**Journal of Computing and Electronic Information Management**
ISSN: 2413-1660 | Vol. 12, No. 3, 2024

# An Improved Harris Hawks Optimization Algorithm for Solving the Permutation Flow Shop Scheduling Problem

**Xiaoyu Li ***

Business School, Sichuan University, Chengdu, China

* **Corresponding author**

**Abstract:** In this paper, an improved Harris Hawks optimization algorithm is proposed to solve the permutation flow shop scheduling problem with the objective of minimizing the completion time. Logistic chaotic mapping and inverse learning strategy are used to generate a high-quality initial population. A golden sine algorithm is introduced to improve the position update method. A nonlinear escape energy factor and adaptive t-distribution strategy are introduced to solve the problem of imbalance between the exploration and exploitation phases of the HHO algorithm. The effectiveness of the improved Harris Hawks optimization algorithm is verified by testing it on the Reeves benchmark test set and comparing it with other algorithms.

**Keywords:** Permutation flow shop scheduling problem; Harris Hawks optimization; Minimize makespan.

## 1. Introduction

The permutation flow shop scheduling problem (PFSP) is a typical class of combinatorial optimization problems, the core of which is to reasonably allocate resources and arrange work processes to achieve the optimization of production indexes under the premise of satisfying specific production constraints and demands. The PFSP has been proved to belong to the NP-hard problem, and with the increasing scale of the problem, the requirements on the solution algorithms in terms of complexity and efficiency have been increasing. Therefore, the study of efficient intelligent algorithms to solve such scheduling problems is a hot research topic in the field of scheduling.

Metaheuristic algorithms are optimization methods inspired by natural phenomena that explore large search spaces to find solutions to complex problems，It is an efficient method for solving shop floor scheduling. The commonly used meta-heuristic algorithms in PFSP are Particle Swarm Optimization algorithm, Genetic Algorithm, Ant Colony Algorithm and so on.

Harris Hawks optimization, a novel meta-heuristic algorithm proposed by Heidari et al. in 2019, has been applied in many fields such as information science, data analysis, engineering design, and wireless communication. Many researchers have improved the HHO algorithm to enhance its performance. Mahapatra et al. introduced an inverse learning mechanism and a stochastic perturbation factor to improve the exploration capability of the algorithm Qu et al. incorporated a nonlinear energy factor of chaotic disturbances into the HHO and introduced an information exchange mechanism to enhance the performance of the algorithm. Gupta et al. proposed an adversarial learning based HHO to make the algorithm jump out of the local optimal solution and improve the global search capability.

In this paper, an Improved Harris Hawks optimization (IHHO) algorithm is proposed for solving PFSP. SPV method is adopted for coding, Logistic mapping and inverse learning strategy are used for population initialization, and Harris Hawks position update strategy is improved by incorporating the golden sine algorithm to enhance the global search capability. Introducing a nonlinear escape energy factor and introducing an adaptive t-distribution strategy balances the algorithm exploration and development capabilities. Comparison experiments of IHHO with other three algorithms are conducted to verify the effectiveness of the algorithm.

## 2. Permutation flow shop scheduling problem

The permutation flow shop scheduling problem (PFSP) is a classic problem in the field of flow shop scheduling. It is a simplified form of the actual production scheduling problem and is an NP-hard problem. In PFSP, there are n workpieces to be processed on m machines in the same order. Each machine can only process one workpiece at any time, and each workpiece can only be processed on one machine at the same time. The optimization goal is to minimize the makespan.

p(i,k) represents the processing time of workpiece i on machine k, C(i,k) represents the completion time of workpiece i on machine k, and represents the makespan. The PFSP mathematical model is as follows:

$$C(1,1) = p(1,1) \tag{1}$$

$$C(i,1) = C(i-1,1) + p(i,1), i = 2,3,\cdots,n \tag{2}$$

$$C(1,j) = C(1,k-1) + p(1,k), k = 2,3,\cdots,m \tag{3}$$

$$C(i,k) = \max\{C(i-1,k) + C(i,k-1)\} + p(i,k)$$
$$i = 2,3,\cdots,n; k = 2,3,\cdots,m \tag{4}$$

$$C_{max} = \max_{i=1,2,\cdots,n} C(i,m) \tag{5}$$

Minimizing makespan:

$$f = \min\{C_{max}\} \tag{6}$$

## 3. Improved harris hawk optimization

### 3.1. Basic Harris Hawks optimization

The Harris Hawks optimization consists of three phases: exploration, transition from exploration to exploitation, and exploitation.

In the exploration phase, the HHO randomly generates initial population individuals. A random number q is generated as a probability value, and the search strategy used is decided based on the size of the probability value in relation to 0.5. The formula is as follows:

$$x_i^{t+1} = \begin{cases} x_{rand}^t - r_1 \times \left|x_{rand}^t - 2r_2 x_i^t\right| & q \geq 0.5 \\ (x_{prey}^t - x_m^t) - r_3(lb + r_4(ub - lb)) & q < 0.5 \end{cases} \quad (7)$$

where t is the current number of iterations, $x_i^{t+1}$ is the individual position at the t+1st iteration, $x_i^t$ is the current individual position, $x_{rand}^t$ is the random individual position, $x_{prey}^t$ is the current optimal individual position, $x_m^t$ is the average position of the current population, $r_1$、$r_2$、$r_3$、$r_4$ and q are random numbers within (0,1), lb and ub are the upper and lower limits of the search range, N is the population size.

$$x_m^t = \frac{1}{N}\sum_{i=1}^{N} x_i^t \quad (8)$$

During the transition phase from exploration to exploitation, the HHO is able to dynamically transition from the exploration phase to the exploitation phase based on the change in the escape energy E of the prey.

$$E = 2 \times E_0 \cdot (1 - \frac{t}{T}) \quad (9)$$

where T is the maximum number of iterations and is the initial escape energy, a random number between (-1,1).

During the exploitation phase, depending on the random number r and the escape energy E, the Harris hawks adopt four different strategies to encircle their prey.

#### 3.1.1. Soft besiege

When $|E| \geq 0.5$ and $r \geq 0.5$, The prey still has more energy to try to escape and a soft besiege strategy is adopted. The position update formula is as follows:

$$x_i^{t+1} = \Delta x^t + E \cdot \left|J \cdot x_{prey}^t - x_i^t\right| \quad (10)$$

$$\Delta x^t = x_{prey}^t - x_i^t \quad (11)$$

Where, $\Delta x^t$ is the distance between the current individual position $x_i^t$ and the prey position $x_{prey}^t$, J is the randomized jump strength of the prey,$J\epsilon(0,2)$.

#### 3.1.2. Hard besiege

When $|E| < 0.5$ and $r \geq 0.5$, The prey is too exhausted to escape and the Harris hawks launch a fierce attack. The location update formula is below:

$$x_i^{t+1} = x_{prey}^t - E \cdot |\Delta x^t| \quad (12)$$

#### 3.1.3. Soft besiege with progressive rapid dives

When $|E| \geq 0.5$ and $r < 0.5$, the prey has enough energy to get away from being surrounded, Harris hawks adopt smarter roundup strategy. The location update formula is below:

$$\begin{cases} Y = x_{prey}^t - E \cdot \left|J \cdot x_{prey}^t - x_i^t\right| \\ Z = Y + S \cdot levy(Q) \end{cases} \quad (13)$$

where S is a Q-dimensional random vector, Q is the dimension of the optimization problem, and levy () is the flight function, which is calculated as follows:

$$levy(x) = 0.01 \times \frac{\mu \cdot \sigma}{|v|^{\frac{1}{\beta}}} \quad (14)$$

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{(\frac{\beta-1}{2})}}\right)^{\frac{1}{\beta}} \quad (15)$$

$$\Gamma(y) = (y-1)! \quad (16)$$

where u, v are random numbers between (0,1), and $\beta$ is a constant 1.5.

$$x_i^{t+1} = \begin{cases} Y, & f(Y) < f(x_i^t) \\ Z, & f(Z) < f(x_i^t) \end{cases} \quad (17)$$

#### 3.1.4. Hard besiege with progressive rapid dives

When $|E| < 0.5$ and $r < 0.5$, the prey's escape energy is low, and Harris hawks adopt a hard besiege with progressive rapid dives strategy. The location update formula is below:

$$\begin{cases} Y = x_{prey}^t - E \cdot \left|J \cdot x_{prey}^t - x_m^t\right| \\ Z = Y + S \cdot levy(Q) \end{cases} \quad (18)$$

$$x_i^{t+1} = \begin{cases} Y, & f(Y) < f(x_i^t) \\ Z, & f(Z) < f(x_i^t) \end{cases} \quad (19)$$

## 3.2. Coding Method

HHO is usually applicable to the solution of continuous problems and cannot be directly solved for discrete optimization problems such as shop scheduling. Therefore, the Smallest Position Value (SPV) proposed by Tasgetiren et al. can be used to efficiently map continuous position values to discrete sequences of workpieces.

The SPV rule works by selecting the workpiece corresponding to the smallest value in the successive position variables as the next workpiece to be machined, then removing it from the list and repeating the process until all workpieces have been arranged. Table 1 gives an example of using SPV rules to convert a set of consecutive values $x_i = [0.43, 2.21, 0.52, 2.87, 0.14, 2.64, 1.03]$ representing the locations of Harris hawks in a 7-dimensional space to an artifact ordering $\pi_i = [5, 1, 3, 7, 2, 6, 4]$.

**Table 1.** Example

| j | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| $x_{i,j}$ | 0.43 | 2.21 | 0.52 | 2.87 | 0.14 | 2.64 | 1.03 |
| $\pi_{i,j}$ | 5 | 1 | 3 | 7 | 2 | 6 | 4 |

## 3.3. Population initialization

The initial population of the basic HHO is generated by randomization, which can lead to a population that may be extremely unevenly distributed in space. Therefore, Logistic chaotic mapping and inverse learning strategy is used to generate the initialized population.

The steps for initializing the population based on Logistic mapping and inverse learning strategy are:

Step 1: Generate the initial solution $X_i$ by logistics mapping.

Step 2: Reverse solution $X_{iobl}$ obtained by reverse learning strategy for each solution $X_i$ in the initial population.

Step 3: Calculate the fitness of the initial individual $X_i$ and the reverse individual $X_{iobl}$. Select the individuals with high fitness to form a new initial population to improve the quality of the initial population.

## 3.4. Golden Sine Algorithm

The Golden Sine Algorithm (Gold-SA) combines the properties of the golden section coefficient and the sine function, using the special relationship between the sine function and the unit circle, the algorithm is able to realize the traversal of all points on the unit circle. The Gold-SA algorithm is introduced into the position update strategy in the global search phase of HHO to improve the global search capability of the algorithm.

The position update by incorporating the golden sine algorithm is as follows:

$$x_i^{t+1} = \begin{cases} x_i^t \times |\sin r_1| + r_2 \times \sin r_1 \times \left|m_1 x_{prey}^t - m_2 x_{prey}^t\right| & q \geq 0.5 \\ (x_{prey}^t - x_m^t) - r_3(lb + r_4(ub - lb)) & q < 0.5 \end{cases} \quad (20)$$

Where, both $r_1$ and $r_2$ are random numbers in $[0,2\pi]$ that determine the distance and direction that individual i moves during the iteration process, golden ratio $\tau = (1-\sqrt{5})/2$, $m_1 = -\pi + (1-\tau) \times 2\pi$, $m_2 = -\pi + \tau \times 2\pi$.

### 3.5. Nonlinear escape energy factor

In HHO, the escape energy of prey depends on three key parameters: $E_0$, t, T. The escape energy factor decreases linearly with the number of iterations, which fails to reflect the complex dynamic process of the roundup behavior of Harris hawks populations in nature. In this paper, a nonlinear decreasing factor of escape energy is introduced, which can better balance the exploration phase and exploitation phase.

The nonlinear escape energy factor is shown in Eq. 21 and the updated prey escape energy is shown in Eq. 22:

$$E_1 = 2 \times (1 - \sin\frac{\pi t}{2T} \times \frac{t}{T}) \quad (21)$$

$$E = E_1 \cdot E_0 \quad (22)$$

where $E_1$ is the energy factor at iteration number t, $E_0$ is the initial escape energy value, and is the random number in the [0,1].

### 3.6. Adaptive t-distribution strategy

The t-distribution is also known as the student distribution, and the probability density function of the t-distribution with n degrees of freedom is shown in Eq. 23:

$$p(x) = \frac{\Gamma(\frac{n+1}{2})}{\sqrt{n\pi}\cdot\Gamma(\frac{n}{2})} \cdot (1 + \frac{x^2}{2})^{-\frac{n+1}{2}}, -\infty < x < \infty \quad (23)$$

where n is the degree of freedom parameter and $\Gamma$ is the Gamma function.

The degree of freedom, n, affects the curve shape; the larger the value of n, the higher the middle of its curve. When n = 1, $t(n=1) \rightarrow C(0,1)$; When $n \rightarrow \infty$, $t(n=\infty) \rightarrow N(0,1)$, where C(0,1) represents the Cauchy distribution and N(0,1) represents the Gaussian distribution. With different degrees of freedom, the t-distribution can fully utilize the advantages of two different distributions, the Cauchy and Gaussian distributions

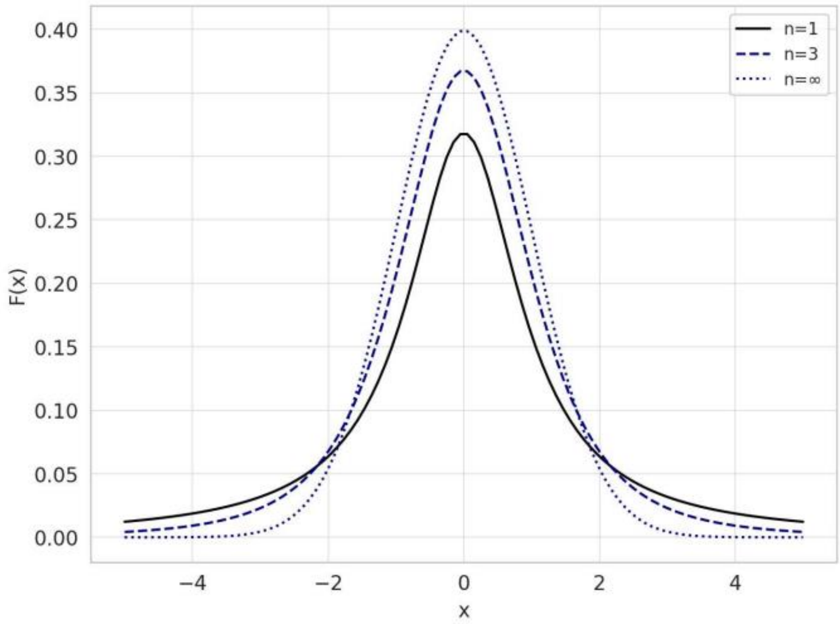


**Fig. 1** t-distribution

The t-distribution strategy is adjusted using the number of iterations as a parameter of the degrees of freedom to perturb the position of individual Harris hawks, and the t-distribution perturbation formula is as follows:

$$x_i^t = x_i + t(iter) \cdot x_i \quad (24)$$

where t(iter) denotes the t-distribution with degrees of freedom as the number of iterations.

## 4. Simulation experiment

To test the optimized performance of IHHO to solve PFSP, it is tested using Reeves test set. The experimental environment is a processor Intel(R) Core (TM) i7-9750H CPU with 2.59 GHz and 16.00 GB of RAM, implemented using Matlab programming language. The population size was set to 30, the number of iterations was set to 500, and the algorithms were run independently 20 times for each operator. Best Relative Error (BRE) and Average Relative Error (ARE) are used to evaluate the performance of the algorithms. The calculation formula is as follows:

$$BRE = \frac{C_{best} - C^*}{C^*} \times 100\% \quad (25)$$

$$ARE = \frac{\bar{C} - C^*}{C^*} \times 100\% \quad (26)$$

where $C_{best}$ and $\bar{C}$ are the best and average results obtained by the algorithm in 20 independent runs of an arithmetic case, respectively, and $C^*$ is the optimal solution known to the arithmetic case.

Tests were performed using the Reeves test set, and IHHO was compared to the Variable Parameters Quantum-inspired Evolutionary Algorithm (VP-QEA), Hybrid Differential Evolution (L-HDE), and Hybrid Genetic Algorithm (HGA).

According to the comparison results in Table 2, it can be seen that IHHO's BRE in solving the 21 Reeves algorithms is better than or equal to the results of the other comparative algorithms for 18 of the 21 algorithms, and in the calculation of the ARE, IHHO achieves 17 optimal values out of the same 21 algorithms, which is better than the other algorithms.

By analyzing Fig. 2, it can be seen that there is a significant difference between both algorithms, VP-QEA and HGA, and the IHHO algorithm, with VP-QEA performing particularly poorly. The performance curve of IHHO lies below the other algorithms in most of the algorithmic solutions, indicating its overall superior performance.The L-HDE algorithm and the IHHO algorithm are very close to each other in terms of BRE and ARE. However, when computing large-scale algorithms such as Rec37, Rec39 and Rec41 algorithms, the performance of IHHO is significantly better than that of L-HDE, which suggests that the algorithm proposed in this study has an advantage in solving large-scale problems.

**Table 2** Test Results

| | VP-QEA | | L-HDE | | HGA | | IHHO | |
|---|---|---|---|---|---|---|---|---|
| | BRE | ARE | BRE | ARE | BRE | ARE | BRE | ARE |
| Rec01 | 0 | 1.05 | 0 | 0 | 0 | 0.14 | 0 | 0 |
| Rec03 | 0 | 0.27 | 0 | 0 | 0 | 0.09 | 0 | 0 |
| Rec05 | 0.24 | 0.99 | 0.242 | 0.242 | 0 | 0.29 | 0 | 0.169 |
| Rec07 | 1.14 | 1.56 | 0 | 0 | 0 | 0.69 | 0 | 0 |
| Rec09 | 0.97 | 2.08 | 0 | 0.026 | 0 | 0.64 | 0 | 0 |
| Rec11 | 0.62 | 2.06 | 0 | 0 | 0 | 1.10 | 0 | 0 |
| Rec13 | 1.34 | 2.48 | 0 | 0.275 | 0.36 | 1.68 | 0 | 0.127 |
| Rec15 | 0.82 | 1.84 | 0 | 0.523 | 0.56 | 1.12 | 0 | 0.372 |
| Rec17 | 2.78 | 3.94 | 0 | 0.363 | 0.95 | 2.32 | 0 | 0.649 |
| Rec19 | 2.34 | 4.25 | 0.287 | 0.702 | 0.62 | 1.32 | 0.294 | 0.585 |
| Rec21 | 1.63 | 3.52 | 0.645 | 1.279 | 1.44 | 1.57 | 0.513 | 1.197 |
| Rec23 | 2.43 | 3.96 | 0.348 | 0.428 | 0.40 | 0.87 | 0.348 | 0.856 |
| Rec25 | 2.62 | 4.61 | 0.557 | 1.082 | 1.27 | 2.54 | 0.631 | 1.124 |
| Rec27 | 2.14 | 4.04 | 0.253 | 0.851 | 1.10 | 1.83 | 0.225 | 0.742 |
| Rec29 | 4.32 | 5.20 | 0.831 | 1.049 | 1.40 | 2.70 | 0.479 | 0.890 |
| Rec31 | 4.63 | 6.14 | 0.427 | 0.644 | 0.43 | 1.34 | 0.353 | 0.602 |
| Rec33 | 1.28 | 2.89 | 0 | 0.244 | 0 | 0.78 | 0.018 | 0.536 |
| Rec35 | 0.09 | 1.61 | 0 | 0 | 0 | 0 | 0 | 0 |
| Rec37 | 6.84 | 7.81 | 2.565 | 3.001 | 3.75 | 4.90 | 2.077 | 2.271 |
| Rec39 | 4.97 | 6.19 | 1.730 | 1.832 | 2.20 | 2.79 | 1.554 | 1.940 |
| Rec41 | 6.99 | 7.80 | 2.661 | 3.350 | 3.64 | 4.92 | 2.362 | 3.146 |
| AVG | 2.295 | 3.538 | 0.502 | 0.757 | 0.863 | 1.601 | 0.421 | 0.724 |

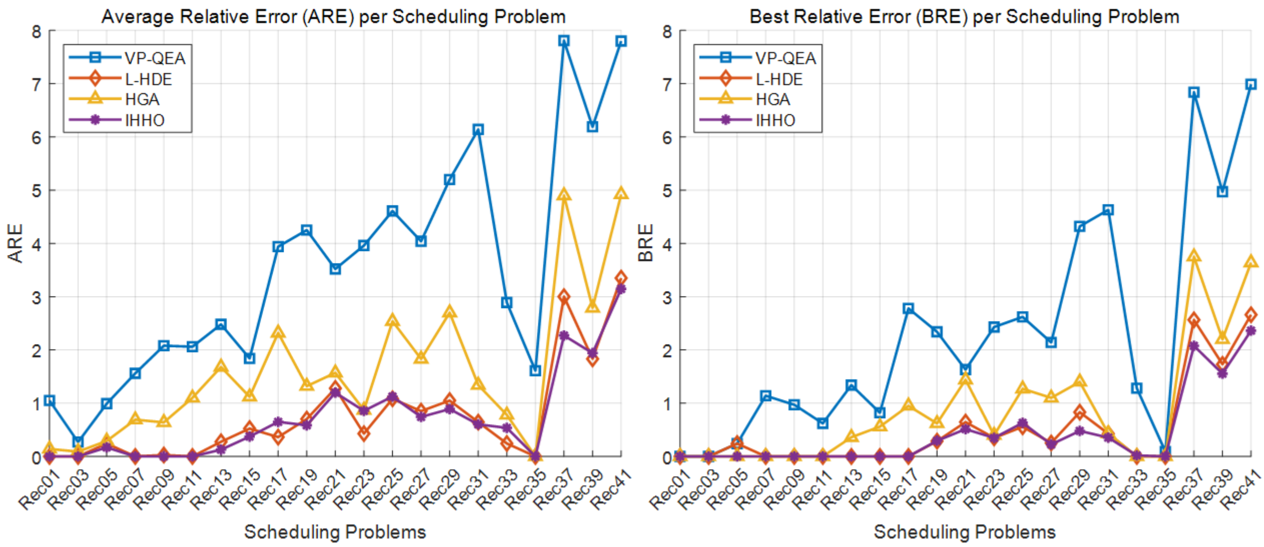


**Fig. 2** Comparison of errors

## 5. Summary

For the permutation flow shop scheduling problem, this paper proposes an improved Harris Hawks optimization algorithm. A coding mechanism based on SPV rules is designed, and Logistic mapping and inverse learning strategies are introduced in the initialization stage to improve the initial population quality, and the Harris Hawks position update strategy is improved by incorporating the golden sine algorithm to enhance the global search capability of the algorithm, accelerate the convergence speed, and improve the algorithm efficiency. To better balance the algorithm's global search ability and local exploitation ability, a nonlinear escape energy factor is introduced, and an adaptive t-distribution strategy is introduced to expand the search range at an early stage to avoid local optimums, and refine the search at a later stage to improve the accuracy of the solution, so as to enhance the overall performance of the algorithm. The effectiveness and superiority of the improved Harris Hawks optimization is verified by testing it on the Reeves test set and comparing it with other algorithms.

# References


[1] Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In Proceedings of ICNN'95-international conference on neural networks (Vol. 4, pp. 1942-1948). ieee.

[2] Holland, J. H. (1992). Genetic algorithms. Scientific american, 267(1), 66-73.

[3] Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. IEEE transactions on systems, man, and cybernetics, part b (cybernetics), 26(1), 29-41.

[4] Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. Future generation computer systems, 97, 849-872.

[5] Mahapatra, S., Dey, B., & Raj, S. (2021). A novel ameliorated Harris hawk optimizer for solving complex engineering optimization problems. International Journal of Intelligent Systems, 36(12), 7641-7681.

[6] Qu, C., He, W., Peng, X., & Peng, X. (2020). Harris hawks optimization with information exchange. Applied mathematical modelling, 84, 52-75.

[7] Gupta, S., Deep, K., Heidari, A. A., Moayedi, H., & Wang, M. (2020). Opposition-based learning Harris hawks optimization with advanced transition rules: principles and analysis. Expert Systems with Applications, 158, 113510.

[8] Deng, G., & Gu, X. (2012). A hybrid discrete differential evolution algorithm for the no-idle permutation flow shop scheduling problem with makespan criterion. Computers & Operations Research, 39(9), 2152-2160.

[9] Tasgetiren, M. F., Sevkli, M., Liang, Y. C., & Gençyilmaz, G. (2004). Particle swarm optimization algorithm for single machine total weighted tardiness problem. In Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753) (Vol. 2, pp. 1412-1419). IEEE.

[10] Zhang, X., & Zhou, H. . (2016). Variable parameters quantum-inspired evolutionary algorithm and its application in permutation flow-shop scheduling problem. Computer Integrated Manufacturing Systems.

[11] Liu, Y., Yin, M., & Gu, W. (2014). An effective differential evolution algorithm for permutation flow shop scheduling problem. Applied Mathematics and Computation, 248, 143-159.

[12] Tseng, L. Y., & Lin, Y. T. (2010). A hybrid genetic algorithm for no-wait flowshop scheduling problem. International journal of production economics, 128(1), 144-152.