Approach of Solving Dual Resource Constrained Multi-Objective Flexible Job Shop Scheduling Problem Based on MOEA/D

Li Xixing
Hubei University of Technology, Wuhan,China

Liu Yi(✉)
Wuhan Donghu University, Wuhan, China
44744518@qq.com

Abstract—With considering the scheduling objectives such as makespan, machine workload and product cost, a dual resource constrained flexible job shop scheduling problem is described. To solve this problem, a multi-objective evolutionary algorithm based on decomposition (MOEA/D) was proposed to simplify the solving process, and an improved differential evolution algorithm was introduced for evolving operation. A special encoding scheme was designed for the problem characteristics, the initial population was generated by the combination of random generation and strategy selection, and an improved crossover operator was applied to achieve differential evolution operations. At last, actual test instances of flexible job shop scheduling problem were tested to verify the efficiency of the proposed algorithm, and the results show that it is very effective.

Keywords—RFID; agricultural automation; sugarcane; growth monitoring

1 Introduction

With the rapid development of China's manufacturing industry and increasingly fierce market competition, manufacturing companies must be able to complete production tasks with high timeliness, high quality and low cost if they want to take advantage of the competition. Meanwhile, with the enhancement of dynamic and uncertain market environment, most products have many varieties, small batches, and short production cycles. To meet the market demand, manufacturing industry needs to develop in the direction of agility and flexibility. These all place higher requirements on the production company's production scheduling optimization theory and program. The Dual Resource Constrained Flexible Job Shop Scheduling Problem (DRCFJSP) is an extension of the Flexible Job Shop Scheduling Problem (FJSP), which considers the process and machine capability constraints in the FJSP. It increases the constraints of the worker's ability on the basis of the FJSP's consideration of process and machine capacity constraints, enhances the flexibility in the scheduling process, is conducive

to the rational use of resources, and is more in line with the actual conditions of work-shop production; so many scholars at home and abroad have studied this problem.

Elmaraghy et.al[1] studied the use of a genetic algorithm (GA)-based scheduling solution to solve the DRCFJSP, and constructed an experimental method for determining the employee level, machine, and worker configuration. Based on this, it is proposed that the six distribution rules and GA have been tested on single-resource-constrained shop scheduling and double-resource-constrained shop scheduling problems. The results show that the proposed method and model can effectively solve the DRCFJSP. Li et.al[2] designed a hybrid algorithm combining ant colony algorithm and simulated annealing algorithm for DRCFJSPs with different workers' capabilities. An adaptive control mechanism based on ant path flow was also proposed to improve global search capabilities. Cao et.al[3] proposed an immune genetic algorithm to solve DRCFJSPs. This algorithm can effectively avoid premature problems under high selection pressure and improve the ability to search for optimal solutions, while accelerating the speed of convergence. Simulation experiments show that the algorithm can effectively obtain the optimal solution of DRCFJSP. Liang et.al[4] constructed a DRCFJSP model with the goal of minimizing the maximum completion time using petri nets, and proposed a hybrid algorithm based on GA and tabu search (Tabu Serch, TS) to solve the model. Lei et.al [5] proposed to use variable neighborhood search algorithm to solve DRCFJSP, and designed different variable neighborhood search strategies for two different sub-problems in DRCFJSP. Huang et.al[6] proposed a three-layer dynamic scheduling system based on time window including dynamic perturbation analysis, dynamic perturbation evaluation and dynamic scheduling method to solve DRCFJSP. Zhang et.al [7] proposed a three-layer coding scheme, and based on discrete particle swarm optimization algorithm, simulated annealing algorithm and variable neighborhood search were used as local search strategies to construct a hybrid discrete particle swarm algorithm to solve DRCFJSP. Lei et. al[8] took environmental factors into account in the DRCFJSP, built a scheduling model with the goal of minimizing carbon footprint and maximum completion time, and proposed a dynamic domain search algorithm to solve the problem. Yazdani et.al[9] proposed using the simulated annealing algorithm and the shock optimization two heuristic algorithms combined with the variable neighborhood search strategy to obtain the DRCFJSP solution. The experimental results show that the two proposed hybrid algorithms have relatively good performance. Li et.al[10] considered the two goals of minimizing the maximum completion time and minimizing the processing cost in the DRCJSP. The branch population genetic algorithm was used, and the compression time window strategy was used to improve the global scheduling ability. Zheng et.al[11] proposed to use knowledge-based fruit fly optimization algorithm to solve the DRCFJSP with the goal of minimizing the maximum completion time, and studied and tested the impact of algorithm parameter settings on its performance.

2   Problem description and model establishment

The research of DRCFJSP is that the N workpieces with multiple processes are processed by W workers to operate the M equipments for processing and scheduling optimization. In the DRCFJSP, there is a process order constraint between the different processes of the same workpiece, which can only be processed in accordance with the predetermined process sequence. Meanwhile, each processing equipment can choose different workers to operate, and each worker has different operation ability and operation efficiency. Therefore, the problem to be solved by the DRCF JSP is to obtain a scheduling scheme that optimizes the predetermined performance index by rationally arranging the processing sequence of the work pieces, completing the processing equipment for each process, and operating the workers of the various processing equipment under conditions that satisfy process constraints, equipment processing capacity constraints, and worker operating capacity constraints.

This paper establishes a scheduling model with the goal of minimizing the maximum completion time, minimizing the processing equipment load, and minimizing the production cost. The model is based on the following assumptions:

a) All processing equipment is available at time 0;
b) All workpieces can be processed starting at time 0;
c) There is no process order constraint between the workpieces;
d) One device can only process one process at a time, and one worker can only operate one device at a time;
e) Does not consider interruptions in the processing of the process;
f) Does not consider transport time, preparation time, release time, etc.

2.1   Variable Definition

Labeling

j Workpiece number, $j \in \{1,...,N\}$;

l Process number, $l \in \{1,...,\theta_j\}$;

k Device number, $k \in \{1,...,M\}$;

r Worker number, $r \in \{1,...,W\}$;

Known variables

N Quantity of workpieces；

M Quantity of processing equipment；

W Quantity of operator；

θ_j Quantity of process in workpiece j, $j \in \{1,...N\}$;

p_{jlk} Basic processing time of process l in workpiece j on device k；

h_{rk} Processing efficiency of worker r operates device k；

d_j Deadline for delivery of workpiece j;

e_k Unit processing time cost of device k;

g_{rk} Unit processing time cost when worker r operating equipment k;

α_j Unit time inventory cos t when workpiece j is advanced;

β_j Unit time punishment cos t when workpiece j is delay;

Decision variables

x_{jlrk} If the process l in the workpiece j is processed by the device k

and the worker r, it is 1, otherwise it is 0;

SM_{tk} The state of machine k at time t, idle is 0, being processed is 1;

SW_{tr} The state of worker r at time t, idle is 0, being processed is 1;

s_{jl} The starting time of process l in workpiece j;

f_{jl} The end time of process l in workpiece j;

c_j The completion time of workpiece j;

2.2 Mathematical model

Based on the above problem description and variable definition, the scheduling op-timization mathematical model for DRCF JSP is established and the corresponding constraint conditions are as follows:

$$\min\{f_1, f_2, f_3\} \tag{1}$$

$$f_1 = \max_{j=1}^{N}\{c_j\} \tag{2}$$

$$f_2 = \sum_{k=1}^{M}\sum_{j=1}^{N}\sum_{l=1}^{\theta_j}\sum_{r=1}^{W} p_{jlk} \times h_{rk} \times x_{jlrk} \tag{3}$$

$$f_3 = \sum_{k=1}^{M}\sum_{j=1}^{N}\sum_{l=1}^{\theta_j}\sum_{r=1}^{W} (e_k + g_{rk}) \times p_{jlk} \times h_{rk} \times x_{jlrk}$$
$$+ \sum_{j=1}^{N} (\alpha_j \times \max\{d_j - c_j, 0\} + \beta_j \times \max\{c_j - d_j, 0\}) \tag{4}$$

$s.t.$

$$c_j = s_{j\theta_j} + \sum_{k=1}^{M} \sum_{r=1}^{W} p_{j\theta_j k} \times x_{j\theta_j rk} \tag{5}$$

$$s_{jl} \geq f_{j(l-1)} = s_{j(l-1)} + \sum_{k=1}^{M} \sum_{r=1}^{W} p_{j(l-1)k} \times x_{j(l-1)rk} \quad \text{suppose } s_{j0} = 0, p_{jok} = 0. \tag{6}$$

$$\sum_{k=1}^{M} \sum_{r=1}^{W} x_{jlrk} = 1, \quad \forall j,l \tag{7}$$

$$SM_{s_{jl}} = 0, SW_{s_{jl}} = 0, \forall j,l \tag{8}$$

Equation (1) is the objective function of the schedule optimization. Equation (2) represents the maximum completion time, the maximum completion time is the completion time of the last process of the last workpiece. Equation (3) is the total equipment load, that is, the sum of all processing equipment cumulative processing time. Equation (4) is the production cost, and the production cost is composed of two parts. The first part is the sum of the cost of processing equipment and workers in the production process. The second part is the sum of the lead time inventory cost and tardiness penalty cost after all the parts are completed. Equation (5) shows that the completion time of the workpiece is the starting processing time of the last process of the workpiece plus its actual processing time. The actual processing time is determined by the basic processing time of the selected equipment and the operating efficiency of the selected operator. Equation (6) reflects the process sequence constraint, and the starting processing time of the process must be later than the processing ending time of the preceding process. Equation (7) indicates that all processes must be completed by only one piece of equipment and one worker. Equation (8) indicates that the selected processing equipment and workers must be in an idle state when all processes are started.

3 Multi-Objective Evolutionary Algorithm Based on Decomposition

Multi-Objective Evolutionary Algorithm Based on Decomposition (MOEA/D) is a new multi-objective optimization framework [12]. The core idea is to decompose a multi-objective optimization problem into a series of single-objective optimization subproblems by an aggregate function. It does not directly approach the real Pareto frontier, but aims to optimize these subproblems in parallel and optimize these subproblems simultaneously by using evolutionary algorithms. The reason why this mechanism is effective is because the optimal solution of each subproblem actually corresponds to the Pareto optimal solution of a given multi-objective optimization problem, and finally a set of Pareto optimal solutions can be obtained. Due to the existence of decomposition operations, this method has great advantages in maintain-

ing the dispersion of the solution, and can be globally optimized by analyzing the information of neighboring sub-problems to avoid falling into local optimum. Compared with Nondominated Sorting Genetic Algorithm II (NSGA-II), MOEA/D can obtain more accurate and better Pareto optimal solution set in faster time.

3.1 Weight vector generation and aggregation method

The main feature of MOEA/D is to deal with multi-objective optimization problems in the form of solving a set of single-target sub-problems. Each single-target sub-problem has its own different weight vector. Each weight vector has one and only one individual in the current population, that is, the number of weight vectors and the population size are consistent. The generation of weight vector is an important link in the implementation of MOEA/D, and the result of using a relatively uniform weight vector is better than the non-uniform weight vector [13]. The common method is to evenly take points on plane $f_1 + f_2 + ... + f_m = 1$ or surface $f_1^2 + f_2^2 + ... + f_m^2 = 1$ to generate a uniform weight vector. The specific implementation steps are as follows: First, a positive integer H is defined by the user, and thus a weight vector element value set $\delta = \{0, \frac{1}{H}, \frac{2}{H}, ..., \frac{H}{H}\}$ or $\delta = \{0, \sqrt{\frac{1}{H}}, \sqrt{\frac{2}{H}}, ..., \sqrt{\frac{H}{H}}\}$ can be obtained. For the weight value $\lambda_i \in \delta, i = 1, 2, .., m$ in each weight vector, $\lambda_i \in \delta$, and $\lambda_1 + \lambda_2 + ... + \lambda_m = 1$ or $\lambda_1^2 + \lambda_2^2 + ... + \lambda_m^2 = 1$. The number of weight vectors G and user-defined positive integers H, the number of sub-objective functions m satisfies $G = C_{H+m-1}^{m-1}$ can be obtained from the condition that the weight vector must satisfy the sum of the unequal weights and the weight values is 1. From this we can see that in MOEA/D, the size of the population is not set at random, but is determined during the process of generating the weight vector.

After generating the uniformly distributed weight vectors through the above method, the corresponding aggregation method is needed to transform the multi-objective optimization problem of the solution to a single-objective optimization problem. The choice of polymerization method has a decisive influence on the performance of MOEA/D. The most common method is the Chebyshev polymerization method. This method has the property of being insensitive to the shape of the Pareto front. This method will be used in this paper. The formula for the Chebyshev polymerization method is:

$$\min g^{te}(\vec{x} \mid \lambda, z^*) = \max\{\lambda_i \mid f_i(\vec{x}) - z_i^* \mid\}$$

Where $\overset{1}{x} \in \Omega$, $\overset{1}{x}$ is the solution vector, Ω is solution space, $z^* = (z_1^*,...,z_m^*)^T$ is the reference point, for each $i = 1, 2,..., m$, there are $z_i^* = \min\{f_i(\overset{1}{x}) \mid \overset{1}{x} \in \Omega\}$.

3.2 Algorithm Architecture

Algorithm initial conditions:

A population $\overset{\longrightarrow}{x_1}, \overset{\longrightarrow}{x_2},..., \overset{\longrightarrow}{x_Q}$ of size Q, where $\overset{u}{x_i}$ is the current solution to the current subproblem i;

Algorithm initial conditions:

$FV_1,..., FV_Q$, FV_i is the objective function value of the individual $\overset{u}{x_i}$, i.e. for each $i = 1,...Q$, $FV_i = F(x_i)$

$\overset{1}{z} = (z_1,..., z_m)^T$, z_i is the optimal value of the currently searched target;

The additional population EP used to record all non-dominated solutions in the search process;

Input:

Multi-objective optimization problems;

Algorithm termination conditions, generally the maximum number of iterations;

H: custom positive integer, combined with the number of sub-objective function M can calculate the population size Q;

Uniformly distributed Q weight vectors: $\lambda_1,..., \lambda_Q$;

T: the number of weight vectors in each domain, user-defined;

Output: EP

Step 1: Initialization

Step 1.1: Set EP=$\varnothing$.

Step 1.2: Use the weight generation method to obtain a set of weight vectors, calculate the Euclidean distance between any two weight vectors, and select the nearest T weight vectors for each weight vector as its domain. For $i = 1,..., Q$, let $B(i) = \{i_1,..., i_T\}$, $\lambda_{i_1},..., \lambda_{i_T}$ is the nearest T weight vectors of λ_i.

Step 1.3: Initialize the population and get the population $\overrightarrow{x_1},..., \overrightarrow{x_Q}$.

Step 1.4: Initialize the reference point $\overset{1}{z} = (z_1,..., z_m)^T$, $z_i = \min\{f_i(\overrightarrow{x_1}),..., f_i(\overrightarrow{x_Q})\}$.

Step 2: Update

for i=1,…,Q do

Step 2.1: Evolutionary operations: perform differential evolution operations on subpopulations of individuals $x_{i_1},...,x_{i_Q}$ from B(i) to obtain new individual *y*.

Step 2.2: Update Reference Point:

for each j=1:m

$$if \ z_j < f_j(y), then \ z_j = f_j(y)$$

end for

Step 2.3: Update the domain solution:

for each $j \in B(i)$

$$if \ g^{te}(y|\lambda_j,z) \le g^{te}(x_j|\lambda_j,z), \ then \ x_j = y, FV_j = F(y)$$

end for

Step 2.4: Update the EP

Remove all $F(y)$-dominated solutions in the EP;

If there is no determinant $F(y)$ in the EP, add $F(y)$ in EP;

end for

Step 3: Determine the termination condition: meet the set termination condition, end and output EP; otherwise, go to step 2.

3.3 coding scheme

Compared with FJSP, DRCFJSP increases the choice of workers based on the consideration of process sequence constraints and equipment selection. Therefore, the two-layer coding of procedures and equipment commonly used in FJSP cannot be applied. This paper proposes a three-layer coding method based on double-level coding in FJSP. To facilitate the description of the coding scheme proposed in this paper, a 3*3*2 dual resource constrained flexible job shop scheduling problem example is given below. The process-equipment processing time matrix is shown in Table 1, and the equipment-worker processing efficiency matrix is shown in Table 2.

Table 1. Process-equipment processing time matrix

	O_{11}	O_{12}	O_{13}	O_{21}	O_{22}	O_{31}	O_{32}	O_{33}
M_1	3	-	4	9	-	7	-	7
M_2	7	5	3	-	4	7	6	-
M_3	6	4	-	7	-	8	8	6

Table 2. Equipment-Worker Process Efficiency Matrix

	M_1	M_2	M_3
W_1	0.8	0.9	-
W_2	-	0.85	0.9

Process-based coding. The first layer is a process-based code used to determine the sequential processing sequence. In this encoding method, the length of the encod-ed chromosome $L = \sum_{j=1}^{N} \theta_j$ is assigned to the same number for all steps of the same workpiece, and the sequential processing order of the steps is determined according to the different positions of the numbers. All processes of the same workpiece are as-signed the same number, and the sequential processing sequence of the process is determined according to the different positions of the number. Fig. 1(a) is an example of a process chromosome, in which two successively appearing 2 represent two dif-ferent processes of the workpiece 2. The first 2 represents the first process of the workpiece 2. The second 2 represents the second process of the workpiece 2. The processes of the workpiece 1 and the workpiece 3 are similar.

O_{21}	O_{31}	O_{11}	O_{32}	O_{22}	O_{12}	O_{33}	O_{13}
2	3	1	3	2	1	3	1

(a) Operation sequence vector

M_1	M_2	M_2	M_3	M_2	M_3	M_1	M_1
1	2	2	3	2	3	1	1

(b) Machine assignment vector

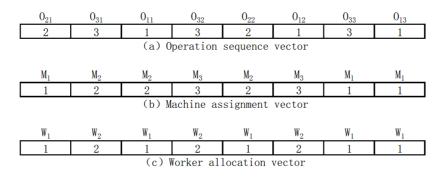| W_1	W_2	W_1	W_2	W_1	W_2	W_1	W_1
1	2	1	2	1	2	1	1

(c) Worker allocation vector

Fig. 1. coding example

Device-based coding. Device-based coding is used to determine the equipment ar-rangement for each process. Figure 1(b) shows a device-encoded chromosome in a regulation. The length of the chromosome is the same as that of the process coded chromosome, and the code of the corresponding position in the process code indicates the processing equipment number of the process. For example, if the first step of the process chromosome O_{21} represents the second workpiece, the code at the position corresponding to the coded chromosome of the device is 1, that is, the processing device with the serial number 1 is selected in the process, and the corresponding basic processing time is 9. In this way, the number of the processing equipment for all pro-cesses can be determined in turn, and the processing equipment selection must com-ply with the processing equipment constraints.

Worker-based coding. Worker-based coding is similar to device-based coding. As shown in Figure 1(c), the number of the corresponding position in the encoded chro-mosome corresponding to the device encoding chromosome is the selected worker number. As shown in Figure 1(b), the first worker with the number 1 corresponds to a worker number of 1. It should be noted that the worker code should comply with the constraint.

Population initialization. Initialization of the population has a great influence on the quality of the solution. The commonly used methods are all randomly generated,

but the quality of the obtained population is not stable. This paper uses the combination of random generation and strategy selection to generate the initial population.

Firstly, we generate chromosomes for the sorting of processes. For process-sequencing chromosomes, we use the mixed strategy proposed by Pezzella et.al[14] and consist of the following three rules:

a) Randomly select a workpiece (Random);
b) Give priority to select the workpiece with the longest remaining processing time (MWR);
c) Prioritize the workpiece with the most remaining operations (MOR).

The number of chromosomes for process sequencing generated by the above three strategies is 20%, 40%, and 40%, respectively.

Secondly, for device allocation and worker distribution of chromosomes, 50% of the chromosomes are generated randomly within the constraint set, and the other 50% are generated using the initial population positioning method proposed by Kacem et.al[15]. Take the example of generating device allocation chromosomes. The specific operation step is to sort the processes in the chromosomes in sequence through the traversal processes, select the device corresponding to the shortest basic processing time in the set of available devices, and adds the current basic processing time to all the basic processing times of the same device (Note that only the available equipment is operated). When the subsequent equipment is selected, the updated basic processing time is referenced until the processing equipment is selected for each process to generate a distribution chromosome. Workers assign chromosomes can be generated using the same method.

Evolutionary operations. Regarding the evolutionary operation in MOEA/D, this paper adopts Differential Evolution (DE) algorithm[16]. As a new type of intelligent algorithm, it has the characteristics of simple principle, less controlled parameters, good robustness and easy implementation, and it has been successfully applied to the workshop scheduling problem[17-21], and got better results. Traditional DE mutations and crossover operators are more suitable for continuous optimization problems and do not apply to discrete DRCFJSPs. This article has improved them to fit DRFCJSPs.

Individual Crossover Operators. Crossover operator $y = G(x_1, x_2)$ means that individuals x_1, x_2 are crossed to get new individual y. In this process, different crossover strategies are adopted for different chromosomes in individuals. The process sequencing chromosomes adopt the POX crossover method[22], and the device-allocated chromosomes and worker-distribution chromosomes adopt the MPX crossover method[23].

Mutation Operation. The difference mutation operation in DE is replaced by an individual crossover operator. First, a subpopulation of individuals $x_{i_1}, \ldots, x_{i_T}$ in neighborhood set $B(i)$ of an individual i is obtained, and three individuals $x_{r_1}^t, x_{r_2}^t, x_{r_3}^t$ are randomly selected, and generate variant individual V_i^{t+1} by the following mutation.

$$V_i^{t+1} = F \otimes G(F \otimes G(x_{r_2}^t, x_{r_3}^t), x_{r_1}^t) \tag{9}$$

Where $r_1, r_2, r_3 \in B(i)$, and $r_1 \neq r_2 \neq r_3 \neq i$, $F \in (0,1)$.
In formula (9)

$$F \otimes G(x_{r_2}^t, x_{r_3}^t) = \begin{cases} G(x_{r_2}^t, x_{r_3}^t), & rand < F; \\ x_{r_2}^t, & Otherwise \end{cases} \tag{10}$$

Among them, $rand$ is the random number on $[0,1]$, it can be seen that the mutation operation is achieved by two individual cross operations with a certain probability.

Cross Operations. The crossover operation is also implemented with individual crossover operators. For individual V_i^{t+1} and individual x_i obtained from mutation operations, the following crossover operations are performed to obtain experimental individual U_i^{t+1}.

$$U_i^{t+1} = Cr \otimes G(V_i^{t+1}, x_i^t) = \begin{cases} G(V_i^{t+1}, x_i^t), & rand < Cr \\ V_i^{t+1}, & Otherwise \end{cases} \tag{11}$$

Among them $Cr \in (0,1)$, the experimental individual U_i^{t+1} obtained as a new individual of the evolutionary operation carries out the following updating reference points and updating the neighborhood solution.

4 Case validation and results analysis

4.1 Test examples and algorithm parameter settings

Since there is no standard test case for DRCFJSP at present, this paper selects 18 commonly used test cases for flexible job shop scheduling problems {MK2, MK4, MK6, Mk8, MK10}[24], {DP2, DP5, DP10, DP15, DP18 }[25] and {LA5, LA10, LA15, LA20, LA25, LA30, LA35, LA40}[26]. On the basis of it, information on worker restraints, equipment, and worker hourly fees was added to expand it into test cases for DRCFJSPs. The value of the random variable used in the extended test case is shown in Table 3. Using the data in this table, five replicates are generated for each case as a group. There are a total of 18 groups of 90 test cases.

Table 3. Random Variables

Variables	Values
Number of workersW	$\lceil M*0.6 \rceil$
Average number of workers per unit [*]	$\lceil W*0.5 \rceil$
Equipment working hourse$_k$	U[80,120]
Worker hour feeg$_{rk}$	U[30,50]
Advance unit time costα$_i$	U[5,10]
Tard time per unit penalty feeβ	U[8,20]
Deadline	U[10,50]

[*] The number of workers that can be operated on a single device is at least 1

Set the parameter H=13 in MOEA/D (population size Q=105), T=20, Gen=100, F=0.5, Cr=0.5. For better comparison, the population size of the NSGA-II algorithm is set to 105 for the comparison, the maximum number of iterations is set to 100, the crossover probability is set to 0.8, and the mutation probability is set to 0.1.

4.2 Calculation Results and Analysis

To avoid the randomness of the calculation results, five calculations are performed for each test case. Finally, the summary of the minimum value, average value, and average value of non-dominated solutions in the sub-objective function in each group of cases is summarized in Table 4.

Table 4. Test results

Instance	MOEA/D							NSGA-II						
	min f$_1$	avg f$_1$	min f$_2$	avg f$_2$	min f$_3$	avg f$_3$	num	min f$_1$	avg f$_1$	min f$_2$	avg f$_2$	min f$_3$	avg f$_3$	num
MK2	3330.40	3641.82	10652.8	10782.8	1924093	1967577	20	3477.97	3698.24	10536	10652.6	1955329	1978015	8
MK4	3118.30	3496.20	10861.3	10939.3	1945546	1973954	22	3170.97	3604.05	10882.9	10942.2	1955147	2000192	10
MK6	3475.23	3802.01	16050.9	16101.1	2855293	2901838	6	3630.75	3823.75	16026.8	16121.2	2885133	2930486	7
MK8	6256.79	6843.47	21002.2	21157.7	4494180	4613173	19	6683.23	7107.2	20869.9	21212.4	4610363	4696595	14
MK10	3731.27	3979.09	20275.5	20609	3866345	3938387	18	3801.2	4235.22	20467.1	20633.1	3939907	3995006	18
DP2	40.10	42.95	134.379	141.303	19968.2	21553.34	11	44.602	47.9671	138.394	145.735	20642	21030.2	10
DP5	101.93	115.14	308.608	323.115	54714.4	57761.44	29	103.699	134.856	308.336	328.223	58037.1	61830.9	33
DP10	117.81	123.39	324	327.415	60638.1	60888	6	126.124	130.377	345.299	353.757	63787.5	64844.7	4
DP15	592.42	630.23	2427.06	2474.65	466370	471870.9	12	601.57	629.644	2413.54	2457.15	466877	471844	6
DP18	435.57	460.02	1866.8	1880.71	366199	370804.5	8	438.578	467.599	1925.1	1977.23	379562	384868	11
LA5	556.17	725.75	2115.6	2154.69	339298	349328.6	25	634.532	712.393	2126.14	2155.54	354294	361797	11
LA10	931.64	1049.20	3609.55	3679.75	629393	654831.8	17	1004.07	1166.75	3611.69	3670.56	658916	668337	14
LA15	1326.12	1476.90	4990.01	5083.62	999088	1025786	10	1450.15	1624.51	4933.43	5139.57	1042038	1086812	17
LA20	1247.74	1490.44	5174.89	5248.84	900273	922293.3	15	1332.96	1564.47	5195.64	5287.27	927971	944416	11
LA25	1884.89	2215.91	7094.05	7171.38	1331036	1381009	13	1972.76	2176.02	7171.94	7291.48	1366495	1394705	9
LA30	2858.04	3199.52	10418.2	10445.8	2133143	2207951	9	3224.83	3733.93	10389	10496.5	2258419	2386551	10
LA35	3801.63	4023.86	15878.8	16008.4	3536337	3571671	8	3958.95	4353.04	15823.7	15925.1	3588919	3685172	9
LA40	2881.26	3150.42	10922.2	11051.9	2034389	2089985	16	2948.1	3273.65	10929.2	11015.6	2073558	2113911	12

Note: The bold value means the best result.

Analyze the data in this table, in the comparison of the minimum value of the sub-objective function value and the average value of the sub-objective function value, the MOEA/D algorithm gets smaller values. Therefore, the MOEA/D algorithm proposed in this paper is superior to the NSGA-II algorithm in the quality of the solution obtained. In addition, comparing the number of non-dominated solutions obtained in Table 4, the number of non-dominated solutions obtained by the MOEA/D algorithm

is large. Therefore, the proposed algorithm also has certain advantages in the distribution of solutions.

5 Conclusion

This paper proposes a scheduling model for DRCFJSPs that minimizes the maximum completion time, minimizes the processing equipment load, and minimizes the production cost. It uses the MOEA/D algorithm framework integrated differential evolution algorithm to solve it. The Chebyshev polymerization method is used in MOEA/D to decompose this multi-objective problem into a single-objective problem to simplify the solution process. In the evolutionary operation, a three-layer coding method based on process ordering, equipment allocation, and worker assignment phase is designed for DRCF JSPs, and a combination of random generation and strategy selection is used to generate initial populations. A new crossover operator was designed and used to improve the mutation and crossover operations in the differential evolution algorithm to make it suitable for solving DRCFJSPs. Finally, the commonly used test cases of flexible job shop problems are extended to test the proposed algorithm, and the NSGA-II algorithm is used as a comparison. The results show that the proposed algorithm has advantages in the quality of solutions and the diversity of solutions.

6 Acknowledgment

7 References

[1] Elmaraghy H A, Patel V, Abdallah I B. Scheduling of manufacturing systems under dual-resource constraints using genetic algorithms. Journal of Manufacturing Systems, 2000, 19(3): 186–201. https://doi.org/10.1016/S0278-6125(00)80011-4

[2] Li S J, Huang S, et al. Adaptive Hybrid ant colony optimization for solving Dual Resource Constrained Job Shop Scheduling Problem[J]. Journal of Software, 2011, 6(4):584-594. https://doi.org/10.4304/jsw.6.4.584-594

[3] Cao X Z, Yang Z H. An Improved Genetic Algorithm for Dual-Resource Constrained Flexible Job Shop Scheduling. Proceedings of the 2011 Fourth International Conference on Intelligent Computation Technology and Automation, 2011(01): 42-45.

[4] Liang D, Liu S, Tao Z. Based on Petri Nets and Hybrid Genetic-Tabu Search Approach to Scheduling Optimization for Dual-Resource Constrained Job Shop[J]. EMEIT-12, 2012.

[5] Lei D, Guo X. Variable neighborhood search for dual-resource constrained flexible job shop scheduling[J]. International Journal of Production Research, 2014, 52(9):2519-2529. https://doi.org/10.1080/00207543.2013.849822

[6] Huang Y, Yan X T, Li J Y. Dual Resource Constrained Job Shop Dynamic Scheduling with Local Time Window[J]. Advanced Materials Research, 2014, 933:708-713. https://doi.org/10.4028/www.scientific.net/AMR.933.708

[7] Zhang J, Wang W, Xu X. A hybrid discrete particle swarm optimization for dual-resource constrained job shop scheduling with resource flexibility[J]. Journal of Intelligent Manufacturing, 2015:1-12.

[8] Lei D, Guo X. An effective neighborhood search for scheduling in dual-resource constrained interval job shop with environmental objective[J]. International Journal of Production Economics, 2015, 159(1):296-303. https://doi.org/10.1016/j.ijpe.2014.07.026

[9] Yazdani M, Zandieh M, Tavakkoli-Moghaddam R, et al. Two meta-heuristic algorithms for the dual-resource constrained flexible job-shop scheduling problem[J]. Scientia Iranica, 2015, 22(3):1242-1257.

[10] Li J, Huang Y, Niu X. A branch population genetic algorithm for dual-resource constrained job shop scheduling problem[J]. Computers & Industrial Engineering, 2016, 102:113-131. https://doi.org/10.1016/j.cie.2016.10.012

[11] Zheng X, Wang L. A knowledge-guided fruit fly optimization algorithm for dual resource constrained flexible job-shop scheduling problem[J]. International Journal of Production Research, 2016(18):1-13. https://doi.org/10.1080/00207543.2016.1170226

[12] Zhang Q, Li H. MOEA/D: A Multi-objective Evolutionary Algorithm Based on Decomposition[J]. IEEE Transactions on Evolutionary Computation, 2007, 11(6):712-731. https://doi.org/10.1109/TEVC.2007.892759

[13] Ishibuchi H, Sakane Y, Tsukamoto N, et al. Adaptation of Scalarizing Functions in MOEA/D: An Adaptive Scalarizing Function-Based Multi-objective Evolutionary Algorithm[M].Evolutionary Multi-Criterion Optimization. Springer Berlin Heidelberg, 2009:438-452.

[14] Pezzella F, Morganti G, Ciaschetti G. A genetic algorithm for the Flexible Job-shop Scheduling Problem[J]. Computers & Operations Research, 2008, 35(10):3202-3212. https://doi.org/10.1016/j.cor.2007.02.014

[15] Kacem I, Hammadi S, Borne P. Approach by localization and multi-objective evolutionary optimization for flexible job-shop scheduling problems[J]. IEEE Transactions on Systems Man & Cybernetics Part C Applications & Reviews, 2002, 32(1):1-13. https://doi.org/10.1109/TSMCC.2002.1009117

[16] Storn R, Price K. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces [J]. Journal of Global Optimization, 1997, 11(4):341-359. https://doi.org/10.1023/A:1008202821328

[17] Wang W L, Fan L X, Xu X L, et al. Multi-objective differential evolution algorithm for flexible Job-Shop batch scheduling problem[J]. Computer Integrated Manufacturing Systems, 2013, 19(10):2481-2492.

[18] Wang W L, Wang L, Wang H Y, et al. Dynamic Job Shop scheduling based on hybrid differential evolution algorithm[J]. Computer Integrated Manufacturing Systems, 2012, 18(3):531-539.

[19] Wang H Y, Zhao Y W, Zhang J L, et al. Batch optimized scheduling of intermingling flow-shop based on hybrid differential evolution algorithm[J]. Computer Integrated Manufacturing Systems, 2013, 19(7):1613-1625.

[20] Wang W L, Xin li X U, Shi L N, et al. An improved differential evolution algorithm for uncertain process shop scheduling problem[J]. Computer Integrated Manufacturing Systems, 2011, 17(3):630-637..

[21] Wang S X, Zhang C Y, Liu Q, et al. Flexible job shop dynamic scheduling under different reschedule periods[J]. Computer Integrated Manufacturing Systems, 2014, 20(10):2470-2478.

[22] Zhang C Y, Rao Y Q, Pei-Gen L I, et al. An improved genetic algorithm for Job-Shop scheduling[J]. Computer Integrated Manufacturing Systems, 2004, 10(8):966-970.

[23] Zhang C. Improved NSGA-II for the Multi-objective Flexible Job-shop Scheduling Problem[J]. Journal of Mechanical Engineering, 2010, 46(11). https://doi.org/10.3901/JME.2010.11.156

[24] Brandimarte P. Routing and scheduling in a flexible job shop by tabu search[J]. Annals of Operations Research, 1993, 41(3):157-183. https://doi.org/10.1007/BF02023073

[25] Dauzère-Pérès S, Paulli J. An integrated approach for modeling and solving the general multiprocessor job-shop scheduling problem using tabu search[J]. Annals of Operations Research, 1997, 70:281-306. https://doi.org/10.1023/A:1018930406487

[26] Hurink J, Jurisch B, Thole M. Tabu search for the job-shop scheduling problem with multi-purpose machines[J]. OR Spectrum, 1994, 15(4):205-215. https://doi.org/10.1007/BF01719451

8   Authors

Li Xixing (1990-), come from Wuhan, Hubei province. PhD, the main research direction for SCM, IOT.(e-mail: 736793603@ qq.com, Affiliation: Hubei Key Laboratory of Modern Manufacturing and Quality Engineering, School of Mechanical Engineering, Hubei University of Technology, Wuhan, 430068,China).

Liu Yi (1983-), come from Wuhan, Hubei province. PhD, the main research direction for the Intelligent Manufacture.IE (e-mail: 44744518@qq.com, Affiliation: School of Mechanical Engineering, Wuhan Donghu University, Wuhan, 430299, China).