

Article

A Novel Production Scheduling Approach Based on Improved Hybrid Genetic Algorithm

Lili Dai ¹, He Lu ^{1,2}, Dezheng Hua ² , Xinhua Liu ^{2,*}, Hongming Chen ², Adam Glowacz ³ ,
Grzegorz Królczyk ⁴  and Zhixiong Li ^{4,5,*} 

¹ Institute of Smart Materials and Applied Technology, Lianyungang Normal College, Lianyungang 222006, China

² School of Mechatronic Engineering, China University of Mining and Technology, Xuzhou 211006, China

³ Department of Automatic, Control and Robotics, AGH University of Science and Technology, 30-059 Kraków, Poland

⁴ Faculty of Mechanical Engineering, Opole University of Technology, 45-758 Opole, Poland

⁵ Yonsei Frontier Lab, Yonsei University, Seoul 03722, Korea

* Correspondence: liuxinhua@cumt.edu.cn (X.L.); zhixiong.li@yonsei.ac.kr (Z.L.)

Abstract: Due to the complexity of the production shop in discrete manufacturing industry, the traditional genetic algorithm (GA) cannot solve the production scheduling problem well. In order to enhance the GA-based method to solve the production scheduling problem effectively, the simulated annealing algorithm (SAA) is used to develop an improved hybrid genetic algorithm. Firstly, the crossover probability and mutation probability of the genetic operation are adjusted, and the elite replacement operation is adopted for simulated annealing operator. Then, a mutation method is used for the comparison and replacement of the genetic operations to obtain the optimal value of the current state. Lastly, the proposed hybrid genetic algorithm is compared with several scheduling algorithms, and the superiority and efficiency of the proposed method are verified in solving the production scheduling.

Keywords: production scheduling; hybrid genetic algorithm; artificial intelligence; sustainable design; discrete manufacturing



Citation: Dai, L.; Lu, H.; Hua, D.; Liu, X.; Chen, H.; Glowacz, A.; Królczyk, G.; Li, Z. A Novel Production Scheduling Approach Based on Improved Hybrid Genetic Algorithm. *Sustainability* **2022**, *14*, 11747. <https://doi.org/10.3390/su141811747>

Academic Editor: Yoshiki Shimomura

Received: 19 July 2022

Accepted: 5 September 2022

Published: 19 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Production scheduling is one of the classic non-deterministic problems, involving aircraft carrier scheduling, port cargo scheduling, parts processing scheduling, and many other fields [1]. In the production field, the scheduling of production workshops, equipment, personnel, and materials is different for different manufacturing enterprises, and the production cycle is also varied [2]. With the emergence of large-scale systems for production and the proposal of a variety of intelligent algorithms, the production scheduling of workshops has attracted extensive attention from managers, and remarkable results have been achieved continuously [3]. In the competitive environment of the new era, enterprises have paid more attention to how to use production scheduling to effectively manage resource allocation, deal with production scheduling, and improve production efficiency [4].

Although there have been many intelligent scheduling algorithms, it is still an important goal to seek more efficient and practical scheduling approaches in the production scheduling of discrete manufacturing industry [5]. Considering the complexity and diversity of discrete production scheduling, there are two algorithms that are used to solve these problems [6]. One is based on traditional unified research, which mainly includes the Lagrange relaxation method [7], the branch-and-bound method [8], and the mathematical programming method [9], etc., which are not effective in practical engineering at present. The other is a heuristic algorithm, mainly including the genetic algorithm, simulated annealing algorithm [10], particle swarm optimization algorithm [11], and ant colony

algorithm [12], etc. For example, Choi et al. [13] proposed a mixed integer programming model and local search algorithm to solve project scheduling in a variety of manufacturing environments. Nie et al. [14] studied the dynamic scheduling problems of job publication by date and proposed a heuristic algorithm and a reactive scheduling strategy based on gene expression coding and applied it to production scheduling. These algorithms are simple in structure and easy to implement, which can achieve satisfactory results in solving production scheduling issues [15].

Among many intelligent algorithms, genetic algorithm (GA), with its special optimization mode, is an effective approach used to solve production scheduling problems, and has achieved some results [16,17]. Pezzella et al. [18] proposed a GA that combines multiple strategies for initial population generation, selection, and reproduction, which is used to solve production scheduling problems. De Giovanni and Pezzella [19] proposed an improved GA for distributed production scheduling problems, which determines the processing path of the workpiece through a greedy decoding process. Zhang et al. [20] used pareto-optimization-based GA and two new objective functions based on setup and synergy costs to solve production scheduling problems. Chamnanlor et al. [21] proposed a hybrid GA based on ant colony algorithm for production scheduling. Meng Yue et al. [22] created a hybrid algorithm of path reconnection for the production scheduling problems, which combined the genetic algorithm, domain structure algorithm, and path reconnection algorithm to further enhance the calculation ability. Zhao et al. [23] used the hybrid genetic simulated annealing algorithm to deal with the production scheduling of flexible jobs. Although simulated annealing factors were added to improve the performance of the algorithm and make the algorithm jump out of local optimal, the influence of adaptive crossover and mutation probability on the final convergence of the algorithm was not considered.

In accordance with the above literature analysis, to better and more effectively solve production scheduling problems, an improved hybrid genetic algorithm (IHGA) is proposed in this work. Different from other hybrid genetic algorithms in selecting individuals for crossover and mutation, the proposed algorithm adopts an adaptive strategy to adjust the size of probability. In the early stages of population iterative reproduction, at first, higher probability is used to choose more individuals, expand the scope of the late optimization, increase species diversity, and reduce the probability of premature phenomena. The premature algorithm is beginning to choose a large amount of high fitness individuals, obtaining a quick local optimal value and not a global optimal value. In the later stage of population iterative reproduction, excellent individuals can be preserved with a low probability, and the convergence of the algorithm relative to the optimal value can be completed as soon as possible. In the simulated annealing factor, using the memory function, the optimal solution is conducted to mutate by probability to obtain as many individuals as possible. A heating strategy is added to avoid falling into local optimum, which determines the new individual compared with the fitness function of the original individual.

The rest of this paper is organized as follows. Section 2 presents a system modeling of the production scheduling. Section 3 describes the operation process of improved hybrid genetic algorithm. In Section 4, experimental analysis is carried out. The application testing of the improved approach is carried out in Section 5. The conclusions and future works are provided in Section 6.

2. Modeling Production Scheduling

The problems of production scheduling in the discrete manufacturing industry usually refer to the determination of the processing sequence of each workpiece under various production requirements and processing constraints, considering the relevant parts in the assembly plan. In view of the characteristics of the discrete manufacturing industry and the actual demand, time arrangement is the main factor. The production scheduling of the discrete manufacturing industry can be described as follows: a batch of workpieces are needed to be produced; the number of the workpieces is n ; each workpiece contains

many processes; the batch of the workpieces is completed by using m sets of machines in the shortest time. The main constraint conditions for production scheduling are as follows:

- (1) At the beginning of production, each workpiece can be randomly selected and processed on the designated machine;
- (2) Each piece of equipment used for production in the workshop can only process one workpiece at any time;
- (3) Each workpiece can only be processed once on each piece of equipment;
- (4) Sudden interruption is forbidden after it has started;
- (5) Any workpiece in the first process is not in order; however, the same workpiece in production has a certain sequence constraint and absolutely cannot be changed;
- (6) The production of the workpiece must conform to the actual process line and needs to be practical;
- (7) The processing time of each workpiece has been determined and does not change with the sorting;
- (8) Auxiliary time for processes such as tool installation and workpiece transportation is not considered.

According to the actual production situation of the discrete manufacturing industry, each workpiece has a certain entry point; however, only after the production of all the relevant workpieces can this be assembled. To improve the production and assembly efficiency, it is necessary to reduce the overall production time of the batch of workpieces to a relative minimum. Mathematical modeling is used to find the completion time of the whole batch of workpieces, that is, the latest time to find the completion of the last workpieces. The mathematical model is as follows:

In terms of the production characteristics of workpieces in a workshop of the discrete manufacturing industry, it can be concluded that the calculation formula for the latest processing time of process P of workpiece j_i is as follows:

$$S_{jip} = \begin{cases} T_{c,t} & (P = 0) \\ \max(E_{jip-1}, T_{c,t}) & (P > 0) \end{cases} \quad (1)$$

where c is machine serial number, t is the current machine.

The processing time $T_{c,t}$ of machine t is:

$$T_{c,t} = S_{jip} + T \quad (2)$$

where T is process P occupies the production time of machine t .

The end time E of the workpiece j_i for process P is:

$$E_{jip} = S_{jip} + T \quad (3)$$

The latest production time for finishing the last step of workpiece is:

$$\max_{0 \leq i < c} T_i \quad (4)$$

The meaning of each variable in the mathematical model is shown in Table 1. The mathematical model describes the production scheduling, and the production time of each plan can be calculated. Then, the time consumption of each plan can be compared to select the best scheduling plan with the shortest time.

Table 1. Variables in the mathematical model.

Name	Meaning of Variable
S_{ij}	Latest start time of process j for workpiece i
E_{ij}	Completion time of process j for workpiece i
t_{ij}	Number of the required machine for workpiece i to perform process j
T_{ij}	Time of process j for workpiece i
P_{ij}	Process of workpiece i using machine j
T_i	Processing time with serial number i
P_i	Process of workpiece with serial number i
j_i	The workpiece i
P	Current operation P_{ji} of j_i
t	Machine number of process P
T	Processing time of process P

3. Improved Hybrid Genetic Algorithm for Production Scheduling

The genetic algorithm, particle swarm optimization algorithm, and simulated annealing algorithm are commonly used in production scheduling. Genetic algorithm has good parallelism and strong global search ability; however, it is easy to fall into local optimal solution. Particle swarm optimization algorithm has a simple structure and an autonomous learning ability; however, its stability is poor and global search is slow. The simulated annealing algorithm has strong parallelism and robustness; however, the parameters are difficult to control, the initial value requires high, and the convergence is slow.

According to the production characteristics and requirements of a discrete industry workshop, the algorithm needs to have high robustness and stability, must be able to meet the better global search ability, and cannot have high requirements for the initial value, therefore the genetic algorithm is chosen as the fundamental, and the simulated annealing algorithm is used to improve its defects.

3.1. Chromosomal Coding

There are many encoding methods of chromosomes in the genetic algorithm, and an appropriate encoding method can improve the efficiency of the global optimal solution and its of finding ability. Due to the complexity and particularity of discrete production scheduling, a procedure-based coding method is chosen in this work, and the encoding mode of chromosomes is shown in Figure 1. The length of chromosomes is related to the number of machines and workpieces. Therefore, if there are m machines, n workpieces, and i process ($0 \leq i < n$) of each machine, the length of the chromosome is $chSize \leq (n \times m)$. Since the number of processes may not be equal for different workpieces, the length of the chromosome should be the sum of the number of processes in all workpieces. In a specific chromosome, the coding rule is that the number of times a workpiece appears in a chromosome indicates the number of processes it needs to be processed, such as, for example, the first occurrence of j_1 in chromosome means the first process of j_1 workpiece, and the fourth occurrence means the fourth process of j_1 , based on which the process of the other workpiece can be deduced.

Chromosome initialization

j_0	j_0	j_1	j_1	j_2	...	...	j_{n-1}	j_{n-2}	j_{n-1}	j_{n-2}
-------	-------	-------	-------	-------	-----	-----	-----------	-----------	-----------	-----------

Figure 1. Discrete scheduling chromosome encoding scheme.

3.2. Design of Fitness Function

In the process of genetic algorithm optimization, the fitness function is mainly used to evaluate the advantages and disadvantages of chromosomes. Under the condition of minimizing the completion time in production scheduling in the discrete manufacturing industry, the value of the fitness function is the maximum completion time obtained in

accordance with the mathematical model created above, and the fitness function can be used to effectively determine the level of chromosomes. In this method, the larger the fitness function value is, the longer the completion time of all the part is; the smaller the fitness of chromosome is, the easier it is to be eliminated in the selection operation. The smaller the fitness function value is, the shorter the completion time of all the workpiece is; the larger the fitness of chromosome is, the easier it is to be selected and inherited by the next generation population.

3.3. Selection Operation

When selecting individuals in a population, excellent individuals are selected with a high probability, while those with low fitness are selected with a low probability. Therefore, the population will evolve in a good direction at the beginning of the iteration. A tournament method is used to select individuals, which removes the best one to enter the offspring population and repeats the operation until the new population size reaches the original population size. The main operation method is used to randomly select Z individuals from the population, let them compete for fitness, and then select the best one from them. In this work, all individuals participating in the championship are the whole population, and Y is the number of randomly selected individuals from the population and $Y = 3$; then, the optimal individual j is selected. The steps are shown in Figure 2.

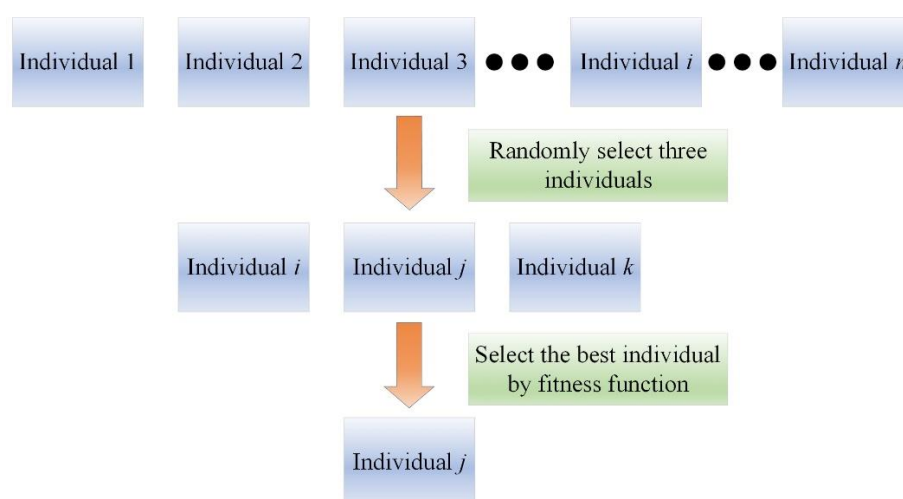


Figure 2. Optimal individual selection process based on tournament method.

There are usually three main ways to select individuals in genetic algorithms. The first is more commonly used in the tournament method described above. The second is the roulette selection method, which calculates the selection probability of individuals through a comparison of the individual fitness value in the population and then determines the composition of the offspring population [24]. To solve the time minimization problem of production scheduling, there is a need to first transform the fitness function into a maximization problem, where the fitness value of each individual is obtained separately. Each fitness value is divided by the sum of the fitness values of all individuals, and the probability of individual selection is obtained. The cumulative probability of all individuals is collated into a roulette wheel, and a new generation population is continuously obtained by generating random numbers between $[0, 1]$ for each roulette selection. The third random traversal sampling method has the same probability of individual selection as the second [25]. The difference lies in whether it is necessary to select individuals with equal distance. For example, if m_{exm} individuals need to be selected, the distance of pointer selection should be $1/m_{exm}$, and the position of the first pointer should be determined by generating a random number in $[0, 1/m_{exm}]$. Through the analysis of the above methods,

the advantages of the tournament method include its low time complexity, easy parallel processing, and low premature phenomenon.

3.4. Crossover Operation

The crossover operation is one of the key operations in continuous evolution, where two paired chromosomes somehow exchange a part of their genes with each other, forming two new individuals to increase population diversity and increase the probability of producing excellent individuals [26]. Different from the previous crossover operation, an adaptive adjustment strategy is adopted in this work to keep the diversity of individuals at the initial stage of population iteration and ensure the convergence of optimization results as soon as possible at the later stage. An OX crossover method based on process coding is adopted in the hybrid algorithm, and the operator crossover diagram is shown in Figures 3 and 4. In the crossover process, the starting and ending positions of the chromosomes of the father and the mother are random. Crossover operation changes the gene sequence of a part on the basis of preserving chromosome gene fragments, which increases the diversity of the population, improves the search ability of the algorithm, and increases the probability of excellent individuals; therefore, the randomness of newly generated individuals is greater.

Step 1: Two chromosomes P_1 and P_2 are randomly selected each time according to the selection operation as the father and mother.

Step 2: One chromosome is selected as the father, a gene fragment is intercepted by randomly determining two positions in the chromosome, and the intercepted gene fragment is used as a progeny chromosome prototype. The operation process is shown in Figure 3.

Step 3: The remaining chromosome is taken as the mother, where the missing codes of the progeny prototype are completed, as shown in Figure 4.

Step 4: The first child is generated through Steps 2 and 3 above, and then the second child is generated by repeating Steps 2 and 3 above.

The crossover probability is P_c , and the probability can be adjusted adaptively by adopting the number of iterations. The crossover probability is closely related to individual fitness and the number of iterations. The calculation formula of the probability is as follows:

$$P_c = \begin{cases} k_1 \times \frac{1}{\sqrt{\text{gen}}} \times \frac{f_{\max} - f'}{f_{\max} - f_{\text{avg}}} & (f' \geq f_{\text{avg}}) \\ k_1 \times \frac{1}{\sqrt{\text{gen}}} & (f' < f_{\text{avg}}) \end{cases} \quad (5)$$

where $f_{\max}$ is the maximum fitness value in the population; f' is the larger fitness value of the two chromosomes that need to be crossed; f_{avg} is the average value of the fitness of all individuals in the population; gen is the number of iterations of the population so far; k_1 is the adjustment parameter of the crossover probability. The value of k_1 increases, and the crossover change increases, which is determined as 0.85 according to simulation experience.

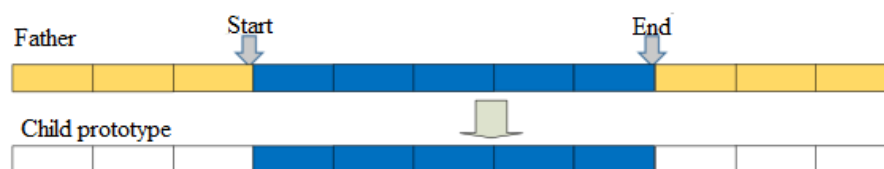


Figure 3. Father chromosomes' generated offspring. The yellow fragment represents the part removed from the Father, the blue is the part passed on to the offspring, and the white is the unassigned portion of the offspring.

3.5. Mutation Operation

Mutation operation has a relatively small probability in the genetic algorithm; however, it also has a significant impact on the more diverse population. Therefore, the mutation probability should be adjusted slightly in the same way as the crossover operation; the

mutation probability has a great relationship with the fitness value of chromosomes and the number of iterations of the population. The mutation operation is mainly based on the location mutation method, which randomly selects chromosomes in the population according to the probability, randomly determines the two locations of chromosomes, and carries out gene exchange to generate new chromosomes in this way. In this algorithm, two pairs of the chromosomes' genes are exchanged, as shown in Figure 5.



Figure 4. Mother chromosome completes the progeny chromosome prototype. In the mother chromosome, the yellow fragment represents the part passed on to the offspring, and the blue is the part removed. In the offspring, the yellow fragment represents the portion that inherits from the mother, and the blue is the portion that inherits from the father.

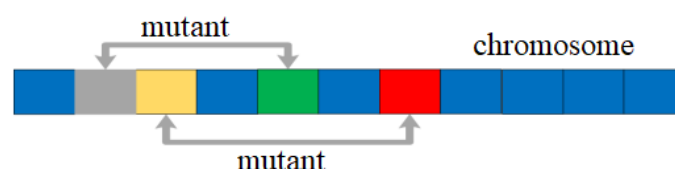


Figure 5. Gene exchange based on positional variation. The blue fragment represents the invariant region, and the other colored fragments are swapped according to the direction of arrows.

The probability of mutation operation is also adjusted adaptively as the number of iterations increases, and the adjustment formula is shown as follows:

$$P_m = \begin{cases} k_2 \times \frac{1}{\sqrt{gen}} \times \frac{f_{\max} - f}{f_{\max} - f_{avg}} & (f \geq f_{avg}) \\ k_2 \times \frac{1}{\sqrt{gen}} & (f < f_{avg}) \end{cases} \quad (6)$$

where f is the fitness value of individuals who may be mutated at present; k_2 is the adjustment parameter of individual mutation, generally between (0, 1). Other variable names in the formula have the same meanings as described in Equation (5).

3.6. Simulated Annealing Operator

In order to improve the local searching ability of the genetic algorithm, a new hybrid genetic algorithm is proposed. A simulated annealing algorithm is created, which is different from the previous genetic simulated annealing algorithm in that the simulated annealing operator is adjusted [27]. In the past hybrid genetic algorithms, the simulated annealing operator lacks the memory function. However, using the memory function saves the optimal solution generated in the annealing process in this work. Moreover, in the simulated annealing factor, the optimal solution is introduced by probability and is mutated to obtain the optimal individual as far as possible, and the heating strategy is added to avoid falling into the local optimum. The fitness function of the new individual is compared with that of the original individual, and whether the original individual replaces the original individual through the result is determined to ensure that the individual is as good as possible.

The optimal value is determined by a comparison in the simulated annealing operator, which is not applicable to the algorithm because the optimal value is highly likely to be ignored. For this IHGA, the optimal value is searched for in the population for several times at each temperature state, and the optimal value in this stage is determined with a certain probability by reaching the specified optimization times. At temperature T_k (k is the number of cooling times), the optimal value in the population is compared with the

randomly selected value, T_1 is the initial temperature; others have similar meanings in turn, and T_{min} is the end temperature, which is set to 0.001 in this work. The cooling formula added in this work is as follows:

$$T_{k+1} = \alpha T_k \quad (7)$$

where α is the cooling coefficient.

In the simulated annealing process, to prevent the temperature from falling too fast at the initial stage of cooling, an appropriate heating strategy is adopted in the annealing process, which is beneficial to increase the acceptance probability of various chromosomes, ensure a more diverse selectivity of chromosomes, and avoid the occurrence of local optimization. The search steps are as follows:

Step 1: The current state bit is set as S , the initial value of the cycle counter is $d = 1$, the initial value of the counter for the new individual of the generation population is $O = 1$, and the length of the Markov chain is L .

Step 2: Select v individuals with the lowest fitness value from the population generated after mutation operation as the initial solution in annealing operation, and make the current state bit $S = v$.

Step 3: Select an individual randomly from the population, determine this state bit as $S' = v'$, and calculate the increment of fitness value as $dE = f(v') - f(v)$.

Step 4: If $dE < 0$, the current status bit is determined as S' ; otherwise, the $\exp(-dE/T_K)$ is taken as the probability of S' , and $S = S'$.

Step 5: Select the current chromosome for mutation, determine the new state of the mutant chromosome $S'' = v''$, and calculate the fitness value of the new chromosome. If $f(v'') \leq f(v')$, the state bit of the mutant chromosome is S'' , that is, $S = S''$; otherwise, $S = S'$, and return to Step 3; If $d > L$ is true, the current internal temperature cycle is terminated, and Step 6 is performed instead.

Step 6: Calculate and compare the fitness values of all new individuals after the completion of the internal cycle. Then, select the individuals with the lowest fitness values to the new population, and take them as the initial solution of the temperature of the next iteration and make $O = O + 1$.

Step 7: The temperature value of the next iteration is obtained through cooling operation. The calculation formula of temperature difference is $\Delta T = T_K - T_{K+1}$. If $\Delta T > 0.5(T_1 - T_2)$, the heating operation is carried out and $T_{K+1} = T_{K+1} + 0.5\Delta T$.

Step 8: Repeat Step 3–Step 7 until the number of generated individuals reaches the population value or the minimum temperature and the operation is terminated.

The pseudocode is shown below (Algorithm 1):

Algorithm 1. The pseudocode of IHGA.

```

1: Begin
2:   Initialize the individuals  $S$  and the temperature  $t$  in the population
3:   While ( $t \geq$  minimum temperature and  $O \leq$  population size)
4:     While ( $d \leq$  chain length)
5:       Update One New to a random individual in the population
6:       Update  $S$  by One New fitness
7:       Update  $S$  by mutating its genes
8:       Add  $S$  to population All New
9:        $d++$ ;
10:    end while
11:   Calculate the fitness of each individual in the population All New
12:   Update  $S$  the individual with the highest fitness of All New
13:   Update  $t$  by judging the cooling speed
14:    $O++$ 
15: end while
16: End

```

By analyzing the operation process of the improved hybrid genetic algorithm (IHGA), the operation process is simply described as shown in Figure 6.

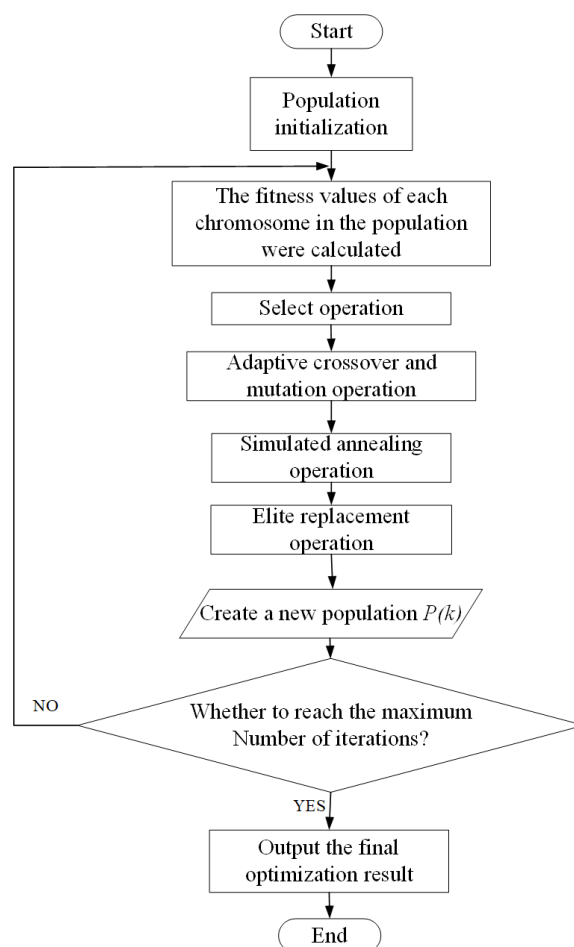


Figure 6. Flowchart of improved hybrid genetic algorithm operation.

Step 1: After analyzing the problem to be solved optimally, N chromosomes (N scheduling schemes displayed in coded format) are randomly encoded to form the initial population of P_0 (the initial effective scheme).

Step 2: According to the fitness function determined by the mathematical model, the fitness values of all chromosomes are obtained (evaluate the advantages and disadvantages of the scheduling scheme).

Step 3: The tournament method was used for the selection operation, and the appropriate winners are selected during iteration to ensure a better population (the scheduling scheme is further optimized).

Step 4: The crossover or mutation operation is carried out by adaptive adjustment strategy (crossover or mutation among scheduling schemes) so that the population diversity is high in the early stage (the diversity of scheduling schemes is improved) and the outstanding individuals are not easily eliminated in the later stage (the high-quality process in the scheduling scheme is retained).

Step 5: Using the simulated annealing factor to strengthen the local search ability, including those selected for probability (high quality), in comparison with the mutations in the annealing stage of the operation using a cooling strategy, will ultimately distinguish the elite individual from the original optimal individuals, who are then compared to determine whether they should be replaced (compare and optimize new scheduling scheme with the initial scheduling scheme).

Step 6: If the number of iterations is compared with the final number of iterations, the final number of iterations cannot be reached, continue to perform the operations of Step 2 to Step 5; If the execution reaches the last generation, the optimal solution is output.

4. Experimental Analysis

To verify the effectiveness of IHGA in solving production the scheduling problems in discrete industries, the method is compared with the improved particle swarm optimization algorithm proposed by Liu Hongming et al. [28], the Quantum Whale optimization algorithm proposed by Yan Xu et al. [29], and traditional genetic algorithm. The FT series proposed by Fisher and Thompson in 1963, and the LA series proposed by Lawrence in 1984, are used for comparison [30]. IHGA is written in C++ language and runs in Visual Studio 2019 software. The operating environment is Windows10 system, the main frequency is 2.60 GHz, and the memory is 8 G on a personal laptop.

The parameters of the IHGA are set as follows: population size $N = 200$, number of evolutionary iterations $G = 100$ (according to the results of the previous algorithm, it will converge and find the optimal solution in about 50 iterations; thus, 100 iterations are used as the convergence criterion to ensure that the optimal solution is found), initial temperature $T_1 = 1000$, ending temperature $T_{\min} = 0.001$, cooling coefficient $\alpha = 0.98$, and Markov chain length $L = 200$ in simulated annealing.

4.1. Comparison of IHGA and Existing Popular Algorithms

For the discrete shop scheduling problem, the improved particle swarm optimization (IPSO) proposed by Liu et al. [28] and the Quantum Whale optimization algorithm (QWOA) proposed by Yan et al. [29] can deal with some production scheduling problems. However, their optimization performance obviously has some shortcomings compared with the hybrid algorithm proposed in this work. Based on two kinds of algorithms related to case set scheduling, and when compared with IHGA, the obtained case set scheduling results, and the comparison of the two aspects, mainly results in the average of the scheduling results of the optimal solution and the solution of the data comparison, as shown in Table 2. The $n \times m$ represents the total number of parts and machine; the product of C^* for the optimal solution has been obtained; the Avg. runs the algorithm ten times to obtain the average of the solutions. According to the comparison results in Table 2, for scheduling case sets FT06, LA01, and LA06, the IHGA algorithm proposed in this work can obtain the known optimal solution due to the excellent local search ability of the simulated annealing operator.

Table 2. Comparison of IHGA, QWOA, and IPSO.

Numerical Example	The Size Is $n \times m$	C^*	QWOA		IPSO		IPSO	
			The Optimal Solution	Avg.	The Optimal Solution	Avg.	The Optimal Solution	Avg.
FT06	6×6	55	55	55	55	55	55	55
FT10	10×10	930	983	1045	976	1027	956	982
FT20	20×5	1165	1223	1313	1206	1222	1188	1209
LA01	10×5	666	666	674	666	666	666	666
LA06	15×5	926	926	927	926	926	926	926
LA16	10×10	945	958	1012	973	1011	945	954

As shown in Table 2, although IHGA did not obtain the known optimal solution for algorithm case set FT10 and FT20, the relevant solutions obtained by IHGA are superior to the other two algorithms, with a significantly stronger optimization capability.

4.2. Comparison of IHGA and Traditional GA

To further confirm that the improved algorithm is superior to the standard genetic algorithm, the algorithm proposed in this work is compared with the genetic algorithm in searching results, as shown in Table 3.

Table 3. Comparison of IHGA and GA optimization results.

Numerical Example	Size ($n \times m$)	C^*	GA		IHGA		Improved Effect/%	
			Optimal Solution	Avg.	Optimal Solution	Avg.	Optimal Solution	Avg.
FT06	6×6	55	55	55.4	55	55	0	0.7
FT10	10×10	930	1020	1050	951	990	6.8	5.7
FT20	20×5	1165	1269	1326	1182	1215	6.9	8.4
LA01	10×5	666	666	668	666	666	0	0.2
LA03	10×5	597	597	658	597	609	0	7.5
LA06	15×5	926	926	927	926	926	0	0.1
LA08	15×5	863	863	928	863	870	0	6.3
LA13	20×5	1150	1150	1210	1150	1161	0	4.1
LA18	10×10	848	885	946	848	868	4.2	8.2

It can be seen from Table 3 that, in the cases of FT06, LA01, LA03, LA06, LA08, and LA13, both algorithms can finally obtain the optimal solution of the scheduling problem; however, the average value obtained by IHGA is superior to GA. The IHGA did not find the optimal solution of scheduling cases in FT10, FT20, and LA18. However, compared with GA, the optimization effect was greatly improved, with the improvement effect ranging from 4.2% to 6.9% and the average improvement effect ranging from 5.7% to 8.4%. In the comparison of the two algorithms, the search result is greatly improved, mainly because the simulated annealing operator improves the local search ability of the genetic algorithm, which greatly improves the search ability of the algorithm proposed in this work.

Taking LA03 in the test datasets as an example, the feasibility and superiority of the algorithm proposed in this work in solving discrete scheduling problems are explained in detail. The specific datasets of LA03 are shown in Table 4. The data in Table 4 show that the processing parts uses the serial number of the machine and the processing time; for example, in the corresponding parts in the Table 2, 1/29 means that, in the process of selecting the Numbers for 1 s process machinery for processing, it takes time for 29, and other data have the same meaning.

Table 4. Details of LA03 datasets.

Artifacts	Process	Each Process Processing Machine and Processing Time				
		Process 0	Process 1	Process 2	Process 3	Process 4
Artifacts 1		1/23	2/45	0/82	4/84	3/38
Artifacts 2		2/21	1/29	0/18	4/41	3/50
Artifacts 3		2/38	3/54	4/16	0/52	1/51
Artifacts 4		4/37	0/54	2/74	1/62	3/57
Artifacts 5		4/37	0/81	1/61	3/68	2/30
Artifacts 6		4/81	0/79	1/89	2/89	3/11
Artifacts 7		3/33	2/20	0/91	4/20	1/66
Artifacts 8		4/24	1/84	0/32	2/55	3/8
Artifacts 9		4/56	0/7	3/54	2/64	1/39
Artifacts 10		4/40	1/83	0/19	2/8	3/7

For example, it can be seen from the comparison of results in Table 3 that both IHGA and GA algorithms can find the optimal solution; however, the average value obtained by IHGA is better than that obtained by GA, which means that IHGA has good robustness. After a period of comparative testing, the convergence of the iterative curves of the

two algorithms, in dealing with the LA03 scheduling problem, is shown in Figure 7. As can be seen from Figure 7, GA not only has a slow convergence speed when searching for the optimal value, but also cannot find the optimal value, and converging occurs at about 630. The IHGA has a relatively fast convergence speed and can jump out of the local optimum in time and find the global optimum value of 597. This indicates that the IHGA proposed in this work is obviously superior to the genetic algorithm and can solve the production scheduling problem of the workshop more effectively.

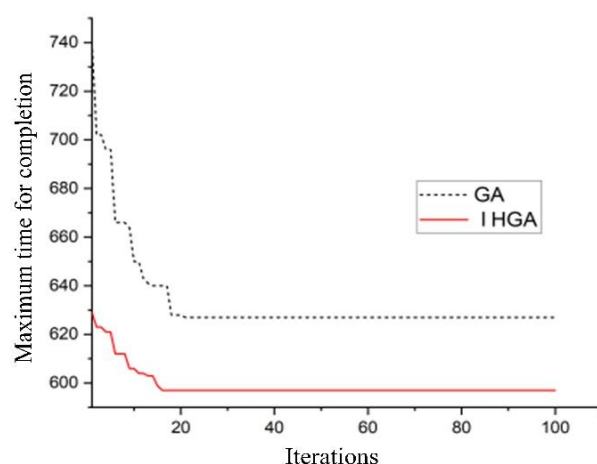


Figure 7. LA03 convergence curve.

Taking LA03 as an example, the IHGA can determine the currently known optimal fitness solution 597, select the chromosome with the best fitness, and obtain the corresponding scheduling Gantt chart according to the process arrangement results of each part in the chromosome, as shown in Figure 8. The Gantt chart can prove that the IHGA proposed in this work can provide an excellent and effective solution for production scheduling in discrete industries.

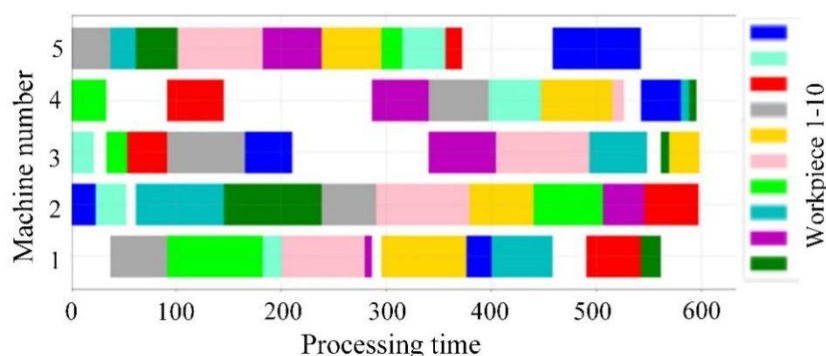


Figure 8. Scheduling Gantt diagram of LA03 obtained by the IHGA.

5. Industrial Field Test

A MES system is developed by the C# object-oriented development language, and the corresponding database is established by an SQL server, where the production scheduling function adopts the proposed IHGA. To verify the feasibility of the MES system, industrial application tests were carried out with the help of a network environment and hardware platform used in the discrete manufacturing industry. The reducer production project of the industry was selected as the basis, and the business process of the designed MES system was verified in a workshop. The specific steps are as follows:

Step 1: Production plan. According to order demands, the production planner makes a specific reducer production project, where the 12 kinds of parts are set in the production process, and the project is distributed after the process and working hours of each part are

modified. The system records the names of the employees who distribute and receive the production plan. The distributed production plan contains basic information.

Step 2: Production scheduling. The team leader of the production unit checks the distributed production plan to conduct scheduling optimization according to the IHGA, and the optimized results through Gantt chart is displayed as shown in Figure 9. The processing sequence and corresponding processing time of different parts on different equipment are showed, and further optimization of the production plan has been realized, including specific production time and operating staff.

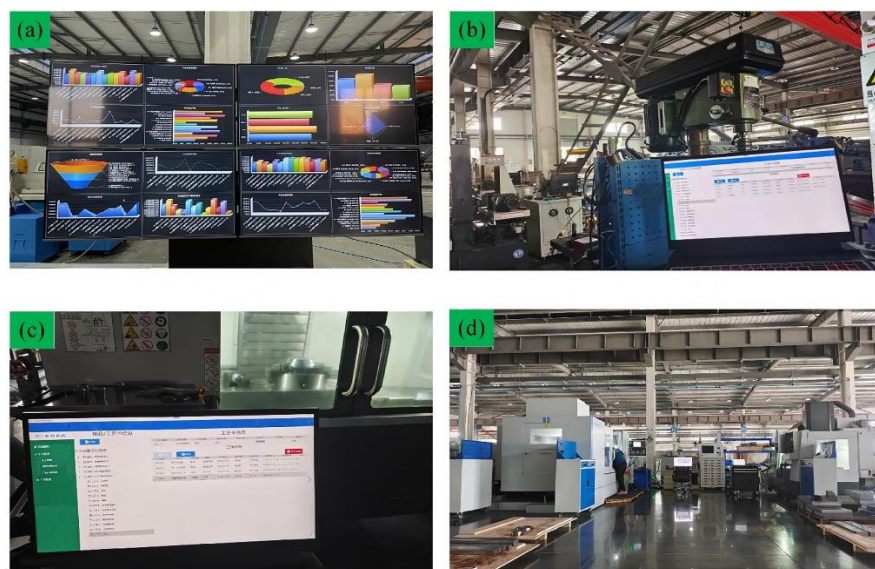


Figure 9. Production scene diagram of discrete manufacturing industry. (a) The workshop produces kanban boards. (b) Gear drilling machine processing process. (c) Machining process of bushing lathe. (d) Bushing lathe workshop.

Step 3: Production and processing. Workshop operators receive the production plan distributed by the team leader, logging in the MES system according to their own authority and obtaining the parts to be produced. Then, the parts are produced according to the time of the production process and process drawings. The system records the name and production time of employees, and the production team leader using the system can see the production status of the process at any time, as shown in Figure 10.

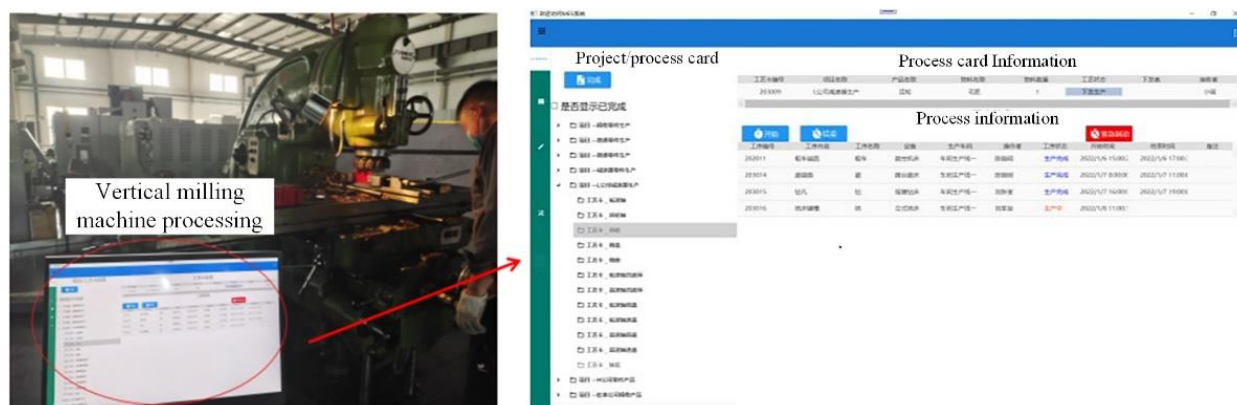


Figure 10. Gear shop operation diagram.

The industrial experiment proves that the IHGA can realize the production scheduling of the industry so that the production of a variety of products can be completed in

a relatively short time, which greatly improves the production efficiency of the workshop. According to the detailed production plan generated after scheduling, the on-site production situation of the workshop is shown in Figure 10. It can be further seen that, when the operator uses a drilling machine and lathe to process the reducer workpiece, the optimized software can run normally and record the production time, equipment name, and employee name to reflect the processing status of the parts and facilitate the traceability of defective products. Real-time display is carried out on a large screen in the workshop to reflect the working efficiency of the staff, the running status of the equipment, and the production situation, etc., and the transparent production of the workshop is realized.

6. Conclusions and Future Works

Firstly, the basic situation of production scheduling in discrete industries is introduced in this paper. Considering that the simulated annealing algorithm can improve the shortcomings of genetic algorithm optimization, an improved hybrid genetic algorithm is proposed to solve the problem of production scheduling. In this work, the shortcomings of the genetic algorithm are analyzed and studied, and the adaptive strategy is adopted to adjust the probability of crossover and variation of genetic operators. The adjustment mainly depends on the number of iterations and fitness, so that the population diversity is high in the early stage and convergence is possible in the later stage. For the added simulated annealing operator, the elitist substitution strategy is adopted to save the optimal solution and avoid the occurrence of the local optimal of the genetic algorithm. The results show that the improved hybrid genetic algorithm has a better optimization ability compared with other scheduling algorithms. Compared with the genetic algorithm, although the optimal solution was not obtained in the three cases, the improvement effect of the optimal solution was 4.2%~6.9%, and the average improvement range was 5.7%~8.4%. The simulated annealing operator played a good local optimization effect in IHGA, which greatly improved the optimization ability of the overall algorithm.

In the actual production process of the discrete industry, production scheduling considers more complex indexes; therefore, the next main task is to study the flexible production scheduling with multiple indexes, improve the production efficiency of the workshop, and reduce resource waste. More cases will be used for simulation comparison to verify the effectiveness of the method.

Author Contributions: Conceptualization, L.D. and H.L.; methodology, D.H., X.L., H.C. and A.G.; software, A.G. and G.K.; validation, G.K. and Z.L.; formal analysis, L.D. and H.L.; investigation, D.H., A.G., G.K. and Z.L.; resources, X.L., D.H. and H.C.; data curation, L.D., H.L. and H.C.; writing—original draft preparation, L.D., X.L. and Z.L.; writing—review and editing, A.G. and G.K.; visualization, A.G.; supervision, Z.L.; project administration, X.L.; funding acquisition, X.L. and Z.L. All authors have read and agreed to the published version of the manuscript.

Funding: The research leading to these results has received funding from the Norwegian Financial Mechanism 2014–2021 under Project Contract No 2020/37/K/ST8/02748.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data that can reproduce the results in this study can be requested from the corresponding author.

Acknowledgments: The authors acknowledge the support from the Natural Science Foundation of the Jiangsu Higher Education Institution of China (22KJD460004), National Science Foundation of China (No. 51975568), Natural Science Foundation of Jiangsu Province (No. BK20191341), Qing Lan Project for Excellent Young Key Teachers of Colleges and Universities of Jiangsu Province (2022) in carrying out this research are gratefully acknowledged.

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

References

- Wang, P.; Xiao, X.C.; Yuan, L.H.; Fang, J. Application and Research in Discrete Manufacturing Industry Based on MES. *China Sci. Technol. Inf.* **2015**, *18*, 119–120.
- Lu, Y.; Feng, Z.J.; Hu, Y. An Automated Guided Vehicle Deals with Scheduling Algorithm for Multiple Transport Request. *Modul. Mach. Tools Autom. Processing Technol.* **2019**, *2*, 157–160. [\[CrossRef\]](#)
- Kumar, N.; Mishra, A. Comparative Study of Different Heuristics Algorithms in Solving Classical Job Shop Scheduling Problem. *Mater. Today Proc.* **2020**, *22*, 1796–1802. [\[CrossRef\]](#)
- Kong, L.; Wang, L.M.; Li, F.I. Green Production Scheduling of Hybrid Flow Shop Based on Machining Matching Characteristics of Machine Tool. *Comput. Integr. Manuf. Syst.* **2019**, *25*, 11.
- Chen, X.H. *Research on Reconfigurable Architecture and Key Technologies of Discrete Manufacturing Execution System Based on Semantic Gateway*; Chongqing University: Chongqing, China, 2014.
- Bian, D.Z.; Guo, J.M.; Wang, M.; Song, F.F. Research on MES System Based on Discrete Shipbuilding. *Manuf. Technol. Mach. Tools* **2020**, *1*, 149–152. [\[CrossRef\]](#)
- Mao, K.; Pan, Q.K.; Pang, X.F. Lagrangian Algorithm for Steelmaking Continuous Casting Production Scheduling Problem. *J. Syst. Eng.* **2014**, *29*, 13.
- Bellabdaoui, A.; Teghem, J. A Mixed-integer Linear Programming Model for The Continuous Casting Planning. *Int. J. Prod. Econ.* **2006**, *104*, 260–270. [\[CrossRef\]](#)
- Bellman, R.E.; Dreyfus, S.E. *Applied Dynamic Programming*; Princeton University Press: Princeton, NJ, USA, 2015.
- Kirkpatrick, S.; Gelatt, D.C., Jr.; Vecchi, M.P. Optimization by simulated annealing. *Science* **1983**, *220*, 671–680. [\[CrossRef\]](#)
- Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
- Dorigo, M.; Gambardella, L.M. Ant colonies for the travelling salesman problem. *Bio. Syst.* **1997**, *43*, 73–81. [\[CrossRef\]](#)
- Choi, I.C.; Choi, D.S. A Local Search Algorithm for Job Shop Scheduling Problems with Alternative Operations and Sequence-dependent Setups. *Comput. Ind. Eng.* **2002**, *42*, 43–58. [\[CrossRef\]](#)
- Nie, L.; Gao, L.; Li, P.; Li, X. A GEP-Based Reactive Scheduling Policies Constructing Approach for Dynamic Flexible Job Shop Scheduling Problem with Job Release Dates. *J. Intell. Manuf.* **2013**, *24*, 763–774. [\[CrossRef\]](#)
- Chen, J.C.; Chen, K.H.; Wu, J.J.; Chen, C.W. A Study of The Flexible Job Shop Scheduling Problem with Parallel Machines and Reentrant Process. *Int. J. Adv. Manuf. Technol.* **2008**, *39*, 344–354. [\[CrossRef\]](#)
- Lu, T.T.; Chen, P.; Wan, X.Y. An Improved Cellular Genetic Algorithm for Solving Multi-objective Flexible Job Shop Scheduling Problem. *Mod. Manuf. Eng.* **2016**, *11*, 41–49. [\[CrossRef\]](#)
- Gao, J.; Gen, M.; Sun, L.; Zhao, X. A Hybrid of Genetic Algorithm and Bottleneck Shifting for Multi-objective Flexible Job Shop Scheduling Problems. *Comput. Ind. Eng.* **2007**, *53*, 149–162. [\[CrossRef\]](#)
- Pezzella, F.; Morganti, G.; Ciaschetti, G. A Genetic Algorithm for The Flexible Job-shop Scheduling Problem. *Comput. Oper. Res.* **2008**, *35*, 3202–3212. [\[CrossRef\]](#)
- De Giovanni, L.; Pezzella, F. An Improved Genetic Algorithm for The Distributed and Flexible Job-shop Scheduling Problem. *Eur. J. Oper. Res.* **2010**, *200*, 395–408. [\[CrossRef\]](#)
- Zhang, R.; Chang, P.C.; Wu, C. A Hybrid Genetic Algorithm for The Job Shop Scheduling Problem with Practical Considerations for Manufacturing Costs: Investigations Motivated by Vehicle Production. *Int. J. Prod. Econ.* **2013**, *145*, 38–52. [\[CrossRef\]](#)
- Chamnanlor, C.; Sethanan, K.; Gen, M.; Chien, C.F. Embedding Ant System in Genetic Algorithm for Re-entrant Hybrid Flow Shop Scheduling Problems with Time Window Constraints. *J. Intell. Manuf.* **2015**, *40*, 1–17.
- Meng, Y.; Zhao, S.H. Solving Job Shop Scheduling Problem with Hybrid Algorithm Integrating Path Reconnection. *Mech. Eng.* **2019**, *12*, 32–36, 39.
- Zhao, W. Application of Simulated Annealing Genetic Algorithm in Job Shop Scheduling. *Comput. Simul.* **2011**, *28*, 361–364.
- Slowik, A.; Bialko, M. Modified version of roulette selection for evolution algorithms—The fan selection. Lecture Notes in Artificial Intelligence; In *International Conference on Artificial Intelligence and Soft Computing*; Springer: Berlin/Heidelberg, Germany, 2004; Volume 3070, pp. 474–479.
- Wenhao, L.; Zongnan, H. Analysis of genetic selection operator performance in job shop scheduling problem. *Manuf. Technol. Mach. Tool* **2011**, *3*, 124–128.
- Shijun, R.; Liang, C. Study on improvement strategy for crossover operation in genetic algorithms. *J. Harbin Univ. Commer.* **2006**, *22*, 60–62.
- Jiahai, W.; Cheng, L. Application of a genetic algorithm combined with simulated annealing in flexible job shop scheduling. *Digit. Technol. Appl.* **2019**, *37*, 133–136.
- Liu, H.M.; Zeng, H.Y.; Zhou, W.; Wang, T. Optimization of Job Shop Scheduling Problem Based on Improved Particle Swarm Optimization. *J. Shandong Univ. Eng. Ed.* **2019**, *49*, 75–82.
- Yan, X.; Ye, C.M.; Yao, Y. Quantum Whale Optimization Algorithm for Job Shop Scheduling. *Comput. Appl. Res.* **2019**, *36*, 975–979.
- Beasley, J.E. OR-Library [DB /OL], (2018-02-01) [2022-09-08]. Available online: <http://people.brunel.ac.uk/~{}mastjib/jeb/info.html> (accessed on 9 August 2022).