

Article

Dual Resource Constrained Flexible Job Shop Scheduling Based on Improved Quantum Genetic Algorithm

Shoujing Zhang ¹, Haotian Du ¹, Sebastian Borucki ² , Shoufeng Jin ¹, Tiantian Hou ¹ and Zhixiong Li ^{3,*}

¹ Xi'an Key Laboratory of Modern Intelligent Textile Equipment, Department of Industrial Engineering, Xi'an Polytechnic University, Xi'an 710600, China; zhangshoujing@xpu.edu.cn (S.Z.); 2018022067@stu.xpu.edu.cn (H.D.); jinshoufeng@xpu.edu.cn (S.J.); 200211006@stu.xpu.edu.cn (T.H.)

² Institute of Electrical Power Engineering and Renewable Energy, Opole University of Technology, 45-758 Opole, Poland; s.borucki@po.edu.pl

³ Yonsei Frontier Lab, Yonsei University, 50 Yonsei-ro, Seodaemun-gu, Seoul 03722, Korea

* Correspondence: zhixiong.li@yonsei.ac.kr

Abstract: Aiming at solving the problem of dual resource constrained flexible job shop scheduling problem (DRCFJSP) with differences in operating time between operators, an artificial intelligence (AI)-based DRCFJSP optimization model is developed in this paper. This model introduces the differences between the loading and unloading operation time of workers before and after the process. Subsequently, the quantum genetic algorithm (QGA) is used as the carrier; the process is coded through quantum coding; and the niche technology is used to initialize the population, adaptive rotation angle, and quantum mutation strategy to improve the efficiency of the QGA and avoid premature convergence. Lastly, through the Kacem standard calculation example and the reliability analysis of the factory workshop processing process example, performance evaluation is conducted to show that the improved QGA has good convergence and does not fall into premature ability, the improved QGA can solve the problem of reasonable deployment of machines and personnel in the workshop, and the proposed method is more effective for the DRCFJSP than some existing methods. The findings can provide a good theoretical basis for actual production and application.

Keywords: dual resource constrained; flexible job shop scheduling; improved quantum genetic algorithm; niching technique; adaptive rotation angle



Citation: Zhang, S.; Du, H.; Borucki, S.; Jin, S.; Hou, T.; Li, Z. Dual Resource Constrained Flexible Job Shop Scheduling Based on Improved Quantum Genetic Algorithm.

Machines **2021**, *9*, 108. <https://doi.org/10.3390/machines9060108>

Academic Editor:
Ratna Babu Chinnam

Received: 13 April 2021
Accepted: 21 May 2021
Published: 24 May 2021

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



Copyright: © 2021 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

Job shop scheduling (JSP) is the basis of intelligent manufacturing management and decision-making, and JSP optimization is the core of advanced manufacturing technology and modern management technology [1,2]. JSP is involved with a set of machines to process a set of work parts. Each work part is formed by a series of processes with sequential constraints. Each process requires only one machine, which is always available and can process one operation at one time without interruption [3]. Flexible job shop scheduling problem (FJSP) is an extension of the traditional JSP, in which several processes of workpieces are allowed to be processed by several machines at the same time. At present, most FJSP models only consider the constraints of machinery and equipment. However, in various complex man-machine systems, about 60–90% of failures are attributed to operator errors [4]. The research on dual resource constrained flexible job shop scheduling problem (DRCFJSP) has theoretical significance and practical value.

Generally, the analytical [5] and simulation methods [6] are mainly used to solve DRCFJSP. However, due to the complexity of worker factors such as the technical level, fatigue level, learning ability, and assignment rules, each manufacturing system presents its special features, and a generic analytical method is not available to establish an accurate model for a specific manufacturing system. Moreover, the simulation method often requires a long time to develop, debug, and run. Fortunately, the meta-heuristic algorithm does not

rely on the unique conditions of the problem and has acceptable computational efficiency, so it is widely used in the DRCFJSP. The genetic algorithm, particle swarm algorithm, and ant colony algorithm are typical representatives. Li et al. [7] used the branch population genetic algorithm to solve the dual resource constraint problem. The elite evolution operator, sector segmentation, and neighborhood search mechanism were adopted to achieve the maximum completion time and optimal cost. Zhong et al. [8] proposed a branch genetic algorithm based on a compressed time window scheduling strategy to reduce the maximum completion time and total processing cost by 7.4% and 4.7%, respectively. Gong et al. [9] designed a new type of hybrid genetic algorithm that uses three-layer chromosome coding to complete the job sequence and reasonable allocation of worker resources. Wu et al. [10] established a model with the goal of minimizing the maximum completion time, and combined the hybrid genetic algorithm with variable neighborhood search to improve the local search capabilities. Zhang et al. [11] proposed a hybrid particle swarm algorithm based on the maximum fitness function and adopted a dynamic search strategy to enhance the local search capability of the particle swarm. Gong et al. [12] used the NSGA-II algorithm framework and the memetic algorithm of neighborhood search to satisfy the balance of minimizing the maximum completion time, the maximum workload of the machine, and the total workload of all machines. Geng et al. [13] used the same method to satisfy the balance of minimizing the maximum completion time, minimizing the total delay time, and minimizing the workload of workers. Zheng et al. [14] proposed a knowledge-based fruit fly optimization algorithm (KGFOA), which used a new coding scheme to solve the DRCFJSP problem with the goal of minimum production time. Yang et al. [15] used the limited search space-based algorithm to solve the DRCFJSP with multi-layer product structure. Li et al. [16] considered scheduling goals such as production cycle, machine workload, and product cost, and proposed a decomposition-based multi-objective evolutionary algorithm (MOEA/D) to solve them. Andrade-Pineda et al. [17] studied the due date changes, delay in arrival, changes in job processing time and rush jobs, etc., and developed a constructive iterative greedy program that can effectively deal with large-scale dual-objective DRCFJSP. Yazdani et al. [18] proposed a hybrid meta-heuristic algorithm to solve DRCFJSP. The algorithm used the variable neighborhood search and simulated annealing algorithm to search in the solution space, and used the randomly generated test questions for computational research to evaluate the performance of the proposed algorithm. Yazdani et al. [19] proposed two meta-heuristic algorithms, namely, simulated annealing algorithm (SA) and vibration reduction optimization algorithm (VDO), to solve the DRCFJSP problem. Defersha et al. [20] regarded skilled setting operators as constrained resources and developed a multi-test/optimal mobile simulated annealing method for the dual resource-constrained flexible job shop scheduling problem (DRC-FJSP). Chatzikonstantinou et al. [21] used a hybrid algorithm between the global search meta-heuristic algorithm and the adaptive greedy operation allocation and scheduling algorithm to solve the problem. Some scholars also used the hybrid artificial bee colony algorithm [22], improved water wave optimization algorithm [23], and bat algorithm [24] to solve DRCFJSP.

To sum up, most of the research on DRCFJSP did not consider the difference of the technical level of the personnel. Differences in the skill level of workers directly lead to differences in the operating time. So, it is vital to consider the differences in the operating time of workers to establish a more accurate production model. At the same time, in the metaheuristic algorithms such as a bat algorithms or Cuckoo Search, the optimization performance partially depends on the initial population. It is possible to calculate the statistic indexes such as the mean, maximum, minimum, and standard deviation to rate the different modifications. With the combination of the quantum mechanics and intelligent algorithms, the quantum genetic algorithm (QGA) [25] is proposed in literature. At present, QGA is used in path planning [26], mechanical structure design [27], and network coverage [28]. It has achieved good results, but there is still a lack of corresponding research on the DRCFJSP.

To solve the problem of manual operation time difference, according to the characteristics of DRCJSP that need to occupy workers and machines at the same time during processing, this paper proposes a combination of personnel information and machine information as a decoding mechanism based on the use of QGA. An example is used to evaluate the feasibility and effectiveness of the proposed algorithm. The analysis results demonstrate that this article establishes a more refined model for adding personnel operating time differences in the DRCJSP solution.

2. Research on DRCFJSP

2.1. DRCFJSP Problem Description

In the DRCFJSP problem, it is assumed that a processing system has n workpieces $J_i (i = 1, 2, \dots, n)$ to be processed. Each workpiece J_i contains n_i processes, and each process is denoted as O_{ij} . Then, each process can be processed by choosing one of the available $M_{ij} = \{1, 2, \dots, m\}$ equipment. There are w multi-skilled workers in total, and each worker masters the operation technology of multiple devices. $W_m (r = 1, 2, \dots, w)$ denote the set of workers that can control equipment m [6–9]. In addition to the impact of processing equipment, workshop production capacity is also affected by human resources. The purpose of scheduling is to rationally arrange the processing sequence of workpieces on each equipment under the dual constraints of equipment and manpower. The schematic diagram of dual-resource intelligent workshop scheduling problem is shown in Figure 1.

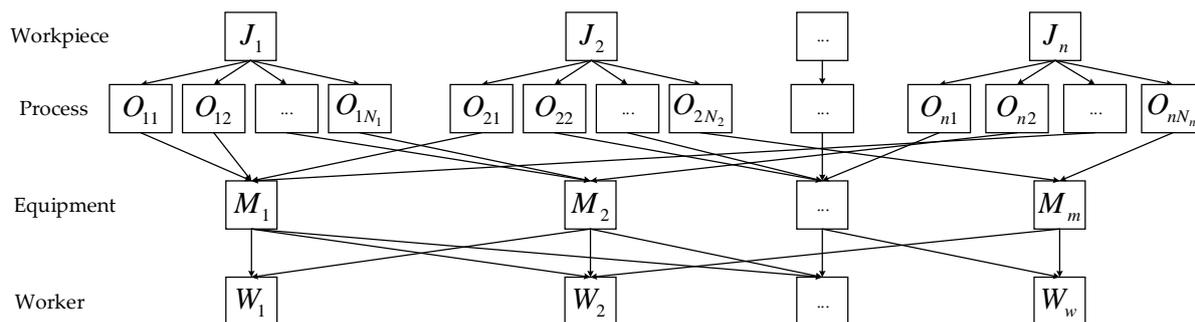


Figure 1. Schematic diagram of DRCFJSP.

The scheduling must meet the following constraints:

- (1) Each workpiece is not allowed to be interrupted while being processed on the equipment;
- (2) Each piece of equipment can only process one of the processes of a workpiece at the same time;
- (3) Each process must be completed in accordance with the processing sequence, that is, the next process of the process route can be processed after one process is completed;
- (4) No priority restrictions between artifacts;
- (5) The transportation time in the production process is not considered;
- (6) Workers are responsible for the loading and unloading operations of process processing, and the processing time is determined by the standard working hours of the machine tool;
- (7) A worker can only operate one piece of equipment at a time.

2.2. Mathematical Model

According to the problem description in Section 2.1 of this article, the following mathematical model is established with the goal of minimizing the maximum completion time. The parameter description is shown in Table 1.

Table 1. Parameter description table.

Parameter	Description
$\max C_i$	The maximum completion time
r_{ij}	The start time of process O_{ij}
T_{ij}	The processing time of O_{ij}
C_{ij}	The end time of O_{ij}
T_{ijmw}	The loading and unloading time of worker w on equipment m
$X_{ijk} = \{0, 1\}$	When the value is 1, it means that the j process of the i workpiece is processed by the r worker on the k equipment.

The objective function $\min f$ is:

$$\min f = \min(\max C_i); i \in \{1, 2, \dots, n\}, \quad (1)$$

S.T.

$$r_{i(j+1)} - (r_{ij} + T_{ij} + T_{ijmw}) \geq 0, \quad (2)$$

$$\sum_{k=1}^m \sum_{r=1}^w X_{ijk} = 1; i \in \{1, 2, \dots, n\}, j \in \{1, 2, \dots, n_i\}, \quad (3)$$

$$C_{ih} - C_{pq} \geq r_{ih} + T_{ih} + T_{ihmw}, \quad (4)$$

Equation (1) indicates that the objective function is to minimize the maximum completion time;

Equation (2) means that each process can only proceed to the next process after the previous process is completed;

Equation (3) means that at the same time, a process can only be processed by one piece of equipment, and one of the w workers can process it;

Equation (4) means that assuming that both processes O_{ih} and O_{pq} are processed by worker w , process O_{ih} can only start process O_{pq} after complete processing, that is, at the same time, the same worker can only be responsible for the processing of one process.

2.3. Improved Quantum Genetic Algorithm to Solve DRCFJSP

QGA is an intelligent optimization algorithm that combines quantum computing and genetic algorithm. It was proposed by Han et al. [25]. It introduced quantum concepts such as the quantum states, quantum gates, quantum state characteristics, and probability amplitudes into the genetic algorithm. QGA is also a probabilistic search algorithm, which uses qubits to represent genes. The gene of the genetic algorithm expresses certain information, and in the quantum genetic algorithm, due to the superposition of the quantum information, the gene expressed by the qubit contains more useful information.

This paper improves on the basis of quantum genetic algorithm, adopts niche technology for population initialization, adaptive rotation angle to dynamically adjust, and adds quantum mutation strategy, which can effectively improve algorithm efficiency and avoid premature convergence of the algorithm.

2.3.1. Quantum Coding

This paper adopts the process coding method, in which the length of the real number code depends on the total number of processes. For example, in the problem that n workpieces are processed by m equipment, the length of the real number code is the sum of the number of operations of all the workpieces [29].

$$L = ([\log_2(m) + 1]) \sum_{i=1}^n n_i \quad (5)$$

In formula (5), $[\log_2(m) + 1]$ represents the length of the binary code corresponding to a decimal code and $\sum_{i=1}^n n_i$ represents the length of the decimal code.

Take 3 workpieces, 3 pieces of equipment, and 3 processes for each workpiece as an example.

Quantum coding:

$$Q(t) = \begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_{27} \\ \beta_1 & \beta_2 & \dots & \beta_{27} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \dots & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \dots & \frac{1}{\sqrt{2}} \end{bmatrix},$$

Binary coding:

$$P(t) = 010011001001010011011001010,$$

Decimal coding:

$$231123312,$$

Among the codes shown above, the decimal code is the code in the order of the process. Each number i represents a process, and the number of occurrences of the same number i is j , which represents the j process of the workpiece i . The processing sequence and processing equipment of the workpiece in this example can be expressed as $P_{21}, P_{31}, P_{11}, P_{12}, P_{22}, P_{32}, P_{33}, P_{13}, P_{23}$.

2.3.2. Niche Initialization Population

The niche technology is to divide each generation of individuals into several categories, select a number of individuals with greater fitness from each category as the outstanding representatives of a category to form a group, and then cross between the population and between different populations. Mutation produces a new generation of individual groups. At the same time, the pre-selection mechanism and the crowding-out mechanism or the sharing mechanism are used to complete the task. The introduction of niche technology can better maintain the diversity of solutions, and at the same time has a high global optimization capability and convergence speed, which is especially suitable for the optimization of complex multimodal functions [30]. The initial specific operations of qubits are as follows, where i represents the serial number of the subpopulation and N represents the total number of subpopulations.

$$\begin{bmatrix} \alpha_k \\ \beta_k \end{bmatrix} = \begin{bmatrix} \sqrt{i/N} \\ \sqrt{(1-i)/N} \end{bmatrix} \quad (6)$$

2.3.3. Adaptive Rotation Angle Adjustment

The quantum revolving door is the core of the quantum genetic algorithm. Its main function is to shift the probability amplitude of each locus in the chromosome to the optimal solution so that there is a greater chance of obtaining the optimal solution. The adjustment operation of the quantum revolving door is:

$$\begin{aligned} \begin{bmatrix} \alpha'_i \\ \beta'_i \end{bmatrix} &= \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} \\ &= \begin{bmatrix} \cos \theta \cdot \alpha_i & -\sin \theta \cdot \beta_i \\ \sin \theta \cdot \alpha_i & \cos \theta \cdot \beta_i \end{bmatrix} \end{aligned} \quad (7)$$

Among them, $\begin{bmatrix} \alpha'_i \\ \beta'_i \end{bmatrix}$ represents quantum chromosome coding; θ represents the rotation angle; its sign (positive or negative) determines the direction of algorithm convergence; and its magnitude determines the speed and efficiency of the algorithm's convergence.

In the quantum genetic algorithm, to determine the direction of the rotation angle of the quantum revolving gate, this paper adopts the general adjustment strategy proposed

in the literature [31], as shown in Table 2. Among them, rotation angle $\theta = S(\alpha_i, \beta_i) \cdot \Delta\theta$. $S(\alpha_i, \beta_i)$ represents the direction of rotation, and it can ensure that the algorithm rotates in a better direction. $\Delta\theta$ represents the rotation angle, and it can control the convergence rate. X_i is the i position of the current chromosome, and b_i is the i position of the optimal chromosome. $f(X_i)$ is the fitness of the current individual, and $f(b_i)$ is the fitness of the optimal individual.

Table 2. Quantum rotation angle update strategy.

X_i	b_i	$f(X_i) > f(b_i)$	$\Delta\theta$	$S(\alpha_i, \beta_i)$			
				$\alpha_i\beta_i > 0$	$\alpha_i\beta_i < 0$	$\alpha_i = 0$	$\beta_i = 0$
0	0	False	0	-	-	-	-
0	0	True	0	-	-	-	-
0	1	False	δ	+1	-1	0	± 1
0	1	True	δ	-1	+1	± 1	0
1	0	False	δ	-1	+1	± 1	0
1	0	True	δ	+1	-1	0	± 1
1	1	False	0	-	-	-	-
1	1	True	0	-	-	-	-

The dynamic rotation angle is used in this paper, and the formula for a value is:

$$\delta = \theta_{\min} + \frac{f_{\max} - f_x}{f_{\max}} (\theta_{\max} + \theta_{\min}) \tag{8}$$

Among them, $\delta \in [0.001\pi, 0.005\pi]$. This dynamic rotation angle can make the algorithm find the optimal solution faster without falling into premature maturity.

2.3.4. Quantum Mutation

Mutation is to make the results jump out of premature and improve local search power. In the quantum genetic algorithm, the mutation operation is realized through the quantum NOT gate.

The qubit mutated through the quantum NOT gate U is:

$$U \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} = \begin{bmatrix} \beta_i \\ \alpha_i \end{bmatrix}$$

2.4. Algorithm Steps

Figure 2 shows the specific flow chart of the algorithm. The algorithm flow is:

Step 1: Initialize the parameters, determine the population size pop , the maximum algebra $MGen$, the number of quantum chromosomes $lenchrom$ and the quantum mutation probability P_m , and the niche technology initializes the population;

Step 2: Measure the population, get the binary code, and get the observation state $P(t)$ according to the initial state $Q(t)$ of the population;

Step 3: Calculate and save, convert the observation state $P(t)$ into a decimal code, obtain a process-based code, obtain a set of solution vectors through decoding technology, evaluate it through the fitness function, and set the population optimal to the global optimal value;

Step 4: Judge whether the termination condition is satisfied, the termination algorithm is satisfied, and *Step5* is executed if it is not satisfied;

Step 5: Update. Calculate the rotation angle of the quantum revolving gate according to the dynamic quantum rotation angle method and perform quantum mutation operation to update the population $Q(t)$;

Step 6: Observe and decode the new population, determine whether the optimal value of the new population is greater than the global optimal value, and set the optimal value of the population to the global optimal value if it is satisfied, then return to *Step 4*.

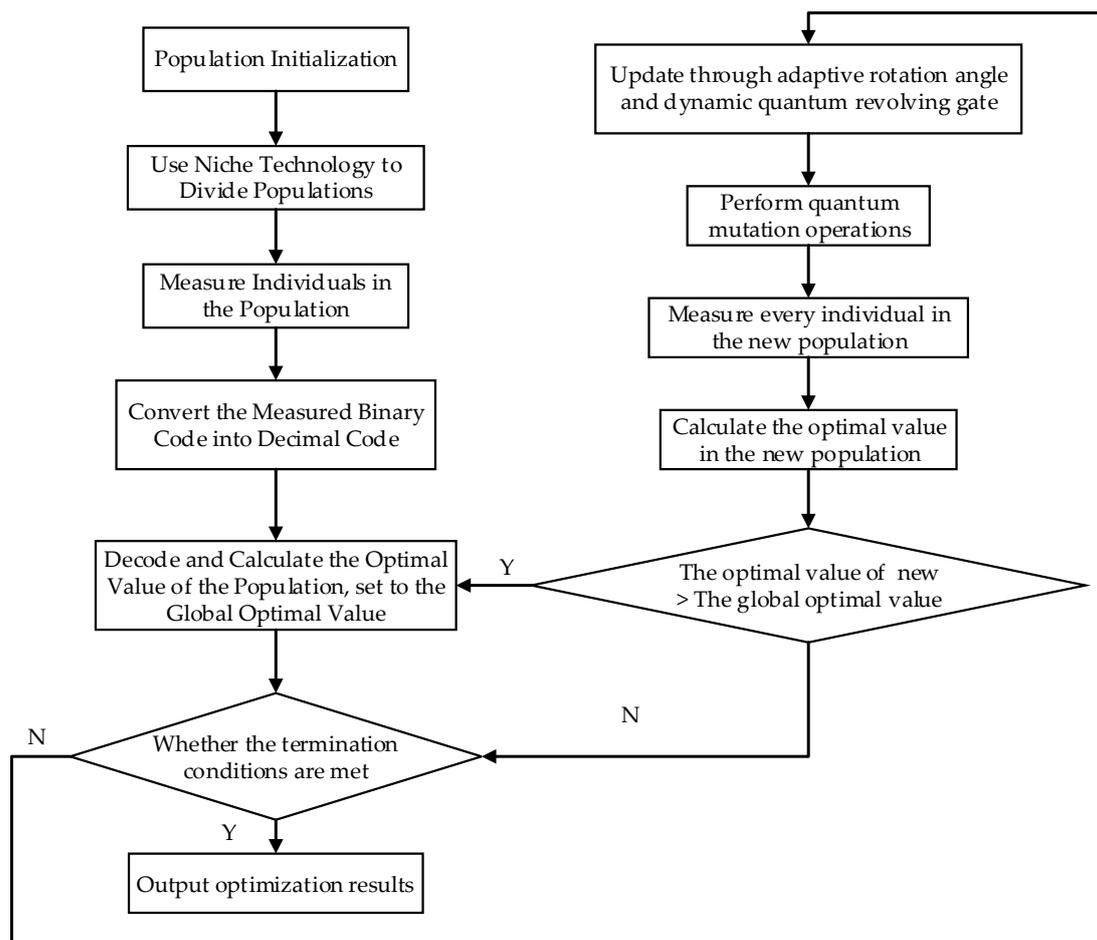


Figure 2. Flow chart of improved quantum genetic algorithm.

3. Results

3.1. Standard Calculation Examples

In order to verify the performance of the algorithm in this paper more comprehensively, five Kacem standard calculation examples are solved [32]; the algorithm runs 30 times to obtain the optimal solution, and it is compared with the particle swarm algorithm [33], heuristic algorithm [34], meta-heuristic algorithm [35] for comparison, as shown in Table 3.

Table 3. Comparison table of Kacem standard examples.

Examples	$n \times m$	Standard Solution	Particle Swarm Algorithm	Heuristic Algorithm	Meta-Heuristic Algorithm	IQGA
Kacem01	4×5	11	11	11	11	11
Kacem02	8×8	14	17	15	14	14
Kacem03	10×7	11	-	13	13	12
Kacem04	10×10	7	8	7	7	7
Kacem05	15×10	11	-	12	14	12

3.2. Actual Calculation Example

In order to demonstrate the effectiveness of the improved algorithm, data verification is carried out based on a certain enterprise example. The company is a production logistics equipment company that selects six types of parts produced by belt conveyors, and different processes of different workpieces can be processed on multiple machines. The improved

quantum genetic algorithm (IQGA) uses MATLAB to program the solution, and the solution results are compared and analyzed with the genetic algorithm (GA), global version of classic particle swarm algorithm (PSO), and the quantum genetic algorithm (QGA).

Table 4 shows whether the j process of the i workpiece can be processed on the machine and the processing time.

Table 4. Processing timetable of each process (min).

		Equipment Number							
		M1	M2	M3	M4	M5	M6	M7	M8
J1	O11	4	3	—	6	—	—	—	—
	O12	—	—	5	—	—	5	—	7
	O13	6	—	8	—	—	6	—	7
	O14	6	—	5	—	—	8	7	—
	O15	—	4	9	—	—	—	4	—
J2	O21	6	5	—	3	—	—	—	—
	O22	—	9	4	—	5	—	6	—
	O23	—	—	—	—	5	—	7	4
	O24	3	—	3	—	—	5	6	—
J3	O31	—	2	6	—	4	—	5	—
	O32	5	—	8	—	—	7	—	5
	O33	8	—	6	—	—	5	5	—
	O34	—	—	—	4	8	—	—	4
J4	O41	—	6	8	—	6	—	7	—
	O42	—	—	7	—	—	2	—	9
	O43	—	—	—	—	4	—	4	4
	O44	—	7	4	—	—	—	5	—
J5	O51	3	5	—	5	—	—	—	—
	O52	—	6	4	—	7	—	8	—
	O53	3	—	7	—	—	4	—	8
	O54	—	—	—	—	4	—	5	3
	O55	—	—	—	3	8	—	—	3
J6	O61	—	5	9	—	6	—	9	—
	O62	—	—	4	—	—	3	—	8
	O63	5	—	9	—	—	6	—	6
	O64	—	5	6	—	—	—	3	—

Table 5 shows whether worker w can use machine m and the loading and unloading time of machine m .

Table 5. Time schedule for worker to operate machine loading and unloading (min).

Worker	M1	M2	M3	M4	M5	M6	M7	M8
W1	2	3	—	—	—	—	—	—
W2	—	2	3	—	5	—	—	—
W3	—	—	—	3	—	4	5	—
W4	3	—	—	—	3	—	—	6
W5	—	—	2	—	—	—	4	4
W6	—	—	—	2	—	3	—	—

The parameters hyperparameters are shown in Table 6. The parameters of the GA are set as follows: the population size is 100, the number of iterations is 100, the crossover rate is 0.8, and the mutation rate is 0.05. The parameters of the PSO are set as follows: the population size is 100, the number of iterations is 100, the inertia weight $w = 0.73$, and the learning factor $c1 = 2$, $c2 = 2.1$. The parameters of the QGA and IQGA are set as: the population size is 100, the number of iterations is 100, and the mutation rate is

0.05. The above algorithm uses the combination of PBX and LOX crossover operators as the crossover operation, and uses the mutual exchange of two segments of genes as the mutation operation [36]. In order to compare IQGA with other algorithms, each algorithm was run 10 times independently, and the results of the three algorithms after running 10 times are shown in Table 7. Among them, *Best* represents the minimum value of the maximum completion time of the algorithm running ten times, *Worst* represents the maximum value of the maximum completion time of the algorithm running ten times, and *Average* represents the average value of the maximum completion time of the algorithm running ten times.

Table 6. Algorithm hyperparameters table.

Algorithm	Population Size	Number of Iterations	Other Parameters
GA	100	100	Crossover rate 0.8, mutation rate 0.05
PSO	100	100	Inertia weight $w = 0.73$, learning factor $c1 = 2$, $c2 = 2.1$
QGA	100	100	Mutation rate 0.05
IQGA	100	100	Mutation rate 0.05

Table 7. Algorithm test results (min).

	Number of Runs	Algorithm			
		GA	PSO	QGA	IQGA
maxC_i	1	58	59	57	55
	2	59	56	56	56
	3	59	57	56	56
	4	59	57	56	55
	5	58	56	57	55
	6	57	56	57	56
	7	58	56	56	55
	8	57	56	56	55
	9	59	56	56	55
	10	59	57	56	55
	Best	57	56	56	55
	Worst	59	59	57	56
	Average	58.3	56.6	56.3	55.3

It can be seen from Table 5 that due to the randomness of the algorithms initial population and the crossover and mutation process, the values of each algorithm for ten runs are different. On the whole, GA and PSO are easy to fall into the local optimum. When the minimum maximum completion times reach 57 min and 56 min, respectively, they cannot continue to converge, and the maximum completion times are 59 min and 59 min, respectively; QGA uses quantum coding to achieve premature convergence can be avoided to a certain extent, making its maximum completion minimum of 56 min; IQGA adds niche technology, dynamic rotation angle, and quantum variation on the basis of QGA, and reaches the optimal value of 55 min in 7 out of 10 runs. Its maximum completion maximum is 56 min. From the average of the calculation results of the four types of algorithms, the average maximum completion times of GA, PSO, QGA, and IQGA are 58.3 min, 56.6 min, 56.3 min, and 55.3 min, respectively, which can also prove the superiority of the algorithm in this paper. It should be noted that although the average maximum completion time of PSO and QGA are close, the variance of the results of the PSO algorithm is large, and it has a significantly worse robustness than QGA.

Figure 3 shows the iterative process of four algorithms, among which the black dots are the iterative optimal solutions of the four algorithms. Within 60 iterations, GA, QGA, PSO, and IQGA can all find the optimal solution. The optimization speed of PSO and GA is significantly slower than that of QGA and IQGA, and GA has not found the optimal

solution, indicating that the QGA and IQGA have their advantages in dealing with the problem of workshop resource scheduling. At the same time, QGA falls into premature, and IQGA algorithm can find the optimal solution more accurately.

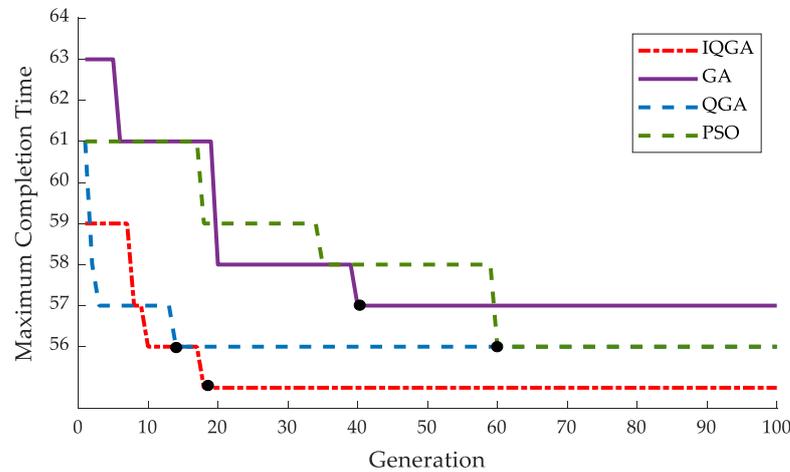


Figure 3. Algorithm iteration process diagram.

After solving the calculation example, Figure 4 is the Gantt chart of the scheduling plan obtained by the converged IQGA. Each solid line frame and the dashed line frame after it are combined to represent a process, and the length of the colored rectangle $O(i,j)$ represents the time required to complete the j process of the i workpiece. The time required for the first process, the length of the dashed box behind, indicates the time for workers R_k to load and unload materials in this process. For example, $O(5,1)$ and $R1$ in the lower left corner indicate that the first process of the fifth workpiece is processed by worker 1 on machine 1, where the solid line frame is the machine processing time, and the dashed line frame is the workers loading and unloading time. The Gantt chart is the arrangement of workers and machines with the shortest and maximum completion time of 55 min.

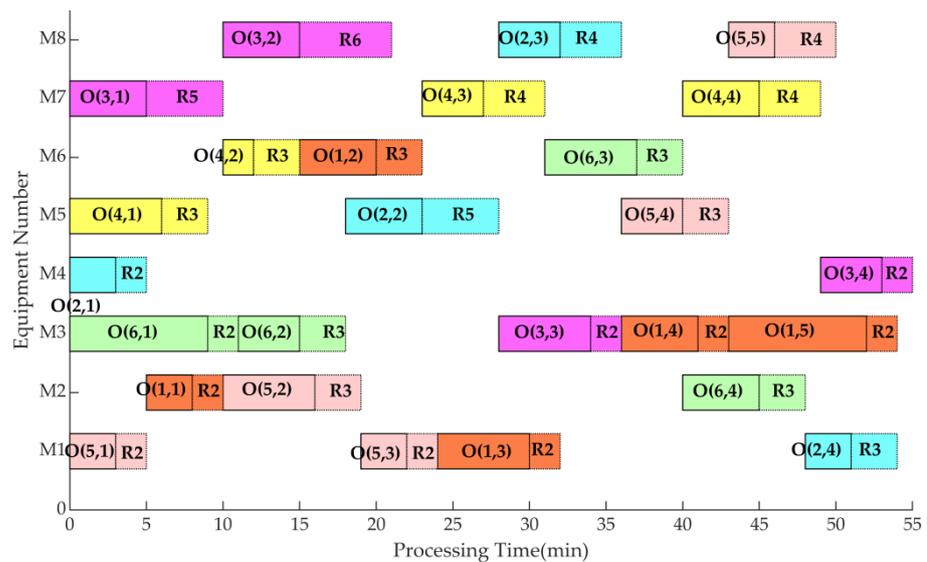


Figure 4. Gantt chart solved by IQGA.

To facilitate comparison, Figures 5–7 are Gantt charts solved by GA, PSO, and QGA algorithms.

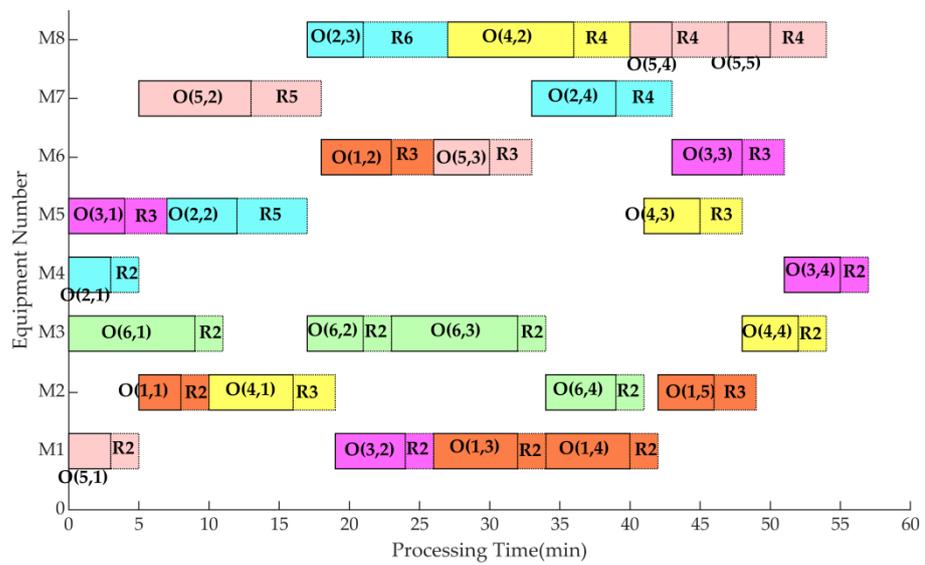


Figure 5. Gantt chart solved by GA.

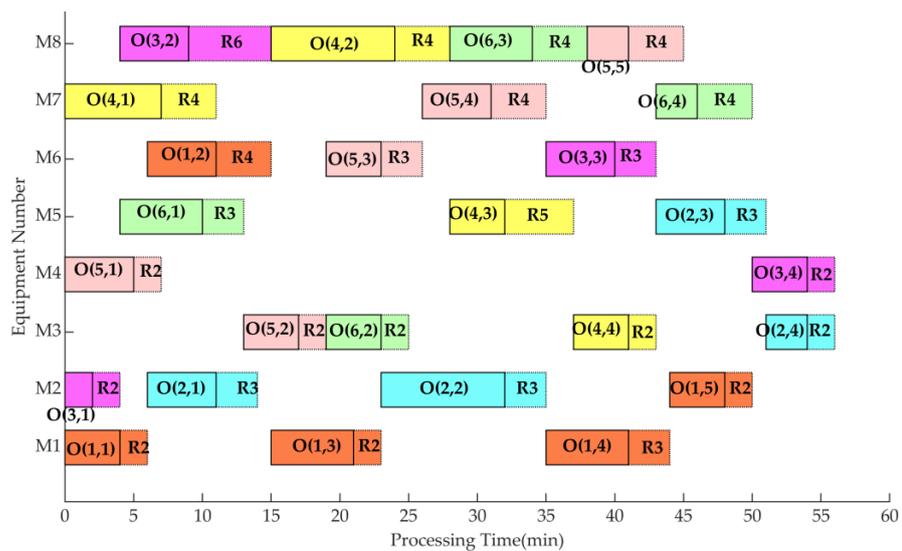


Figure 6. Gantt chart solved by PSO.

Through the comparison of Gantt chart results, it is possible to more clearly compare the calculation results of different algorithms and the difference between the arrangement strategies of the employees and the staff. Among them, the maximum completion time after QGA and PSO solution is 56 min, the maximum completion time after GA solution is 57 min, the Gantt chart after IQGA solution is more closely arranged for machines and employees, and the maximum completion time for IQGA solution is shorter.

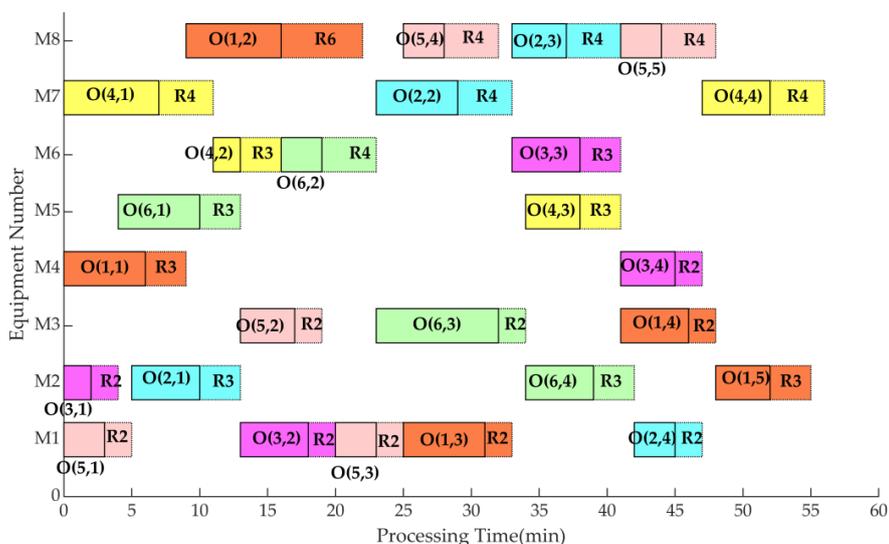


Figure 7. Gantt chart solved by QGA.

4. Discussion

The maximum completion time is an important indicator to evaluate the DRCFJSP problem. In this study, firstly, a DRCFJSP model with different operating time for personnel was established, and then IQGA was used to solve the problem and compared with the algorithms in the literature [33–35] through standard calculation examples; it was found that with the complexity of the matrix, the error between the IQGA result and the standard solution is due to the randomness of the quantum mutation operation, but compared with the other three types of algorithms, the results have higher accuracy. Therefore, the reliability analysis of the standard calculation example proves the superiority of the algorithm in this paper.

In the subsequent example analysis, it can be seen from Table 5 that IQGA can find the optimal solution more stable. In the table, both the degree of optimization of the optimal solution and the frequency of finding the optimal solution ten times are significantly better than those in the table. For the other two algorithms, it can be seen from Figure 3 that the initial population strategy of the niche makes the initial population of IQGA superior to GA, PSO, and QGA; the adaptive rotation angle and quantum mutation strategy allow IQGA to converge to the optimal value of 55 min faster and better, which proves IQGA can effectively solve the DRCFJSP. Finally, by solving the Gantt chart of the four algorithms of GA, PSO, QGA, and IQGA, it can be clearly seen that the Gantt chart solved by the IQGA algorithm is arranged more closely and the total production time is shorter.

5. Conclusions

In the workshop production process, employees occupy an increasingly important position. In this paper, considering the difference in staff operating time, with the goal of minimizing the maximum completion time, a more accurate DRCFJSP model is established. At the same time, a process-based quantum code is designed for the DRCFJSP problem, and then the niche initialization population, dynamic rotation angle, and quantum mutation strategy are used to improve the quantum genetic algorithm to improve the efficiency of the algorithm and avoid premature convergence of the algorithm. Finally, a calculation example is used to analyze.

The Kacem standard calculation example shows that the algorithm used in this article is numerically closer to the standard solution than the algorithm in the literature [33–35]. The four algorithms in Kacem01 can all find the optimal solution; in Kacem02, IQGA optimizes 17.6% and 6.7% of the time compared to the algorithms in literature [33,34], respectively; in Kacem03, IQGA optimizes 7.7% and 7.7% in comparison with the algo-

rithms in literature [34,35]; in Kacem04, IQGA optimizes 12.5% of the time compared to the algorithm in [33]; and in Kacem05, IQGA optimizes 14.3% of the time compared to the algorithm in [35]. The Kacem standard calculation example shows that the IQGA has good accuracy. In the reliability analysis of the factory workshop processing process example, IQGA, GA, PSO, and QGA are run ten times, respectively, and the minimum maximum completion times obtained are: 55 min, 57 min, 56 min, and 56 min. From the average of the calculation results of the four types of algorithms, the average maximum completion times of GA, PSO, QGA, and IQGA are 57.8 min, 56.6 min, 56.3 min, and 55.3 min, respectively, which can also prove the superiority of the algorithm in this paper. It should be noted that although the average maximum completion times of PSO and QGA are close, the variance of the results of the PSO algorithm is large, and it has a significantly worse robustness than QGA. The above analysis shows that the improved quantum genetic algorithm in this paper has good convergence and will not fall into premature. The findings of this research help to reduce the maximum completion time of the workshop production model taking into account differences in personnel, thereby improving the processing efficiency of the entire processing system.

However, the above research still has some shortcomings. In this paper, the goal of the DRCFJSP model is only the maximum completion time, and it does not consider the real-time update of factory production, that is, it does not consider the dynamic demand. Therefore, the model proposed in this article has certain limitations.

Based on the above problems, the DRCFJSP problem under dynamic data will be studied in the future, and multiple variables will be integrated into the solution model to improve the applicability of the positioning method. At the same time, in terms of algorithm performance, the algorithm performance can be improved by optimizing the algorithm call function and running time.

Author Contributions: Conceptualization, Z.L. and S.J.; methodology, S.Z.; software, S.B.; validation, H.D., T.H. and S.Z.; formal analysis, S.Z.; investigation, S.J.; resources, Z.L.; data curation, S.B.; writing—original draft preparation, S.J.; writing—review and editing, Z.L. and S.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Ethical review and approval were waived for this study, due to this research does not involve any aspects related to life sciences and biology.

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

Acknowledgments: The authors would like to acknowledge the support of the Scientific Research Program funded by the Shaanxi Provincial Education Department (no.17JK0321), Xi'an Key Laboratory of Modern Intelligent Textile Equipment (no.2019220614SYS021CG043). and National Key Research and Development Program of China (no. 2019YFB1707205). Thanks to Xi'an Typical Industries Co., Ltd. for providing data support.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Liu, C.; Chang, C.; Tseng, C. Actor-Critic Deep Reinforcement Learning for Solving Job Shop Scheduling Problems. *IEEE Access* **2020**, *8*, 71752–71762. [[CrossRef](#)]
2. Qamhan, A.A.; Ahmed, A.; Al-Harkan, I.M.; Badwelan, A.; Al-Samhan, A.M.; Hidri, L. An Exact Method and Ant Colony Optimization for Single Machine Scheduling Problem with Time Window Periodic Maintenance. *IEEE Access* **2020**, *8*, 44836–44845. [[CrossRef](#)]
3. Berti, N.; Finco, S.; Battaia, O.; Delorme, X. Ageing workforce effects in Dual-Resource Constrained Job-Shop Scheduling. *Int. J. Prod. Econ.* **2021**, *237*, 108151. [[CrossRef](#)]
4. Kumar, P.; Gupta, S.; Gunda, Y. Estimation of Human Error Rate in Underground Coal Mines through Retrospective Analysis of Mining Accident Reports and Some Error Reduction Strategies. *Saf. Sci.* **2020**, *123*, 104555. [[CrossRef](#)]
5. Meng, L.; Zhang, C.; Ren, Y.; Zhang, B.; Lv, C. Mixed-integer Linear Programming and Constraint Programming Formulations for Solving Distributed Flexible Job Shop Scheduling Problem. *Comput. Ind. Eng.* **2020**, *142*, 106347. [[CrossRef](#)]

6. Wang, Y.; Cen, H.; Yang, O. Optimal Configuration for Workshop Manufacturing System under Dual Resource Constraints. *Int. J. Simul. Model.* **2018**, *17*, 180–189. [[CrossRef](#)]
7. Li, J.; Huang, Y.; Niu, X. A Branch Population Genetic Algorithm for Dual-Resource Constrained Job Shop Scheduling Problem. *Comput. Ind. Eng.* **2016**, *102*, 113–131. [[CrossRef](#)]
8. Zhong, Q.; Yang, H.; Tang, T. Optimization Algorithm Simulation for Dual-Resource Constrained Job-Shop Scheduling. *Int. J. Simul. Model.* **2018**, *17*, 147–158. [[CrossRef](#)]
9. 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](#)]
10. Wu, R.; Li, Y.; Guo, S.; Xu, W. Solving the Dual-Resource Constrained Flexible Job Shop Scheduling Problem with Learning Effect by a Hybrid Genetic Algorithm. *Adv. Mech. Eng.* **2018**, *10*, 1–14. [[CrossRef](#)]
11. Zhang, J.; Jie, J.; Wang, W.; Xu, X. A Hybrid Particle Swarm Optimisation for Multi-Objective Flexible Job-Shop Scheduling Problem with Dual-Resources Constrained. *Int. J. Comput. Sci. Math.* **2017**, *8*, 526–532. [[CrossRef](#)]
12. Gong, X.; Deng, Q.; Gong, G.; Wei, L.; Qing, R. A Memetic Algorithm for Multi-Objective Flexible Job-Shop Problem with Worker Flexibility. *Int. J. Prod. Res.* **2018**, *56*, 2506–2522. [[CrossRef](#)]
13. Geng, K.; Ye, C.; Liu, L. Research on Multi-Objective Hybrid Flow Shop Scheduling Problem with Dual Resource Constraints Using Improved Memetic Algorithm. *IEEE Access* **2020**, *8*, 104527–104542. [[CrossRef](#)]
14. Zheng, X.; Wang, L. A Knowledge-Guided Fruit Fly Optimization Algorithm for Dual Resource Constrained Flexible Job-Shop Scheduling Problem. *Int. J. Prod. Res.* **2016**, *54*, 5554–5566. [[CrossRef](#)]
15. Yang, G.; Chung, B.; Lee, S. Limited Search Space-Based Algorithm for Dual Resource Constrained Scheduling Problem with Multilevel Product Structure. *Appl. Sci.* **2019**, *9*, 4005. [[CrossRef](#)]
16. Li, X.; Yi, L. Approach of Solving Dual Resource Constrained Multi-Objective Flexible Job Shop Scheduling Problem Based on MOEA/D. *Int. J. Online Eng.* **2018**, *14*, 75–89.
17. Andrade-Pineda, J.; Canca, D.; Gonzalez-R, P.; Calle, M. Scheduling a Dual-Resource Flexible Job Shop with Makespan and Due Date-Related Criteria. *Ann. Oper. Res.* **2020**, *291*, 5–35. [[CrossRef](#)]
18. Yazdani, M.; Zandieh, M.; Tavakkoli-Moghaddam, R. A hybrid meta-heuristic algorithm for dual resource constrained flexible job shop scheduling problem. *Ind. Manag. Stud.* **2014**, *12*, 43–74.
19. Yazdani, M.; Zandieh, M.; Tavakkoli-Moghaddam, R.; Jolai, F. Two meta-heuristic algorithms for the dual-resource constrained flexible job-shop scheduling problem. *Sci. Iran.* **2015**, *22*, 1242–1257.
20. Defersha, F.; Obimuyiwa, D.; Yimer, A. Multiple-Trial/Best-Move Simulated Annealing for Flexible Job Shop Scheduling with Scarce Setup-Operators. In Proceedings of the 2020 7th International Conference on Soft Computing & Machine Intelligence (ISCM), Stockholm, Sweden, 14–15 November 2020; pp. 61–67.
21. Chatzikonstantinou, I.; Giakoumis, D.; Tzovaras, D. A new shopfloor orchestration approach for collaborative human-robot device disassembly. In Proceedings of the 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Leicester, UK, 19–23 August 2019; pp. 225–230.
22. Gong, G.; Chiong, R.; Deng, Q.; Gong, X. A Hybrid Artificial Bee Colony Algorithm for Flexible Job Shop Scheduling with Worker Flexibility. *Int. J. Prod. Res.* **2020**, *58*, 4406–4420. [[CrossRef](#)]
23. Li, H.; Zhu, H. Modified Water Wave Optimization for Energy-Conscious Dual-Resource Constrained Flexible Job Shop Scheduling. *Int. J. Perform. Eng.* **2020**, *16*, 916.
24. Xu, H.; Bao, Z.; Zhang, T. Solving Dual Flexible Job-Shop Scheduling Problem Using a Bat Algorithm. *Adv. Prod. Eng. Manag.* **2017**, *12*, 5–16. [[CrossRef](#)]
25. Narayanan, A.; Moore, M. Quantum-Inspired Genetic Algorithms. In Proceedings of the IEEE International Conference on Evolutionary Computation, Nagoya, Japan, 20–22 May 1996; pp. 61–66.
26. Gao, L.; Liu, R.; Wang, F.; Wu, W.; Bai, B.; Yang, S.; Yao, L. An Advanced Quantum Optimization Algorithm for Robot Path Planning. *J. Circuits Syst. Comput.* **2020**, *29*, 4–13. [[CrossRef](#)]
27. Sabahi, F. Quantum Genetic LMI-based H_∞ Control with Time Delay. *Int. J. Ind. Electron. Control Optim.* **2020**, *3*, 1–10.
28. Tian, Y.; Hu, W.; Du, B.; Hu, S.; Nie, C.; Zhang, C. IQGA: A Route Selection Method Based on Quantum Genetic Algorithm-Toward Urban Traffic Management under Big Data Environment. *World Wide Web* **2019**, *22*, 2129–2151. [[CrossRef](#)]
29. Sun, J.; Xu, L. Cloud-Based Adaptive Quantum Genetic Algorithm for Solving Flexible Job Shop Scheduling Problem. In Proceedings of the 2019 IEEE 7th International Conference on Computer Science and Network Technology, Dalian, China, 19–20 October 2019; pp. 1–5.
30. Li, H.; Zou, P.; Huang, Z.; Zeng, C.; Liu, X. Multimodal optimization using whale optimization algorithm enhanced with local search and niching technique. *Math. Biosci. Eng. MBE* **2019**, *17*, 1–27. [[CrossRef](#)]
31. Ning, T.; Guo, C.; Chen, R.; Jin, H. A Novel Hybrid Method for Solving Flexible Job-Shop Scheduling Problem. *Open Cybern. Syst. J.* **2016**, *10*, 13–19. [[CrossRef](#)]
32. Kacem, I.; Hammadi, S.; Borne, P. Approach by localization and Multi-objective Evolutionary Optimization for Flexible Job-Shop Scheduling Problems. *IEEE Trans. Syst. Man Cybern. Part C* **2002**, *32*, 1–13. [[CrossRef](#)]
33. Nouiri, M.; Bekrar, A.; Jemai, A.; Niar, S.; Ammari, A. An Effective and Distributed Particle Swarm Optimization Algorithm for Flexible Job-Shop Scheduling Problem. *J. Intell. Manuf.* **2018**, *29*, 603–615. [[CrossRef](#)]

-
34. Ziaee, M. A Heuristic Algorithm for Solving Flexible Job Shop Scheduling Problem. *Int. J. Adv. Manuf. Technol.* **2014**, *71*, 519–528. [[CrossRef](#)]
 35. Luan, F.; Cai, Z.; Wu, S.; Ma, J.; Li, F.; Yang, J. Improved Whale Optimization Algorithm of Scheduling Problem for Low Carbon Workshop. *Mech. Sci. Technol. Aerosp. Eng.* **2020**, *39*, 721–728.
 36. Cui, Q.; Wu, X.; Yu, J. Improved Genetic Algorithm Variable Neighborhood Search for Solving Hybrid Flow Shop Scheduling Problem. *Comput. Integr. Manuf. Syst.* **2017**, *23*, 1917–1927.