

# JOB-SHOP SCHEDULING PROBLEM BASED ON IMPROVED CUCKOO SEARCH ALGORITHM

Hu, H. X.<sup>\*,\*\*\*\*</sup>; Lei, W. X.<sup>\*\*</sup>; Gao, X.<sup>\*\*\*,#</sup> & Zhang, Y.<sup>\*</sup>

<sup>\*</sup> College of Computer and Information, Hohai University, Nanjing 211100, China

<sup>\*\*</sup> School of Public Administration, Hohai University, Nanjing 211100, China

<sup>\*\*\*</sup> College of Electrical Engineering, Nanjing Institute of Industry Technology, Nanjing 210023, China

<sup>\*\*\*\*</sup> Department of Electrical Engineering, Tibet Agricultural and Animal Husbandry College, Tibet 860000, China

E-Mail: hexuan\_hu@hhu.edu.cn, njlylxx4239@126.com, 2013100784@niit.edu.cn, yzh@hhu.edu.cn

(<sup>#</sup> Corresponding author)

## Abstract

This paper makes an in-depth exploration into the job-shop scheduling problem (JSP). After reviewing the related literature, the local search mechanism of the particle swarm algorithm (PSA) and the large-span search principle of standard cuckoo search algorithm (CSA) were combined into an improved cuckoo search algorithm (ICSA), which is capable of both local search and global search. Later, several simulation experiments were carried out on the LA type typical library proposed by Lawrence, and the stability and accuracy of the ICSA was contrasted with those of the PSA and the genetic algorithm (GA) based on the means and variances in multiple iterations. After the comparison, a convergence analysis of the ICSA was specially designed for our model. The results demonstrate that the ICSA provides a better tool for solving the JSP than other algorithms. The research findings lay a solid theoretical basis for the JSP in the actual production process.

(Received, processed and accepted by the Chinese Representative Office.)

**Key Words:** Job-Shop Scheduling Problem (JSP), Improved Cuckoo Search Algorithm (ICSA), Numerical Simulation

## 1. INTRODUCTION

In actual practice, the job-shop scheduling problem (JSP) can be extended to various aspects of industrial production, including but not limited to transport route planning, product manufacturing and mobile communication equipment fabrication. With the trend of multi-type, small-batch customization in the manufacturing industry, the focal point of the JSP has shifted to fulfilling the individualized demand and delivery time of different products. The JSP has been proved as a typical NP-hard problem, as its solution needs to satisfy multiple nonlinear constraints and consider continuous and discrete variables. Hence, both the industry and the academia have been pursuing an efficient and accurate algorithm to solve the problem.

The existing JSP-solving algorithms mainly fall into three categories: analytical algorithms (AAs), heuristic algorithms (HAs), and swarm intelligence algorithms (SIAs). In general, only a few simple JSP models can derive analytical solutions via AAs, while the clear majority only provides approximate solutions through HAs and SIAs. In this paper, the local search mechanism of particle swarm algorithm (PSA) was combined with the large-span search mechanism of cuckoo search algorithm (CSA) into an improved cuckoo search algorithm (ICSA). Specifically, the expression of the ICSA was constructed, the coding format was designed for the minimum makespan of the JSP, and the flow of the ICSA was presented in details. The proposed algorithm was demonstrated as accurate and effective through numerical simulations. The research findings shed new light on the research into the JSA algorithms and the efficient scheduling in the actual production process [1-5].

## 2. PROBLEM DESCRIPTION

The JSP can be described as follows. Suppose  $n$  jobs are being processed on  $m$  machines. Let  $O_{ij}$  be the processing of job  $i$  on machine  $j$ . The corresponding processing time is denoted as  $T_{ij}$ . For each job, the sequence of processing machines is predetermined. In addition to machine sequence, the single machine is subjected to several constraints. For example, some jobs can only be processed on a certain machine, all machines are continuous, and the buffer zones between the machines are unlimited. If all jobs share the same processing sequence, the JSP can be converted into a flow-shop scheduling problem (FSP). If the same processing sequence applies to all jobs on the same machine, the FSP can be further transformed into replaceable FSP.

The JSP has been proved to be a typical NP-hard problem. The complexity is mainly reflected in the following aspects. First, the JSP involves mixed-integer nonlinear programming, and its solution is often coded in both continuous and discrete variables; Second, the number of possible permutations and combinations rises in a geometrical manner; Third, the lack of global information tends to lure any algorithm into the local optimal trap, making it extremely hard to find the global optimal solution [6-12].

To solve the JSP, it is often necessary to obtain certain optimal values, i.e. the solutions of the objective function. In general, there are three objectives of JSP optimization: to minimize the maximum makespan of all jobs, to minimize the tardiness of all jobs, and to minimize the value of penalty function, which considers the minimum earliness or tardiness of all jobs. Here, the minimum makespan of all jobs is taken as the optimization objective. In other words, all jobs should be optimized without considering the production cost, such that the maximum makespan can be minimized and the efficiency of the production line can be enhanced across the board. The minimization of the maximum makespan can be expressed by the general model below:

$$\min \max_{1 \leq g \leq m} \left\{ \max_{1 \leq i \leq n} T_{ig} \right\} \quad (1)$$

Constraints:

$$T_{ig} - p_{ig} + M(1 - a_{ihg}) \geq T_{ih} \quad (i = 1, 2, \dots, n; h, g = 1, 2, \dots, m) \quad (2)$$

$$T_{jg} - T_{ig} + M(1 - a_{ijg}) \geq p_{jg} \quad (i, j = 1, 2, \dots, n; g = 1, 2, \dots, m) \quad (3)$$

$$C_{ig} \geq 0 \quad (i = 1, 2, \dots, n; g = 1, 2, \dots, m) \quad (4)$$

$$x_{ijg} = 0, 1 \quad (i, j = 1, 2, \dots, n; g = 1, 2, \dots, m) \quad (5)$$

Among them, Eq. (1) is the objective function, i.e. the minimization of the maximum makespan of all jobs; Eq. (2) is the operation sequence of any job on any machine, i.e. the technical constraint on the jobs; Eq. (3) is the sequence of machines of each job through the production line; Eq. (4) requires that the makespan of a job at any machine is greater than zero; Eq. (5) is the value of the binary selection variable: job  $i$  is not processed on machine  $j$  if  $x_{ijg} = 0$ , and is processed on that machine if  $x_{ijg} = 1$ .

## 3. ICESA

Inspired by the Lévy flight mechanism, the CSA is a combination of the GA and PSO. This search strategy is known for its high flexibility, because it can control population aggregation and diversity with a few parameters. However, the previous studies on the CSA concentrate heavily on the optimization of continuous space. There is little report on multidimensional discrete-continuous search space. Therefore, the standard CSA cannot be directly applied to solve our model.

There are two goals for the improvement of the standard CSA: the improved algorithm must avoid the local optimum trap and realize large-scale global search, and acquire the ability to find sufficiently accurate solution in the local domain. To achieve the goals, the Lévy flight mechanism of the CSA was combined with the group search principle of the PSA into a new algorithm based on the standard CSA. The improvement was carried out through Eqs. (6) and (7) below.

$$v_i^{(t+1)} = w \cdot v_i^{(t)} + c \cdot r_1 \cdot (p_g^{(t)} - x_i^{(t)}) + \alpha \cdot step \quad (6)$$

$$x_i^{(t+1)} = \text{round}(x_i^t + v_i^{(t+1)}) \quad (7)$$

where  $v_i^{(t+1)}$  is the velocity in the PSA after the  $(t+1)^{\text{th}}$  iteration;  $w$  is the inertia weight factor in the PSA;  $v_i^{(t)}$  is the velocity before the  $t^{\text{th}}$  iteration;  $c$  is the social learning factor in the PSA;  $r_1$  is a random real number between 0 and 1;  $p_g^{(t)}$  is the local optimal solution of all populations after the  $t^{\text{th}}$  iteration;  $x_i^{(t)}$  is the flight position of particle  $i$  after the  $t^{\text{th}}$  iteration;  $\alpha$  is the step length factor of the CSA;  $step \sim L^D(\lambda)$ , with  $L^D(\lambda)$  obeying the Lévy random distribution in the  $D^{\text{th}}$  dimension;  $x_i^{(t+1)}$  is the flight position of particle  $i$  after the  $(t+1)^{\text{th}}$  iteration;  $\text{round}$  is the rounded output [13-21].

### 3.1 Discrete performance tests

The ICSA received discrete performance tests before being applied to solve our model. The tests aim to verify if the new algorithm can track the global optimum or have the global search ability amidst parameter changes, and to roughly determine the variation in the algorithm performance with parameter changes.

For any discrete combinatorial optimization problem, an algorithm must be able to optimize 1D discrete points before finding the optimal solution to the problem. Considering this, 1D discrete points optimization tests were designed to disclose how the improved algorithm changes with the parameters.

Let  $x \in [1, 2, 3, 4, 5]$  be a set of 1D discrete points. Suppose the global optimal solution is  $p_g = 3$  and the initial solution is  $x_0 = 1$ . Three parameters  $w$ ,  $c$  and  $\alpha$  should be taken into account. During the tests, the test parameter should be increased gradually, while the other two parameters should be constant, so that the tester can observe the effect of the test parameter on the algorithm performance and its ability to track global optimal solution and achieve global random search.

#### (1) Discrete test on step length factor $\alpha$

In standard BCA, the value of step length factor  $\alpha$  is positively correlated with the scale of the problem. The larger the step length factor, the stronger the global search ability; the smaller the step length factor, the stronger the local search ability. Here, a discrete test is designed to disclose the effect of step length factor  $\alpha$  on the IBCA. The test parameters were set as follows:  $w = 0.9$ ,  $c = 2$ ,  $p_g = 3$  (global optimal solution),  $x_0 = 1$  (initial solution),  $popsiz = 30$  (number of individuals in the population). Then, the value of  $\alpha$  was increased step by step to reveal its impact on algorithm performance (Fig. 1).

As shown in Fig. 1, when the step length factor  $\alpha$  was small, the position of  $x$  in most of the 500 iterations converged to the global optimal solution, owing to the followability of the social learning part of the particle swarm; with the increase in the step length factor  $\alpha$ , the Lévy flight mechanism of the CSA came into play, leading to significant oscillation of the position of  $x$  around the global optimal solution. This process can be understood as the gradual transition from local fine search to global search. Overall, the oscillation of  $x$  position is relatively frequent near the global optimal solution, due to the followability of the PSA. This means the search is emphasized on local neighbourhoods. To sum up, the step length factor  $\alpha$  is related to the mutation degree of the algorithm. The greater the factor, higher the

mutation degree during the variation, and the stronger the global search ability; the smaller the factor, lower the degree of mutation, the stronger the followability of the global optimal solution, i.e. the easier the convergence.

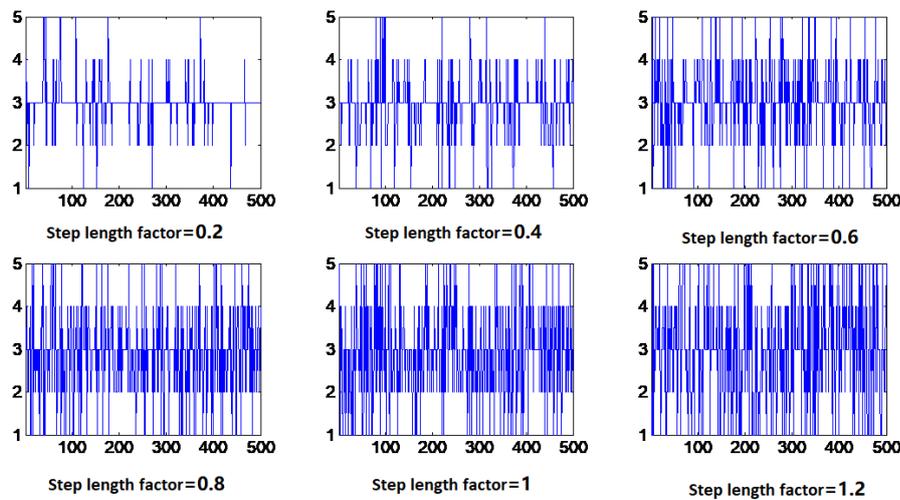


Figure 1: Discrete test on step length factor  $\alpha$ .

(2) Discrete test on velocity inertia factor  $w$

In the PSA, the value of velocity inertia factor  $w$  is positively correlated with the search span. The larger the velocity inertia factor, the stronger the global search ability; the smaller the velocity inertia factor, the stronger the local search ability. Here, a discrete test is designed to disclose the effect of velocity inertia factor  $\alpha$  on the IBCA. The test parameters were set as follows:  $p_g = 3$  (global optimal solution),  $x_0 = 1$  (initial solution),  $c = 2$ ,  $\alpha = 0.6$ ,  $popsiz = 30$  (number of individuals in the population). Then, the value of  $w$  was increased step by step to reveal its impact on algorithm performance (Fig. 2).

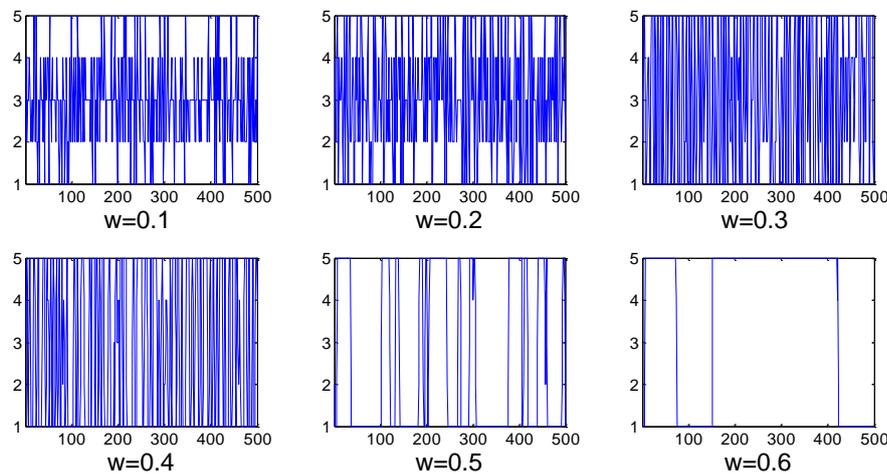


Figure 2: Discrete test on velocity inertia factor  $w$ .

As shown in Fig. 2, the increase of the velocity inertia factor  $w$  from 0.1 to 0.6 aggravated the random oscillation and uncertain of the system. The algorithm exhibited random changes after  $w$  surpassed 0.3, and the system lost its stability after  $w$  exceeded 0.5. It can be seen that  $w$  can add to the randomness of the system. The greater the value of  $w$ , the larger the randomness of the system. For a specific object, the system is no longer stable when  $w$  is greater than a certain threshold. Hence, the global random search ability of the algorithm can be enhanced through proper use of the pulse value of  $w$ , so that the algorithm can avoid the local optimal value.

(3) Discrete test on social learning factor  $c$ 

In the PSA, the value of social learning factor  $c$  is related to the followability of the global optimal solution. The factor amplifies the negative feedback on the distance to the global optimal solution. If the social learning factor has a large value, the solution of the algorithm after each iteration will oscillate violently around the global optimal solution, and the search will focus on the neighbourhoods far away from the global optimal solution; if the social learning factor has a small value, the solution of the algorithm after each iteration will oscillate slightly around the global optimal solution, and the search will focus on the neighbourhoods near the global optimal solution. Here, a discrete test is designed to disclose the effect of social learning factor  $c$  on the IBCA. The test parameters were set as follows:  $p_g = 3$  (global optimal solution),  $x_0 = 1$  (initial solution),  $w = 0.1$ ,  $\alpha = 0.6$ ,  $popsiz = 30$  (number of individuals in the population). Then, the value of  $c$  was increased step by step to reveal its impact on algorithm performance (Fig. 3).

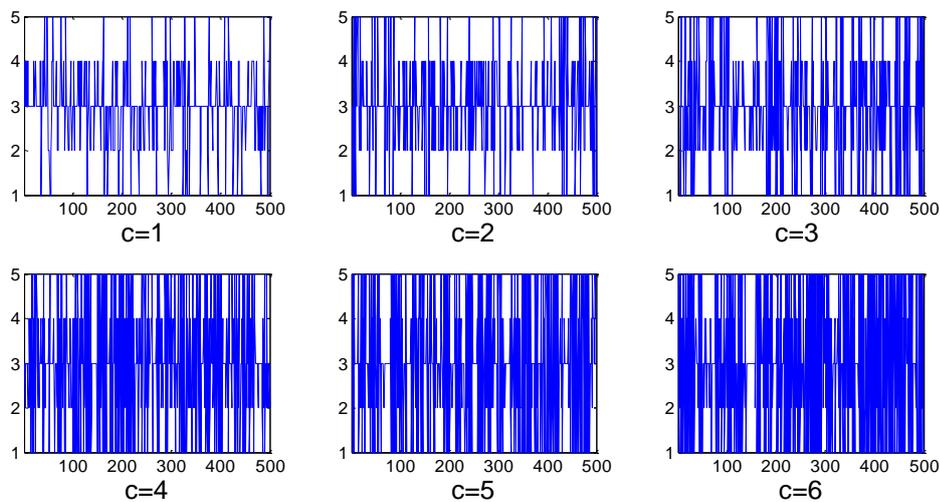


Figure 3: Discrete test on social learning factor  $c$ .

As shown in Fig. 3, the local search of the algorithm gradually shifted to global search with the gradual increase of the social learning factor  $c$  from 1 to 6, adding to the degree of divergence of the system. Therefore, the social learning factor can strike a balance between the aggregation and diversity of the algorithm. The algorithm focuses on the local fine search when  $c$  has a small value, and global random search when  $c$  has large value.

In summary, there are three control parameters in the ICSA, namely, step size factor  $\alpha$ , velocity inertia factor  $w$  and social learning factor  $c$ . These parameters have different effects on the search performance of the algorithm. Specifically, the step size factor  $\alpha$  is positively correlated with the mutation degree of the algorithm. The velocity inertia factor  $w$  is related to the randomness of the algorithm: the greater the value of  $w$ , the higher the randomness of the search and the stronger the instability of the algorithm; the inverse is also true. The social learning factor  $c$  signifies strong global search ability if it has a high value, and strong local search ability if it has a small value. From the above summary, the following strategies were developed for parameter setting:

- First, the step length factor  $\alpha$  should be increased properly in the initial phase to enhance the mutation degree of the algorithm; the factor should be gradually reduced in the later phase for the convergence of the algorithm.
- Second, the algorithm should be adjusted via the pulse value of the velocity inertia factor  $w$  throughout the search process, so as to bolster the random search ability.
- Third, the social learning factor  $c$  should be increased properly in the initial phase and gradually reduced in the later phase.

## 4. ICOSA-BASED SOLUTION OF THE JSP

### 4.1 Solution encoding

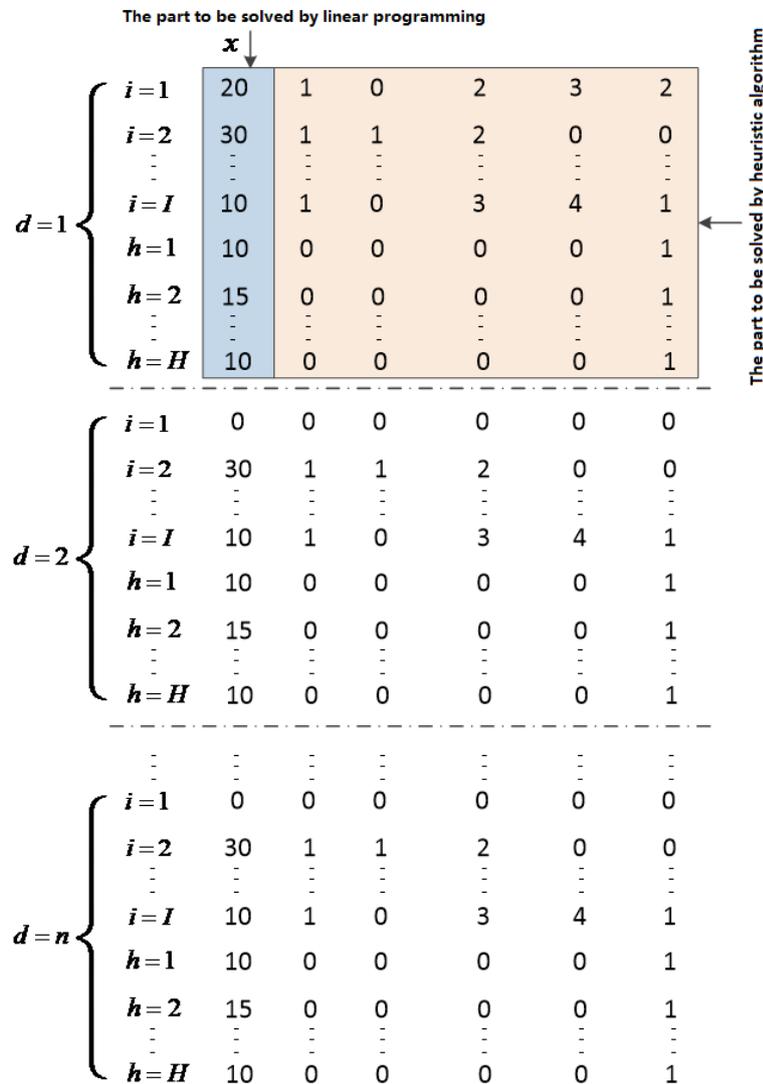


Figure 4: Coded format of the solution.

Fig. 4 shows the coded format of a solution of the ICOSA. The solution consists of  $D$  submatrices, each of which contains  $I+H$  rows. Each submatrix can be further divided into two parts: the part to be solved by linear programming and the part to be solved by heuristic algorithm. Thus, the proposed algorithm is a matheuristics algorithm rather than a pure metaheuristic algorithm. The latter has many disadvantages in solving continuous variables, e.g. high randomness, numerous iterations and poor accuracy. As a result, the discrete variables were heuristically solved through metaheuristic algorithm (the ICOSA) and then the continuous variables were accurately solved by linear programming.

### 4.2 Steps of the ICOSA

The ICOSA differs from the CSA in the steps to generate new solutions. In the standard CSA, the new solutions can be generated in two ways: Lévy flight mechanism and the selection based on discovery probability  $p_a$ . By contrast, the ICOSA simplifies the computing procedure and enhances the operation efficiency through the integration of Lévy flight mechanism with some functions of the PSA. Through the adjustment of its three-parameter kernel, the ICOSA

can control the mutation degree and randomness of the algorithm, and keep the balance between global search and local search. The flow of the ICSEA is illustrated in Fig. 5 below.

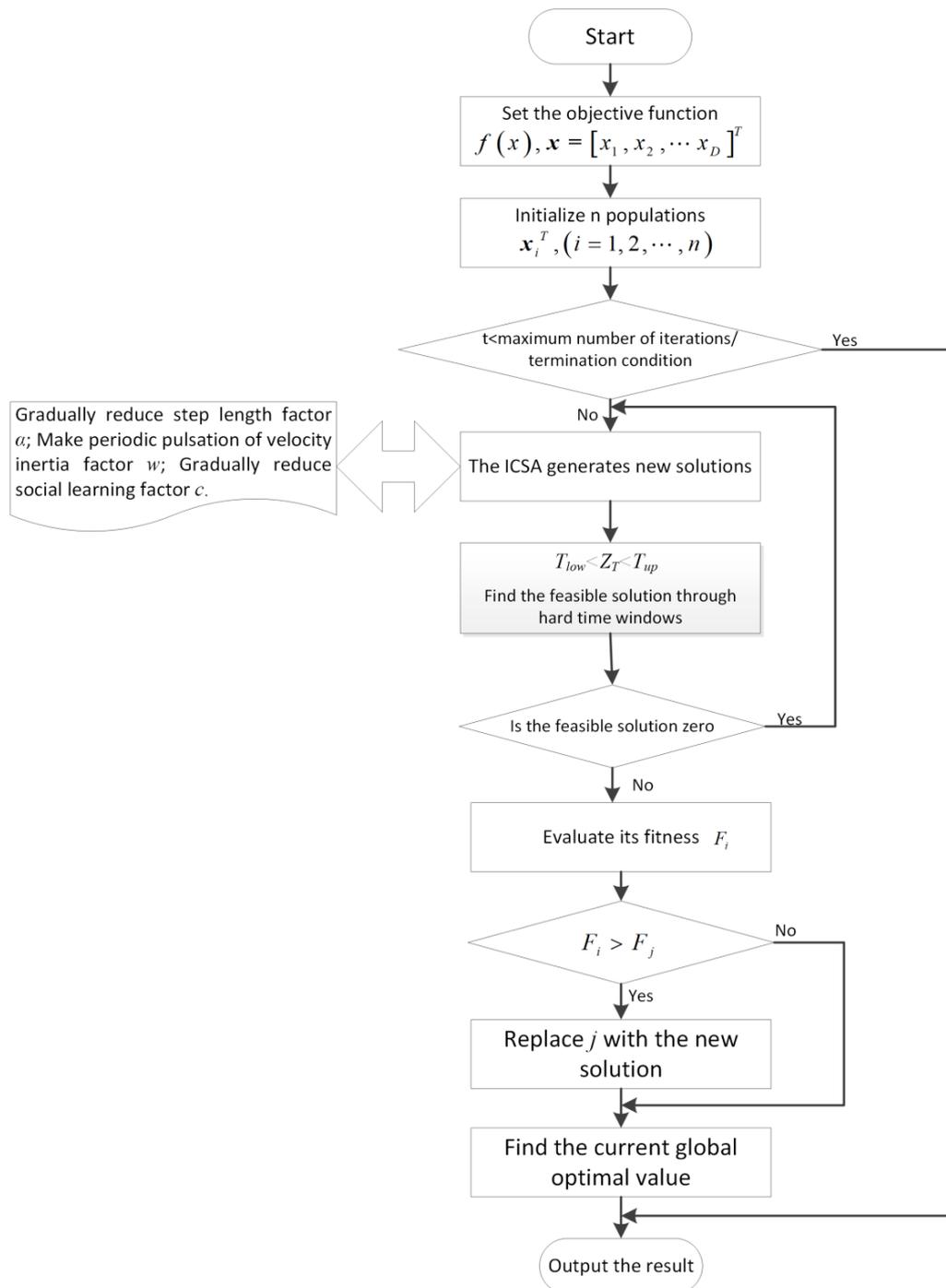


Figure 5: Flow of the ICSEA.

### 4.3 Simulation experiments

#### (1) Comparison with other algorithms

Over the years, many typical case libraries have been developed for the JSP through the study on similar problems. Here, the LA type library proposed by Lawrence in 1984 is adopted for contrasting the proposed ICSEA algorithm with the PSA and the GA. The simulation experiments were conducted in Matlab on a computer (Windows 10; 2.8 GHz Intel Core i5 Dual Core, 4 GB RAM). The experimental parameters were configured as follows.

In the ICSA, the number of nests  $n = 30$  and the discovery probability  $Pa = 0.25$ . Considering the wide search range for the global optimal solution in the initial phase, the step length factor  $\alpha$ , velocity inertia factor  $w$  and social learning factor  $c$  were set to relatively high values. During the operation, the values of these parameters were gradually reduced to lower the mutation degree of the algorithm and elevate the importance of local search; in the later phase, the three parameters were minimized to ensure the convergence.

In the PSA, the number of populations  $n = 30$ , the learning factor  $c_1 = 0.75$ , the social factor  $c_2 = 0.24$  and the inertia factor  $w = 0.56$ .

In the GA, the number of populations  $n = 30$ , the crossover probability  $P_c = 0.9$  (single-point crossover); the mutation probability  $P_m = 1/n$  (bit-by-bit mutation).

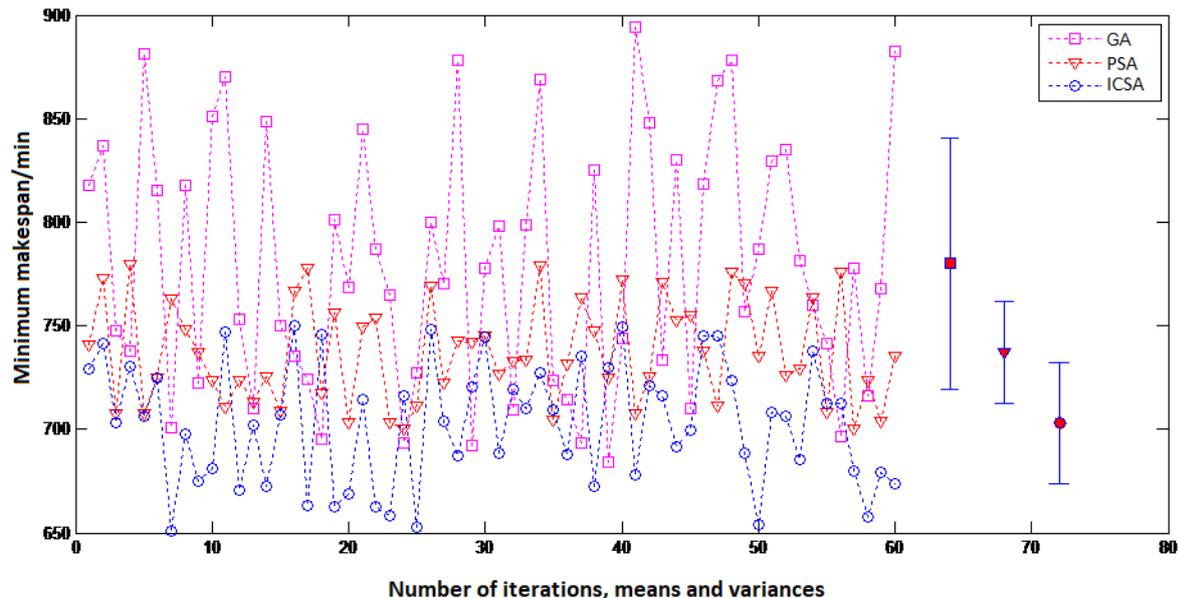


Figure 6: Comparison between different algorithms.

Fig. 6 compares the means and variances of the three algorithms after 60 iterations. The GA had the worst performance. Its means and variances were greater than those of the other two algorithms, indicating that the mean makespan is long and the results differ greatly between iterations. The poor stability of the algorithm can be attributed to the individual coding differences. The PSA exhibited a moderate performance. The medium means and low variances show that the algorithm is relatively stable. Besides, the results differed slightly between iterations. However, the PSA failed to overcome its bottleneck and achieve further optimization. The proposed ICSA boasts advantages over the other two algorithms in that it acquired both the global optimal solution and the local optimal solution. The leading edge comes from the integration between the local fine search of the PSA and the large-span search of the standard CSA.

## (2) Analysis of algorithm convergence

Based on the comparative analysis, a convergence analysis of the ICSA was specially designed for our model. During the analysis, the ICSA was run independently on our model 30 times (100 iterations each time). The evolution diagrams were superimposed into Fig. 7. According to Fig. 7, the ICSA's convergence curves were basically the same, ranging between 660 min and 710 min. Being a swarm intelligence algorithm, the ICSA can find some comparatively good plans in the initial phase; with the increase in the number of iterations, more and more better plans for makespan minimization are discovered by individuals in the populations and shared across the populations. In this way, the algorithm can look for even better optimization plans.

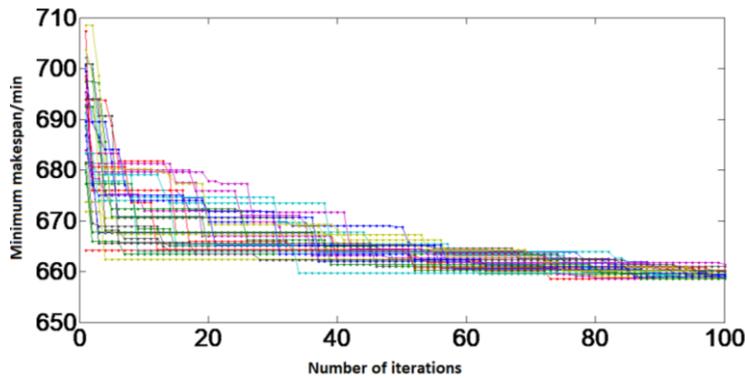


Figure 7: Convergence analysis of the ICSA.

## **5. CONCLUSIONS**

This paper makes an in-depth exploration into the JSP. First, the related literature was reviewed in details to obtain the status and problems of JSP research. Then, the author decided to develop an ideal solution to the minimum total makespan model of typical JSP. Then, the local search mechanism of the PSA and the large-span search principle of standard CSA were combined into an ICSA, which is capable of both local search and global search. Targeting the minimum makespan model, the algorithm structure was designed as a mixed integer nonlinear structure program involving both continuous and discrete variables. Next, the parameters of the ICSA were subjected to discrete simulations to determine the value range of the step length factor  $\alpha$ , velocity inertia factor  $w$  and social learning factor  $c$ . This is because the optimal status of the algorithm can be achieved through proper setting of the here parameters. After that, several simulation experiments were carried out on the LA type typical library proposed by Lawrence, and the stability and accuracy of the ICSA was contrasted with those of the PSA and the GA based on the means and variances in multiple iterations. After the comparison, a convergence analysis of the ICSA was specially designed for our model. The results demonstrate that the ICSA provides a better tool for solving the JSP than other algorithms. The research findings lay a solid theoretical basis for the JSP in the actual production process.

## **ACKNOWLEDGEMENTS**

This work is supported by National Natural Science Foundation of China (Grant No: 51667017), Key Research Projects of Tibet Autonomous Region for Innovation and Entrepreneur (Grant No. Z2016D01G01/01).

## **REFERENCES**

- [1] Pinedo, M. L. (2012). *Scheduling: Theory, Algorithms, and Systems*, 4<sup>th</sup> edition, Springer Science+Business Media, LLC, New York
- [2] Wang, Y.; Zhu, L.-B.; Wang, J.-W.; Qiu, J.-F. (2015). An improved social spider algorithm for the flexible job-shop scheduling problem, *Proceedings of the 2015 International Conference on Estimation, Detection and Information Fusion*, 157-162, doi:[10.1109/ICEDIF.2015.7280181](https://doi.org/10.1109/ICEDIF.2015.7280181)
- [3] Wang, L.; Zhou, G.; Xu, Y.; Wang, S.-Y.; Liu, M. (2012). An effective artificial bee colony algorithm for the flexible job-shop scheduling problem, *The International Journal of Advanced Manufacturing Technology*, Vol. 60, No. 1-4, 303-315. doi:[10.1007/s00170-011-3610-1](https://doi.org/10.1007/s00170-011-3610-1)
- [4] Meijer, D. K. F.; Geesink, H. J. H. (2017). Consciousness in the universe is scale invariant and implies an event horizon of the human brain, *NeuroQuantology*, Vol. 15, No. 3, 41-79, doi:[10.14704/nq.2017.15.3.1079](https://doi.org/10.14704/nq.2017.15.3.1079)

- [5] Singh, S.; Singh, K. P. (2015). Cuckoo search optimization for job shop scheduling problem, *Proceedings of Fourth International Conference on Soft Computing for Problem Solving. Advances in Intelligent Systems and Computing*, Vol. 335, 99-111, doi:[10.1007/978-81-322-2217-0\\_9](https://doi.org/10.1007/978-81-322-2217-0_9)
- [6] Zhang, X.-H. (2018). Research on the evaluation and decision-making of product premium based on electroencephalography (EEG) testing technology, *NeuroQuantology*, Vol. 16, No. 3, 68-74, doi:[10.14704/nq.2018.16.3.1199](https://doi.org/10.14704/nq.2018.16.3.1199)
- [7] Zheng H.-Q.; Zhou, Y.-Q. (2012). A novel cuckoo search optimization algorithm base on Gauss distribution, *Journal of Computational Information Systems*, Vol. 8, No. 10, 4193-4200
- [8] Zhou, Y.-Q.; Zheng, H.-Q.; Luo, Q.-F.; Wu, J.-Z. (2013). An improved cuckoo search algorithm for solving planar graph coloring problem, *Applied Mathematics & Information Sciences*, Vol. 7, No. 2, 785-792, doi:[10.12785/amis/070249](https://doi.org/10.12785/amis/070249)
- [9] Zhang, H.-C.; Qu, Y.-H.; Yang, A.; Meng, F.-Q.; Ma, H.-F. (2015). Numerical simulation of refining rolling process based on adaptive BEM parallel algorithm, *Academic Journal of Manufacturing Engineering*, Vol. 13, No. 4, 81-87
- [10] Huang, X.-R.; Dai, W.; Du, B.-G. (2016). Resource-constrained project scheduling problem for large complex equipment: a hybrid approach using Pareto genetic algorithm and interval-valued intuitionistic fuzzy sets, *Academic Journal of Manufacturing Engineering*, Vol. 14, No. 1, 12-21
- [11] Ong, P. (2014). Adaptive cuckoo search algorithm for unconstrained optimization, *The Scientific World Journal*, Vol. 2014, Paper 943403, 8 pages, doi:[10.1155/2014/943403](https://doi.org/10.1155/2014/943403)
- [12] Nguyen, T. T.; Vo, D. N. (2015). Modified cuckoo search algorithm for short-term hydrothermal scheduling, *International Journal of Electrical Power & Energy Systems*, Vol. 65, 271-281, doi:[10.1016/j.ijepes.2014.10.004](https://doi.org/10.1016/j.ijepes.2014.10.004)
- [13] Thammano, A.; Phuang, A. (2013). A hybrid artificial bee colony algorithm with local search for flexible job-shop scheduling problem, *Procedia Computer Science*, Vol. 20, 96-101, doi:[10.1016/j.procs.2013.09.245](https://doi.org/10.1016/j.procs.2013.09.245)
- [14] Nouri, M.; Jemai, A.; Ammari, A. C.; Bekrar, A.; Niar, S. (2013). An effective particle swarm optimization algorithm for flexible job-shop scheduling problem, *Proceedings of 2013 International Conference on Industrial Engineering and Systems Management (IESM)*, 6 pages
- [15] Zaccone, R.; Sacile, R.; Fossa, M. (2017). Energy modelling and decision support algorithm for the exploitation of biomass resources in industrial districts, *International Journal of Heat and Technology*, Vol. 35, No. 1 (Special issue), S322-S329, doi:[10.18280/ijht.35Sp0144](https://doi.org/10.18280/ijht.35Sp0144)
- [16] Liang, C. H.; Zeng, S.; Li, Z. X.; Yang, D. G.; Sherif, S. A. (2016). Optimal design of plate-fin heat sink under natural convection using a particle swarm optimization algorithm, *International Journal of Heat and Technology*, Vol. 34, No. 2, 275-280, doi:[10.18280/ijht.340217](https://doi.org/10.18280/ijht.340217)
- [17] Sen, G. D.; Sharma, J.; Goyal, G. R.; Singh, A. K. (2017). A multi-objective PSO (MOPSO) algorithm for optimal active power dispatch with pollution control, *Mathematical Modelling of Engineering Problems*, Vol. 4, No. 3, 113-119, doi:[10.18280/mmep.040301](https://doi.org/10.18280/mmep.040301)
- [18] Zhou, W.; Bu, Y.-P.; Zhou, Y.-Q. (2014). Combining CA and PSO to solve flexible job shop scheduling problem, *The 26<sup>th</sup> Chinese Control and Decision Conference (2014 CCDC)*, 1031-1036, doi:[10.1109/CCDC.2014.6852316](https://doi.org/10.1109/CCDC.2014.6852316)
- [19] Yang, X.-S.; Deb, S. (2009). Cuckoo search via Lévy flights, *World Congress on Nature & Biologically Inspired Computing (NaBIC 2009)*, 210-214, doi:[10.1109/NABIC.2009.5393690](https://doi.org/10.1109/NABIC.2009.5393690)
- [20] Behnke, D.; Geiger, M. J. (2012). *Test instances for the flexible job shop scheduling problem with work centers*, Research Report RR-12-01-01, Helmut-Schmidt University, Hamburg
- [21] Mastrolilli, M.; Gambardella, L. M. (2000). Effective neighbourhood functions for the flexible job shop problem, *Journal of Scheduling*, Vol. 3, No. 1, 3-20, doi:[10.1002/\(SICI\)1099-1425\(200001/02\)3:1<3::AID-JOS32>3.0.CO;2-Y](https://doi.org/10.1002/(SICI)1099-1425(200001/02)3:1<3::AID-JOS32>3.0.CO;2-Y)