applied sciences

MDPI

Article

Hybrid Salp Swarm Algorithm for Solving the Green Scheduling Problem in a Double-Flexible Job Shop

Changping Liu [1,*], Yuanyuan Yao [2] and Hongbo Zhu [3]

[1] Faculty of Management Engineering, Huaiyin Institute of Technology, Huaian 223200, China
[2] College of Economics and Management, Suzhou Chienshiung Institute of Technology, Taicang 215411, China; tracy_yao11@163.com
[3] Faculty of Mathematics and Physics, Huaiyin Institute of Technology, Huaian 223003, China; zhb8151@hyit.edu.cn
[*] Correspondence: lcpzwy@hyit.edu.cn; Tel.: +86-0517-8355-9603

Abstract: Green scheduling is not only an effective way to achieve green manufacturing but also an effective way for modern manufacturing enterprises to achieve energy conservation and emission reduction. The double-flexible job shop scheduling problem (DFJSP) considers both machine flexibility and worker flexibility, so it is more suitable for practical production. First, a multi-objective mixed-integer programming model for the DFJSP with the objectives of optimizing the makespan, total worker costs, and total influence of the green production indicators is formulated. Considering the characteristics of the problem, three-layer salp individual encoding and decoding methods are designed for the multi-objective hybrid salp swarm algorithm (MHSSA), which is hybridized with the Lévy flight, the random probability crossover operator, and the mutation operator. In addition, the influence of the parameter setting on the MHSSA in solving the DFJSP is investigated by means of the Taguchi method of design of experiments. The simulation results for benchmark instances show that the MHSSA can effectively solve the proposed problem and is significantly better than the MSSA and the MOPSO algorithm in the diversity, convergence, and dominance of the Pareto frontier.

Keywords: double flexible job-shop scheduling; multi-objective optimization; hybrid salp swarm algorithm; green shop scheduling

check for updates

Citation: Liu, C.; Yao, Y.; Zhu, H. Hybrid Salp Swarm Algorithm for Solving the Green Scheduling Problem in a Double-Flexible Job Shop. *Appl. Sci.* **2022**, *12*, 205. https://doi.org/10.3390/app12010205

Academic Editor: Chun-Cheng Lin

Received: 1 October 2021
Accepted: 20 December 2021
Published: 25 December 2021

1. Introduction

Traditional job shop scheduling problems usually do not consider human factors. With increasing research on job shop scheduling problems, researchers believe that human factors should be considered in production scheduling and operation management activities [1–4]. Based on the flexible job shop scheduling problem (FJSP), Gong et al. [5] proposed a double-flexible job shop scheduling problem (DFJSP) for the first time, which is more in line with the actual production situation. That is, each process of each workpiece can be processed by multiple machines, and each machine can be operated by multiple workers. The DFJSP is an NP hard combinatorial optimization problem that is an extension of FJSP. At present, there is no polynomial time algorithm to solve the problem [5].

In recent years, with the increasingly severe environmental problems, research on green manufacturing has attracted a lot of attention. For example, He and Liu [6] studied the issue of reducing energy consumption for the job-shop scheduling problem, constructed the biobjective model of minimizing both the energy consumption and the makespan, and used the Tabu search algorithm to optimize the model. Mouzon et al. [7] proposed a multi-objective mathematical programming model of minimizing the energy consumption and total completion time. By solving the model, we can get a set of non-dominated solutions to determine the most efficient production sequence. The model only considers the energy saving by shutting down non-bottleneck or idle equipment, and it does not consider the energy saving of running machines and equipment. Dai and Tang et al. [8] modeled an energy-efficient scheduling with makespan and energy consumption. The results showed that the relationship between the two objectives was apparently one of conflict. Similar results were also proved in the study of Liu and Huang [9]. Mansouri et al. [10] developed a mixed integer linear multi-objective optimization model to find the Pareto frontier comprised of makespan and total energy consumption, and they defined lower bounds for the objectives and proposed a heuristic for the problem. In addition, some scholars have also carried out similar studies related to green production (see references [11–15], etc.). The above research results mainly focus on the theoretical model, which is quite different from the actual green manufacturing and production scheduling. Green production indicators need to be used to build scheduling models closer to the actual situation.

On the other hand, green scheduling is an important part of green manufacturing, and some scholars have studied it. For example, in reference [16], green job shop scheduling is defined as efficiency increasing, energy saving, emission reduction, and consumption reduction, which improve economic benefits and lead to a green manufacturing process through resource allocation, operation sequencing, and reasonable optimization of the operation mode. Wang et al. [17] analyzed the complexity of the green job shop scheduling problem and the corresponding processing mechanism and summarized and reviewed the representative research on green job shop scheduling from the problem, method, and application levels. To solve the flexible job shop scheduling problem, Lei et al. [18] proposed a hybrid frog-leaping algorithm that considers the energy consumption index. Wu et al. [19] took completion time and energy consumption as the optimization objectives, studied the flexible job shop problem with multi-speed machines, analyzed the energy consumption distribution of machines, established the energy consumption calculation model, and solved it by the NSGA-II algorithm. Dong and Ye [20] constructed a scheduling model for green manufacturing collaborative optimization of the semiconductor wafer distributed heterogeneous factory and proposed an improved gray wolf algorithms to solve the model. To solve the process scheduling problem, Yao et al. [21] designed a multi-objective salp swarm algorithm to solve the process scheduling problem of a TFT-LCD panel array that considers energy saving. Nanthapodej et al. [22] proposed a variable neighborhood strategy adaptive search method to minimize energy consumption while also considering job priority and makespan control for parallel-machine scheduling problems. In addition, some researchers have also carried out similar studies related to green scheduling (see references [23–25], etc.). To sum up, the current research on green scheduling mostly focuses on optimizing green production indicators, such as energy consumption and carbon emissions, while there are few studies considering human factors in green scheduling; Moreover, there are still many indicators related to resource consumption and environmental impact, such as emissions of toxic substances and wastes, noise, workpiece chip recycling, etc. The research on green scheduling considering these factors is also rare. At present, there are still many tasks on green workshop scheduling to deal with, such as constructing new models closer to practical production scheduling using green production indicators and developing more efficient algorithms to solve the green scheduling problem.

In this paper, a double-flexible job shop green scheduling problem model is constructed with the objectives of minimizing the maximum completion time, the total labor cost, and the total green index. A multi-objective hybrid salp swarm algorithm is proposed to solve the complex job shop scheduling problem, such as the DFJSP. By testing a benchmark example and comparing it with two typical algorithms, the feasibility and effectiveness of the algorithm designed in this paper in solving the double flexible job shop green scheduling problem are verified.

2. Green Scheduling Problem Model of Double Flexible Job Shop

2.1. Problem Description

The DFJSP can be described as follows: n workpieces are processed on m parallel machines of different types, which can be operated by l workers with different skill levels. The workpiece i includes r_i processes, and the processing sequence of each process is given in advance. The *j*-th process O_{ij} of the workpiece *i* can be processed on any machine in the set of optional processing machines. At the same time, each machine can be operated by any worker in the set of optional workers. The solution of the DFJSP includes three sub-problems: machine selection, worker selection, and process scheduling. Its scheduling goal is to select the most suitable machine and matching operators for each process and determine the optimal processing sequence and start-up time of each process on each machine so that some performance indexes of the production system can be optimized.

The double-flexible job shop green scheduling problem constructed in this paper has three optimization objectives: minimizing the maximum completion time, total labor cost, and total green index. Among them, the maximum completion time represents the final completion time of all jobs and all processes, which reflects the production efficiency. The total labor cost is an important part of the production cost of an enterprise. By reducing the total labor cost, the operating cost of an enterprise can be reduced. The total green index in this paper includes energy consumption, noise, workpiece chip recycling, and safety, making it a comprehensive green production index. Minimizing the total green index can help enterprises save energy, reduce emission, and reduce the impact on the environment.

The assumptions and constraints to be met are as follows: (1) Each machine can handle only one process at a time. (2) Each process can be processed only once on one machine. (3) Each worker can operate only one machine at a time. (4) Each process can be processed only once by one worker. (5) There are sequential constraints between the same workpiece processes, and there are no sequential constraints between different workpiece processes. (6) Interruption is not allowed after a process starts. (7) All machines and workpieces are ready at zero hour. (8) The installation and adjustment time of the workpiece is not considered. According to the skill levels, workers are divided into three types. In addition, the processing time, the labor cost, and the energy consumption corresponding to each process, as well as the attribute values of various green indexes of the machine, are known and fixed.

2.2. Mathematical Model of the DFJSP

The mathematical symbols in the model and their meanings are as follows:

m: Number of machines;

n: Number of work pieces;

l: Number of workers;

i, h: Workpiece serial number; $i, h = 1, 2, \cdots, n$;

r_i: Total number of operations of workpiece i;

j, g: Operation sequence number, $j, g = 1, 2, \cdots, r$;

k: Machine serial number, $k = 1, 2, \cdots, m$;

s: Worker serial number, $s = 1, 2, \cdots, l$;

O_{ij}: The j-th process of workpiece i;

C_{ij}: Completion time of process O_{ij};

C_k: Completion time of machine k;

P_{ijks}: Processing time of the process O_{ij} operated by worker s on machine k;

L_{ijks}: Processing cost of the process O_{ij} operated by worker s on machine k;

EI_{ijks}: Green index value of the process O_{ij} operated by worker s on machine k;

E_{ijks}: Energy consumption of the process O_{ij} operated by worker s on machine k;

N_{ijk}: Noise generated by the process O_{ij} on machine k;

R_{ijk}: Chip recovery rate of the process O_{ij} processed on machine k;

S_{ijk}: Safety of the process O_{ij} processing on machine k;

$$x_{ijk} = \begin{cases} 1, \text{theprocess } O_{ij} \text{ is processed on machine } k \\ 0, \text{other} \end{cases}$$

$$x_{ijks} = \begin{cases} 1, \text{the process } O_{ij} \text{ is processed on machine k and operated by the worker s} \\ 0, \text{other} \end{cases}$$

Based on the research in [5], this paper constructs the following model where C_M represents the maximum completion time target, L_C represents the total labor cost, and G represents the total green index.

Target function:

$$MinC_M = \max_{1 \leq k \leq m} (C_k) \tag{1}$$

$$MinL_C = \sum_{i=1}^{n} \sum_{j=1}^{r_i} \sum_{k=1}^{m} \sum_{s=1}^{l} \left(L_{ijks} \times x_{ijks} \right) \tag{2}$$

$$MinG = \sum_{i=1}^{n} \sum_{j=1}^{r_i} \sum_{k=1}^{m} \sum_{s=1}^{l} \left(EI_{ijks} \times x_{ijks} \right) \tag{3}$$

$$EI_{ijks} = E_{ijks}{}' + N_{ijk}{}' + \left(1/R_{ijk} \right)' + \left(1/S_{ijk} \right)' \tag{4}$$

Constraints:

$$C_{ij} - C_{i(j-1)} \geq \ P_{ijks} x_{ijks}, \quad \forall i,j \ ; i = 1,2,\cdots,n; j = 2,3,\cdots,r_i \tag{5}$$

$$\sum_{k=1}^{m} \sum_{s=1}^{l} x_{ijks} = 1, \forall i,j; i = 1,2,\cdots,n; j = 1,2,\cdots,r_i \tag{6}$$

$$\begin{gathered} \left[\left(C_{hg} - C_{ij} - t_{hgk} \right) x_{hgk} x_{ijk} \geq 0 \right] \vee \left[\left(C_{ij} - C_{hg} - t_{ijk} \right) x_{hgk} x_{ijk} \geq 0 \right], \\ \forall i,j,h,g,k; i,h = 1,2,\cdots,n; j = 1,2,\cdots,r_i; g = 1,2,\cdots,r_h; k = 1,2,\cdots,m; \end{gathered} \tag{7}$$

Here, Equations (1)–(3) represent the objective functions of maximum completion time, total labor cost, and total green index, respectively. Equation (4) defines the green index value EI_{ijks} of the process O_{ij} on machine k operated by worker s. The value of this index is determined by four factors: energy consumption, noise, chip recovery, and safety. The lower the energy consumption and noise value, the better and the higher the chip recovery rate and safety index value, so the reciprocal is taken for calculation. In addition, because the measurement units and meanings of the above four green indicators are different, they cannot be calculated directly. Therefore, it is necessary to normalize the four indicators by the min–max standardization method. The calculation formula of the min–max standardization method is $x' = \frac{x - x_{min}}{x_{max} - x_{min}}$, where x' represents the new value after x normalization, x_{max} is the maximum value of x indicator, and x_{min} is the minimum value of x indicator. Equation (5) is a process constraint to ensure that the workpiece is processed according to the specified process. Equation (6) ensures that each process can be assigned only to a machine in the set of optional processing machines for the process and the machine can be operated only by a worker in the set of optional workers. Equation (7) ensures that each machine can handle only one working procedure at a time.

3. DFJSP Solution Based on the Multi-Objective Hybrid Salp Swarm Algorithm

3.1. Salp Swarm Algorithm

The salp swarm algorithm (SSA) is a population intelligent optimization algorithm proposed by scholars Mirjalili and Gandomi in 2017 [26]. This stochastic optimization algorithm simulates the movement and foraging behavior of the salp swarm group in the ocean, and its effectiveness is verified by solving function optimization problems and engineering design problems. The salp swarm algorithm divides the salp chain into two parts: the first half of the individuals as leaders and the second half of the individuals as followers. It is assumed that there is a food source F in the search space, which is the search target of the group. The position update of the leader is only related to the position of the food source, while the position update of the follower is only related to the position of the salp swimming in front of it. The multi-objective salp swarm algorithm (MSSA) is a multi-objective swarm intelligent optimization algorithm based on the SSA, using the same population division and individual location update methods and referring to the external archive access mechanism of the MOPSO algorithm [27]. The basic idea is as follows: N salp individuals are randomly generated as the initial population; then, the objective function value of these salp individuals is calculated, and all non-dominated individuals in the population are identified according to the Pareto dominance relationship. If the number of individuals in the current external archive does not exceed the specified number, the non-dominated solution is directly saved to the archive. Otherwise, the redundant individuals are deleted from the archive-dense area before a new non-dominated solution is added. When performing the deletion operation, each solution in the archive needs to be sorted and selected by roulette. After updating the archive, according to the sorting value, an individual is selected as the food source F from the sparse area of the archive by the roulette method. Next, the value of c is updated, and the individual position in the population of salp is updated according to the food source F. The above steps are repeated until the termination conditions are met. The key operations involved in the MSSA are the update and maintenance of the archive and the selection of the food source F.

According to the above analysis, the MSSA is easy to implement, because it only has two main control parameters: c and archive size. In addition, the non-dominated solutions obtained so far are stored in the archive, so even if the whole population degenerates, the non-dominated solutions found will not be lost. Moreover, food source F is selected from the list of non-dominated solutions with the least number of neighboring solutions, which leads the search toward the less crowded regions of the Pareto optimal front obtained and improves the coverage of solutions found [26]. As a result of these reasons, the MSSA is logically able to find accurate Pareto optimal solutions with high distribution across all objectives, and it is proved experimentally on benchmark problems.

3.2. Multi-Objective Hybrid Salp Swarm Algorithm for Solving the DFJSP

3.2.1. Algorithm Framework

According to the characteristics of the DFJSP, this paper designs a multi-objective hybrid salp swarm algorithm (MHSSA) to solve the double-flexible job shop green scheduling problem. The flowchart of the MHSSA is shown in Figure 1. The main operators for the studied problems are encoding and decoding, individual position update of the OS part, and crossover and mutation operation of MA and WA parts.

3.2.2. Encoding and Decoding Mechanisms

The double-flexible job shop green scheduling problem needs to solve three subproblems: machine selection, worker selection, and process scheduling. According to the characteristics of the DFJSP, the multi-objective hybrid salp swarm algorithm designed in this paper needs three layers of salp individuals to represent: operation sequence (OS), machine assignment (MA), and worker assignment (WA). The length of each layer is equal to the total number of processes of all workpieces, as shown in Figure 2. The OS part adopts random key coding based on ascending arrangement, and the corresponding process sequencing is obtained by arranging the element values of each position in ascending order. The specific operation method is shown in the decoding process of the OS part in Figure 2. The MA part adopts machine codes arranged according to processes, which are arranged according to the order of workpieces and workpiece processes, and it is expressed by integers representing the machine number of the processing machine selected in the current process. The WA part adopts worker codes arranged according to processes, which are arranged according to the order of workpieces and workpiece processes, and it is expressed by integers representing the worker type number selected in the current process. For example, in Figure 2, the first numeral, 2, in the MA section indicates that the process O_{11} is arranged to be processed on the second machine, and the fourth numeral, 1, indicates that the process O_{21} is arranged to be processed on the first machine. The first number, 3, in the WA part indicates that the process O_{11} has selected the third type of worker to operate. The other numbers represent the same meaning as above.

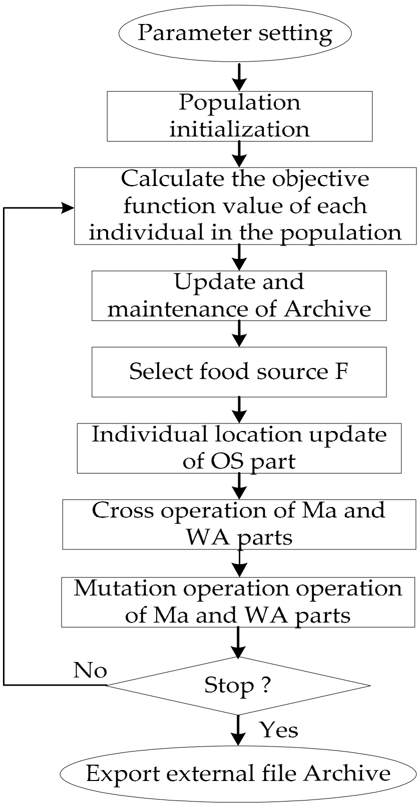
Parameter setting
Population initialization
Calculate the objective function value of each individual in the population
Update and maintenance of Archive
Select food source F
Individual location update of OS part
Cross operation of Ma and WA parts
Mutation operation operation of Ma and WA parts
No
Stop ?
Yes
Export external file Archive

Figure 1. Flowchart of a multi-objective hybrid salp swarm algorithm.

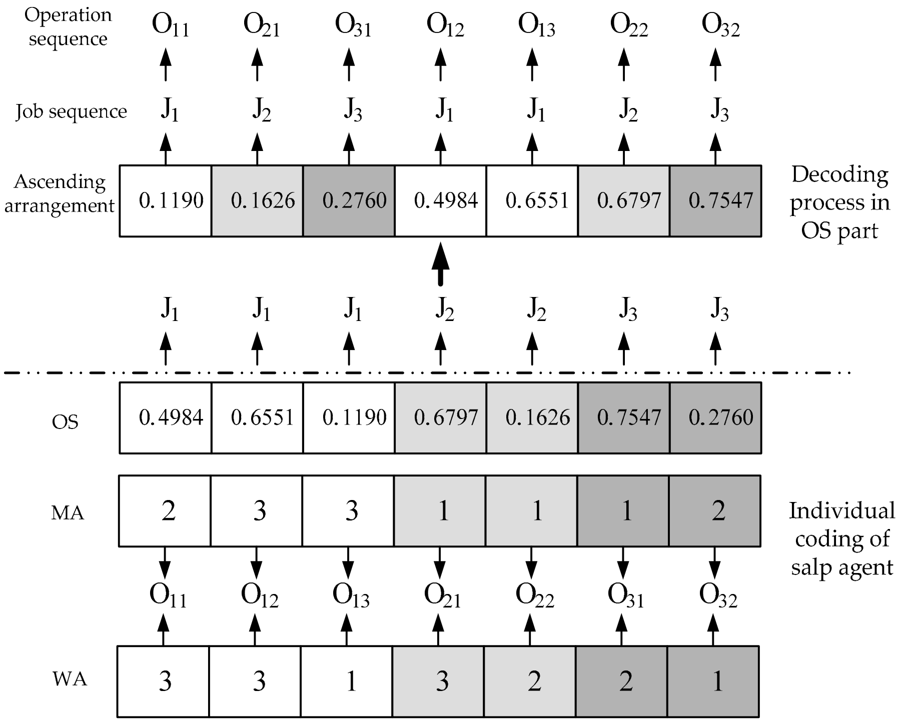
Operation sequence
O_{11} O_{21} O_{31} O_{12} O_{13} O_{22} O_{32}
Job sequence
J_1 J_2 J_3 J_1 J_1 J_2 J_3
Ascending arrangement
0.1190 0.1626 0.2760 0.4984 0.6551 0.6797 0.7547
Decoding process in OS part
J_1 J_1 J_1 J_2 J_2 J_3 J_3
OS
0.4984 0.6551 0.1190 0.6797 0.1626 0.7547 0.2760
MA
2 3 3 1 1 1 2
Individual coding of salp agent
O_{11} O_{12} O_{13} O_{21} O_{22} O_{31} O_{32}
WA
3 3 1 3 2 2 1

Figure 2. Individual coding and OS part decoding process of the salp agent.

To sum up, the following information can be interpreted from Figure 2: in the scheduling scheme represented by the salp individual, process O_{11} is processed first, and it is operated by a third type of worker on the second machine. The second process to be processed is O_{21}, which will be processed by a third type of worker on the first machine. Then, other processes will be operated.

The processing sequence of each process and the corresponding processing machines and operators can be determined from the individual code of salp. Then, the method in [5] is adopted to decode and generate activity scheduling for the salp individual. The specific steps are as follows:

(1) Determine the machine set required to process each workpiece and the process set processed on each machine.

(2) Determine the allowable starting processing time AS of each process, that is, the processing end time of the previous process.

(3) Adopt a process insertion method to determine the earliest starting processing time S of each process. The method is as follows: Find all interval idle time periods $[TS, TE]$ on the machine where the current process is located and judge whether each interval idle time period meets the insertion conditions from left to right. If so, insert it into the current idle time period, $S = max(TS, AS)$. Otherwise, $S = max(LM, AS)$, where LM represents the end time of the last operation of the current machine.

(4) Calculate the processing completion time C of each operation, where $C = S + P$ and P represents the processing time of the operation.

By the above decoding process, the processing sequence of the workpieces on each machine and the corresponding start time and completion time can be obtained. After calculation, the corresponding objective function value can be obtained.

3.2.3. Individual Location Update Operations in OS Part

In the SSA algorithm, the foraging chain of salp is divided into two parts: leader and follower. Therefore, the individual position update of the population includes the individual position update of leaders and the individual position update of followers. At the same time, combined with the coding method of the OS part, the following methods are used to update the location of the OS part of the salp population.

The position update of the individual leader is related to the position of the food source F, and the position update method is shown in Equation (8).

$$x_j^i = \begin{cases} F_j + c \cdot L & rand < 0.5 \\ F_j - c \cdot L & rand \geq 0.5 \end{cases} \tag{8}$$

Suppose there are N individuals in the population and the OS part of each individual has D-dimensional variables (i.e., the total number of processes). x_j^i represents the j-dimensional variable value of the OS part of the i-th salp individual;

$i = 1, 2, \cdots, N/2$; $j = 1, 2, \cdots, D$; and F_j represents the j-dimensional variable value of the OS part of the food source. The correlation coefficient c is the most important parameter in the SSA, which is used to balance global exploration and local development. See Equation (9), where t is the current number of iterations, t_{max} is the maximum number of iterations of the population, and the value of c decreases from 2 to 0 with the iteration process. L is the Lévy flight step size, which can effectively prevent premature convergence of the algorithm. The calculation method [28] is shown in Equations (10)–(12), where β is a parameter between [1,2] and taken here to be equal to 1.5, u and v obey normal distribution, and $rand$ is a random number between [0,1]. When $rand < 0.5$, x_j^i changes in the positive direction of the food source F_j; otherwise, it changes in the negative direction.

$$c = 2 \cdot \exp\left(-\left(\frac{4t}{t_{\max}}\right)^2\right) \tag{9}$$

$$L = \frac{u}{\sqrt[\beta]{|\nu|}} \tag{10}$$

$$u \sim N\left(0, \sigma_u^2\right), v \sim N\left(0, \sigma_v^2\right) \tag{11}$$

$$\sigma_u = \sqrt[\beta]{\frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma[(1+\beta)/2]\beta 2^{(\beta-1)/2}}},\ \sigma_v = 1 \tag{12}$$

The individual position update of the follower is only related to the position of the salp individual in front of it, which is updated according to Equation (13), where x_j^i represents the variable value of the j-th dimension of the OS part of the i-th salp individual, $i = (N/2)+1, (N/2)+2, \cdots, N, j = 1, 2, \cdots, D$.

$$x_j^i = \frac{x_j^i + x_j^{i-1}}{2} \tag{13}$$

3.2.4. Cross Operations in MA and WA Parts

According to the characteristics of the coding mode of the salp individual, the crossover and mutation operators of the genetic algorithm are embedded in the MSSA to update the position of the salp individual in MA and WA parts. The crossover operation mainly helps obtain better offspring by enabling the exchange of excellent gene fragments between parent chromosomes. To avoid generating infeasible solutions, MA and WA parts are regarded as a whole for the crossover operation. Meanwhile, since the MA and WA parts must ensure that the sequence of each element remains unchanged, the random probability crossover operation method is adopted, as shown in Figure 3. Two individuals are randomly selected in the population, and a random number *rand* on [0,1] is generated from left to right for each position element. If *rand* is less than the crossover probability (CR), the position element remains unchanged; otherwise, the position element is crossed to produce two new individuals. The above steps are repeated until all individuals complete the crossover operation.

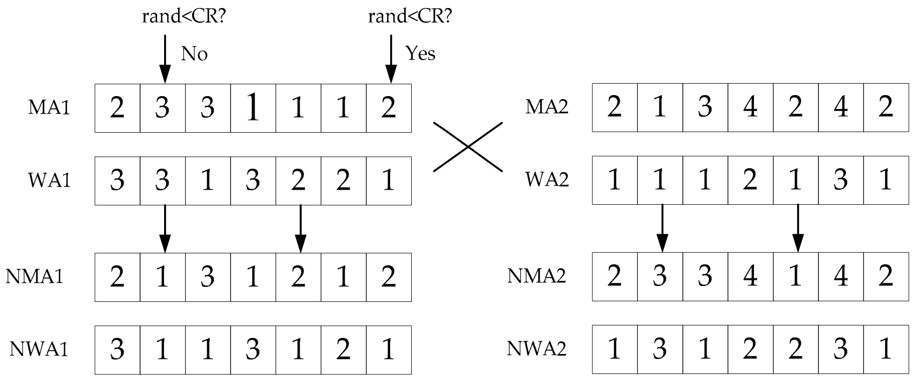

Figure 3. Random probability crossover operation.

3.2.5. Mutation Operations in MA and WA Sections

Usually, mutation operators are used to jump out of the local optimal solution and increase the diversity of the population. In this paper, a random number *rand* is generated for each individual in the salp population. If *rand* is less than the mutation probability (MR), the mutation operation is performed. The specific method of mutation operation for MA and WA parts is as follows: randomly select a process, randomly select a device in the optional processing machine set of the process, and randomly select an operator in the corresponding optional worker set.

4. Simulation Experiment and Analysis

4.1. Experimental Setup and Test Examples

To verify the effectiveness of the MHSSA in solving the double-flexible job shop green scheduling problem, two multi-objective optimization algorithms, the MSSA [26] and the MOPSO [27], are selected for comparison, and each algorithm runs independently 20 times.

The simulation platform includes the Win7 operating system, a CPU of 2.60 GHz, and 4G memory; MATLAB R2016a is used for programming.

In this paper, the DFJSP test example constructed in Reference [5] is selected for numerical experiments. The test problem is improved on the basis of the existing Brandimarte benchmark example [29]. Nine groups of data (DFJSP 01-09) are selected for testing the MHSSA in this paper; we can download these test data from https://pan.baidu.com/s/1mhHfv6K (accessed on 15 December 2021) [5] that in this example, with a decrease in the processing time, the labor cost will increase and the energy consumption will decrease. In addition, the example considers complete worker flexibility; that is, each machine can be operated by any worker.

4.2. Performance Evaluation Index

Since the measurement units of the three optimization objectives in this paper (maximum completion time, total labor cost, and total green index) are different, in order to evaluate the performance index of each algorithm, it is necessary to normalize the data of each target value. In this paper, three evaluation indexes SP, IGD, and Ω are selected. Since the optimal Pareto frontier of the tested problem is unknown, the non-dominated solution set obtained from the operation results of the three algorithms is approximated as the optimal Pareto frontier.

The SP index is used to measure the uniformity of the distribution of non-inferior solutions on the Pareto frontier. The calculation process is shown in Equation (14). Where $d_i = min_j\left(\left|f_1^i(x) - f_1^j(x)\right| + \left|f_2^i(x) - f_2^j(x)\right| + \left|f_3^i(x) - f_3^j(x)\right|\right), i, j = 1, \cdots, |PF|$, $\overline{d}$ is the mean of all d_i, and $|PF|$ is the number of non-inferior solutions on the Pareto frontier. A smaller SP value means better uniformity of the distribution of non-inferior solutions. In particular, when the SP value is 0, the non-inferior solutions are equally uniformly distributed on the Pareto frontier.

$$SP \triangleq \left[\frac{1}{|PF| - 1}\sum_{i=1}^{|PF|}\left(\overline{d} - d_i\right)^2\right]^{1/2} \tag{14}$$

IGD is a comprehensive index for simultaneously evaluating the diversity and frontier convergence of the obtained Pareto non-inferior solutions. The calculation is shown in Equation (15). $|PF^*|$ is the number of non-inferior solutions on the optimal Pareto frontier, and $dist(x, PF)$ is the Euclidean distance between a solution on the optimal Pareto frontier and the nearest solution on the obtained Pareto frontier. The smaller the IGD value, the better the performance of the algorithm.

$$IGD = \frac{\sum_{x \in PF^*} dist(x, PF)}{|PF^*|} \tag{15}$$

Ω is the dominant index, which is the percentage of the number of non-inferior solution sets found by the algorithm falling into the optimal Pareto frontier. The calculation process is shown in Equation (16), where H_k represents the set of non-dominated solutions obtained by the k-th algorithm, and $|P(S)|$ represents the number of non-dominated solutions in the set S. The larger the value of Ω, the better the dominance of the Pareto frontier found by the algorithm.

$$\Omega_{H_k} = \frac{\left|P(\bigcup_i H_i) \backslash P(\bigcup_{i \neq k} H_i)\right|}{\left|P(\bigcup_i H_i)\right|} \times 100\% \tag{16}$$

4.3. Parameter Setting

The MHSSA proposed in this paper mainly involves three parameters: population size N, crossover probability CR, and mutation probability MR. To obtain the best combination of factor levels, the Taguchi method is used for experimental design to explore the influence of algorithm parameters on the solution effect [30], and DFJSP 01 is used as a test example.

The specific process is to set four factor levels for each parameter. According to the number of experimental groups, the number of experimental factors, and the number of factor levels, the orthogonal table of $L_{16}\left(4^3\right)$ is adopted, as shown in Table 1. For each factor-level combination in the orthogonal table, the MHSSA runs independently 20 times and records the obtained non-dominated solutions. Since the index Ω represents the percentage of the number of non-inferior solution sets falling into the optimal Pareto frontier, which can reflect the dominance of the algorithm, the index Ω is used as the response variable (RV). The larger the RV value, the better the performance of the parameter combination. The experimental results are shown in Table 2. The average value of response characteristics (ARV) and the importance ranking of each parameter are shown in Table 3.

Table 1. Factor level setting of MHSSA parameters.

Factor	Factor Level			
	1	2	3	4
N	50	100	150	200
CR	0.2	0.5	0.7	0.9
MR	0.1	0.2	0.3	0.4

Table 2. Orthogonal table and response eigenvalues.

Experimental Serial Number	Factor			RV
	N	CR	MR	
1	1	1	1	0.0098
2	1	2	2	0.0098
3	1	3	3	0.0392
4	1	4	4	0.0196
5	2	1	2	0.0196
6	2	2	1	0.0392
7	2	3	4	0.0392
8	2	4	3	0.0784
9	3	1	3	0.0686
10	3	2	4	0.0686
11	3	3	1	0.0882
12	3	4	2	0.0490
13	4	1	4	0.0882
14	4	2	3	0.1471
15	4	3	2	0.1275
16	4	4	1	0.1078

Table 3. Response table.

Factor Level	*N*	*CR*	*MR*
1	0.0196	0.04655	0.06125
2	0.0441	0.066175	0.051475
3	0.0686	0.073525	0.083325
4	0.11765	0.0637	0.0539
Delta	0.09805	0.026975	0.03185
Rank	1	3	2

As can be seen from Table 3, the population size *N* is the most significant among the three parameters. The *N* value is too large, the convergence speed of the algorithm is slow, and the running time is too long. A small *N* value may cause premature convergence of the algorithm. The variation probability *MR* is the second most significant, and the factor significance level of the crossover probability CR is the lowest. According to the above parameters, the main parameters of the MHSSA are set to N = 200, CR = 0.7, and MR = 0.3, which are the best combination of factor levels for simulation. Other parameters of the algorithm are set as follows: the maximum number of iterations t_{max} = 100, and the external file *Archive_size* = 100.

4.4. Test Results and Analysis

To verify the effect of the proposed HMSSA to solve the DFJSP, nine test problems were selected to simulate the MHSSA. Each algorithm ran 20 times independently, and a group of [SP, IGD, Ω] values was recorded each time. Table 4 lists the mean (Avg) and standard deviation (Std) of each evaluation index of the three algorithms, and the optimal results of each index are marked in bold. It can be seen from the IGD and Ω indexes in Table 4 that the diversity, convergence, and dominance of the non-inferior solutions on the Pareto frontier obtained by the MHSSA are the best in the comparison algorithms and that the solution effects of the other two algorithms are not very different. For the SP index, the difference between the three algorithms is not obvious. For all test problems, the MHSSA designed in this paper has the best uniformity of the distribution of non-inferior solutions on the Pareto frontier.

Table 4. Simulation test results of three algorithms.

Table	**Evaluation Index**	**MSSA**		**MOPSO**		**HMSSA**	
		Avg	**Std**	**Avg**	**Std**	**Avg**	**Std**
DFJSP 01 (10 × 6 × 3)	SP	0.0562	0.0154	**0.0488**	**0.0150**	0.0691	0.0223
	IGD	0.1912	0.0199	0.1882	0.0158	**0.0940**	**0.0100**
	Ω	0.0025	0.0066	0.0025	0.0052	**0.0449**	**0.0319**
DFJSP 02 (10 × 6 × 3)	SP	0.0650	0.0179	0.0647	0.0239	**0.0609**	**0.0164**
	IGD	0.1864	0.0168	0.1854	0.0237	**0.1042**	**0.0155**
	Ω	0.0029	0.0063	0.0075	0.0101	**0.0397**	**0.0355**
DFJSP 03 (15 × 8 × 3)	SP	**0.0677**	**0.0169**	0.0761	0.0297	0.0692	0.0150
	IGD	0.1971	0.0133	0.1947	0.0208	**0.1178**	**0.0139**
	Ω	0.0038	0.0063	0.0091	0.0106	**0.0371**	**0.0389**
DFJSP 04 (15 × 8 × 3)	SP	**0.0619**	**0.0189**	0.0708	0.0259	0.0649	0.0148
	IGD	0.1867	0.0204	0.1950	0.0146	**0.1029**	**0.0121**
	Ω	0.0016	0.0039	0.0032	0.0061	**0.0452**	**0.0303**
DFJSP 05 (15 × 4 × 3)	SP	0.0638	0.0202	0.0676	0.0316	**0.0599**	**0.0121**
	IGD	0.1398	0.0133	0.1382	0.0087	**0.0755**	**0.0075**
	Ω	0.0058	0.0070	0.0065	0.0082	**0.0377**	**0.0304**
DFJSP 06 (10 × 15 × 3)	SP	0.0603	0.0121	**0.0570**	**0.0138**	0.0678	0.0151
	IGD	0.1809	0.0208	0.1848	0.0184	**0.1032**	**0.0072**
	Ω	0.0084	0.0126	0.0074	0.0084	**0.0342**	**0.0285**
DFJSP 07 (20 × 5 × 3)	SP	**0.0626**	**0.0174**	0.0672	0.0275	0.0690	0.0144
	IGD	0.1800	0.0229	0.1809	0.0157	**0.0951**	**0.0101**
	Ω	0.0052	0.0082	0.0057	0.0076	**0.0391**	**0.0317**
DFJSP 08 (20 × 10 × 3)	SP	0.0590	0.0195	0.0618	0.0173	**0.0566**	**0.0078**
	IGD	0.1722	0.0151	0.1857	0.0161	**0.0950**	**0.0104**
	Ω	0.0072	0.0104	0.0011	0.0026	**0.0417**	**0.0375**
DFJSP 09 (20 × 10 × 3)	SP	0.0604	0.0110	0.0690	0.0226	**0.0574**	**0.0148**
	IGD	0.1777	0.0154	0.1801	0.0158	**0.0975**	**0.0111**
	Ω	0.0038	0.0048	0.0053	0.0096	**0.0409**	**0.0289**
Total	SP	**0.0619**	**0.0034**	0.0648	0.0081	0.0639	0.0052
	IGD	0.1791	0.0165	0.1814	0.0170	**0.0984**	**0.0113**
	Ω	0.0046	0.0023	0.0054	0.0026	**0.0401**	**0.0036**

Furthermore, the significant differences between the algorithms were tested using the Wilcoxon signed-rank test for relevant samples. For the three evaluation indexes of the test problem, the three algorithms are compared in pairs. The test results are shown in Table 5. The MHSSA has significant differences from the comparison algorithm in IGD and Ω indexes, and the effect is the best. For the SP index, the three algorithms have no significant difference. In conclusion, the MHSSA proposed in this paper can effectively solve the dual-flexible job shop green scheduling problem. On the whole, the MHSSA is obviously superior to the other two comparison algorithms in terms of diversity, convergence, and dominance of non-inferior solutions on the Pareto frontier.

Table 5. Wilcoxon signed-rank test (the MHSSA vs. the MSSA and the MOPSO).

Evaluation Index	MSSA		MOPSO	
	p Value	Sig ($p < 0.05$)	*p* Value	Sig ($p < 0.05$)
SP	0.477	N	0.515	N
IGD	0.008	Y	0.008	Y
Ω	0.008	Y	0.008	Y

5. Conclusions

In this paper, a multi-objective hybrid salp swarm algorithm is proposed to solve the dual-flexible job shop green scheduling problem with the objectives of optimizing the maximum completion time, total labor cost, and total green index. The specific research contents include the following:

(1) According to the characteristics of the DFJSP, a three-layer individual coding and decoding method of salp is designed, and it is decoded into an active scheduling scheme. In the optimization process of the algorithm, the individual position of the OS part of the individual salp is updated based on the Lévy random walk, and the random probability crossover operation and corresponding mutation operation are performed on the MA and WA parts, which effectively improves the optimization accuracy and efficiency of the algorithm.

(2) In the part of numerical simulation experiment, the Taguchi method was used to study the influence of the parameter setting on scheduling results and the best factor-level combination of algorithm parameters was identified. Using a simulation test for the benchmark example of the DFJSP, the MHSSA was compared with the MSSA and the MOPSO to verify that the MHSSA can effectively solve the dual-flexible job shop green scheduling problem.

In the next step, the DFJSP will be studied more extensively. In terms of the green scheduling problem, considering that the machine has a variety of speeds and other indicators, green manufacturing will be achieved by an actual production strategy.

Author Contributions: In this article, C.L. and Y.Y. designed the study. Their contributions are as follows: Conceptualization, C.L. and Y.Y.; methodology, C.L.; software, Y.Y.; validation, Y.Y.; writing—original draft preparation, Y.Y.; writing—review and editing, C.L.; visualization, H.Z.; supervision, H.Z.; funding acquisition, C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by The National Social Science Foundation of China, grant number 17BSH040, 20BGL006 and the Natural Science Foundation of the Jiangsu Higher Education Institutions of China, grant number 20KJB520036.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data used to support the findings of this study are included within the article.

Acknowledgments: The authors are thankful to the anonymous reviewers and the editor for their valuable comments and constructive suggestions that have led to an improved version of this paper.

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

References

1. Boudreau, J.; Hopp, W.; McClain, J.-O.; Thomas, L.J. On the interface between operations and human resources management. *Manu. Serv. Opera. Manag.* **2003**, *5*, 179–202. [CrossRef]
2. Gino, F.; Pisano, G. Toward a theory of behavioral operations. *Manu. Serv. Opera. Manag.* **2008**, *10*, 676–691. [CrossRef]
3. Neumann, W.-P.; Dul, J. Human factors: Spanning the gap between OM and HRM. *Int. J. Opera. Prod. Manag.* **2010**, *30*, 923–950. [CrossRef]
4. Ye, C. Research on a new model of behavioral production scheduling based on learning effect. *Enterp. Econ.* **2015**, *35*, 5–10.
5. Gong, G.; Deng, Q.; Gong, X.; Liu, W.; Ren, Q. A new double flexible job-shop scheduling problem integrating processing time, green production, and human factor indicators. *J. Clean. Prod.* **2018**, *174*, 560–576. [CrossRef]
6. He, Y.; Liu, F.; Cao, H.-J.; Li, C.-B. A Bi-objective model for job-shop scheduling problem to minimize both energy consumption and makespan. *J. Cent. South Univ. Tech.* **2005**, *12*, 167–171. [CrossRef]
7. Mouzon, G.; Mehmet, B.-Y.; Twomey, J. Operational methods for minimization of energy consumption of manufacturing equipment. *Int. J. Prod. Res.* **2007**, *45*, 4247–4271. [CrossRef]
8. Dai, M.; Tang, D.; Giret, A.; Salido, M.-A.; Li, W.-D. Energy-efficient scheduling for a flexible fow shop using an improved genetic-simulated annealing algorithm. *Rob. Comput. Integr. Manuf.* **2013**, *29*, 418–429. [CrossRef]
9. Liu, C.-H.; Huang, D.-H. Reduction of power consumption and carbon footprints by applying multi-objective optimisation via genetic algorithms. *Int. J. Prod. Res.* **2014**, *52*, 337–352. [CrossRef]
10. Mansouri, S.-A.; Aktas, E.; Besikci, U. Green scheduling of a two-machine fowshop: Trade-off between makespan and energy consumption. *Eur. J. Oper. Res.* **2016**, *248*, 772–788. [CrossRef]
11. Subaï, C.; Baptiste, P.; Niel, E. Scheduling issues for environmentally responsible manufacturing: The case of hoist scheduling in an electroplating line. *Int. J. Prod. Econ.* **2006**, *99*, 74–87. [CrossRef]
12. Fang, K.; Uhan, N.; Zhao, F.; Sutherland, J.W. A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction. *J. Manuf. Syst.* **2011**, *30*, 234–240. [CrossRef]
13. Diaz, N.; Redelsheimer, E.; Dornfeld, D. Energy consumption characterization and reduction strategies for milling machine tool use. In Proceedings of the 18th CIRP International Conference on Life Cycle Engineering, Braunschweig, Germany, 2–4 May 2011; pp. 263–267.
14. Wang, S.; Lu, X.; Li, X.-X.; Li, W.-D. A systematic approach of process planning and scheduling optimization for sustainable machining. *J. Clean. Prod.* **2015**, *87*, 914–929. [CrossRef]
15. Zhang, R.; Chiong, R. Solving the energy-efficient job shop scheduling problem: A multi-objective genetic algorithm with enhanced local search for minimizing the total weighted tardiness and total energy consumption. *J. Clean. Prod.* **2016**, *112*, 3361–3375. [CrossRef]
16. Gahm, C.; Denz, F.; Dirr, M.; Tuma, A. Energy-efficient scheduling in manufacturing companies: A review and research framework. *Eur. J. Opera. Res.* **2016**, *248*, 744–757. [CrossRef]
17. Wang, L.; Wang, J.; Wu, C. Advances in green shop scheduling and optimization. *Control Deci.* **2018**, *33*, 385–391.
18. Lei, D.-M.; Zheng, Y.-L.; Guo, X.-P. A shuffled frog-leaping algorithm for flexible job shop scheduling with the consideration of energy consumption. *Int. J. Prod. Res.* **2017**, *55*, 3126–3140. [CrossRef]
19. Wu, X.; Sun, Y. Flexible job shop green scheduling problem with multi-speed machine. *Comput. Integr. Manu. Sys.* **2018**, *24*, 862–875.
20. Dong, J.; Ye, C. Research on collaborative optimization of green manufacturing in semiconductor wafer distributed heterogeneous factory. *Appl. Sci.* **2019**, *9*, 2879–2903. [CrossRef]
21. Yao, Y.-Y.; Ye, C.-M. IMSSA for TFT-LCD panel array process scheduling problem considering energy saving. *China Mechan. Eng.* **2019**, *30*, 2994–3003.
22. Nanthapodej, R.; Liu, C.; Nitisiri, K.; Pattanapairoj, S. Variable neighborhood strategy adaptive search to solve parallel-machine scheduling to minimize energy consumption while considering job priority and control makespan. *Appl. Sci.* **2021**, *11*, 5311–5332. [CrossRef]
23. He, Y.; Liu, F.; Cao, H.-J.; Li, C.-B. Scheduling optimization of mechanical processing system for green manufacturing model. *Chin. J. Mechan. Eng.* **2007**, *43*, 27–33. [CrossRef]
24. Li, X.-X.; Huang, X.-M.; Liu, J.-X.; Liu, F. Optimization simulation for job-shop scheduling for reducing manufacturing energy consumption. *J. Sys. Simu.* **2016**, *28*, 114–120.
25. Lu, C.; Gao, L.; Li, X.-Y.; Pan, Q.; Wang, Q. Energy-efficient permutation flow shop scheduling problem using a hybrid multi-objective backtracking search algorithm. *J. Clean. Prod.* **2017**, *144*, 228–238. [CrossRef]
26. Mirjalili, S.; Gandomi, A.-H.; Mirjalili, S.-Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Soft.* **2017**, *114*, 163–191. [CrossRef]

27. Coello, C.-A.; Pulido, G.-T.; Lechuga, M.-S. Handling multiple objectives with particle swarm optimization. *IEEE Trans. Evolut. Comput.* **2004**, *8*, 256–279. [CrossRef]
28. Liu, C.-P.; Ye, C.-M. Bat algorithm with the characteristics of lévy flight. *CAAI Trans. Intell. Sys.* **2013**, *8*, 240–246.
29. Paolo, B. Routing and scheduling in a flexible job shop by tabu search. *Ann. Opera. Res.* **1993**, *41*, 157–183.
30. Montgomery, D.-C. *Design and Analysis of Experiments, 7th ed*; John Wiley & Sons: Hoboken, NJ, USA, 2008; pp. 50–60.