applied sciences

MDPI

Article

Flexible Job Shop Scheduling Optimization for Green Manufacturing Based on Improved Multi-Objective Wolf Pack Algorithm

Jian Li [1,2,*], Huankun Li [1], Pengbo He [1], Liping Xu [1,2], Kui He [1,2] and Shanhui Liu [3]

[1] School of Mechatronics Engineering, Henan University of Science and Technology, Luoyang 471000, China; lhkkk545@163.com (H.L.); hepengbo666@163.com (P.H.); xlpzz@163.com (L.X.); hekui@haust.edu.cn (K.H.)
[2] Collaborative Innovation Center of Machinery Equipment Advanced Manufacturing, Luoyang 471000, China
[3] Faculty of Printing, Packaging Engineering and Digital Media Technology, Xi'An University of Technology, Xi'an 710048, China; shanhuiliu@xaut.edu.cn
* Correspondence: li_jian@haust.edu.cn

Abstract: Green manufacturing has become a new production mode for the development and operation of modern and future manufacturing industries. The flexible job shop scheduling problem (FJSP), as one of the key core problems in the field of green manufacturing process planning, has become a hot topic and a difficult issue in manufacturing production research. In this paper, an improved multi-objective wolf pack algorithm (MOWPA) is proposed for solving a multi-objective flexible job shop scheduling problem with transportation constraints. Firstly, a multi-objective flexible job shop scheduling model with transportation constraints is established, which takes the maximum completion time and total energy consumption as the optimization objectives. Secondly, an improved wolf pack algorithm is proposed, which designs individual codes from two levels of process and machine. The precedence operation crossover (POX) operation is used to improve the intelligent behavior of wolves, and the optimal Pareto solution set is obtained by introducing non-dominated congestion ranking. Thirdly, the Pareto solution set is selected using the gray relational decision analysis method and analytic hierarchy process to obtain the optimal scheduling scheme. Finally, the proposed algorithm is compared with other algorithms through a variety of standard examples. The analysis results show that the improved multi-objective wolf pack algorithm is superior to other algorithms in terms of solving speed and convergence performance of the Pareto solution, which shows that the proposed algorithm has advantages when solving FJSPs.

Keywords: flexible job shop scheduling problem; multi-objective wolf pack algorithm; transportation time; maximum completion time; energy consumption

check for updates

Citation: Li, J.; Li, H.; He, P.; Xu, L.; He, K.; Liu, S. Flexible Job Shop Scheduling Optimization for Green Manufacturing Based on Improved Multi-Objective Wolf Pack Algorithm. *Appl. Sci.* **2023**, *13*, 8535. https://doi.org/10.3390/app13148535

Academic Editor: Wei Li

Received: 18 June 2023
Revised: 18 July 2023
Accepted: 21 July 2023
Published: 24 July 2023

1. Introduction

The manufacturing industry is an important part of the modern economy and an important symbol with which to measure a country's comprehensive national strength. While the manufacturing industry is moving forward, a series of problems such as energy depletion, environmental pollution and global warming have become the focus of attention in the world today, and the manufacturing industry is facing the new challenge of green transformation. Green manufacturing, as a new modern manufacturing model that comprehensively considers environmental impacts and resource consumption, aims to reduce the negative impact of manufacturing on the environment and improve resource utilization. As the most basic production unit in the manufacturing industry, job shops play an irreplaceable role in the manufacturing industry. Reasonable workshop scheduling can make the conversion of products more efficient and maximize the utilization of resources, so as to reduce the cost of enterprises and improve production efficiency. As an extension of the traditional job shop scheduling problem (JSP), the FJSP is one of the core issues in the field of green manufacturing process planning. Since it was proposed, it has become a classic NP hard combinatorial optimization problem in computer science and operations research [1]. In the beginning, FJSP tended to focus on single-objective optimization for minimizing maximum completion time, economic cost, energy consumption, total delay time, and total overrun time [2–4]. With the continuous development of the manufacturing industry and the continuous optimization of product performance, focusing solely on single-objective optimization can no longer meet the development requirements of today's manufacturing industry. Therefore, the research on multi-objective flexible job shop scheduling problems (MOFJSP), which is a deeper step compared to FJSP, has become a hot topic in the industry.

For MOFJSP, many scholars have improved the traditional intelligent algorithms such as Genetic Algorithm (GA), Ant Colony Algorithm (ACO), Particle Swarm Algorithm (PSO) and Artificial Bee Colony Algorithm (ABC). For example, An et al. [5] established a multi-objective mathematical model to minimize the maximum completion time, total delay time, total production cost and total energy consumption, and proposed a hybrid multi-objective evolutionary algorithm based on the Pareto elite storage strategy to solve it. Zheng et al. [6] proposed a fruit fly collaborative multi-objective optimization algorithm to solve green scheduling with the objective of minimizing the maximum completion time and minimum total carbon emissions. Luo et al. [7] used an improved multi-objective Gray Wolf algorithm for optimization, with minimizing the maximum completion time and total energy consumption as the optimization objective. Wu et al. [8] optimized from the aspect of operations management, with minimizing the maximum completion time, energy consumption and machine switching times as the optimization objectives, and adopted an improved Non-Dominated Sorting Genetic Algorithm (NSGA-II) for optimization. Zhao et al. [9] took minimizing the maximum completion time, energy consumption and noise as the optimization objectives of the multi-objective mathematical model, and embedded the improved simulated annealing algorithm into the imperialist competition algorithm to overcome the premature convergence problem of the imperialist competition algorithm. Hasani et al. [10] introduced the NSGA-II to solve the multi-objective mathematical model aiming at production cost and energy consumption. Zhu et al. [11] designed a gray wolf algorithm with a new coding method and job priority repair mechanism for MOFJSP with priority constraints. Caldeira et al. [12] proposed a multi-objective discrete Jaya algorithm to optimize the flexible job shop, with minimizing the maximum completion time, total machine workload and key machine workload as optimization indicators. Chen Kui et al. [13] established a flexible job shop scheduling model considering transportation time, proposed a hybrid discrete particle swarm optimization algorithm for optimization, and introduced a competitive learning mechanism and random restart algorithm to avoid premature algorithms. Huang et al. [14] proposed an improved NSGA-III algorithm, which introduced the reference-based niche selection mechanism to improve the diversity of the algorithm, and was used to solve the MOFJSP with the goal of minimizing the maximum completion time, total machine load, maximum machine load and machine energy consumption. Mehdi et al. [15] used a mixed integer linear programming model to solve the green flowshop scheduling problem with the objective of minimizing the maximum completion time and total carbon emissions. Chen et al. [16] proposed an improved non-dominated sorting genetic algorithm to solve the hybrid process shop scheduling problem under time-of-use and step tariff system with the optimization objective of minimizing the maximum completion time with respect to the total shop energy consumption. Liu et al. [17] established a multi-objective mathematical model of flexible workshop with crane transportation constraints to minimize the maximum completion time and energy consumption, and optimized the model by combining a genetic algorithm with a firefly swarm optimization algorithm. However, with the deepening complexity of the mathematical model of MOFJSP, the traditional intelligent optimization algorithm often has some disadvantages in solving MOFJSP, such as slow running speed and fast algorithm convergence, and is easy to fall into local optimization in the iterative process. With the continuous updating of the new intelligent algorithm, it provides a new idea for solving the MOFJSP problem more efficiently.

The Wolf Pack Algorithm (WPA) is a pack intelligence optimization algorithm that simulates the division of labor and collaboration of wolves in nature to capture prey [18], with strong global search capability and computational robustness, and is used to solve problems such as multi-distribution center vehicle path [19], Traveling Salesman Problem (TSP) [20], and unmanned helicopter route path planning [21]. However, for workshop scheduling problems, there are still fewer WPA-related applications involved. In this paper, an improved Multi-Objective Wolf Pack Algorithm (MOWPA) is designed to solve the Multi-Objective Flexible job shop green scheduling mathematical model with the optimization objectives of minimizing the maximum completion time and minimum energy consumption, and generate the Pareto optimal solution set. The gray relational decision analysis method and Analytic Hierarchy Process (AHP) are introduced to select the Pareto solution set, and a new scheme is proposed to effectively solve the multi-objective flexible job shop problem. The contribution of this article can be summarized in three aspects: (1) In order to be more in line with actual production and processing, FJSP is extended according to the definition of FJSP, and a MOFJSP with transportation constraints is established. (2) An improved multi-objective wolf swarm algorithm is designed. The crossover and mutation operations of the genetic algorithm and pox crossover operations are introduced to improve the three intelligent behaviors of the wolf swarm algorithm. To comply with the multi-objective problem constraints, the WPA update method is designed and combined with non-dominated congestion ranking to solve the optimal Pareto solution set. (3) In order to facilitate the decision-maker to better select a scheduling scheme, this paper introduces the gray relational decision analysis method and analytic hierarchy process to calculate the Pareto solution set, and the decision-maker selects a scheduling scheme that is more consistent with the workshop processing according to the calculation results.

The framework of the rest of the paper is as follows. In Section 2, a brief description of the FJSP definition is given and a mathematical model of MOFJSP with transportation constraints is developed based on the problem definition and constraints. In Section 3, an improved multi-objective wolf pack algorithm is proposed and a detailed description of the algorithm if solving MOFJSP is given. In Section 4, simulation tests and analyses are conducted. A method for selecting the Pareto solution set is introduced in conjunction with an actual job shop. Additionally, a comparison between the MOWPA algorithm and the NSGA-II algorithm is performed to further validate the effectiveness of the proposed method. Finally, Section 5 summarizes the entire text.

2. Problem Description and Modeling

2.1. Problem Description

The FJSP problem can be described as follows: n workpieces are processed on m machines. Each workpiece has multiple processing processes, and each process can be executed on more than one machine. All processes of n workpieces are scheduled on m machines according to a specified processing sequence. The processing time and energy consumption values of the processes vary depending on the selected processing machines [22]. The following assumptions are made to establish the mathematical model:

(1) Each workpiece must be processed in the previous process before it can be processed in the next process;
(2) Each process of each workpiece can only be processed on one machine;
(3) The workpiece will not be interrupted during processing;
(4) At the same time, each machine can only process one workpiece, and each workpiece can only be processed by one machine;
(5) At the initial moment, all workpieces and machines are ready;
(6) For the first process of each workpiece, transportation time and energy consumption are not considered;

(7) The idle start time of each machine is the end time of the last process, and the idle end time is the start time of the first process;
(8) During the transportation of workpieces, problems such as transportation failures are not considered.

To establish the mathematical model for the maximum completion time and total workshop energy consumption, the following symbolic descriptions are provided, as presented in Table 1.

Table 1. Symbol descriptions.

	Symbols	Definitions
Parameters	N	Workpieces
	J	Working sequence
	M	Machines
	$O_{i,j}$	The j-th process of the i-th workpiece
	$T_{i,j,k}$	Processing time of $O_{i,j}$ on machine k
	$T_{i(j-1)jmk}$	Time for transporting N from machine tool M_m to machine tool M_k between operation $O_{i,j-1}$ and $O_{i,j}$ of workpiece N
	$U_{i,j,k}$	Integer variable, takes 0 or 1 if $O_{i,j}$ is processed on machine k, otherwise 0
	P_c^k	Machining power of machine tool k
	P_{idle}^k	Standby power of machine tool k
	$P_{i(j-1)jmk}$	Power for transporting N from machine tool M_m to machine tool M_k between operation $O_{i,j-1}$ and $O_{i,j}$ of workpiece N
Variables	$S_{i,j,k}$	Starting processing time of $O_{i,j}$ at machine k
	$F_{i,j,k}$	$O_{i,j}$ end processing time on machine k
	C_i	Completion time for workpiece i
	T_c^k	Machining time of machine tool k
	T_{idle}^k	Standby time of machine tool k
	E_k	Total energy consumption of machine tool k
	E_c^k	Machining energy consumption of machine tool k
	E_{idle}^k	Standby energy consumption of machine tool k
	E_{trans}	Total transportation energy consumption

2.2. Mathematical Model Building

Workshop energy consumption comprises machine tool energy consumption and transportation energy consumption. In Figure 1, which represents a simplified model of the input power of a machine tool during the machining process [23], the energy consumption of a single piece of equipment can be divided into four states: starting state, processing state, no-load state, and stop state. During equipment start-up and shut-down, there is a significant fluctuation in power, but the duration is short. Frequent start–stop operations can negatively impact the machine's lifespan and processing quality. Typically, a machine performs only one start–stop operation, which is not the primary factor affecting energy consumption. Hence, it is not considered in the model. Consequently, only the processing state and no-load state of a single equipment are taken into account when considering energy consumption.

The machining energy consumption is determined by the machining power and machining time of the machine tool.

$$E_c = \sum_k^m E_c^k = \sum_k^m P_c^k \times T_c^k \tag{1}$$

The no-load energy consumption is generated by the idling state of the machine tool before the workpiece is processed, and is determined by the no-load time and no-load power of the machine tool.

$$E_{idle} = \sum_k^m E_{idle}^k = \sum_k^m P_{idle}^k \times T_{idle}^k \tag{2}$$

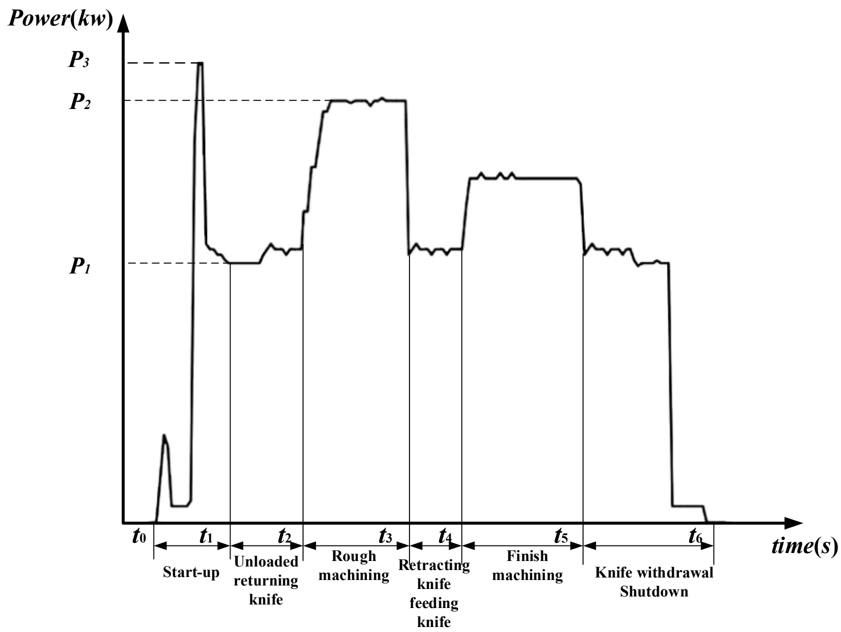


Figure 1. Energy consumption diagram of machine tool processing.

During the entire processing process, workpieces need to be transported from one machine tool to another for processing, necessitating transportation. Assuming a constant transportation power, the transportation time is determined by the distance between the two machine tools. The transportation energy consumption can be calculated as the product of the transportation power and the transportation time.

$$E_{trans} = \sum_{i=1}^{n}\sum_{j=1}^{q} T_{i(j-1)jmk} \times P_{i(j-1)jmk} \tag{3}$$

The total energy consumption of the workshop is thus obtained as:

$$E = E_c + E_{idle} + E_{trans} = \sum_{k}^{m} P_c^k \times T_c^k + \sum_{k}^{m} P_{idle}^k \times T_{idle}^k + \sum_{i=1}^{n}\sum_{j=1}^{q} T_{i(j-1)jmk} \times P_{i(j-1)jmk} \tag{4}$$

A multi-objective mathematical model is established with the maximum completion time f_1 and total energy consumption of the workshop f_2.

$$\begin{aligned} &\min f_1 = C = \max_{1\le i\le n} C_i \\ &\min f_2 = E \end{aligned} \tag{5}$$

The constraints are as follows:

$$S_{i,j} \ge F_{i,j-1} + T_{i(j-1)jmk} \tag{6}$$

$$\sum_{k=1}^{m} U_{i,j,k} = 1, i \in N \tag{7}$$

$$F_{i,k} \le S_{i\prime,k}, i \in N, i\prime \in N, k \in M \tag{8}$$

$$C_i = F_{i,q} \tag{9}$$

where Equation (6) indicates that the start time of the process of the workpiece is greater than the end time of the previous process plus the transportation time of the workpiece operation; Equation (7) indicates that each process can only be processed on one machine;

Equation (8) indicates that the machine can only start processing the next workpiece after finishing processing one workpiece; Equation (9) indicates that the processing time of the workpiece is the completion time of the last process, q is the final process of the workpiece. The specific operation is shown in Figure 2.

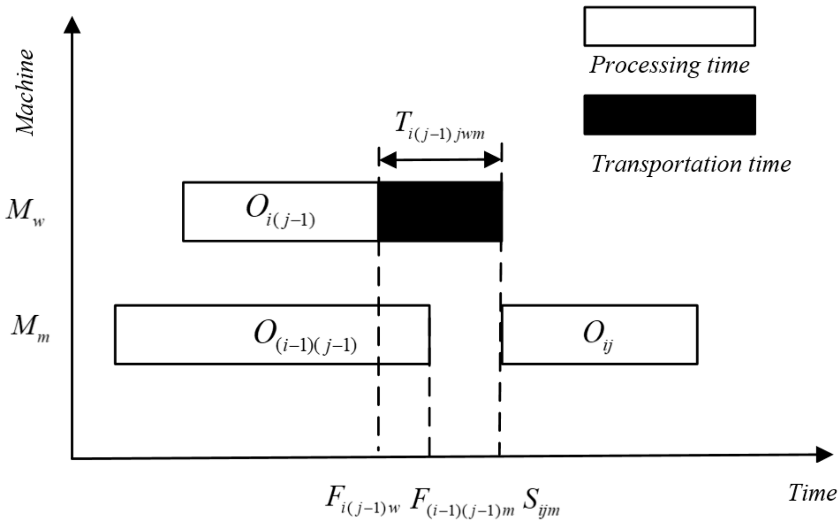


Figure 2. Constraint Description Gantt Chart.

3. Improved Multi-Objective Wolf Pack Algorithm Design

The WPA is an intelligence optimization algorithm inspired by the behavior of wolves preying on their prey. The WPA algorithm abstracts three intelligent behaviors: wandering behavior, calling behavior, and siege behavior. In the algorithm, the head wolf represents the best wolf, and a wolf pack renewal method is employed to retain the best wolves and eliminate the inferior ones [24]. Originally designed for solving continuous function optimization problems, WPA has been found to suffer from the drawbacks of falling into local optima and premature convergence. To address these limitations and leverage the characteristics of the MOFJSP problem, three intelligent algorithms within WPA have been improved to expand the search range and obtain the global optimal Pareto solution set.

3.1. Encoding and Decoding

According to the discrete characteristics of the FJSP problem, a two-level coding method is adopted, that is, the encoded individual vector is composed of two parts: process sequencing vector and machine selection vector. Additionally, the code length of the process layer and the machine layer are equal, so that the process code and the machine code correspond to each other. The coding method is shown in Table 2.

Table 2. Code segment.

Process layer	1	1	2	3	3	1	2	2	3
Machine layer	1	2	2	1	2	2	2	1	3

The first row of the table represents the process order, where the number represents the name of the workpiece and the number of times it appears represents the process of the workpiece. For example, if "1" means workpiece 1, the first occurrence of "1" means the first process of workpiece 1, and the second occurrence of "1" means the second process of workpiece 1, and so on. The second row is the machine selection problem for the machining process. The machines that can be processed by each machining process correspond to a set of machines, and each number indicates the index of the location of its machine set. For example, the processing machine set for process $O_{2,1}$ is $[M_2, M_4]$ (M_2 and M_4 denote machine 2 and machine 4, respectively), and 2 means that its processing machine is the second position in the machine set, which is M_4, indicating that process $O_{2,1}$ is processed on machine 4.

3.2. Population Initialization

The quality of the initial solution directly affects the performance of the algorithm. Random initialization is a widely used method which ensures diversity in the initial population but does not guarantee the quality of the solutions. In the case of MOFJSP optimization, three rules are employed to generate the initial population: the minimization of maximum completion time method, the minimization of energy consumption value method, and the random generation method. The population size for each rule is set at 40%, 40%, and 30%, respectively, aiming to improve the quality of the initial solutions.

3.3. Non-Dominated Crowding Ranking

The non-dominated crowding ranking method is used to calculate the level of individuals, stratify them, and calculate the crowding degree between individuals at the same level. This realizes the preservation of optimal solutions and elimination of inferior solutions for wolf packs, allowing wolf packs to update the position of artificial wolves during the iteration process. The non-dominated sorting is shown in Figure 3.

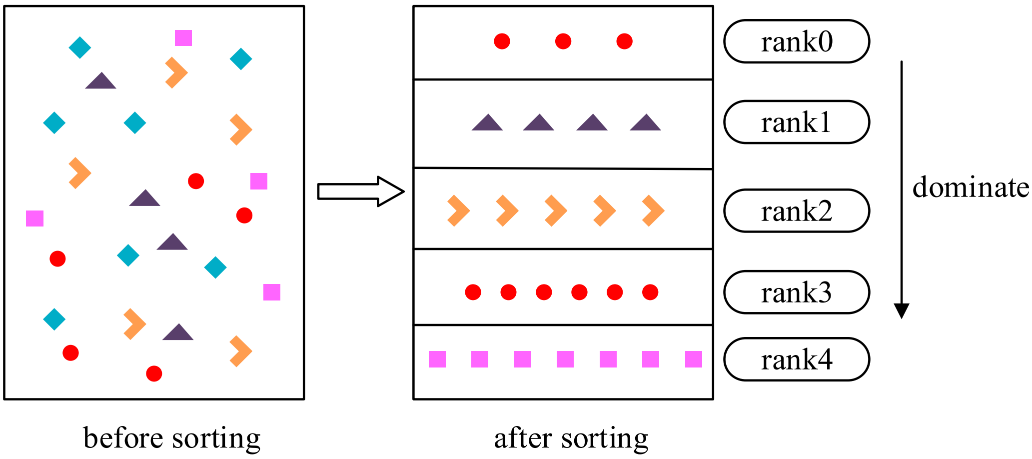


Figure 3. Non-dominated sorting.

After individual stratification, it is necessary to distinguish the individuals of the same layer. The crowding distance is used to distinguish the advantages and disadvantages among individuals. The formula for calculating the crowding distance of individuals is shown in Formula (10). The individuals with larger crowding distance are far away from other individuals. According to the crowding distance, the distribution uniformity of solution set can be judged.

$$P[i]_{dis\tan ce} = \frac{P[i+1]\bullet f_1 - P[i-1]\bullet f_1}{f_1^{\max} - f_1^{\min}} + \frac{P[i+1]\bullet f_2 - P[i-1]\bullet f_2}{f_2^{\max} - f_2^{\min}} \tag{10}$$

where $P[i]_{\text{distance}}$ denotes the crowding distance of an individual: $P[i]\bullet f_1$ and $P[i]\bullet f_2$ represent two objective function values of individual i;$f_1^{\max}$, $f_1^{\min}$ denote the maximum and minimum values of the objective function f_1, respectively; $f_2^{\max}$, $f_2^{\min}$ denote the maximum and minimum values of the objective function f_2, respectively.

3.4. Intelligent Behavior Design

For each of the three intelligent behaviors in WPA, the crossover and variation operators from the genetic algorithm are incorporated to maintain the diversity of feasible solutions and enhance the local search capability of the algorithm. Additionally, the elite retention strategy and non-dominated ranking method are employed to improve the algorithm's ability to seek promising solutions. Considering the encoding method and features of FJSP, efficient crossover and mutation operations have been specifically designed to prevent the generation of illegal solutions and ensure the validity of the solutions after applying intelligent behaviors. The wandering behavior incorporates a double-layer mutation, the summoning behavior utilizes the POX crossover [25], and the besieging behavior incorporates a mutation operator.

Wandering behavior: take the process wandering and machine wandering in two ways. For wandering walking, as shown in Figure 4, first, according to the process code, the walking step length $step_{a1}$ is defined as the number of individual position vectors for the detection wolf to walk, $step_{a1}$ process codes containing different workpieces are randomly extracted, randomly sorted, and then the sorted codes are placed in the spare position of the original process code in order.

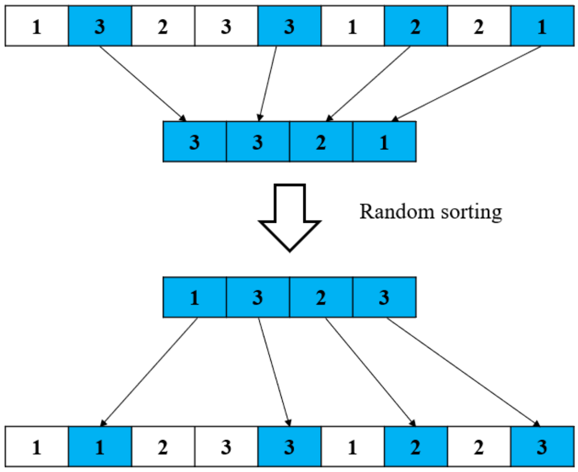


Figure 4. Process wandering behavior.

For machine code wandering operation, assuming machine wandering step $step_{a2}= 1$, the process of any one processing machine set of no less than two machines is randomly selected in its corresponding machine set, as shown in Figure 5.

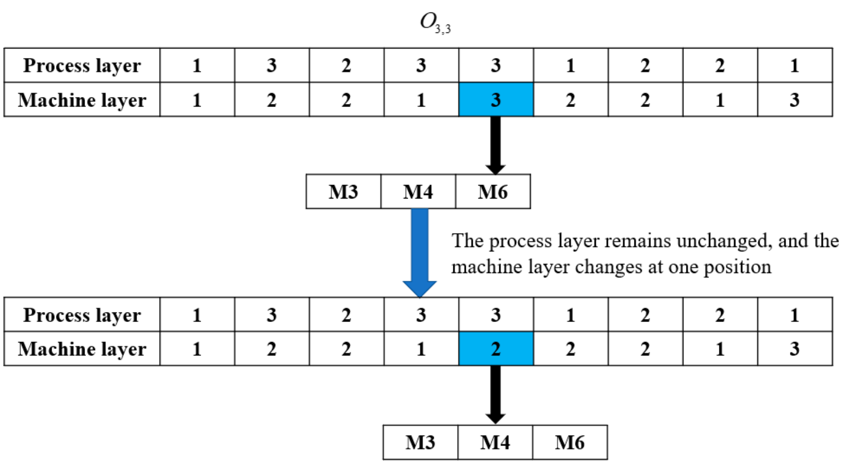


Figure 5. Machine wandering behavior.

(2) Calling behavior. The wolf pack is ranked using the non-dominated crowding degree ranking method, and one of the solution sets is randomly selected from the optimal Pareto solution set as X_{leader}. The POX crossover operation is then performed as follows: the workpiece serial numbers are randomly assigned to two non-empty and complementary sets Q_1 and Q_2, the workpiece serial numbers containing the set Q_1 are selected from the parent X_1, the position of each workpiece serial number is kept unchanged, and copied to

the child $X_1{}'$. The set Q_2 workpiece serial numbers are selected from the parent X_{leader}, and these are inserted to the vacant positions of the child $X_1{}'$ in order. Similarly, the workpiece serial numbers from the parent X_{leader} containing the set Q_1 are selected, while the position of each workpiece serial number is kept unchanged, and then copied to the child $X_{\text{leader}}{}'$. The set Q_2 workpiece serial numbers from the parent X_1 are selected and inserted into the vacant positions of the child $X_{\text{leader}}{}'$, in order. The POX crossover operation is shown in Figure 6.

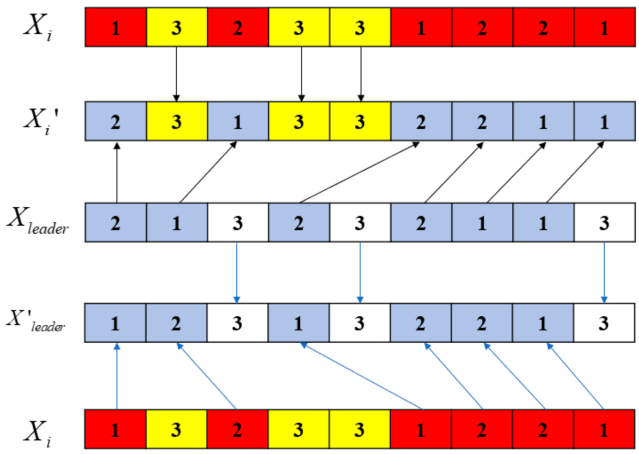


Figure 6. Schematic diagram of POX crossover operation.

The siege behavior is only for the process code, and the machine code can be transformed accordingly. Similar to the improved wandering behavior, the siege step size is set to $step_c$ and defined as an integer. For example, the process code of artificial wolf X_i is [1, 1, 2, 3, 3, 1, 2, 2, 3] and its individual code number is 9. The siege step $step_c$ will be taken as a random number of [0, 9]. Due to the large setting of the siege step size, it is easy for the value to jump out of the optimal solution range. Generally, the step size is set to be 1/3 to 1/2 of the number of individual codes.

The wolf pack update mechanism is achieved by using a non-dominated crowding sorting method after conducting a siege behavior to remove the R artificial wolves with the lowest odor concentration value (i.e., the higher objective function value) and generate R artificial wolves. Generally $R \in [M/(2 \times \beta), M/\beta]$, β is the population update proportion factor, and M is the number of artificial wolves.

3.5. Algorithm Flow

To sum up, the flow chart of the MOWPA algorithm steps is shown in Figure 7, and the details are described as follows:

Step 1: Initialize the algorithm parameters.

Step 2: Set the external file $Q = \varnothing$, calculate the objective function value of each artificial wolf in the initial population, layer the individuals through rapid non-dominated sorting, and update the external files set.

Step 3: Calculate the fitness value, select some of the better artificial wolves to perform the double walk behavior of process coding and machine coding for the detection wolves, update the location of the detection wolves and judge whether the number of walks reaches the maximum number of walks $T_{\max}$; if so, go to step 4.

Step 4: The remaining artificial wolves are selected as the fierce wolves, and the detecting wolves initiate the summoning behavior, and the POX crossover with the fierce wolves is randomly selected among the detection wolves to calculate the prey odor concentration value perceived by each artificial wolf and update the location of the fierce wolves.

Step 5: The detection wolf teams up with the fierce wolf to execute the siege behavior. In this behavior, the artificial wolf position with the best fitness value for each optimized subgoal is randomly selected as the target for the siege. After the siege behavior is completed, each artificial wolf position is updated, the optimized objective function value is calculated and recorded, and the Pareto better solution is obtained. The external profile set is then updated by sorting the individuals.

Step 6: Renewing populations according to the survival of the strongest.

Step 7: Determine whether the algorithm has reached the termination condition. If it has, output a set of optimal solutions from the Pareto optimal solution set. Otherwise, proceed to step 3.

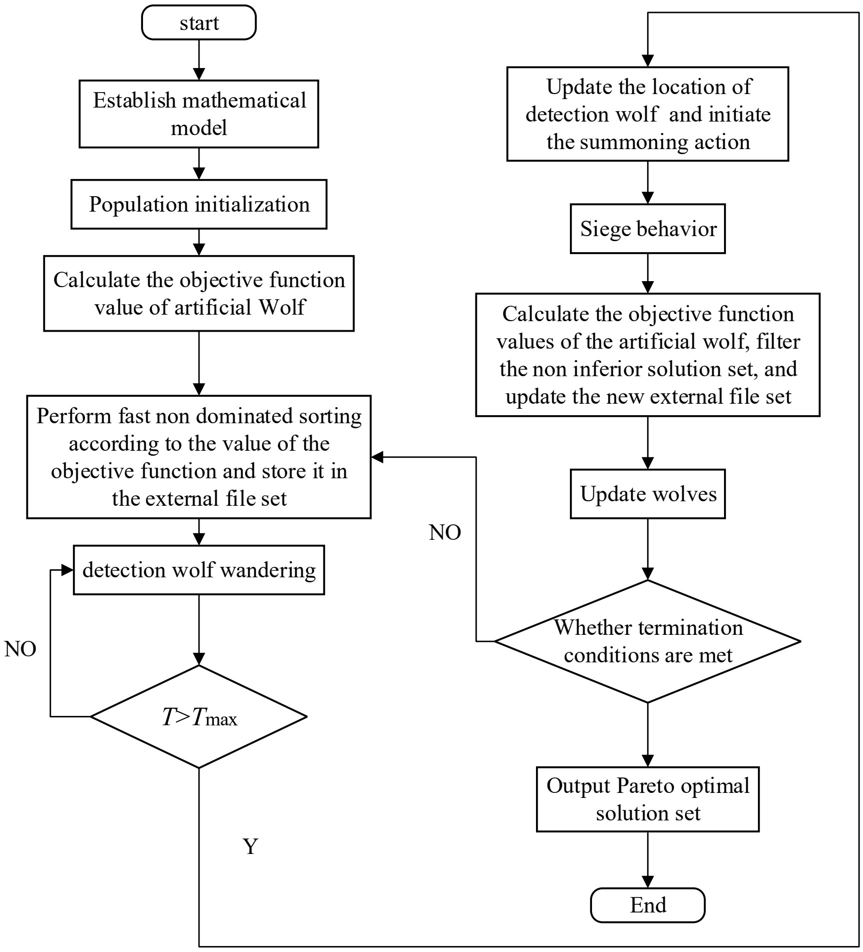


Figure 7. Flow chart of MOWPA algorithm.

4. Simulation Testing and Analysis

4.1. Test Example

In this paper, the Brandimarte example [26] is adopted as a benchmark case. However, since the model in this study incorporates energy consumption as an index, additional data need to be generated and extended accordingly. Random data within a reasonable range were generated, and the corresponding values are presented in Table 3. The table includes transportation energy consumption and transportation time, which have been standardized to a unified dimension.

Table 3. Energy Consumption for Machine Processing.

Machine Power	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}
Processing power	2	1.8	1.6	2.4	2.4	4.1	3.5	4.1	2.8	2.7
standby power	0.5	0.6	0.3	0.4	0.4	0.6	0.8	0.9	0.3	0.4

The transport time of the workpiece in each machine is shown in Table 4. The data in the table express the time required for the workpiece to be transported from machine n to machine M. The transporting time was set as a random integer in the [1,5] interval. The transporting power of the transporting equipment is fixed and its value is $P_{\text{trance}} = 1.89$, which is the unit time power.

Table 4. Transportation time.

Machines	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}
M_1	0	2	1	2	4	3	4	3	2	3
M_2	2	0	2	2	3	2	3	4	4	2
M_3	1	2	0	3	2	4	3	1	5	2
M_4	2	2	3	0	4	4	3	4	3	2
M_5	4	3	2	4	0	1	4	4	3	4
M_6	3	2	4	4	1	0	4	3	2	2
M_7	4	3	3	3	4	4	0	5	1	3
M_8	3	4	1	4	4	3	5	0	4	1
M_9	2	4	5	3	3	2	1	4	0	3
M_{10}	3	2	2	2	4	2	3	1	3	0

The MOWPA algorithm parameters were configured according to Table 5 using MK04 as the test data. By applying the MOWPA algorithm for optimization, the maximum completion time and workshop energy consumption values were obtained, as presented in Table 6. The table displays 11 sets of Pareto solutions generated by the MOWPA algorithm. Each set comprises the maximum completion time and the total energy consumption. The energy consumption values in each set represent the corresponding energy consumption values during the processing stages.

Table 5. Parameter of MOWPA algorithm.

Parameter Name	Numerical Value
Population number	200
Iterations	100
External archive collection size	100
Maximum number of walking	10
Procedure walking step	6
Machine Walking Steps	4
Siege steps	6
Detection wolf scale factor	0.4
Update scale factor	0.3

Table 6. Pareto solution set of MOWPA.

Serial Number	Maximum Completion Time f_1	Total Workshop Energy Consumption f_2	Energy Consumption of Each Part		
			Transportation Energy Consumption	Processing Energy Consumption	Standby Energy Consumption
1	87	1332.60	359.10	856.50	117.00
2	89	1320.76	347.76	854.60	118.40
3	90	1311.08	343.98	836.50	130.60
4	91	1300.91	357.21	836.20	130.50
5	92	1270.43	334.53	828.70	107.20
6	93	1267.99	342.09	826.80	99.10
7	94	1259.91	338.31	824.60	97.00
8	95	1242.32	336.42	805.10	100.80
9	100	1219.09	323.19	765.50	130.40
10	101	1210.43	320.53	763.30	126.60
11	106	1189.47	308.07	743.80	137.60

4.2. Selection of Pareto Optimal Solution Set

For multi-objective problems, there are multiple solutions in the resulting set of Pareto solutions, which makes it difficult for the decision maker to select a better scheduling solution to process the product. Therefore, in order to facilitate the decision maker to better select an optimal scheduling solution from the Pareto solution set, a combination of AHP and gray correlation decision analysis was used to select the Pareto solution set. First, the weight value of each index was obtained through the analytic hierarchy process. After obtaining the weight value of each objective, the gray correlation decision analysis method was used to obtain the optimal solution under the current weight from a group of optimal solutions. The advantage of combining AHP with gray correlation decision analysis is that it not only combines the subjectivity of the decision maker to assign weights to the goal according to the current conditions, but also quantitatively analyses the data obtained from the optimal solution to select the optimal solution.

(1) Calculate the weight value

In order to find the weight value using the hierarchical analysis method, firstly, the judgment matrix is established by the decision makers such as schedulers by comparing the importance of each index and by quantifying the judgment matrix of each index. The weights of the two indices under the maximum completion time and energy consumption are solved according to the nine-level scale method and in combination with production practice. If the order a_{21}= 2, then the indices f_2 (total energy consumption) are slightly more important than the indices f_1 (maximum completion time). The resulting judgment matrix and the weights of each indicator are shown in Table 7.

Table 7. Index judgement matrix and weight value.

Indicators	f_1	f_2	**Weights** ω
f_1	1	1/2	1/3
f_2	2	1	2/3

(2) Data normalization

The purpose of data normalization is to eliminate the difference between variables due to different dimensions and thus eliminate the influence on the results. The normalization method used here is shown in Formula (11).

$$N_{i,j} = \frac{Y_{i,j} - Y_j^{\min}}{Y_j^{\max} - Y_j^{\min}} \tag{11}$$

where $N_{i,j}$ is the matrix after $Y_{i,j}$ normalization and $Y_{i,j}$ is the raw data, representing the j-th objective function value of the group i data, and $Y_j^{\max}$ and $Y_j^{\min}$ are the maximum and minimum values of the j column of the original matrix Y respectively.

(3) Calculation of gray correlation coefficient

The gray correlation coefficient $\gamma_{i,j}$ reflects the degree of association between the j-th indicator of the i-th data set and the ideal value.

$$\gamma_{i,j} = \frac{N_j^{\min} + \rho N_j^{\max}}{N_{i,j} + \rho N_j^{\max}} \tag{12}$$

where $N_j^{\min}$ and $N_j^{\max}$ are, respectively, the minimum and maximum values in the index data group after normalization. ρ is the resolution coefficient, generally taken as 0.5.

(4) Calculation of gray correlation degree

The gray relational degree is the product of the gray relational coefficient and the corresponding weight. The weight value of each indicator has been obtained from the AHP. The calculation method of the gray relational degree is shown in Formula (13).

$$R_i = \sum_{j=1}^{m} \gamma_{i,j} w_j \tag{13}$$

By calculation, the data are shown in Equation (14). Where Y is the raw data, the first and second columns correspond to the maximum completion time and total energy consumption of the workshop, respectively. N is the normalized matrix corresponding to the original data obtained by data normalization, γ is the gray correlation coefficient matrix, ω is the weight matrix of the two objectives obtained by analytic hierarchy process, $\omega = [0.33, 0.67]$, R is the gray correlation matrix.

$$Y = \begin{bmatrix} 87 & 1332.60 \\ 89 & 1320.76 \\ 90 & 1311.08 \\ 91 & 1300.91 \\ 92 & 1370.43 \\ 93 & 1267.99 \\ 94 & 1259.91 \\ 95 & 1242.32 \\ 100 & 1219.09 \\ 101 & 1210.43 \\ 106 & 1189.47 \end{bmatrix}, N = \begin{bmatrix} 0.000 & 1.000 \\ 0.105 & 0.917 \\ 0.158 & 0.850 \\ 0.211 & 0.779 \\ 0.263 & 0.566 \\ 0.316 & 0.549 \\ 0.368 & 0.492 \\ 0.421 & 0.369 \\ 0.684 & 0.207 \\ 0.737 & 0.146 \\ 1.000 & 0.000 \end{bmatrix}, \gamma = \begin{bmatrix} 1.000 & 0.333 \\ 0.826 & 0.353 \\ 0.760 & 0.370 \\ 0.704 & 0.391 \\ 0.655 & 0.469 \\ 0.613 & 0.477 \\ 0.576 & 0.504 \\ 0.543 & 0.575 \\ 0.422 & 0.707 \\ 0.404 & 0.733 \\ 0.333 & 1.000 \end{bmatrix} \tag{14}$$

The weight matrix $\omega = [0.33, 0.67]$ and the data of Equation (14) are substituted into Equation (13) to obtain the correlation matrix R, as shown in Formula (15).

$$R = [0.556\ 0.511\ 0.500\ 0.495\ 0.531\ 0.522\ 0.528\ 0.564\ 0.612\ 0.650\ 0.778]^T \tag{15}$$

The larger the value of R, the better the effect of the corresponding solution under this weight. From the correlation matrix R, it can be seen that the 11th group of data has the largest correlation of 0.778, which corresponds to a maximum completion time of 106 and a total shop floor energy consumption of 1189.47. Therefore, the scheduling solution corresponding to the 11th group of scheduling optimization results is selected for processing, and its scheduling Gantt chart is shown in Figure 8.

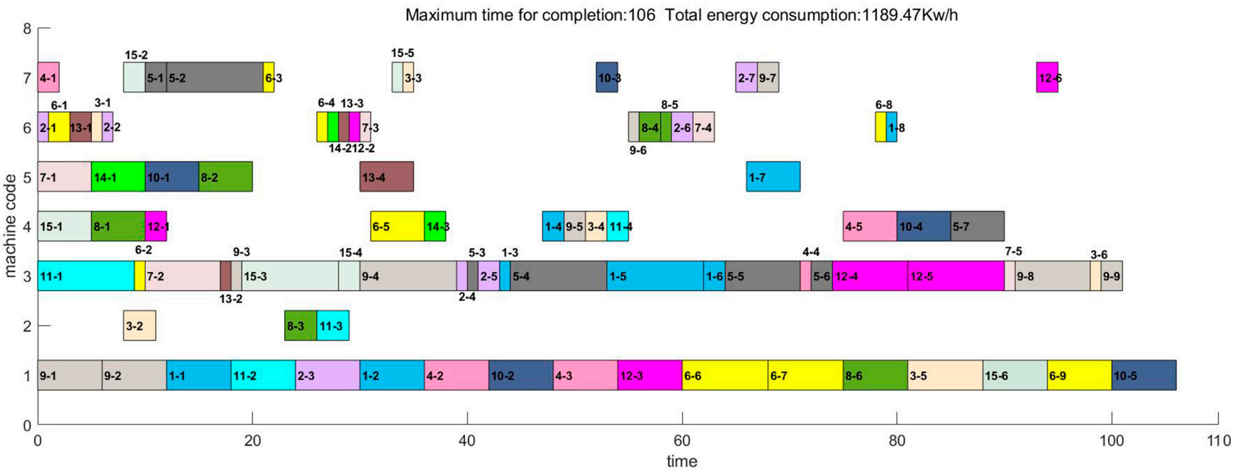


Figure 8. Optimal decision processing Gantt chart.

4.3. Algorithm Performance Evaluation

The parameter settings of the MOWPA algorithm are shown in Table 5. To verify the performance of the algorithm, this article aims to establish a multi-objective mathematical model that considers transportation time and energy consumption to minimize the maximum completion time and minimum energy consumption. MOWPA and NSGA-II are used for solving.

The Pareto optimal solution set obtained through simulation is shown in Table 8. The data format in the Pareto solution set column in the table is (x; y), where x represents the maximum completion time and y represents the total energy consumption of the workshop. From the table, It is evident that the solved range distribution exhibits greater width and uniformity.

Table 8. Pareto solution set table for MOWPA algorithm and NSGA-II algorithm.

Test Data	The Set of Pareto Solutions Obtained by MOWPA	The Set of Pareto Solutions Obtained by NSGA-II
MK01	(46;525.53), (52;516.62), (58;513.43) (47;522.11), (51;518), (45;526.91) (44;530)	(49;546.97), (50;539.37), (51;532.77) (52;524.79), (58;501.11)
MK02	(37;526.91), (38;519.49), (39;514.91) (40;513.09), (41;512.51), (42;512.11) (43;511.47), (44;510.89), (45;509.49)	(38;539.11), (39;534.44), (40;530.68) (43;527.56), (44;522.98)
MK03	(208;3,452.10), (209;3,430.66) (212;3,418.42), (213;3,399.43) (215;3,384.83)	(210;3,462.53), (211;3,439.23) (214;3,410.55), (217;3,398.75) (218;3,375.26)
MK04	(87;1,332.6), (89;1,320.76), (90;1,311.08) (91;1,300.91), (92;1,270.43), (93;1,267.99) (94;1,259.91), (95;1,242.32) (100;1,219.09), (101;1,210.43) (110;1,189.47)	(89;1,357.97), (93;1,337.47), (94;1,313.83) (100;1,289.51), (102;1,276.07), (104;1,259.23) (109;1,251.44)
MK05	(180;1,632.63), (183;1,630.21) (185;1,626.91), (186;1,624.62) (189;1,623.4)	(182;1,669.95), (183;1,652.78) (184;1,644.08), (186;1,625.29)
MK06	(108;1,740.01), (109;1,719.93) (110;1,699.66), (111;1,696.25) (112;1,689.56), (113;1,688.45) (114;1,685.01), (115;1,683.09)	(108;1,799.12), (109;1,750.59) (110;1,740.88), (112;1,730.54) (113;1,720.03), (115;1,690.67)
MK07	(144;1,745.22), (145;1,738.03) (146;1,735.03), (150;1,730.65) (155;1,724.42), (160;1,720.72) (162;1,719.44)	(146;1,766.53), (147;1,754.86) (149;1,750.31), (150;1,743.33)

In order to compare the convergence performance of the two algorithms, this article uses the widely used Coverage (C) [6] and Inverted Generational Distance (IGD) [27] in multi-objective optimization problems as evaluation algorithm indicators. Their meanings and formulas are as follows.

$$C(F_1, F_2) = \frac{|\{sol_2 \in F_2 | \exists sol_1 \in F_1 : sol_1 \succ sol_2\}|}{|F_2|} \tag{16}$$

where F_1 and F_2 are the Pareto fronts by the two algorithms, respectively, and $|F_2|$ is the size of F_2. The larger $C(F_1, F_2)$ is, the better the surface F_1 is. For example, $C(F_1, F_2) = 1$ means that all solutions in F_2 are dominated by F_1, and $C(F_1, F_2) = 0$ means that there is no solution in F_1 that can dominate F_2.

$$IGD(F_1, F*) = \frac{1}{|F*|} \sum_{sol_1 \in F*} \min_{sol_2 \in F_1} d(sol_1, sol_2) \tag{17}$$

where, F^* is the non-dominated solution set of the first frontier, $|F^*|$ is the size of F^*, $d(sol_1, sol_2)$ representing the Euclidean distance between sol_1 and sol_2. The smaller the

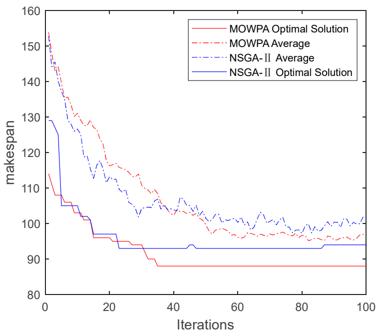
$IGD(F_1, F^*)$, the better the F_1. In this paper, the F^* of each example is formed by averaging the non-dominated solution set obtained after each algorithm runs 20 times, respectively. The results are shown in Table 9.

Table 9. Comparison results of calculation examples.

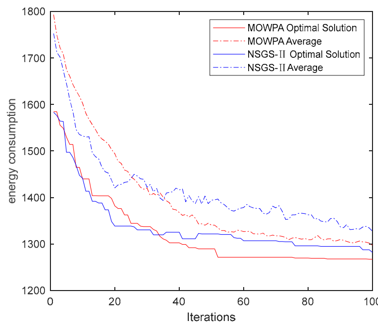
Test Data	*C* (MOWPA, NSGA-II)	*IGD* (MOWPA)	*IGD* (NSGA-II)
MK01	1.00	1.2070	4.8498
MK02	1.00	1.3956	1.2903
MK03	0.60	1.1363	2.3480
MK04	1.00	3.9281	4.5960
MK05	1.00	0.5362	3.0486
MK06	1.00	3.6433	8.4704
MK07	1.00	1.5487	0.8967

Table 9 reveals that in the Brandimarte case, the solutions obtained by the MOWPA algorithm dominate the solutions obtained by the NSGA-II algorithm in the majority of cases. Only in the case of MK03 did there exist individual solutions in NSGA-II that are not dominated. However, considering the overall results, it is evident that the MOWPA algorithm outperforms the NSGA-II algorithm. This implies that the Pareto frontier generated by MOWPA is superior to that of NSGA-II. Additionally, based on the IGD index, it can be observed that the IGD value of MOWPA is consistently smaller than that of NSGA-II in most cases. This indicates that the proposed MOWPA algorithm exhibits better convergence performance compared to the NSGA-II algorithm.

Figure 9 illustrates the population iteration diagram of the maximum completion time and total energy consumption values obtained using the MOWPA algorithm and the NSGA-II algorithm with MK04 data. In Figure 9a, which displays the population iterations for the maximum completion time, the red solid and dashed lines represent the optimal and average values, respectively, obtained by the MOWPA algorithm. Similarly, the blue solid and dashed lines represent the optimal and average values, respectively, obtained by the NSGA-II algorithm. The MOWPA algorithm achieves a stable optimal maximum completion time after approximately 35 generations, while the NSGA-II algorithm achieves this after around 25 generations. Although the MOWPA algorithm has a slower convergence rate compared to the NSGA-II algorithm, it provides better solution accuracy. Figure 9b represents the population iteration diagram for total energy consumption. The red solid line and dotted line correspond to the optimal and average values, respectively, obtained by the MOWPA algorithm. Similarly, the blue solid line and dotted line represent the optimal and average values, respectively, obtained by the NSGA-II algorithm. The MOWPA algorithm maintains a stable optimal solution around 52 generations, whereas the NSGA-II algorithm exhibits more fluctuation. Overall, the MOWPA algorithm demonstrates superior speed and precision compared to the NSGA-II algorithm in terms of both maximum completion time and total energy consumption.



(**a**)



(**b**)

Figure 9. Iterative curves with different objectives; (**a**) Maximum completion time; (**b**) Total energy consumption.

5. Conclusions

This paper explores the multi-objective flexible job shop scheduling problem, taking into account transportation time and energy consumption. We establish a multi-objective mathematical model for a flexible job shop with transportation constraints, where the optimization objectives are maximizing the maximum completion time and minimizing total energy consumption. To address the characteristics of the MOFJSP problem, we propose an enhanced multi-objective wolf pack algorithm as a solution approach. The improvements include designing a coding scheme, introducing a mixed initialization strategy, incorporating crossover and mutation operators, and applying a non-dominated sorting method. Moreover, we extend the traditional algorithm to solve for the optimal Pareto solution set, making it more relevant to real production scenarios. To evaluate the importance of each index, we combine the AHP with the gray relational decision analysis method. This approach allows us to compare the significance of different factors using AHP and quantify their respective weights. Subsequently, we employ gray relational analysis for decision analysis. By combining qualitative and quantitative methods, we can obtain the optimal processing scheme from the Pareto solution set under the current weight settings. This enables enterprises to select the optimal scheduling strategy from a variety of solutions. To verify the algorithm's performance, we compare the proposed algorithm with the non-dominated sorting genetic algorithm. The comparison results demonstrate the superior performance of the proposed algorithm in terms of solving performance and solution distribution. It provides a better decision-making basis for flexible job shop scheduling.

However, this method can only be applied to static flexible job shop scheduling problems. When dealing with dynamic flexible job shop scheduling problems, such as machine failures or changes in workpiece quantities, the data needs to be reprocessed and the calculation process is complicated. Therefore, in the future research, the solution method of the algorithm and the accuracy and efficiency of the algorithm will be further improved, so that it can be more in line with actual production workshop processing.

Author Contributions: Conceptualization, J.L. and H.L.; Data curation, H.L. and P.H.; Funding acquisition, J.L., L.X., K.H. and S.L.; Methodology, J.L.; Writing—original draft, J.L. and H.L.; Writing—review and editing, L.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key R&D Program of China [grant number 2018YFB1701205]; and the Technology Innovation Leading Program of Shanxi Province [grant number 2020QFY03-08].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Garey, M.R.; Sethi, J.R. The complexity of flow shop and job shop scheduling. *Math. Oper. Res.* **1976**, *1*, 117–129. [CrossRef]
2. Shen, L.; Dauzère, P.; Stéphane; Neufeld, J.S. Solving the flexible job shop scheduling problem with sequence-dependent setup times. *Eur. J. Oper. Res.* **2018**, *265*, 503–516. [CrossRef]
3. Homayouni, S.M.; Fontes, D.; Gonalves, J.F. A multistart biased random key genetic algorithm for the flexible job shop scheduling problem with transportation. *Int. Trans. Oper. Res.* **2020**, *30*, 688–716. [CrossRef]
4. Peng, K.; Pan, Q.K.; Gao, L.; Li, X.; Das, S.; Zhang, B. A multi-start variable neighbourhood descent algorithm for hybrid flow shop rescheduling. *Swarm Evol. Comput.* **2019**, *45*, 92–112. [CrossRef]
5. An, Y.; Chen, X.; Zhang, J.; Li, Y. A hybrid multi-objective evolutionary algorithm to integrate optimization of the production scheduling and imperfect cutting tool maintenance considering total energy consumption. *J. Clean. Prod.* **2020**, *268*, 121540. [CrossRef]
6. Zheng, X.; Wang, L. A Collaborative Multiobjective Fruit Fly Optimization Algorithm for the Resource Constrained Unrelated Parallel Machine Green Scheduling Problem. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *48*, 790–800. [CrossRef]
7. Luo, S.; Zhang, L.; Fan, Y. Energy-efficient scheduling for multi-objective flexible job shops with variable processing speeds by grey wolf optimization. *J. Clean. Prod.* **2019**, *234*, 1365–1384. [CrossRef]

8. Wu, X.; Sun, Y. A Green Scheduling Algorithm for Flexible Job Shop with Energy-Saving Measures. *J. Clean. Prod.* **2018**, *172 Pt 3*, 3249–3264. [CrossRef]
9. Peng, Z.; Zhang, H.; Tang, H.; Feng, Y.; Yin, W. Research on flexible job-shop scheduling problem in green sustainable manufacturing based on learning effect. *J. Intell. Manuf.* **2022**, *33*, 1725–1746. [CrossRef]
10. Hasani, A.; Hosseini, S. A bi-objective flexible flow shop scheduling problem with machine-dependent processing stages: Trade-off between production costs and energy consumption. *Appl. Math. Comput.* **2020**, *386*, 125533. [CrossRef]
11. Zhu, Z.; Zhou, X. An efficient evolutionary grey wolf optimizer for multi-objective flexible job shop scheduling problem with hierarchical job precedence constraints. *Comput. Ind. Eng.* **2020**, *140*, 106280. [CrossRef]
12. Caldeira, R.H.; Gnanavelbabu, A. A Pareto based discrete Jaya algorithm for multi-objective flexible job shop scheduling problem. *Expert Syst. Appl.* **2021**, *170*, 114567. [CrossRef]
13. Kui, C.; Li, B. Research on FJSP based on improved particle swarm optimization algorithm considering transportation time. *J. Syst. Simul.* **2021**, *4*, 845–853.
14. Xiabao, H.; Shuling, W. High dimensional multi-objective flexible job shop scheduling considering low carbon. *J. Wuhan Univ. Technol. Inf. Manag. Eng. Ed.* **2019**, *41*, 592–598.
15. Foumani, M.; Smith-miles, K. The impact of various carbon reduction policies on green flowshop scheduling. *Appl. Energy* **2019**, *249*, 300–315. [CrossRef]
16. Chen, W.; Wang, J.; Yu, G.; Hu, Y. Energy-Efficient Hybrid Flow-Shop Scheduling under Time-of-Use and Ladder Electricity Tariffs. *Appl. Sci.* **2022**, *12*, 6456. [CrossRef]
17. Liu, Z.; Guo, S.; Wang, L. Integrated green scheduling optimization of flexible job shop and crane transportation considering comprehensive energy consumption. *J. Clean. Prod.* **2019**, *211*, 765–786. [CrossRef]
18. Chen, X.; Cheng, F.; Liu, C.; Cheng, L.; Mao, Y. An improved Wolf pack algorithm for optimization problems: Design and evaluation. *PLoS ONE* **2021**, *16*, e0254239. [CrossRef]
19. Yong, Y.; Huizhen, Z. Wolf swarm algorithm for vehicle routing problem with multiple distribution centers. *Comput. Appl. Res.* **2017**, *34*, 2590–2593.
20. Shan, G.; Meng, L. Greedy stochastic adaptive gray wolf optimization algorithm for solving TSP problems. *Mod. Electron. Technol.* **2019**, *42*, 50–54.
21. Liwen, W.; Shuyi, S.; Qingxian, W.; Zengliang, H. Unmanned Helicopter Trajectory Planning Based on Improved Wolf Pack Algorithm. Systems Engineering and Electronics Technology 2023.07.22. Available online: http://kns.cnki.net/kcms/detail/11.2422.TN.20221228.1122.005.htm (accessed on 17 June 2023).
22. Shikui, Z. Two-Level Neighborhood Search Hybrid Algorithm for Flexible Workshop Scheduling Problem. *J. Mech. Eng.* **2015**, *51*, 175–184.
23. Jinliang, S.; Fei, L.; Dijian, X.; Guorong, C. Energy-saving Decision-making Model and Practical Method of CNC Machine Tool in No-load Operation. *China Mach. Eng.* **2009**, *20*, 1344–1346.
24. Husheng, W.; Fengming, Z.; Lushan, W. A new swarm intelligence algorithm—The wolf swarm algorithm. *Syst. Eng. Electron. Technol.* **2013**, *35*, 2430–2438.
25. Chaoyong, Z.; Yunqing, R.; Xiangjun, L.; Peigen, L. A Genetic Algorithm Based on POX Intersection for Solving Jobs Shop Scheduling Problem. *China Mech. Eng.* **2004**, *23*, 83–87.
26. Brandimarte, P. Routing and scheduling in a flexible job shop by tabu search. *Ann. Oper. Res.* **1993**, *41*, 157–183. [CrossRef]
27. Czyzzak, P.; Jaszkiewicz, A. Pareto simulated annealing—A metaheuristic technique for multiple-objective combinatorial optimization. *J. Multi-Criteria Decis. Anal.* **1998**, *7*, 34–47. [CrossRef]