative Research Groups of the Natural Science Foundation of Hebei Province (Grant No. E2020203174) and Key Project of National Natural Science Foundation of China, Project (Grant No. U20A20332).

Conflict of interest

The authors declare that they have no conflicts of interest.

Ethics declaration

The authors declare that the ethics committee approval was waived for the study.

References

1. Liu, X.X., Liu, C.B. and Tao, Z., Study on Scheduling Optimization for Flexible Job Shop. *Applied Mechanics and Materials*, 2010, 26: 821–825. <https://doi.org/10.4028/www.scientific.net/AMM.26-28.821>
2. Józefowska, J. and Zimniak, A., Optimization tool for short-term production planning and scheduling. *International Journal of Production Economics*, 2008, 112(1): 109–120. <https://doi.org/10.1016/j.ijpe.2006.08.026>
3. Qu, M., Zuo, Y., Xiang, F. and Tao, F., An improved electromagnetism-like mechanism algorithm for energy-aware many-objective flexible job shop scheduling. *The International Journal of Advanced Manufacturing Technology*, 2022, 119(7-8): 4265–4275. <https://doi.org/10.1007/s00170-022-08665-8>
4. Park, M.J. and Ham, A., Energy-aware flexible job shop scheduling under time-of-use pricing. *International Journal of Production Economics*, 2022, 248: 108507. <https://doi.org/10.1016/j.ijpe.2022.108507>
5. Liu, L., Jiang, T., Zhu, H. and Shang, C., A New Interior Search Algorithm for Energy-Saving Flexible Job Shop Scheduling with Overlapping Operations and Transportation Times. *Axioms*, 2022, 11(7): 306. <https://doi.org/10.3390/axioms11070306>
6. Eberhart, R. and Kennedy, J., A new optimizer using particle swarm theory. in *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 1995, 39–43. IEEE. <https://doi.org/10.1109/MHS.1995.494215>
7. Mirjalili, S., Mirjalili, S.M. and Lewis, A., Grey Wolf Optimizer. *Advances in Engineering Software*, 2014, 69: 46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
8. Mirjalili, S. and Lewis, A., The Whale Optimization Algorithm. *Advances in Engineering Software*, 2016, 95: 51–67. <https://doi.org/10.1016/j.advengsoft.2016.01.008>
9. Chen, F., Li, X. and Yang, X., Multi-objective Flexible Job Shop Scheduling based on Improved NSGA2 Algorithm. *Industrial Engineering Journal*, 2018, 21(2): 55–61.
10. Rajalakshmi, S. and Kanmani, S., A comprehensive review on recent intelligent metaheuristic algorithms. *International Journal of Swarm Intelligence*, 2022, 7(2): 182. <https://doi.org/10.1504/IJSI.2022.123076>
11. Aghaee, Z., Ghasemi, M.M., Beni, H.A., Bouyer, A. and Fatemi, A., A survey on meta-heuristic algorithms for the influence maximization problem in the social networks. *Computing*, 2021, 103(11): 2437–2477. <https://doi.org/10.1007/s00607-021-00945-7>
12. Shen, L., Zhen, G. and Zhu, H., Improved Migratory Bird Migration Algorithm for Flexible Job Shop Scheduling. *Journal of Physics: Conference Series*, 2022, 2219(1): 012036. <https://doi.org/10.1088/1742-6596/2219/1/012036>

13. Chen, R., Yang, B., Li, S. and Wang, S., A self-learning genetic algorithm based on reinforcement learning for flexible job-shop scheduling problem. *Computers & Industrial Engineering*, 2020, 149: 106778. <https://doi.org/10.1016/j.cie.2020.106778>
14. Wang, S.X., Zhang, C.Y. and Jin, L.L., A Hybrid Genetic Algorithm for Flexible Job-Shop Scheduling Problem. *Advanced Materials Research*, 2014, 889-890: 1179–1184. <https://doi.org/10.4028/www.scientific.net/AMR.889-890.1179>
15. Gu, X.L., Huang, M. and Liang, X., A Discrete Particle Swarm Optimization Algorithm With Adaptive Inertia Weight for Solving Multiobjective Flexible Job-shop Scheduling Problem. *IEEE Access*, 2020, 8: 33125–33136. <https://doi.org/10.1109/ACCESS.2020.2974014>
16. Anuar, N.I., Fauadi, M.M., Saptari, A. and Hao, X., Improved Multi-Objective Particle Swarm Optimization For Job-Shop Scheduling Problems. *Journal of Advanced Manufacturing Technology (JAMT)*, 2020, 14(3): 33–49.
17. Zhao, X.H., Wei, Y.F., Wang, K.F. and Ni, Y.Q., Research on Flexible Job Shop Scheduling Problem Based on Improved Ant Colony Algorithm. *Modular Machine Tool & Automatic Manufacturing Technique*, 2022, 2: 165–168. <https://doi.org/10.13462/j.cnki.mmtamt.2022.02.038>
18. Luan, F., Wu, S.Q., Li, F.K., Jia, J.Y. and Cai, Z.Y., A Whale Swarm Optimization Algorithm for Solving Flexible Job Shop Scheduling Problem. *Mechanical Science and Technology for Aerospace Engineering*, 2020, 39(2): 241–246. <https://doi.org/10.13433/j.cnki.1003-8728.20190114>
19. Yuan, Y., Xu, H. and Yang, J.D., A hybrid harmony search algorithm for the flexible job shop scheduling problem. *Applied Soft Computing*, 2013, 13(7): 3259–3272. <https://doi.org/10.1016/j.asoc.2013.02.013>.
20. Li, X., Peng, Z., Du, B., Guo, J., Xu, W. and Zhuang, K., Hybrid artificial bee colony algorithm with a rescheduling strategy for solving flexible job shop scheduling problems. *Computers & Industrial Engineering*, 2017, 113: 10–26. <https://doi.org/10.1016/j.cie.2017.09.005>
21. Tian, Y., Tian, Y.N. and Liu, X., Review on Algorithms for Flexible Job Shop Scheduling Problem. *Journal of Yan'an University(Natural Science Edition)*, 2021, 40(3): 64–70. <https://doi.org/10.13876/J.cnki.ydnse.2021.03.064>
22. Lv, S.Y. and Yu, P., A Review of Green Flexible Job-shop Scheduling Problem. *Logistics Engineering and Management*, 2022, 44(5): 107–111.
23. Luo, X., Qian, Q. and Fu, Y., Review of Application of Genetic Algorithms for Solving Flexible Job Shop Scheduling Problems. *Computer Engineering and Applications*, 2019, 55(23): 15–21.
24. Xue, J. and Shen, B., Dung beetle optimizer: a new meta-heuristic algorithm for global optimization. *The Journal of Supercomputing*, 2023, 79(7): 7305–7336. <https://doi.org/10.1007/s11227-022-04959-6>
25. Xue, J. and Shen, B., A novel swarm intelligence optimization approach: sparrow search algorithm. *Systems Science & Control Engineering*, 2020, 8(1): 22–34. <https://doi.org/10.1080/21642583.2019.1708830>
26. Xia, W. and Wu, Z., An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems. *Computers & Industrial Engineering*, 2005, 48(2): 409–425. <https://doi.org/10.1016/j.cie.2005.01.018>
27. Sun, J. and Xu, L., Disruption management for solving FJSP with improved genetic algorithm. *Journal of Physics: Conference Series*, 2020, 1656(1): 012013. <https://doi.org/10.1088/1742-6596/1656/1/012013>

28. Sule, B. and Lawal, I., An Improve Object-oriented Approach for Multi-objective Flexible Job-shop Scheduling Problem (FJSP). *International Journal of Computer Science and Information Technology*, 2019, 11(5): 91–108. <https://doi.org/10.5121/ijcsit.2019.11508>
29. Xu, W., Liang, J., Gao, Z., Yu, F. and Sheng, S., Improved Harmony Search Algorithm for Solving FJSP. *Computer Applications and Software*, 2022, 39(6): 273–279,335. (Chinese)
30. Tian, M., Zhang, G. and Liu, R., Solve FJSP Considering Transport Time via Particle Swarm Genetic Hybrid Algorithm. *Operations Research and Management Science*, 2019, 28(4): 78–88.
31. Shi, X., Li, Y., Deng, D. and Long, W., Self-adaptive Multistage GA-IWO for Solving Flexible Job Shop Scheduling Problem. *Journal of Mechanical Engineering*, 2019, 55(6): 223–232.
32. Jiang, T., *Mk Series Calculation Example of Flexible Job-shop Scheduling Problem*, CSDN.
33. Yan, X., Ye, C.M. and Yao, Y.Y., Solving Job-Shop scheduling problem by quantum whale optimization algorithm. *Application Research of Computers*, 2019, 36(4): 975–979. <https://doi.org/10.19734/j.issn.1001-3695.2017.10.0985>
34. Tian, Y.N., Tian, Y., Liu, X. and Zhao, Y.L., An Improved Grey Wolf Algorithm for Flexible Job Shop Scheduling Problem. *Computer and Modernization*, 2022, 8: 78–85.
35. Jia, P. and Wu, T., A Hybrid Genetic Algorithm for Flexible Job-shop Scheduling Problem. *Journal of Xi'an Polytechnic University*, 2020, 34(5): 80–86. <https://doi.org/10.13338/j.issn.1674-649x.2020.05.013>
36. Guo, Y., 222. Baidu Netdisk 2023 05/2023.

Author's biography

Dr. Shuangji Yao is currently an associate professor at Yanshan University, China. His research interests cover aspects of Technology and theory of special carrying equipment and Special robot technology. He is a member of IEEE.

Yunfei Guo graduated in Yanshan University(China), He has a master degree of Mechanical Engineering in Vehicle Engineering from Yanshan University. He is specialized in flexible job-shop scheduling problem and dung beetle optimizer.

Botao Yang is a graduate student currently pursuing a master degree of mechanical engineering in vehicle engineering at Yanshan University(China), He is extremely passionate about control algorithm and the data science domain.

You Lv is a Senior Engineer. He is mainly committed to the research of processing and manufacturing technology of complex parts, and is currently the CTO of Shenyang Xinbaolu Aviation Technology Co., Ltd. in China

Dr. Marco Ceccarelli is currently a professor at University of Rome Tor Vergata, Italy. He was the Chair of the IFToMM (International Federation for the Promotion of Mechanism and Machine Science) and Editor in Chief of journal Robotics

Xiaoshuang Dai is an Engineer. She is mainly committed to the processing and manufacturing process design and production planning of complex parts, and is now working in the technical department of Shenyang Xinbaolu Aviation Technology Co., Ltd. in China.

Dr. Giuseppe Carbone is currently a professor at University of Calabria, Italy. He is Chair of the IFToMM TC on Robotics and Mechatronics and Editor in Chief of ROBOTICA Journal (Cambridge University Press).

Appendixes

Appendix A

Table 5. Processing information 1.

W	P	Available machine numbers	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11
	1	[37,12,13,22,21,11,12,13,22,21]	13	14	15								24
	2	[23,24,15,16,18,19,20]											
1	3	[8,9,10,23,24,15,16,18,19,20]								35	37	35	
	4	[1,2,3,11,12,13,22,21]	11	12	13								22
	5	[4,5,6,7,23,24,15,16,18,19,20]			21	23	25	24					
	6	[23,24,15,16,18,19,20]											
	7	[23,24,15,16,18,19,20]											
2	8	[17,4,5,6,7,8,9,10,14]				4	6	7	8	7	5	6	
	9	[1,2,3,11,12,13,21,22]	3	2	2								7
	10	[23,24,15,16,18,19,20]											
	11	[23,24,15,16,18,19,20]											
	12	[23,24,15,16,18,19,20]											
3	13	[17,4,5,6,7,8,9,10,14]				4	5	6	7	7	5	6	
	14	[1,2,3,11,12,13,21,22]	3	4	5								12
	15	[23,24,15,16,18,19,20]											
	16	[17,4,5,6,7,8,9,10,14]				7	6	7	8	9	7	6	
	17	[23,24,15,16,18,19,20]											
4	18	[17,4,5,6,7,8,9,10,14]				4	3	3	2	3	5	4	
	19	[23,24,15,16,18,19,20]											
	20	[1,2,3,11,12,13,21,22]	15	16	17								26
	21	[1,2,3,11,12,13,21,22]	2	3	4								11
	22	[17,4,5,6,7,8,9,10,14]				25	24	24	23	24	26	22	
5	23	[17,4,5,6,7,8,9,10,14]				40	43	41	42	45	42	43	
	24	[17,4,5,6,7,8,9,10,14]				9	7	9	9	8	7	11	
	25	[1,2,3,11,12,13,21,22]	3	4	5								12
	26	[17,4,5,6,7,8,9,10,14]				19	17	19	18	18	17	21	
	27	[17,4,5,6,7,8,9,10,14]				35	32	34	34	35	31	36	
6	28	[17,4,5,6,7,8,9,10,14]				9	7	9	9	8	7	11	
	29	[1,2,3,11,12,13,21,22]	3	4	5								12
	30	[17,4,5,6,7,8,9,10,14]				7	5	7	7	6	6	8	
	31	[23,24,15,16,18,19,20]											
7	32	[23,24,15,16,18,19,20]											
	33	[17,4,5,6,7,8,9,10,14]				36	34	36	36	35	35	37	
	34	[17,4,5,6,7,8,9,10,14]				4	5	6	7	7	5	6	
	35	[1,2,3,11,12,13,22,21]	13	14	15								24
8	36	[23,24,15,16,18,19,20]											
	37	[8,9,10,23,24,15,16,18,19,20]								35	37	35	

	38	[1,2,3,11,12,13,22,21]	11	12	13								22
	39	[4,5,6,7,23,24,15,16,18,19,20]			21	23	25	24					
	40	[23,24,15,16,18,19,20]											
	41	[23,24,15,16,18,19,20]											
9	42	[17,4,5,6,7,8,9,10,14]				4	6	7	8	7	5	6	
	43	[1,2,3,11,12,13,21,22]	3	2	2								7
	44	[23,24,15,16,18,19,20]											
	45	[23,24,15,16,18,19,20]											
	46	[23,24,15,16,18,19,20]											
10	47	[17,4,5,6,7,8,9,10,14]				4	5	6	7	7	5	6	
	48	[1,2,3,11,12,13,21,22]	3	4	5								12
	49	[23,24,15,16,18,19,20]											
	50	[17,4,5,6,7,8,9,10,14]				7	6	7	8	9	7	6	
	51	[23,24,15,16,18,19,20]											
11	52	[17,4,5,6,7,8,9,10,14]				4	3	3	2	3	5	4	
	53	[23,24,15,16,18,19,20]											
	54	[1,2,3,11,12,13,21,22]	15	16	17								26
	55	[1,2,3,11,12,13,21,22]	2	3	4								11
	56	[17,4,5,6,7,8,9,10,14]				25	24	24	23	24	26	22	
	57	[17,4,5,6,7,8,9,10,14]				40	43	41	42	45	42	43	
12	58	[17,4,5,6,7,8,9,10,14]				9	7	9	9	8	7	11	
	59	[1,2,3,11,12,13,21,22]	3	4	5								12
	60	[17,4,5,6,7,8,9,10,14]				19	17	19	18	18	17	21	
	61	[17,4,5,6,7,8,9,10,14]				35	32	34	34	35	31	36	
13	62	[17,4,5,6,7,8,9,10,14]				9	7	9	9	8	7	11	
	63	[1,2,3,11,12,13,21,22]	3	4	5								12
	64	[17,4,5,6,7,8,9,10,14]				7	5	7	7	6	6	8	
	65	[23,24,15,16,18,19,20]											
14	66	[23,24,15,16,18,19,20]											
	67	[17,4,5,6,7,8,9,10,14]				36	34	36	36	35	35	37	
	68	[17,4,5,6,7,8,9,10,14]				4	5	6	7	7	5	6	
	69	[1,2,3,11,12,13,22,21]	13	14	15								24
	70	[23,24,15,16,18,19,20]											
15	71	[8,9,10,23,24,15,16,18,19,20]								35	37	35	
	72	[1,2,3,11,12,13,22,21]	11	12	13								22
	73	[4,5,6,7,23,24,15,16,18,19,20]			21	23	25	24					
	74	[23,24,15,16,18,19,20]											
	75	[23,24,15,16,18,19,20]											
16	76	[17,4,5,6,7,8,9,10,14]				4	6	7	8	7	5	6	
	77	[1,2,3,11,12,13,21,22]	3	2	2								7
	78	[23,24,15,16,18,19,20]											
	79	[23,24,15,16,18,19,20]											
	80	[23,24,15,16,18,19,20]											
17	81	[17,4,5,6,7,8,9,10,14]				4	5	6	7	7	5	6	
	82	[1,2,3,11,12,13,21,22]	3	4	5								12
	83	[23,24,15,16,18,19,20]											
	84	[1,2,3,11,12,13,22,21]	13	14	15								24
18	85	[23,24,15,16,18,19,20]											
	86	[8,9,10,23,24,15,16,18,19,20]								35	37	35	

	87	[1,2,3,11,12,13,22,21]	11	12	13									22
	88	[4,5,6,7,23,24,15,16,18,19,20]			21	23	25	24						
	89	[23,24,15,16,18,19,20]												
	90	[23,24,15,16,18,19,20]												
19	91	[17,4,5,6,7,8,9,10,14]				4	6	7	8	7	5	6		
	92	[1,2,3,11,12,13,21,22]	3	2	2									7
	93	[23,24,15,16,18,19,20]												
	94	[23,24,15,16,18,19,20]												
	95	[23,24,15,16,18,19,20]												
20	96	[17,4,5,6,7,8,9,10,14]				4	5	6	7	7	5	6		
	97	[1,2,3,11,12,13,21,22]	3	4	5									12
	98	[23,24,15,16,18,19,20]												
	99	[17,4,5,6,7,8,9,10,14]				25	24	24	23	24	26	22		
	100	[17,4,5,6,7,8,9,10,14]				40	43	41	42	45	42	43		
21	101	[17,4,5,6,7,8,9,10,14]				9	7	9	9	8	7	11		
	102	[1,2,3,11,12,13,21,22]	3	4	5									12
	103	[17,4,5,6,7,8,9,10,14]				19	17	19	18	18	17	21		
	104	[17,4,5,6,7,8,9,10,14]				35	32	34	34	35	31	36		
22	105	[17,4,5,6,7,8,9,10,14]				9	7	9	9	8	7	11		
	106	[1,2,3,11,12,13,21,22]	3	4	5									12
	107	[17,4,5,6,7,8,9,10,14]				25	24	24	23	24	26	22		
	108	[17,4,5,6,7,8,9,10,14]				40	43	41	42	45	42	43		
23	109	[17,4,5,6,7,8,9,10,14]				9	7	9	9	8	7	11		
	110	[1,2,3,11,12,13,21,22]	3	4	5									12
	111	[17,4,5,6,7,8,9,10,14]				19	17	19	18	18	17	21		
	112	[17,4,5,6,7,8,9,10,14]				35	32	34	34	35	31	36		
24	113	[17,4,5,6,7,8,9,10,14]				9	7	9	9	8	7	11		
	114	[1,2,3,11,12,13,21,22]	3	4	5									12
	115	[17,4,5,6,7,8,9,10,14]				25	24	24	23	24	26	22		
	116	[17,4,5,6,7,8,9,10,14]				40	43	41	42	45	42	43		
25	117	[17,4,5,6,7,8,9,10,14]				9	7	9	9	8	7	11		
	118	[1,2,3,11,12,13,21,22]	3	4	5									12
	119	[17,4,5,6,7,8,9,10,14]				19	17	19	18	18	17	21		
	120	[17,4,5,6,7,8,9,10,14]				35	32	34	34	35	31	36		
26	121	[17,4,5,6,7,8,9,10,14]				9	7	9	9	8	7	11		
	122	[1,2,3,11,12,13,21,22]	3	4	5									12
	123	[17,4,5,6,7,8,9,10,14]				25	24	24	23	24	26	22		
	124	[17,4,5,6,7,8,9,10,14]				40	43	41	42	45	42	43		
27	125	[17,4,5,6,7,8,9,10,14]				9	7	9	9	8	7	11		
	126	[1,2,3,11,12,13,21,22]	3	4	5									12
	127	[17,4,5,6,7,8,9,10,14]				19	17	19	18	18	17	21		
	128	[17,4,5,6,7,8,9,10,14]				35	32	34	34	35	31	36		
28	129	[17,4,5,6,7,8,9,10,14]				9	7	9	9	8	7	11		
	130	[1,2,3,11,12,13,21,22]	3	4	5									12

Table 6. Processing information 2.

W	P	M12	M13	M14	M15	M16	M17	M18	M19	M20	M21	M22	M23	M24
1	1	25	26								20	19		
	2				20	21		15	16	17			26	27

	3				24	25		19	20	18		30	31
	4	23	24										
	5				20	21		15	16	17		26	27
	6				18	19		13	14	15		24	25
2	7				28	29		23	24	25		34	35
	8			11			15						
	9	8	9								4	5	
	10				7	8		4	5	6		11	12
	11				12	13		9	10	11		16	17
3	12				20	21		17	18	19		24	25
	13			10			14						
	14	13	14								8	9	
	15				6	7		3	4	5		10	11
	16			12			16						
	17				18	19		13	14	15		24	25
4	18			10			14						
	19				18	19		13	14	15		24	25
	20	27	28								22	23	
	21	12	13								7	8	
	22			36			40						
5	23			51			55						
	24			15			21						
	25	13	14								8	9	
	26			16			31						
6	27			32			48						
	28			15			21						
	29	13	14								8	9	
	30			13			18						
	31				20	21		15	16	17		26	27
7	32				28	29		23	24	25		34	35
	33			42			47						
	34			10			14						
	35	25	26								20	19	
	36				20	21		15	16	17		26	27
8	37				24	25		19	20	18		30	31
	38	23	24										
	39				20	21		15	16	17		26	27
	40				18	19		13	14	15		24	25
	41				28	29		23	24	25		34	35
9	42			11			15						
	43	8	9								4	5	
	44				7	8		4	5	6		11	12
	45				12	13		9	10	11		16	17
	46				20	21		17	18	19		24	25
10	47			10			14						
	48	13	14								8	9	
	49				6	7		3	4	5		10	11
	50			12			16						
	51				18	19		13	14	15		24	25
11	52			10			14						
	53				18	19		13	14	15		24	25
	54	27	28								22	23	
	55	12	13								7	8	
	56			36			40						
12	57			51			55						
	58			15			21						
	59	13	14								8	9	
	60			16			31						
13	61			32			48						

	62			15			21						
	63	13	14							8	9		
	64			13			18						
14	65				20	21		15	16	17		26	27
	66				28	29		23	24	25		34	35
	67			42			47						
	68			10			14						
	69	25	26								20	19	
15	70				20	21		15	16	17		26	27
	71				24	25		19	20	18		30	31
	72	23	24										
	73				20	21		15	16	17		26	27
	74				18	19		13	14	15		24	25
16	75				28	29		23	24	25		34	35
	76			11			15						
	77	8	9								4	5	
	78				7	8		4	5	6		11	12
	79				12	13		9	10	11		16	17
17	80				20	21		17	18	19		24	25
	81			10			14						
	82	13	14								8	9	
	83				6	7		3	4	5		10	11
	84	25	26								20	19	
18	85				20	21		15	16	17		26	27
	86				24	25		19	20	18		30	31
	87	23	24										
	88				20	21		15	16	17		26	27
	89				18	19		13	14	15		24	25
19	90				28	29		23	24	25		34	35
	91			11			15						
	92	8	9								4	5	
	93				7	8		4	5	6		11	12
	94				12	13		9	10	11		16	17
20	95				20	21		17	18	19		24	25
	96			10			14						
	97	13	14								8	9	
	98				6	7		3	4	5		10	11
	99			36			40						
21	100			51			55						
	101			15			21						
	102	13	14								8	9	
	103			16			31						
22	104			32			48						
	105			15			21						
	106	13	14								8	9	
	107			36			40						
23	108			51			55						
	109			15			21						
	110	13	14								8	9	
	111			16			31						
24	112			32			48						
	113			15			21						
	114	13	14								8	9	
	115			36			40						
25	116			51			55						
	117			15			21						
	118	13	14								8	9	
26	119			16			31						
	120			32			48						

	121			15	21		
	122	13	14			8	9
	123			36	40		
27	124			51	55		
	125			15	21		
	126	13	14			8	9
	127			16	31		
28	128			32	48		
	129			15	21		
	130	13	14			8	9



AIMS Press

©2024 the Author(s), licensee by AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0/>).