Research on Scheduling Methods for Hybrid Flow Shop Based on Intelligent Optimization Algorithms

Mingwei Xu, Yao Dai

***Abstract*— The development of the manufacturing industry relies heavily on the supply of a large amount of energy, especially traditional energy sources such as oil, coal, and natural gas. However, with the increasing environmental awareness of people, there is also a growing concern about the limited availability and environmental impact of traditional energy sources. Improving workshop productivity has become one of the urgent challenges in the manufacturing industry, and production scheduling technology is the key to addressing this issue. However, due to the complexity of the Hybrid Flow Shop Scheduling Problem (HFSP), even precise algorithms struggle to solve small-scale problems. Therefore, this study adopts a novel metaheuristic algorithm to investigate HFSP and designs effective workshop scheduling strategies.**

***Index Terms*—Hybrid flow shop, marine predator algorithm, local search, shop scheduling**

I. INTRODUCTION

The Marine Predators Algorithm (MPA) was proposed by Afshin Faramarzi et al. in 2020 [1]. It is designed for solving complex problems in multidimensional spaces, particularly well-suited for multimodal and high-dimensional optimization problems.

There have been several improvements made to the MPA, including enhancements at different stages of the algorithm. For instance, Fan et al. [2] introduced a logic opposition-based learning strategy during the population initialization stage to generate more accurate solutions. In the optimization stage, they adopted new position updating rules, inertia weight factors, and non-linear step size control parameter updates to enrich the diversity of the population. The proposed method was evaluated and tested using the CEC2020 functions, 23 standard instances, and 4 application problems, and the experimental results demonstrated excellent performance. Moreover, to overcome the drawback of poor optimization ability, Wang et al. [3] updated the population using reverse difference evolution and incorporated a dual-population mechanism for global exploration. They integrated an adaptive parameter strategy based on the t-distribution during the development stage, and updated the ocean memory using a greedy strategy. Based on these improvements, the algorithm was tested using 10 benchmark functions and the CEC2017 functions, and the results showed significant enhancements. Finally, the effectiveness and robustness of the algorithm were validated through its application to a problem in the automotive field.

Manuscript received April 03, 2024

Mingwei Xu, School of Computer Science and Technology, Tiangong University, Tianjin, China
Yao Dai, School of Computer Science and Technology, Tiangong University, Tianjin, China

Additionally, the MPA algorithm has also been applied in the field of solving the Hybrid Flow Shop Scheduling Problem (HFSP). For instance, Yao et al. [4] developed a decentralized MPA to solve the job shop scheduling problem. Firstly, the algorithm discretized the population position vectors based on certain rules. Then, by introducing opposition-based learning, the diversity of the population was enhanced. Furthermore, an adaptive strategy was introduced to balance the exploration and exploitation of the algorithm. Finally, the effectiveness of the proposed algorithm was validated through testing on standard scheduling benchmarks. Liu et al. [5] developed an effective hybrid particle swarm algorithm for solving the permutation flow shop scheduling problem with finite buffers between continuous machines. Yu et al. [6] introduced a comprehensive evaluation index function to address the load balancing of HFS devices and minimize the maximum processing time in the workshop. Pan et al. [7] proposed an efficient artificial bee colony algorithm to minimize completion time for HFSP. Li et al. [8] presented a new hybrid artificial bee colony algorithm for solving HFSP with finite buffers.

In this paper, the HFSP is solved by improving the Marine Predator Algorithm (IMPA).

II. IMPROVEMENT OF MARINE PREDATORS ALGORITHM

A. *Marine Predators Algorithm*

This section provides a comprehensive introduction to the MPA and the theoretical enhancements made to the IMPA.

During the exploration stage of the MPA algorithm, which occurs when the iteration count is less than one-third of the maximum iteration count, the individuals' step lengths or velocities for updating their positions are increased, thereby enhancing the population's exploratory capability. The positions of the predators are updated using equations (2.1) and (2.2).

$$stepsize_i = RB \otimes (Elite_i - RB \otimes X_i) \quad (2.1)$$

$$X_i = X_i + P \cdot R \otimes stepsize_i \quad (2.2)$$

$$i = 1, 2, ..., N, iter < iter_{max}/3 \quad (2.3)$$

where, the stepsize matrix represents the step lengths, RB denotes the Brownian motion matrix, is the elite matrix as shown in equation (2.4), X is the matrix of prey randomly initialized as shown in equation (2.5), p=0.5 is a constant, R is a random number between 0 and 1, N is the number of individuals in the population, d is the dimension size, iter represents the current iteration count, and is the maximum iteration count.

$$Elite = \begin{bmatrix} X_{1,1}^I & X_{1,2}^I & \cdots & X_{1,d}^I \\ \vdots & \vdots & \vdots & \vdots \\ X_{n,1}^I & X_{n,2}^I & \cdots & X_{n,d}^I \end{bmatrix}_{n\times d} \quad (2.4)$$

$$X = \begin{bmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1,d} \\ \vdots & \vdots & \vdots & \vdots \\ X_{n,1} & X_{n,2} & \cdots & X_{n,d} \end{bmatrix}_{n\times d} \quad (2.5)$$

During the transition phase between exploration and exploitation, which serves as an intermediate stage of optimization, half of the individuals in the population are designated for exploration. The positions of these individuals are updated using equations (2.6) and (2.7). Conversely, the other half of the population is allocated for exploitation, and their positions are updated using equations (2.9) and (2.10).

$$stepsize_i = RL \otimes (Elite_i - RL \otimes X_i), i = 1,2,...,N/2 \quad (2.6)$$

$$X_i = X_i + P \cdot R \otimes stepsize_i \quad (2.7)$$

$$i = 1,2,...,N/2, iter_{max}/3 < iter < 2iter_{max}/3 \quad (2.8)$$

$$stepsize_i = RB \otimes (RB \otimes Elite_i - X_i), i = N/2,...,N \quad (2.9)$$

$$X_i = Elite_i + P \cdot CF \otimes stepsize_i \quad (2.10)$$

$$i = N/2,...,N, iter_{max}/3 < iter < 2iter_{max}/3 \quad (2.11)$$

where RL is a random vector matrix used to simulate the Levy distribution. CF is an adaptive parameter used to control the step length of the predators, and its computation is described by equation (2.12).

$$CF = (1 - \frac{iter}{iter_{max}})^{\frac{2iter}{iter_{max}}} \quad (2.12)$$

In the final one-third iteration of the development phase, the low-speed ratio stage, the movement speed of predators is faster than the prey. Update using formulas (2.13) and (2.14).

$$stepsize_i = RL \otimes (RL \otimes Elite_i - X_i) \quad (2.13)$$

$$X_i = Elite_i + P \cdot CF \otimes stepsize_i \quad (2.14)$$

$$i = 1,2,...,N, 2iter_{max}/3 < iter < iter_{max} \quad (2.15)$$

Some environmental issues can also lead to changes in predator behavior, such as the formation of eddies and the effect of Fish Aggregating Devices (FADs), which are mathematically represented by equation (2.16).

$$X_i = \begin{cases} X_i + CF[X_{min} + R \otimes (X_{max} - X_{min})] \otimes U, if\ r \le FADs \\ X_i + [FADs(1-r) + r](X_{r_1} - X_{r_2}), if\ r > FADs \end{cases} \quad (2.16)$$

Where $FADs = 0.2$ is a constant probability; R is a random vector matrix between 0 and 1; U is a matrix with elements either 0 or 1; r is a uniformly distributed random number in the range [0, 1]; X_{max} and X_{min} are matrices consisting of lower and upper bounds for dimensions; r_1 and r_2 are random indices of the prey matrix.

Marine predators possess good memory capabilities to help them recall successful foraging locations. This feature is simulated through memory conservation in MPAs.

B. Improved Marine Predators Algorithm

This paper employs chaotic mapping initialization. Due to the highly random nature of chaotic mapping [9], the initialization performance surpasses that of the original initialization method. The mathematical formula for Sine chaotic mapping is given by equation (2.17).

$$x_{n+1} = \frac{a}{4}\sin(\pi x_n), a \in [0,4] \quad (2.17)$$

where, a range from 0 to 4, and x_1 takes values between 0 and 1.

Because the second stage of the MPA algorithm is prone to getting trapped in local optima, it is optimized by integrating the PSO algorithm [10]. This is represented by equation (2.18).

$$X^{t+1} = X^t + V^{t+1} \quad (2.18)$$

where, X^t represents the particle swarm matrix at time t, and V^t represents the velocity matrix corresponding to the population at time t. The update formula for the velocity matrix is given by equation (2.19).

$$V^{t+1} = W \cdot V^t + r_1 \otimes C_1(P_{best} - X^t) + r_2 \otimes C_2(G_{best} - X^t) \quad (2.19)$$

where r_1 and r_2 are random number matrices ranging from 0 to 1; P_{best} represents the individual historical best position; G_{best} represents the global best position of the population; C_1 and C_2 are the cognitive and social learning factors, respectively.

III. HFSP Mathematical Model

The mathematical model is provided below. To facilitate an accurate description of the related issues, Table 3-1 lists the relevant parameter symbols used.

Table 1 Symbols and meanings related to HFS

symbol	Significance
i	index of jobs.
j	index of stages.
k	index of machines.
n	total number of jobs.
s	total number of stages.
m	total number of machines.
m_j	total number of parallel machines for stage j.
$K_j = \{k_1, k_2, ..., k_{m_j}\}$	set of equipment, $k \in K$.
$O_{i,j}$	processing stage j for job i.
$p_{i,j}$	*Processing time required for workpiece i in processing stage j*
$B_{i,j}$	Start time of workpiece i in operation j
$E_{i,j}$	End time of workpiece i in operation j
C_i	Completion time of workpiece i
C_{max}	Maximum completion time of all workpieces
M	A very large positive number
$X_{i,j,k}$	Binary variable taking the value 1 if the j operation of workpiece i is assigned to be processed on machine k, and 0 otherwise.

The mathematical model for HFS is presented below [7]:

$$\exists m_j > 1, j \in J \quad (3.1)$$

$$\sum_{j=1}^{s} m_j = m, j \in J \quad (3.2)$$

$$\sum_{k \in K_j} X_{i,j,k} = 1, \forall i \in I, j \in J \quad (3.3)$$

$$E_{i,j} = B_{i,j} + p_{i,j}, i \in I, j \in J \quad (3.4)$$

$$B_{i,j+1} - B_{i,j} \geq p_{i,j}, i \in I, j \in \{1,2,...,s-1\} \quad (3.5)$$

$$\sum_{j \in J} p_{i,j} = C_i, i \in I \quad (3.6)$$

where constraint (3.1) ensures that the number of machines in a processing stage is greater than 1; constraint (3.2) indicates that the sum of machines for all operations equals *m*; constraint 3.3) states that only one machine from a stage can be selected for processing any operation of a workpiece; constraint (3.4) is used to calculate the completion time of any workpiece at any processing stage; constraint (3.5) ensures that a workpiece must finish processing in the previous stage before moving to the next one; constraint (3.6) represents the completion time constraint for workpiece *i*.

The fitness function for the basic HFSP problem studied in this paper is given by equation (3.7), which minimizes the workshop completion time as the optimization objective.

$$F = Minimize\ C_{max} \quad (3.7)$$

IV. EXPERIMENTS AND RESULTS ANALYSIS

The test benchmarks are derived from the reC01-42 dataset proposed by Reeves in 1995 [11], with a selection of the first 10 odd-numbered benchmarks. The number of devices corresponding to each process is outlined in Table 3-3. The instances are denoted by R01-R10.

Table 2 Number of Machines for Each Operation

stage	machine number
1	2
2	3
3	3
4	2
5	3
6	3
7	2
8	2
9	3
10	2
11	3
12	2
13	2
14	2
15	3

Table 3-4 presents the experimental findings. The first column denotes the names of the test benchmarks, while the second column indicates their respective scale. Columns three through five exhibit the results for the corresponding algorithms. Optimal values achieved by each algorithm for every test benchmark are highlighted in bold. It is evident from the results that the enhanced IMPA algorithm consistently outperforms others across all ten test benchmarks, followed by the PSO algorithm, with GA algorithm trailing behind. Additionally, the PSO algorithm also secures the optimal outcome for the reC05 instance.

In benchmarks R01-R03, where the scale is relatively small, the overall differences among the three algorithms are minimal, resulting in closely comparable outcomes. However, in benchmarks R04-R09, as the number of processes increases, the enhanced IMPA algorithm exhibits superior performance compared to the other two algorithms. Finally, in the R10 instance, the disparity between the results of IMPA and PSO algorithms is relatively small. In summary, the IMPA algorithm demonstrates favorable performance in solving these ten instances, achieving smaller makespan durations. The PSO algorithm also delivers relatively satisfactory results, while the GA algorithm exhibits inferior solving capabilities.

Figure 1 is the Gantt chart for the R01 benchmark. Each rectangular block in the chart corresponds to each operation of the workpiece, and workpiece information is labeled on each block. Below each block are the processing details for the workpiece operation. The vertical axis represents the device number. In actual production, examining the Gantt chart provides a clear overview of all processing information.

Table 3 the results of the GA, PSO, and IMPA algorithms

Benchmarks	n×s×m	GA	PSO	IMPA
		Makespan	Makespan	Makespan
R01	20×5×13	722	703	**685**
R02	20×5×13	625	613	**598**
R03	20×5×13	697	**693**	**693**
R04	20×10×25	1019	992	**973**
R05	20×10×25	935	902	**898**
R06	20×10×25	930	879	**871**
R07	20×15×37	1321	1299	**1288**
R08	20×15×37	1296	1271	**1251**
R09	20×15×37	1329	1304	**1282**
R10	30×10×25	1332	1272	**1251**

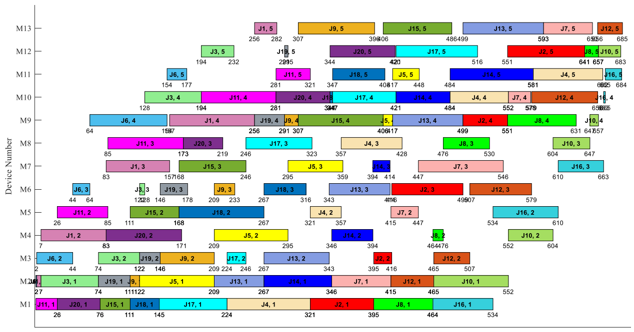

Figure 1 Process Flowchart for Benchmark R01

V. CONCLUSION

This article employs the MPA algorithm to solve the HFSP problem. By integrating chaotic initialization strategy with PSO and GA algorithms into each optimization stage of the MPA algorithm, it aims to overcome its tendency to fall into local optima. Finally, through experimental verification, the improved IMPA algorithm achieves the best results compared to GA and PSO algorithms, as demonstrated on Reeves' 10 benchmark tests.

VI. ACKNOWLEDGMENT

The authors would like to thank the Professor He for supporting the research work.

REFERENCES

[1] Faramarzi, A., Heidarinejad, M., Mirjalili, S., & Gandomi, A. H. (2020). Marine Predators Algorithm: A nature-inspired metaheuristic. *Expert systems with applications*, *152*, 113377.

[2] Fan, Q., Huang, H., Chen, Q., Yao, L., Yang, K., & Huang, D. (2022). A modified self-adaptive marine predators algorithm: framework and engineering applications. *Engineering with Computers*, 1-26.

[3] Wang, Y., Wang, W., Yang, Y., & Zhou, H. (2023). Improved marine predator algorithm with multi-strategy fusion and its engineering application. Computer Integrated Manufacturing Systems.

[4] Yao, X. (2023). Research on job shop scheduling problem based on chaotic improved marine predator algorithm [D]. Yancheng Institute of Technology.

[5] Liu, B., Wang, L., & Jin, Y. H. (2008). An effective hybrid PSO-based algorithm for flow shop scheduling with limited buffers. *Computers & Operations Research*, *35*(9), 2791-2806.

[6] Yu, M., & Liu, D. (2021). Improved PSO-GA algorithm for solving hybrid flow shop scheduling problem. Journal of Wuhan University of Technology (Transportation Science & Engineering Edition), 45(03), 586-590.

[7] Pan, Q. K., Wang, L., Li, J. Q., & Duan, J. H. (2014). A novel discrete artificial bee colony algorithm for the hybrid flowshop scheduling problem with makespan minimisation. *Omega*, *45*, 42-56.

[8] Li, J. Q., & Pan, Q. K. (2015). Solving the large-scale hybrid flow shop scheduling problem with limited buffers by a hybrid artificial bee colony algorithm. *Information Sciences*, *316*, 487-502.

[9] Mansouri, A., & Wang, X. (2020). A novel one-dimensional sine powered chaotic map and its application in a new image encryption scheme. *Information Sciences*, *520*, 46-62.

[10] Hu, H., Ge, J., & Wei, Y. (Year). Optimization strategy for electric vehicle charging based on improved multi-objective particle swarm algorithm. Journal of Electrical Engineering, 1-10.

[11] Reeves, C. R. (1995). A genetic algorithm for flowshop sequencing. Computers & Operations Research, 22(1), 5-13.