Research on Flexible Job Shop Scheduling Problem Based on an Enhanced Walrus Optimization Algorithm

Yunjie Shen *, Ning Cui

School of Business Administration, Liaoning Technical University, Huludao, China.

*shen.edu@outlook.com 125000

Abstract. This paper proposes an enhanced Walrus Optimization Algorithm (WWaOA) to address the Flexible Job Shop Scheduling Problem (FJSP) in the steel pipe industry, where complex processes and small orders cause inefficiencies. WWaOA uses chaotic initialization, a Whale Optimization Operator, and crossover-mutation to improve global and local search. Tested on the Brandimarte dataset (MK01-MK10), WWaOA outperforms ICSO, BEDA, GWO, and WaOA in minimizing makespan, offering a robust solution for production scheduling. Future research will extend WWaOA to multi-objective scheduling problems.

Keywords: Flexible Job Shop Scheduling Problem (FJSP), Walrus Optimization Algorithm (WWaOA), chaotic initialization, Whale Optimization Operator, crossover-mutation, makespan minimization, Brandimarte dataset, production scheduling, steel pipe industry.

1 Introduction

Chinese manufacturers are facing challenges due to changing market dynamics. Under the impetus of "Made in China 2025" and Industry 4.0, steel pipe manufacturers must enhance efficiency through digital transformation. They need to address complex production requirements and implement advanced scheduling solutions. However, traditional heuristic algorithms tend to get trapped in local optima, while existing metaheuristic algorithms have significant limitations, such as high sensitivity to initialization. Solving these limitations is imperative; therefore, this paper proposes a new optimization method (WWAOA) to address these issues. Building upon the walrus optimization algorithm (WaOA) introduced in 2023 [1], the WWAOA incorporates whale optimization operators, crossover and mutation operations, and chaotic initialization to resolve the shortcomings of current scheduling algorithms, thereby supporting smart manufacturing strategies.

In steel pipe manufacturing, traditional scheduling struggles with complex processes and small orders, leading to inefficiencies. For example, a large manufacturer faces issues like long wait times and mismatched production rhythms, reducing efficiency. Customization demands further complicate scheduling, causing delays. Statistics show

S. M. Ghadiri et al. (eds.), *Proceedings of the 2025 4th International Conference on Engineering Management and Information Science (EMIS 2025)*, Advances in Computer Science Research 121,
https://doi.org/10.2991/978-94-6463-736-6_16

16% of orders are delayed, with 80% due to poor process coordination and resource allocation.

1.1 Analysis of the Flexible Job Shop Scheduling Problem

The definition of flexible job shop scheduling is typically as follows:

There are n workpieces to be processed, forming the set J={J1, J2, ⋯, Jn}. Each workpiece has multiple processing steps, denoted by O_{ij} as the J_i j-th process of workpiece Ji. Workpiece Ji has n_i processes, and each process has a fixed processing time. There is a set of workers W={W_1, W_2, ⋯, W_w}, and a set of machines M={M_1, M_2, ⋯, M_m}in the production line. Each process is to be performed on a designated machine, and the goal of scheduling is to assign each process to a corresponding machine, sequence the processes on the machines, and assign workers to operate the machines. To address this specific issue, reducing production time and optimizing production scheduling have become key to enhancing the enterprise's competitiveness. By minimizing the completion time, the waiting time between processes can be effectively reduced, equipment utilization can be improved, and production costs can be lowered, thereby enhancing the enterprise's market responsiveness and order delivery capability. Therefore, in the subsequent research, we will focus on the optimization metric of minimizing completion time, designing targeted scheduling optimization methods to solve this typical scheduling problem in the actual production of manufacturing enterprises.

1.2 Problem Model and Symbol Definitions

In the problem model of this article, the meanings of parameter symbols are shown in Table 1.

Table 1. Symbol Meanings of the Model

Symbol	Explanation
n	Total number of workpieces to be processed
m	Total number of machines
O_{ij}	The j-th operation of workpiece i
n_i	Total number of operations for workpiece i
S_{ij}	Start time of operation O_{ij}
T_i	Completion time of workpiece i
T_{ij}	Completion time of operation O_{ij}
T_{ijk}	Processing time of operation O_{ij} on machine k
X_{ijk}	0-1 decision variable (1 if the j-th operation of workpiece i is processed on machine k, otherwise 0)

The optimization function adopted in this paper is as follows:

$$f_1 = minmax(C_i\ ), i = 1,2, \dots, n \tag{1}$$

The constraint functions in this model are as follows:

$$S_{ij} + T_{ijk} \leq T_{ij}, \forall i, j, k \tag{2}$$

$$\sum_{k=1}^{m} x_{ijk} = 1, \forall i, j \tag{3}$$

$$T_{ij} \leq T_i, \forall i, j \tag{4}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{n_i} X_{ijk} \leq 1, \forall k \tag{5}$$

$$S_{ij} \geq 0, T_{ij,} \geq 0, T_{ijk} \geq 0, \forall i, j, k \tag{6}$$

$$S_{i(j+1)} \geq T_{ij}, \forall i, j \tag{7}$$

$$X_{ijk} * M_k = X_{ijk} \tag{8}$$

Constraints in the Model: Equation (1) represents the minimization of the maximum job completion time. Formula (2): Processing cannot be interrupted once started. Formula (3): Each operation is processed on only one machine. Formula (4): Operation completion time ⩽ total workpiece completion time. Formula (5): A machine processes only one operation at a time. Formula (6): Start time, duration, and completion time are non-negative. Formula (7): Operations follow the specified sequence. Formula (8): Operations are assigned to available machines only.

2 Key Improvements in WWaOA:

2.1 Chaotic Initialization

A well-initialized population in metaheuristic algorithms reduces premature convergence, avoids local optima, accelerates convergence, and improves solution quality. Using chaotic properties, initial individuals are more evenly distributed, enhancing population diversity. The Logistic chaotic map, a classic and effective method, excels in performance and is easy to explain, reducing communication costs. In formula (9).

$$X_{i+1} = a x_i (1 - x_i) \tag{9}$$

x_i: Represents the value of xx at the ii-th iteration, with the domain $x_i \in (0,1)$.a: A critical parameter that determines the behavior of the system. Typically, $a \in (0,4)$,When a=4, the system enters a fully chaotic state. In this state, the trajectory of the iterations will be distributed within the interval (0,1), exhibiting high complexity and unpredictability.

2.2 Whale Optimization Operator

In the global exploration phase of the Walrus Optimization Algorithm (WOA) [1], to avoid issues such as premature convergence or insufficient exploration capability in certain scenarios, this paper introduces a method that allows each virtual walrus individual to have more search paths to better adjust its position. Instead of relying solely on random numbers to determine the next position of a virtual walrus, this approach enhances the algorithm's global search performance. The spiral exploration formula based on the Whale Optimization Operator is as follows (Formula 10):

$$\vec{X}(t+1) = \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) \quad (10)$$

Explanation of Formula (10):$\vec{X}(t+1)$: Represents the new solution position obtained by the walrus individual.b: A constant that defines the shape of the spiral exploration function. In this paper, b=1.l: A random number uniformly distributed in the interval $[-1,1]$.$e^{bl} \cdot \cos(2\pi l)$: Represents the core of the spiral function, whose shape determines the search path of the walrus individual.$\vec{X}^*(t)$: Represents the position of the current best solution.$\vec{D}'$: Represents the distance between the current walrus individual and the best solution. The calculation method for $\vec{D}'$is shown in Formula (11).

$$\vec{D}' = \left|\vec{X}^*(t) - \vec{X}(t)\right| \quad (11)$$

2.3 Crossover and Mutation

To improve the W-WaOA algorithm's ability to solve FJSP [2], crossover and mutation operations are added. Crossover combines parent genes to create new individuals, while mutation randomly modifies genes. The crossover probability is 0.9, and mutation uses Levy flight (β = 1.5) to perturb the encoding. Infeasible solutions generated during these processes are repaired by analyzing and adjusting the entire encoding to ensure feasibility.

2.4 WWaOA Algorithm Steps

Step 1: Initialization. Set the maximum iterations E, population size P, initial chaotic map parameter a, spiral search parameters, and annealing control parameters T0 and r. **Step 2**: Optimal Individual Determination. Identify the current iteration's optimal individual as the leader XL. **Step 3**: Exploration Phase. Simulate foraging behavior, with the strongest walrus guiding group movement. Update individual positions using the specified formula. **Step 4**: Spiral Exploration Phase. Enhance global exploration by incorporating the spiral search path from the Whale Optimization Algorithm. **Step 5**: Migration Phase. Allow the walrus population to randomly migrate to explore additional solution spaces. **Step 6**: Simulated Annealing. Compare a random number with r. If within r's range, include the suboptimal solution; otherwise, discard it. **Step 7**: Exploitation Phase. Narrow the search range to simulate local search and improve precision. **Step 8**: Comparison Phase. Sort the population by objective function values and update the global leader XL. **Step 9**: Termination Condition Check. Check if the iteration limit or other termination conditions are met. Output the current optimal solution, XL.

3 Experimental Testing

To verify the effectiveness of the proposed WWaOA algorithm, this paper tests it using the Brandimarte flexible job shop scheduling dataset MK01-MK10. The testing software is Python 3.5, and the testing environment is an AMD Ryzen 7 7840H. Both WaOA and WWaOA are tested with a population size and iteration count of 100. The WWaOA algorithm is compared with common algorithms such as ICSO [3], BEDA [4], GWO [5], and WaOA. Both WaoA and WWaOA are run ten times, Take the average value in Table 2

Table 2. Comparison Chart of Algorithms on the Brandimarte Test Set

Cases	Scale	LB	ICSO	BEDA	GWO	WaOA		WWaOA	
	Min	Min	Min	Min	Min	Min	Mean	Min	Mean
Mk01	10×6	36	40	42	41	42	43	**40**	**40**
Mk02	10×6	24	28	32	29	30	34.8	**28**	**28**
Mk03	15×8	204	204	209	204	204	204	**204**	**204**
Mk04	15×8	48	67	69	69	75	76.4	**67**	**67**
Mk05	15×4	168	176	210	178	186	187.4	**176**	**176**
Mk06	10×15	33	73	101	69	70	75.6	**67**	**67**
Mk07	20×5	133	149	195	150	158	158	**149**	**149**
Mk08	20×10	523	523	577	523	536	540	**523**	**523**
Mk09	20×10	299	334	472	324	329	353.7	**316**	**316**
Mk10	20×15	165	259	323	240	246	271.7	**230**	**230**

4 Conclusion

This paper investigates the flexible job shop scheduling problem (FJSP) with the objective of minimizing the maximum completion time (makespan) and proposes a WWaOA algorithm to solve it. The algorithm employs a chaotic initialization strategy to initialize the population, updates the population through continuous optimization encoding and crossover-mutation operations, and utilizes a greedy insertion method for decoding. To enhance search efficiency, the algorithm introduces a spiral search strategy during the global search phase and incorporates a simulated annealing algorithm during the local search phase. Comparative experiments with various optimization algorithms on the Brandimarte test set (including MK01 to MK10) demonstrate that the WWaOA algorithm exhibits significant advantages in solving FJSP problems.

The practical implementation of WWaOA in industrial settings shows promising potential for addressing real-world scheduling challenges, particularly in steel pipe manufacturing where production flexibility is crucial. Our analysis indicates potential efficiency improvements of up to 25% in makespan reduction when compared to current scheduling methods, though implementation may require careful integration with existing manufacturing execution systems (MES) and staff training. The algorithm's performance on the Brandimarte test set suggests robust scalability for medium-sized manufacturing operations, though computational requirements may increase significantly for larger-scale applications.

Future research directions will focus on three key areas: (1) extending the algorithm to handle real-time production disturbances and dynamic rescheduling scenarios common in industrial environments, (2) improving computational efficiency for large-scale manufacturing systems with thousands of operations, and (3) developing hybrid approaches that combine WWaOA with deep learning techniques to enhance adaptation to specific industrial contexts. Additionally, investigation into multi-objective optimization will address practical trade-offs between makespan, energy consumption, and equipment utilization in modern smart manufacturing environments.

References

1. Pavel T, Mohammad D. A new bio-inspired metaheuristic algorithm for solving optimization problems based on walruses behavior. [J]. Scientific reports,2023,13(1):8775-8775.
2. Chen R, Yang B, Li S, et al. A self-learning genetic algorithm based on reinforcement learning for flexible job-shop scheduling problem [J]. Computers & Industrial Engineering, 2020, 149
3. Jiang Tianhua Cat swarm optimization algorithm for solving flexible job shop scheduling problem [J]. Computer Engineering and Applications, 2018, 54 (23): 259-263
4. Li Xiulin, Lu Jianxia, Chai Guozhong, etc Hybrid bee colony algorithm for solving flexible job shop scheduling problem [J]. Computer integrated manufacturing system, 2011, 17(7): 1495-1500.
5. ZIAEE M. A heuristic algorithm for solving flexible job shop scheduling problem [J]. The international journal of advanced manufacturing technology, 2014, 71 (1-4): 519-528.