[1]Long-Yan Xu,
[1]Yi-Fan Zhao,
[3]Peng Li1,
[2]Ming Li,
[1]Ya-Hong Zhai*,
[4]Li-Ming Huang

Application of Improved Sparrow Search Algorithm to Flexible Job Shop Scheduling Problem

Journal of Electrical Systems

Abstract: - When the reality of energy saving and reduction of enterprises' emissions are a concern, research on the investigation of establishing mathematical models to optimize the production time has been studied. In this article, an enhanced sparrow optimization method is proposed. First, two-layer coding is employed for workpieces and machines according to the model requirements. Secondly, the three-dimensional chaotic mapping scheme is presented to improve the population heterogeneity of the algorithm, and the adaptive inertia weight balance algorithm is implemented to offset the speed of the convergence and its probability. Finally, the Cauchy mutation scheme is adopted to help the algorithm jump out of the local optimum. Simulated data is run to check the superiority of the proposed method. So, through the simulations and comparisons of 10 kinds of test datasets, the outcomes suggest that the solution quality of the enhanced sparrow optimization method has been effectively advanced, and its good global optimization ability is shown, which can provide scheduling strategies for workshop productions. One of the successes of the ISSA algorithm is its superior search accuracy.

Keywords: Sparrow Search Algorithm; Flexible job shop scheduling, Maximum completion time; Cauchy mutation

1 Introduction

With the diversification of customer needs and rapid development of production technology, the complexity of job shop manufacturing has increased, so efficient production planning is one of the key factors in improving the competitiveness of a company. A flexible job shop scheduling problem (FJSSP) depends completely on optimization processes and deals with an assignment problem that combines production planning with workpieces and machine sequencing. An FJSSP is an extended version of a conventional job shop scheduling problem (JSSP), where each operation is assigned to a machine and the operation processes are sequenced. The JSSP belongs to the NP-hard problem class [1]. Similarly, the FJSSP is very tough to resolve, so the FJSSP also belongs to the NP-hard problem class.

In recent years, the FJSSP has received comprehensive attention For example, Wenchong et al. [2] proposed a super heuristic cross-entropy method to resolve the two-stage distributed assembly FJSSP. Deng et al. [3] established a mathematical model considering new job arrivals, machine failures, job cancellations, and alteration of operation processing time, and employed the Monte Carlo Tree Search (MCTS) method to resolve the model. Li et al. [4] enhanced the genetic algorithm to resolve the shortcoming of insufficient flexibility in batch processing of workpieces. Liu et al.[5] suggested the mixed attribute domain search and genetic algorithm to resolve the contradiction between the waiting time and solution quality of the algorithm when the FJSSP model was solved. Wenxiang et al. [6]

[1]School of Electrical Information Engineering, Hubei University of Automotive Technology, sShiyan 442002, China
[2]Institute of Automotive Engineers, Hubei University of Automotive Technology, Sshiyan 442002, China
[3]Zhe jJiang Hospital, hHangzhou 310052, China
[4]Dongfeng Motor Chassis Systems Co.,Ltd, Shiyan 442002, China
*Correspondence: longtian1119@163.com

established the FJSSP problem considering workpiece job outsourcing and utilized fuzzy hierarchical analysis. Chen et al.[7] improved the NSGA-II method to resolve the FJSSP. Dong et al. [8] introduced the total tariff into the shop scheduling model and implemented an adaptive learning rate mechanism to guide the estimation method to determine the optimal solution. Li et al. [9] mixed the genetic algorithm and taboo search algorithm to resolve the FJSSP problem. Zhenwei et al. [10] studied the uncertain FJSSP with job priority requirements. Changxing et al. [11] constructed an FJSSP model with maximum completion time and machine load as optimization objectives, and a mixed elite retention mechanism and summation search algorithm. Berend et al. [12] suggested a new FJSSP model utilizing optimization based on quantum computing. Xin et al. [13] investigated the FJSSP with AGV strategy utilizing multiple genetic algorithms. Ye et al. [14] proposed an adaptive hybrid optimization method employing reinforcement learning.

Wang et al. [15] proposed an invasive weed optimization algorithm embedded based on the gray wolf algorithm, which was utilized to enhance the capabilities of the global and local search and enhance the quality of the initial solution. Wang et al. [16] suggested a new encoding methodology and optimal subpopulation-based genetic algorithm to resolve the flexible workshop scheduling problem. Jia et al. [17] suggested a hybrid Pareto method based on omniscient particle swarm optimization and dynamic forbidden search to enhance the population diversity and convergence speed. Yu et al. [18] proposed a strategy for producing the initial population by utilizing full search encoding for the initial population to enhance the quality of the solution. Xu et al. [19] suggested a new discrete bat method. Wang et al. [20] suggested an enhanced whale algorithm using the variable neighborhood of the critical path. Edilson et al. [21] suggested an optimization based on the hybrid particle swarm method and a stochastic re-climbing method to resolve the FSSP. Wang et al. [22] designed an advanced ant colony method to optimize the maximum completion time of the flexible scheduling shop. Zheng et al. [23] suggested a novel coding scheme of knowledge rule-guided Drosophila optimization method to resolve the dual-resource constrained FJSSP.

In this paper, the improved sparrow search method is applied to resolve the mathematical model to optimize the production time and the proposed method is effectively verified by the simulations based on production examples.

2 The description of the flexible job shop scheduling problem

The FJSSP is described as follows: given a set of n independent jobs J = { J_1 , J_2 , . . . , J_n} and a set of m machines M = {M_1, M_2, . . . , M_m}, a job J_i is scheduled by a set of operations O_i={O_1,O_2,…,O_n} in the given order, where i denotes the number of operations in J_i . The operation $O_{i,j}$ could be conducted by any machine in the given set of machines $M_{i,j}$ ∈ M. The processing time of an operation depends on its machine assignment if it is assigned to a machine M_K, $P_{i,j,k}$ represents the processing time of operation $O_{i,j}$.

1. At t = 0, all workpieces are machined with the same priority.
2. Since the processes of each workpiece are different, the processes of each workpiece do not constitute a constraint.
3. Once a given job starts processing on a given machine, interruption is not allowed, and processing will continue until completion (no preemption).
4. Only one workpiece is allowed to be processed by a single machine at the same time.
5. The start time of any operation is greater than or equal to the release time of the previous phase, ignoring machine setup time and workpiece transport time.

With the maximum completion time as a decision variable, denoted by C, Eq. (1) presents it.

$$C = \max_{1 \le i \le N} C_i = \max_{1 \le i \le N}(\max_{1 \le j \le N_i} C_{ij}) \tag{1}$$

3 Sparrow Search Algorithm (SSA)

Producers with bigger fitness scores have priorities for obtaining food during foraging in the sparrow search algorithm (SSA). The location of the discoverer is updated in each run and is described below.

The population size of sparrows is denoted by N, D denotes the spatial dimension, and a vector with D dimension whose median score is in [-1,1] in each dimension is randomly produced as the initial operator, the individual positions of sparrows are denoted by $x_i = (x_{i1}, x_{i2}, \ldots, x_{id})$, i ∈ [1, N], d ∈ [1, d] and x_{id} denotes the position of the i^{th} sparrow in the d-dimensional space. Eq. (2) is updated by the discoverer position.

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^{t} \cdot exp(\frac{-i}{\alpha \cdot iter_{\max}}) & R_2 < ST \\ X_{i,j}^{t} + Q \cdot L & R_2 \ge ST \end{cases} \tag{2}$$

where t denotes the current iteration numbers, T denotes the highest number of iterations, α represents a random number between (0,1), Q denotes a random number distributed normally, L denotes a matrix of size 1 × d, R_2∈ [0, 1] represents the warning score, ST ∈ [0.5, 1] denotes the safety score.

Once R_2 < ST, the population is not in danger and the scouts continue searching. On the other hand, once R_2 ≥ ST, the vigilant in the flock has pinpointed a predator and instantly alerted the other sparrows, and then the flock executes anti-predatory actions, tunes its search plan, and flies rapidly to the safe region. During foraging, all sparrows, except the producers, act as followers in search of the best foraging area. Eq. (3) presents the position update of the follower.

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot exp(\frac{X_{\text{worst}} - X_{i,j}^{t}}{i^2}) & i > N/2 \\ X_p^{t+1} + |X_{i,j}^{t} - X_p^{t+1}| A^{+}.L & i \le N/2 \end{cases} \tag{3}$$

where X_{worst} denotes the global worst position for the t^{th} iteration. X_p^{t+1} represents the global best position of the t+1th iteration, A represents an l×d matrix where each element is randomly set to 1 or -1, A^{+} $A^{+} = A^{\mathrm{T}}(AA^{\mathrm{T}})^{-1}$. Once i > N/2, it means that the i^{th} joiner has not yet got food and needs to fly to other places to forage for food. Once i ≤ N/2, the follower at i^{th} the place is near the global optimal position and forages around randomly. The initial position of individuals randomly generated in the population is formulated as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{best}^{t} + \beta |X_{i,j}^{t} - X_{best}^{t}| & f_i \ne f_g \\ X_{i,j}^{t} + K\left(\frac{|X_{i,j}^{t} - X_{worst}^{t}|}{(f_i - f_w) + \varepsilon}\right) & f_i = f_g \end{cases} \tag{4}$$

where β represents the step control parameter, a random number distributed normally whose mean and variance are 0 and 1, respectively, K represents the movement direction of sparrows and takes values randomly in [-1,1], ε denotes a very small constant, f_i denotes the adaptation of the i^{th} sparrow, f_g and f_w denote the best and worst adaptations of the current sparrow population.

Once $f_i = f_g$, the i^{th} sparrow is located at the edge of the population, and then the sparrow in this position is vulnerable to predators, when $f_i = f_g$, the sparrow at the ith place is located in the population center, and since the sparrow is aware of the existence of threats, needs to avoid being captured by approaching other sparrows.

4 The improved SSA algorithm

The SSA algorithm has some shortcomings: the initialized population is not rich enough, the global search ability is poor, local optimums are easy to appear, the solution accuracy is often not high enough, and premature convergence is easy to occur. To better deal with these issues affecting the performance of the SSA method, the following improvements are made in this paper.

4.1 Encoding and decoding

Two layers of real coding are adopted. The first layer is the OS code for the procedure, and the second layer is the MS code for the machine. The coding length of each layer is the total number of processes, the OS code represents the serial number of the workpiece, and the number of times appears as the number of processes of the workpiece, the MS code represents the first step from the first workpiece, and the production equipment is assigned one by one, and the order of the equipment is recorded (1,2, ..., k). As shown in Figure 1, four workpieces are produced on three machines.

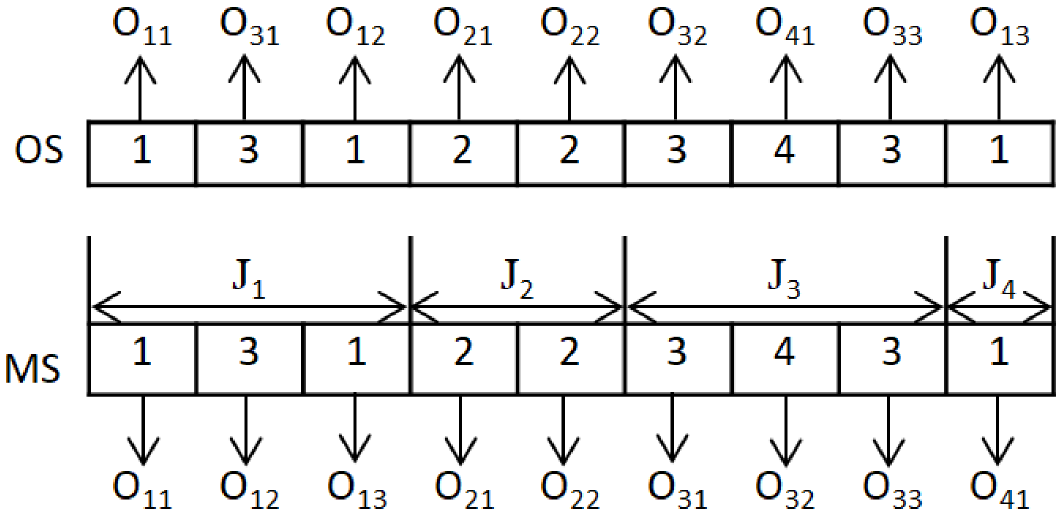

Figure 1 Work sequence and machine selection of two-level coding process.

First, all processes are added to the corresponding equipment in the order of sorting workpieces. Then, the process sequence is traversed and the process equipment is assigned according to the earliest start time of the workpiece (i.e., the maximum of the end time of the immediately preceding process and the finish time of the previous process of the equipment). Finally, the active scheduling solution is obtained by using the plug-in decoding method.

4.2 Population initialization

The quality of the initial solution parameters determines the quality of the algorithm's solution, taking a random number of generations to ensure the population's diversity, but it influences greatly the algorithm in the later solution. Therefore, we choose to adopt the hybrid initialization method to enhance the algorithm's performance. [24] suggested that 30% of them are generated using a random search strategy, and then the processing machine with a short waiting time starts producing the artifact. If the machine waits the same processing time, it continues processing by randomly generating the machine, Otherwise, the operation continues. The remaining 70% is generated using a global search strategy, and the same as above for selecting the processing machine.

4.3 Initialization of chaotic sequence

When complex optimization problems are under investigation, SSA copes with poor population diversity as a disadvantage in runs. In recent years, chaotic mappings have been utilized in many optimization fields as an alternative to pseudo-random number generators. It is indicated in [25] that the population initialized using chaotic sequences affects the whole process and often achieves better results than pseudo-random generators. For example, in [26], chaotic sequences were used to dynamically enhance the size of the population to evade immature convergence, in [27], chaotic sequences were utilized for the initial population production and the performance of the variation operator. Chaotic sequences could be mapped by distinct chaotic models such as Tent map, Logistic map, Kent map, and Cubic map. [28] shows that cubic mapping presents better uniformity and higher flexibility than other mappings. Thus, a cubic map chaotic sequence is used for the initialization of the population in this paper. Thus, the chaotic maps' ergodicity and initial sensitivity improve the diversity of SSA populations. Eqs. (5) and (6) present them.

$$y_{i+1} = 4y_i^3 - 3y_i \tag{5}$$

$$X_i = X_{lb} + \frac{(X_{ub} - X_{lb}) \times (y_{i+1})}{2} \tag{6}$$

where $-1<y_i<1, y_i = 0$, i = 0, 1. . .N. X_{lb} and X_{ub} are the upper and lower bounds, and N is the size of the population. First, let D represent dimension, and randomly generate a D-dimensional vector in [-1,1] in each dimension as the first operator. Then, Eq. (5) was employed to iterate over each dimension of the first operator to attain the rest of the (N-1) operators. Lastly, Eq. (6) was employed to map the operators' scores produced by the three mappings onto the individual sparrows. The recent literature presents some contributions. Doush et al. [29] introduced the flow shop scheduling with blocking which is thought of as a substantial issue in scheduling settings that could be easily applied to real cases. A method called the harmony search algorithm (HSA) was proposed to minimize the total flow time. Doush et al. [30] suggested the island neighboring heuristics harmony search algorithm (INHS) to overcome the blocking flow-shop scheduling problem to diversify the population. Hence, the performance of the proposed algorithm was increased. Wadallah et al. [31] reviewed the recent research articles related to the SSA covering distinct extensions to avoid convergence issues. Besides, the SSA with a multi-objective version was reviewed. Lastly, the critical analysis of the main research gaps in the convergence behavior of SSA was discussed.

4.4 Adaptive inertia weights

In the SSA scouts need a larger search space to find food sources, and whether SSA could determine the optimal solution depends primarily on the scouts' searchability. The individual's location within the search range is randomly distributed. Once there exist no neighboring sparrows close to the former scout, a random search scheme will be performed. Note that this model not only slows down the convergence rate but also reduces the convergence precision with limited iteration numbers. To further improve the algorithm's performance, adaptive inertia weights are introduced in Eq. (2).

In Eq. (2), the largest factor affecting the producer's location is $\frac{i}{\propto \cdot \mathrm{iter}_{max}}$, and when α has a larger random value, the range of values taken may progressively decrease from (0, 1) to roughly (0, 0.4) as i increases. Therefore, an adaptive control factor is introduced to extend the search range of the producer as presented in Eq. (7).

$$\omega = \omega_0 + \mathrm{c}^{\mathrm{t}} \tag{7}$$

where ω0 = 1 denotes the initial weight, c denotes the adaptive factor of ω which could be set according to the real problem, and t denotes the current iteration numbers. To keep the value of ω small, increase the search range of producers, and enhance the algorithm's global search capability, the value of c is initially set to 0.9. Thus, Eq. (8) presents the update of the discoverer position.

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t \cdot exp\left(\frac{-i}{\omega \cdot \alpha \cdot iter_{\max}}\right) & R_2 < ST \\ X_{i,j}^t + Q \cdot L & R_2 \geq ST \end{cases} \tag{8}$$

To avoid predators during foraging, 10-20% of sparrows were picked as scouts. The presence of scouts could aid sparrow populations to obtain better SSA results. When the scouts' number (SN) grows large, it would help advance the sparrows' global optimization. Nevertheless, when the SN grows small, it would help speed up the SSA's convergence. Thus, an adaptive update equation for the scout number is suggested, as presented in Eq. (9), which can be reduced nonlinearly during the iterative process.

$$SN = SN_{max} - round\left((SN_{max} - SN_{min})\frac{iter}{iter_{max}}\right) \tag{9}$$

where $SN_{\max}$ is the highest score for the number of scouts, $SN_{\max}$ indicates the smallest score of the number of scouts, the round mapping was utilized to round the numbers, $iter$ denotes the current iteration, $iter_{max}$indicates the highest number of iterations.

4.5 Cauchy mutation strategy

In the later iterations of the SSA, sparrows progressively converge toward the optimal individuals, resulting in insufficient population heterogeneity and an inclination for the method to converge prematurely. To solve the issue, the Cauchy mutation scheme was presented. The individual with the best current fitness was picked for mutation. Afterward, their positions were compared based on before and after positions, and then the better position was selected to put into a subsequent run. Eqs. (10) and (11) present the Cauchy mutation scheme.

$$U_{best}^{t+1} = \mathrm{X}_{best}^{t}[1 + \tau_1\, Cauchy\,(0, \vartheta^2) + \tau_2\, Gauss\,(0, \vartheta^2)] \tag{10}$$

$$\vartheta = \begin{cases} 1 & f(X_{best}) < f(X_i) \\ exp\left(\dfrac{f(X_{best}) - f(X_i)}{|f(X_{best})|}\right) & f(X_{best}) \geq f(X_i) \end{cases} \tag{11}$$

where X_{best} denotes the position of the elite individual, U_i^{t+1}represents the position of the elite individual after mutation, and ϑ^2represents the standard deviation of the Cauchy-Gaussian mutation scheme. Cauchy $\vartheta^2$is a random variable that satisfies the Cauchy distribution, Gauss $\vartheta^2$is a random variable that satisfies the Gaussian distribution. $\tau_1 = 1 + \mathrm{t}^2/\mathrm{T}^2{}_{\max}$ and $\tau_2 = 1 - \mathrm{t}^2/\mathrm{T}^2{}_{\max}$ are dynamic parameters that are adaptively tuned with the number of iterations. In Eq.(10), the initial stage τ1 is large, allowing the algorithm to explore a larger range of optimal solutions with a larger variation step. τ_2 is a small variation step, which facilitates the method of searching for a near-optimal solution. When the search process is run,τ_1 reduces while τ_2 keeps rising.

4.6 Mutation operator

During the search process of the SSA, the sparrow population's heterogeneity decreases, which may lead to the algorithm's premature convergence. To surmount the SSA's local optimum issue, the mutation operator was suggested to boost the population diversity. So, it helps resolve the local optimum problem. Particularly, just one individual random variation in each iteration has little impact on the algorithm's convergence.

$$x_r = lb + (ub - lb).rand(1, D) \tag{12}$$

where r denotes a random individual in each iteration, and the position of sparrow r will be initialized by variation.

The improved SSA algorithm performs the following flow.

1. Parameter setting: number of populations, the highest number of iterations, number of discoverers, number of scouts, etc., initialized utilizing mapped chaotic sequences.
2. Rank fitness scores to determine the current best and worst individuals.
3. Determine the investigator according to Eq. (8).
4. Perform a position update according to Eq. (3).
5. Identify the Watchman and perform the location update according to Eq. (4).

6. Select elite individuals according to Eq. (10) and Apply adaptive mutation to them.
7. Obtain the current updated location.
8. Decide whether the termination condition is met, and if so, move to the next step, otherwise execute step 2.
9. Output the optimal result.

5 Simulation Experiments

5.1 Experimental platform and parameter settings

The Improved SSA algorithm (ISSA) was conducted by using MATLAB R2019b with a CPU of i5-10500, a main frequency of 3.10GHZ, a computer memory of 8GB, a population size P of 100, and a maximum iteration $t_{\max}$ of 600.

5.2 Analysis of simulation results

For the 10 benchmark cases (Kacem01 and Kacem08, MK1-MK8) in [1], each algorithm was run 10 times independently.

Table 1 depicts that the best denotes the optimal score, the min denotes the smallest score, the average represents the average value, and the RPD represents the relative percentage difference, the parameters used for the RPD are the optimal score and the minimum score, and the equation is presented by RPD=100×(best-min)/min, the smaller its value suggests that the stability and uniformity of the algorithm are better, and the value obtained by this algorithm is more representative. ISSA, SSA, WOA, GWO, and PSO denote advances in the sparrow, whale gray wolf, and particle swarm optimization algorithms. The RPD is the average of the five algorithms under 10 cases. The results suggest that the values obtained by the ISSA algorithm are relatively representative, and the performance of the ISSA algorithm is more stable.

Table 1 Comparison results of the benchmark algorithm

Example and size	ISSA			SSA			WOA			GWO			PSO		
	best	avg	RPD%	best	avg	RPD%	best	avg	RPD%	best	avg	RPD%	best	avg	RPD%
Kacem1 7×8	33	35	0	56	37	47	45	38	8	51	35	14	39	35	5
Kacem2 10×10	39	43	0	44	45	0	47	49	0	39	43	5	43	54	41
MK1 10×6	59	62	18	60	65	12	65	71	39	72	77	10	84	90	0
MK2 10×6	44	44	0	46	50	4	47	53	28	47	52	0	53	67	0
MK3 15×8	402	402	0	404	404	0	420	422	0	408	419	32	411	430	25
MK4 15×8	109	117	24	114	123	24	110	120	14	127	130	37	127	130	0
MK5 15×4	190	191	0	192	192	0	218	224	41	190	227	35	216	245	12
MK6 10×15	116	120	24	124	133	0	147	140	12	132	149	21	140	153	26
MK7 20×5	182	190	0	192	201	17	182	217	24	185	225	28	211	222	16

MK8 20×10	598	598	0	618	639	33	621	636	49	616	621	25	598	598	0
$\overline{\text{RPD}}$ %	6.6			13.7			21.5			20.7			12.5		

Table 2 shows the relative lift rates, where Δ_{best} is the optimal value of the lift rate and Δ_{avg} is the mean of the lift rate, for example, for Kacem1, the optimal value of the lift rate of the ISSA relative to SSA is calculated by $\Delta_{\text{best}} = 100\times(\text{SSA}_{\text{best}} - \text{ISSA}_{\text{best}})/\text{SSA}_{\text{best}}$. For the other four algorithms, the ISSA obtains a certain lifting rate and obtains a maximum lifting rate of 43.8% and a minimum lifting rate of 6.25% for Kacem1. A maximum lifting rate of 20.5% and a minimum lifting rate of 6.6% are achieved for Kacem2. A maximum lifting rate of 31.1% a minimum lifting ratio of 1.7% for MK1, and the highest lifting ratio of 34.3% and a minimum lifting rate of 4.3% for MK2. A maximum lift rate of 34.3% and a minimum lift rate of 4.3% for MK3. A maximum lift rate of 6.5% and the smallest lift ratio of 0.49% are achieved for MK4. The highest lift ratio of 14.2% and the smallest lift ratio of 0.9% is gained for MK5. The highest lift ratio of 23.5% and the smallest lift ratio of 0.5% for MK6 is received. The highest ratio was 21.1% and the minimum lift rate was 3.3% for MK7. The maximum lift rate was 15.6% and the minimum lift rate was 1.6% for MK8. Therefore, the maximum lift rate was 6.4% and the minimum lift rate was 2.9%.

Table 2 Comparison of experimental results

Example and sizes		Kacem1 7×8	Kacem2 10×10	MK1 10×6	MK2 10×6	MK3 15×8	MK4 15×8	MK5 15×4	MK6 10×15	MK7 20×5	MK8 20×10
Relative	Δ_{best}	41	12.8	1.7	4.3	0.49	4.4	1	3.3	5.2	3.2
SSA（%）	Δ_{avg}	5.4	6.6	4.6	12	0.49	4.9	0.5	9.8	5.4	6.4
Relative	Δ_{best}	26.7	20.5	10.1	6.4	4.3	0.9	12.8	21.1	-	3.7
WOA（%）	Δ_{avg}	7.9	12.2	12.7	17	4.7	2.5	14.7	14.3	12.4	6
Relative	Δ_{best}	35.3	-	22	6.4	1.4	14.1	-	12.1	1.6	2.9
GWO（%）	Δ_{avg}	-	-	19.4	15.4	4.1	10	15.9	19.5	15.6	3.7
Relative	Δ_{best}	15.4	9.3	29.7	17	2.18	14.2	12	17.1	13.7	-
PSO（%）	Δ_{avg}	-	20.4	31.1	34.3	6.5	10	23.5	21.5	14	-

Figures 2 and 3 compare the convergence curves of the five algorithms tested for the two instances of Kacem1 and Kacem2, respectively. The performance of the ISSA algorithm is bigger than the other four algorithms, and for Kacem1, the quality of the optimal solution is ISSA, PSO, SSA, WOA, and GWO when ordered from strong to weak, for Kacem2, the quality of the optimal solution is ISSA, PSO, GWO, SSA, and WOA when ordered from strong to weak. In Figure 2, the ISSA obtains the largest optimal solution of 63 and the smallest of 33, where the difference is 30, which is the largest when compared with the other four algorithms, in Figure 3, ISSA obtains the largest optimal solution of 133 and the smallest of 41, with a difference of 92, and ISSA obtains the largest difference from the initial state to the convergence state when compared with the others. In brief, it shows that the ISSA has a faster convergence speed and stronger searchability.

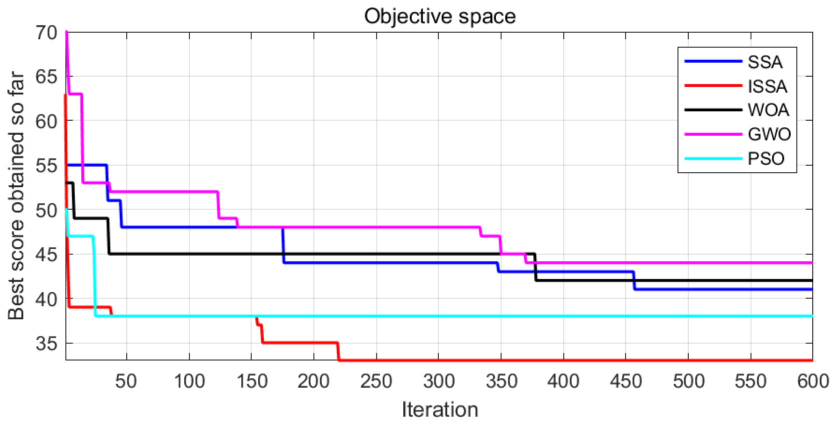

Figure 2 The convergence curve of Kacem1

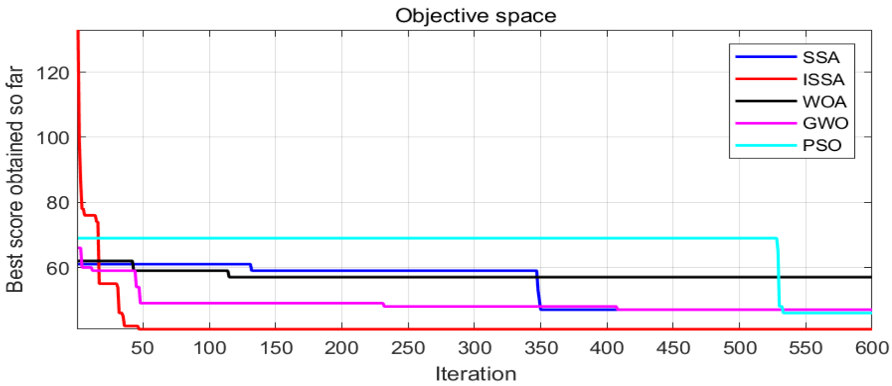

Figure 3 The convergence curve of Kacem2

Figures 4 and 5 show the Gantt charts obtained by the ISSA algorithm for Kacem1 testing, where the horizontal axis is the production time, the vertical axis is the processing equipment, different colors represent different workpieces, and the serial number represents the production process of that workpiece. The effect of the workpiece scheduling generated by the optimized algorithm can be visualized in Figure 3. As shown in Figures 4 and 5, 10 workpieces are processed on 10 machines, and the final optimal processing time is obtained as 39 and 47 seconds, respectively.

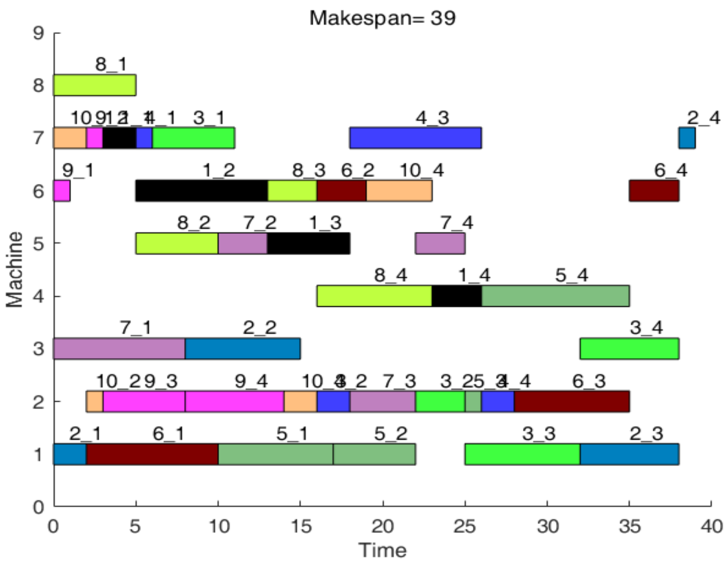

Figure 4 The Gantt chart of Kacem1

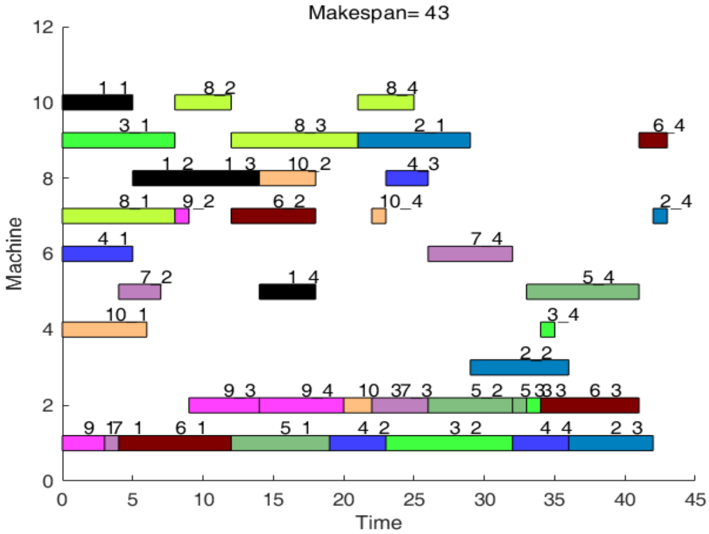

Figure 5 The Gantt chart of Kacem2

6 Conclusion

When the reality of energy saving and reduction of enterprises' emissions are a concern, research on the investigation of establishing mathematical models to optimize the production time has been studied. In this article, an enhanced sparrow optimization method is proposed. First, two-layer coding is employed for workpieces and machines according to the model requirements. Secondly, the three-dimensional chaotic mapping scheme is presented to improve the population heterogeneity of the algorithm, and the adaptive inertia weight balance algorithm is implemented to offset the speed of the convergence and its probability. Finally, the Cauchy mutation scheme is adopted to help the algorithm jump out of the local optimum. Simulated data is run to check the superiority of the proposed method. So, through the simulations and comparisons of 10 kinds of test datasets, the outcomes suggest that the solution quality of the enhanced sparrow optimization method has been effectively advanced, and its good global optimization ability is shown, which can provide scheduling strategies for workshop productions.

The SSA is a swarm intelligence-based optimization, and the following improvements are made to the SSA method for the characteristics of FJSS: 1. two-layer real number encoding is employed, 2. chaotic mapping instead of the pseudo-random number generator is utilized to enrich the initialized population, 3. adaptive inertia weights are employed to enhance the algorithm's search ability, 4. the Cauchy mutation scheme is implemented to avoid the algorithm's premature convergence.

The comparison suggests that the ISSA algorithm presents superiority regarding search accuracy, convergence performance, stability, and exploration over WOA, PSO, and GWO and has an ability not to fall into a local optimum.

Funding

This study was supported by Hubei Provincial Department of Education Key Research Plan Project (No. D20211802), .Hubei Provincial Science and Technology Department Key R&D Plan Project (No. 2022BEC008).

Conflict of Interest: None

References

[1] Brucker, P., Schlie, R. Job-shop scheduling with multi-purpose machines[J]. Computing,1990,45(4).
[2] Luo, W.; Qian, B.; Hu, R.; Zhang, C.; Xiang, F. Hyperheuristic cross-entropy algorithm for distributed assembly flexible job shop scheduling problem. Control Theory and Applications 2021, 38, 1551–1568.
[3] Li, K.; Deng, Q.; Zhang, L.; Fan, Q.; Gong, G.; Ding, S. An effective MCTS-based algorithm for minimizing makespan in dynamic flexible job shop scheduling problem. Computers & Industrial Engineering 2021, 155.
[4] Li, T.; Zhu, B.; Zhang, k. Flexible job shop scheduling considering setup time and job batching. Mechanical & Electrical Engineering Technology 2021, 50, 40–44.
[5] Liu, B.; Yuan, X.; Ming, X. Flexible job shop synchronization scheduling method based on genetic algorithm and variable neighborhood search. Machine Design & Research 2021, 37, 177–182,189.
[6] Xu, W.; Hu, Y.; Luo, W.; Wang. L..; Wu, R. A multi-objective scheduling method for distributed and flexible job shop based on hybrid genetic algorithm and tabu search considering operation outsourcing and carbon emission. Computers & Industrial Engineering 2021, 157.
[7] Chen, Z.; Chen, S.; Chen, H. Large-scale flexible Job shop scheduling problem based on improved multi-population NSGA-II algorithm. Transducer and Microsystem Technologies 2021, 40, 51–54.
[8] Dong, Y.; Hu, R.; Yao, Y. To solve the green pipeline scheduling problem with order-dependent setup time. Information Technology 2021, pp. 1–6.
[9] Li, X.; Gao, L. An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem. International Journal of Production Economics 2016, 174, 93–110.
[10] Zhu, Z.; Zhou, X. A multi-objective multi-micro-swarm leadership hierarchy-based optimizer for uncertain flexible job shop scheduling problem with job precedence constraints. Expert Systems with Application 2021, p. 182.
[11] Song, C.; Ran, J.; W, C. Study on flexible Job shop Scheduling problem based on hybrid multi-objective genetic algorithm. Journal of Mechanical Electrical Engineering 2021.
[12] Denkena, B.; Schinkel, F.; Pirnay, J.; Wilmsmeier, S. Quantum algorithms for process parallel flexible job shop scheduling. CIRP Journal of Manufacturing Science and Technology 2021, 33, 100–114.
[13] Jin, X.; Wang, L. Flexible Job shop scheduling with AGV constraints based on multi-population genetic algorithm. Journal of Changzhou Institute of Technology 2021, 34, 16–21.
[14] Ye, J.; Xu, D.; Hong, H.; Lai, Y.; Jiang, M. AHOA: Adaptively Hybrid Optimization Algorithm for Flexible Job-shop Scheduling Problem. Algorithms and Architectures for Parallel Processing 2022, 13155, 271–287.
[15] Shuai, Q.; Yao, X.F. A Hybrid Genetic Algorithm for Flexible Job-Shop Scheduling Problem. Industrial Engineering Journal 2013.
[16] Wang, J.; Zhu, H.; Zhou, J. Optimal subpopulation genetic algorithm for flexible flow shop scheduling problem. Application Research of Computers 2012, 29, 442–444,526.
[17] Arab Bafrani H, Shahrokhi Kahnooj H, Khaksar S. A Combination of Art, Creativity and Neuroscience. kurmanj 2023; 5 (1) :1-12
[18] Yu, Z.; Yuan, Y.; Li, X. Improved initial population genetic algorithm for flexible job-shop scheduling. Machinery Design & Manufacture 2016, pp. 258–260,264.
[19] Xu, H.; Zhang, T. A novel discrete Bat algorithm for flexible flow shop scheduling problem. Computer Engineering and Applications 2016, pp. 262–265.
[20] Wang, S.; Li, Y.; Li, X. Flexible job shop scheduling method based on Whale swarm algorithm. Journal of Chongqing University (Natural Science Edition) 2020, 43, 1–11.

[21] Rodrigues Kato, E.R.; de Aguiar Aranha, G.D.; Tsunaki, R.H. A new approach to solve the flexible job shop problem based on a hybrid particle swarm optimization and Random-Restart Hill Climbing. Computers Industrial Engineering 2018, 125, 178–189.
[22] Wang, L.; Cai, J.; Li, M.; Liu, Z. Flexible Job Shop Scheduling Problem Using an Improved Ant Colony Optimization. Scientific programming 2017, 2017.
[23] Juma Alsuwaidi S J J, Mohamed Sultan A A. Strategies and Challenges to Enhance the Impact of E-Service Quality on Institutional Excellence: A proposed framework. sjamao 2023; 5 (4) :1-10.
[24] Zhang, Z.; Qian, B.; Hu, R.; Wang, L.; Xiang, F. Low carbon distributed assembly flow shop scheduling based on multidimensional EDA algorithm. Control and Decision 2022, 37, 1367–1377.
[25] Liu, X.; He, Y.; Lu, F.; Wu, C.; Cai, X. Chaotic Crow algorithm based on differential evolution strategy is used to solve discount 0-1 knapsack problem. Journal of Computer Applications 2018, 38, 10.
[26] Lu, X.; Mu, X.; Zhang, J.; Wang, Z. Chaotic Sparrow search optimization algorithm. Journal of Beijing University of Aeronautics and Astronautics 2021.
[27] Yang, Z.; Zheng, Z.; Ao, R.; Wang, G.; Zong, X.; Li, P. Industrial intrusion detection method based on improved Sparrow algorithm. Information Technology and Network Security 2021, 40, 8.
[28] Xie, J. Global bifurcation, and chaos of cubic mapping. Journal of Southwest Jiaotong University 1994, pp. 30–38.
[29] Kacem, I., Hammadi, S., Borne, P. Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems, in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 32, no. 1, pp. 1-13, Feb. 2002, doi: 10.1109/TSMCC.2002.1009117.Brandimarte, P. Routing and scheduling in a flexible job shop by tabu search. Annals of Operations Research 1993, 41, 157–183.
[30] Doush, I. A., Al-Betar, M. A. Awadallah, M. A., Santos, E., Hammouri, A. I., Mafarjeh, M., AlMeraj, Z. Flow shop scheduling with blocking using modified harmony search algorithm with neighboring heuristics methods, Applied Soft Computing, 2019, Volume 85, 105861.
[31] Doush, I. A., Al-Betar, M. A., Awadallah, M. A., Alyasseri, Z. A. A., Makhadmeh, S. N., El-Abd, M. Island neighboring heuristics harmony search algorithm for flow shop scheduling with blocking, Swarm and Evolutionary Computation, 2022, Volume 74, 101127.
[32] Wadallah, M. A., Al-Betar, M. A., Doush, I. A., Makhadmeh, S. N., Al-Naymat, G., Recent Versions and Applications of Sparrow Search Algorithm, Archives of Computational Methods in Engineering, 2023, 2831-2858, 30, 5