FLEXIBLE DYNAMIC SCHEDULING BASED ON IMMUNE ALGORITHM

Jianjun Yu, Shudong Sun, Jinghui Hao
School of Mechatronic Engineering, Northwestern Polytechnical University, China; Email: npu_yjj@163.com

Abstract: The Job-Shop Scheduling Problem is simplified from the shop production though it is NP-hard, and the actual Scheduling is more complicated than it. The complex flexible manufacturing cell dynamic scheduling (FMCDS) which takes much disturb into account is researched in. The machine may suddenly turn into malfunction during production, and the jobs may go into or be canceled from scheduling for some reason. Each kind of part can be disintegrated into a lot of batch which includes some parts, and batch operation can choice one from some machines. Furthermore, several objects are optimized, which include time, cost and machine use rate. The FMCDS model is set up, and a dynamic evaluation based immune algorithm and rolling-disturb hybrid rescheduling policy are initially brought forward to solve FMCDS. The proposed algorithm, policy and method are applied to Xi'an Aviation Manufacture Corporation, and can fulfill its FMCDS.

Key words: Flexible manufacturing cell dynamic scheduling, Immune algorithm, Temporary solution set, Rolling-disturb hybrid rescheduling policy.

1. INTRODUCTION

Most of the scheduling methods idealize the actual shop condition to reduce the calculation difficulty. However, flexible manufacturing cell dynamic scheduling（FMCDS）is closer to the production practice, thus the research for it is of more theory significance and engineering value, but FMCDS has many characteristics, such as complex modeling, complicated calculation, multi-constraint, multi-objective and dynamic disturb, and belongs to the combinatorial optimization category, and is proved to be the typical NP-hard problem[1,2]. In the paper, the immunity idea is imported into FMCDS, and a novel immune algorithm is put forward to solve the FMCDS.

Please use the following format when citing this chapter:
Yu, Jianjun, Sun, Shudong, Hao, Jinghui, 2006, in International Federation for Information Processing (IFIP), Volume 207, Knowledge Enterprise: Intelligent Strategies In Product Design, Manufacturing, and Management, eds. K. Wang, Kovacs G., Wozny M., Fang M., (Boston: Springer), pp. 887-895.

2. THE MODEL OF FMCDS

In the flexible manufacturing cell, all operations may be completed on several choice machines, and the processing tasks can randomly be added or canceled at any moment during the processing, and the machines may encounter the unexpected breakdown. FMCDS can be described as follows:

The flexible manufacturing cell has H machines, M_1, M_2, ⋯, M_H, and may process N types of work pieces, L_1, L_2, ⋯, L_N, and these types have $N_{p,1}$, $N_{p,2}$, ⋯ $N_{p,N}$ batches respectively which are respectively made up of J_1, J_2, ⋯, J_N constraint operations with the technologic constraint. Supposing that the batch number and the batch size of N type of work pieces are ($L_{i,j}$, $P_{i,j}$) , ⋯, ($L_{t,N}$, $P_{t,N}$) respectively, where, $L_{i,j}$ is the j-th batch of i-th work piece and $P_{i,j}$ is the batch size of $L_{i,j}$.

The objective functions include time, cost and machine utilization ratio, etc, and they may be embodied with two sub-objective functions as follows:

① Minimize the penalty due to the earliness/tardiness

$$F_1 = \sum_{i=1}^{N_p} \sum_{j=1}^{N_{p,i}} [b_{i,j} \times \max(0, D_{i,j} - T_{f,i,j}) + a_{i,j} \times \max(0, T_{f,i,j} - D_{i,j})] \qquad (1)$$

② Maximize the machine utilization ratio

$$F_2 = \frac{\sum_{m=1}^{H} \sum_{i=1}^{N_p} \sum_{j=1}^{N_{p,i}} \sum_{k=1}^{J_i} T_{i,j,k} \times z_{m,i,j,k} \times c_m}{\sum_{m=1}^{H} (T_{f,m} - T_{s,m}) \times c_m} \qquad (2)$$

Supposing that w_1 and w_2 are the weight coefficients, then the chief objective function is defined as follows

$$F = \min(w_1 F_1 + w_2 F_2) \qquad (3)$$

The constraint conditions:

① resource constraint: The next batch operation may start on the same machine until current batch operation has finished

$$\begin{cases} T_{s,i,j,k} \geq T_{f,p,q,r} \\ z_{m,i,j,k} = z_{m,p,q,r} = 1 \end{cases} \text{or} \begin{cases} T_{s,p,q,r} \geq T_{f,i,j,k} \\ z_{m,i,j,k} = z_{m,p,q,r} = 1 \end{cases} \qquad (4)$$

② technical constraint: The next batch operation of same batch may start until current batch operation has finished

$$T_{s,i,j,k+1} \geq T_{f,i,j,k} \tag{5}$$

$$z_{m,i,j,k} = \begin{cases} 1 & k\text{-th operation of } j\text{-th batch of } i\text{-th type of work piece is allocated on } m\text{-th machine} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

$$T_{s,m} = \min(T_{s,i,j,k} \in \forall o_{i,j,k} \wedge z_{m,i,j,k} = 1) \tag{7}$$

$$T_{f,m} = \max(T_{f,i,j,k} \in \forall o_{i,j,k} \wedge z_{m,i,j,k} = 1) \tag{8}$$

$o_{i,j,k}$——k-th batch operation of j-th batch of i-th type of work piece

$T_{s,i,j,k}$——the start time of $o_{i,j,k}$

$T_{f,i,j,k}$——the completion time of $o_{i,j,k}$

$T_{i,j,k}$——the processing time of $o_{i,j,k}$

H——the total number of the machines

N_p——the type number of the work pieces

$N_{p,i}$——the batch number of i-th work piece

J_i——the operation number of every batch of i-th work piece

$a_{i,j}$——the penalty coefficient due to tardiness of j-th batch of i-th type of work piece

$b_{i,j}$——the penalty coefficient due to earliness of j-th batch of i-th type of work piece

c_m——the importance coefficient of m-th machine

3. IMMUNE ALGORITHM

Immune algorithm is a new intelligent algorithm enlightened by the biologic immunity system[3, 4]. In order to avoid the algorithm plunging the local optimization, the concept of the temporary solution set is imported into the algorithm to accelerate the algorithm to evolve rapidly for the preponderant direction. The mechanism of the vaccine extraction and inoculation is adopted to lead the algorithm search. The flow of the algorithm is show in Fig.1. The detailed steps are as follow:

step 1 Antigen identification and antibody coding

Every type of work piece is divided into several batches and every operation of every batch may choose one from several machines, therefore new code method showed in Fig.2 is developed.

Step 2 Create solution colony

N_m excellent antibodies are picked up from the memory to form the new generation antibody colony along with the created antibodies.

Step 3 Antibody evaluation

The antibody concentration is adopted to restrain the antibodies which are of large scale but aren't the optimal solution, and the information entropy is regarded as the index to measure the antibody affinity, and the expectant reproduction ratio is applied as the standard of evaluating the antibody.

Supposing that there are N antibodies and every antibody has M genes, and every gene may choose one from S symbols, $k_1, k_2, \cdots, k_s$. The information entropy of N antibodies is defined as follows

$$H(N) = \frac{1}{M} \sum_{i=1}^{M} H_i(N) \tag{9}$$

Where, $H_i(N) = \sum_{j=1}^{S} -p_{ij} \log p_{ij}$, $H_i(N)$ is information entropy of i-th gene of N antibodies, and p_{ij} is the probability by which i-th genes of N antibodies are the symbol k_j.

The antibody affinity: indicating the comparability between two antibodies. The antibody affinity between v-th antibody and w-th antibody is defined as follows

$$a_{v,w} = \frac{1}{1 + H(2)} \tag{10}$$

Where, $H(2)$ is information entropy of v-th antibody and w-th antibody. Similarly, the affinity between antigen and antibody is defined as follows

$$a_v = \frac{1}{(1 + d_v)} \tag{11}$$

Where, d_v is the difference degree between the antigen and the antibody. The antibody concentration C_v of v-th antibody is defined as follows

$$C_v = \frac{1}{N} \sum_{w=1}^{N} b_{v,w} \tag{12}$$

Where, $b_{v,w} = \begin{cases} 1 & \alpha \cdot a_v \leq a_w \leq \beta \cdot a_v \\ 0 & \text{otherwise} \end{cases}$, α and β are the adjustment coefficients.

Finally the expectant reproduction ratio e_v is defined as follows

$$e_v = \frac{a_v \prod_{w=1}^{N} (1 - h_{v,w})}{\sqrt{C_v}} \tag{13}$$

Where, $h_{v,w} = \begin{cases} a_{v,w} & \alpha \cdot a_v \leq a_w \leq \beta \cdot a_v \\ 0 & \text{otherwise} \end{cases}$

Step 4 Vaccine extraction

p_{ij} is the probability by which i-th genes are the symbol k_j in the colony.

$$p_{i,j} = \frac{1}{N} \sum_{j=1}^{N} \mathrm{a}_j \tag{14}$$

Where, $a_j = \begin{cases} 1 & g(i) = k_j \\ 0 & \text{otherwise} \end{cases}$, g($i$) is the symbol of i-th genes in the colony, and N is the antibody quantity of the colony.

Let the symbol k_j in the allele, whose probability is bigger than the set valve value and is the biggest, as the vaccine segment of the allele, and the vaccine extracted finally is B=（b_1，b_2，…，b_n），where, $b_j = \begin{cases} k_j & \max(p_{i,j}) \geq T_a \\ 0 & \text{otherwise} \end{cases}$, and T_a is the valve value.

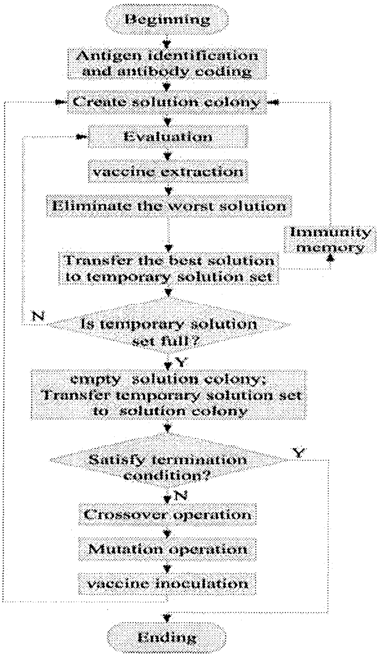
Beginning
Antigen identification and antibody coding
Create solution colony
Evaluation
vaccine extraction
Eliminate the worst solution
Immunity memory
Transfer the best solution to temporary solution set
N
Is temporary solution set full?
Y
empty solution colony; Transfer temporary solution set to solution colony
Satisfy termination condition?
Y
N
Crossover operation
Mutation operation
vaccine inoculation
Ending

Figure 1. Algorithm flow.

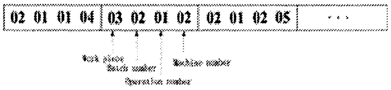

Figure 2. Antibody code.

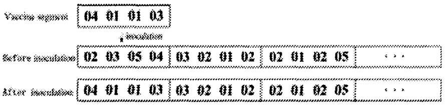

Figure 3. Vaccine inoculation.

Step 5 Eliminate the worst solution and transfer the best solution

Solution colony is ranked according to e_v, then the antibody with least e_v will be eliminated, and at the same time the antibody with biggest e_v will be transferred to the temporary solution set as the temporary optimal solution.

Step 6 Judge whether the temporary solution set has been full

If the temporary solution set has been full, then continue to carry out next step, else its size doesn't reach N_t, therefore go to step 3, and reevaluate the rest solution colony.

Step 7 Empty solution colony and transfer temporary solution set to colony

Abandon the rest solution colony, and transfer temporary solution set to solution colony. At the same time, N_m excellent antibodies are picked from the temporary solution set to be saved in the memory. Finally, the temporary solution set is eliminated.

Step 8 Judge whether the termination condition is satisfied

If satisfied, then terminate, else continue.

Step 9 Crossover operation and mutation operation

The crossover operation doesn't change the machine allocation of the batch operation, but only change the order of the batch operation. The mutation operation is used to change the machine allocation.

Step 10 Vaccine inoculation

The vaccine inoculation operation is shown in Fig.3.

4. FLEXIBLE DYNAMIC SCHEDULING STRATEGY

In the flexible dynamic scheduling, the processing tasks are likely to be inserted or canceled in the midway, and the machine breakdown is likely to appear unexpectedly, so it's necessary to import the flexible dynamic scheduling strategy [5].

4.1 Rolling window technology

The rolling window technology is put forward according to the rolling optimization principle of the forecast control. It isn't the optimization through whole course and isn't optimization only once, but the rolling optimization in the finite period of time. At every sampling time, the optimization performance indexes only involve the finite time from the beginning to the future. When next sampling time appears, the optimization time processes forward accordingly, and this performs repeatedly.

4.2 Scheduling task pool

The scheduling task pool is divided into three consecutive sub-pools. The pending scheduling task pool keeps the work pieces which have been planned but not been scheduled. The window task pool keeps the work pieces which need to be been scheduled and optimized. The completion task pool keeps the work pieces which are completed. The flow course of the work pieces in scheduling pool is shown as Fig.4.

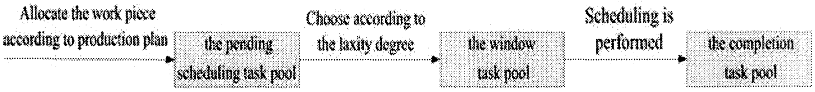
Allocate the work piece according to production plan
the pending scheduling task pool
Choose according to the laxity degree
the window task pool
Scheduling is performed
the completion task pool

Figure 4. Flow course of the work pieces in scheduling pool.

4.3 Rolling-disturbance hybrid rescheduling

Because the flexible and dynamic characteristic of the system, rolling-disturbance hybrid rescheduling strategy is put forward. If the disturbance doesn't appear in the scheduling system, then the rescheduling will be rolled periodically. However, if disturbance appears unexpectedly, the rescheduling must be carried out immediately to respond to the events.

5. THE SCHEDULING INSTANCE

In the manufacturing cell of Xi'an Aviation Engine Manufacturing Corporation, there are 15 machines and 10 types of work pieces, and every type of work piece includes several batches. Machines, work pieces and batches are numbered as table 1. The technical information is as table 2.

The scheduling of 20 work piece batches is carried out by the immune algorithm put forward, and the optimized scheduling solution is gained, and the evolution curve of the algorithm is shown in Fig.5. It's revealed from the evolution curve that the constringency ability of the algorithm for the scheduling problem is strong and the evolution efficiency is high. The Gantt chart of the first optimized scheduling solution is shown in Fig.6.

The disturbances will appear unexpectedly during the production. Work piece 1 is canceled at time 16, a new batch of work piece type 10 is added at time 18, and the breakdown appears on machine 13 at time 20. Consequently rescheduling is carried out, and the Gantt chart of the whole dynamic scheduling is shown in Fig.7.

Table 1. Work piece number in the manufacturing cell

Work piece type	Batch quantity	Work piece number
1	2	1 2
2	2	3 4
3	4	5 6 7 8
4	3	9 10 11
5	1	12
6	1	13
7	3	14 15 16
8		17
9	2	18 19
10	1	20

Table 2. Technical information in the manufacturing cell

Work piece type	Batch quantity	Due date	Operation number	Choice machine	Processing time
1	2	38	1	1, 2, 11, 12	3
			2	3, 4, 5	8
			3	6, 8, 9, 10	4
			4	7, 13, 4	6
			5	11, 12	2
			6	15	5
			7	11, 12, 13, 14	7
2	2	28	1	1, 2, 11, 12	2
			2	8, 9, 10	3
			3	11, 12	4
			4	7, 13, 14	7
3	4	26	1	3, 4, 5	5
			2	7, 11, 12	4
			3	7	8
4	3	18	1	3, 4, 5	9
			2	6, 8, 9, 10	5
5	1	14	1	1, 2	6
			2	13, 14, 15	2
			3	7, 10	5
6	1	20	1	8, 9, 10	2
			2	6	6
			3	11, 12, 15	3
			4	3, 4, 5	1
			5	10	4
7	3	19	1	2	5
			2	11, 12, 13, 14	3
8	1	28	1	1	4
			2	11, 12	1
			3	15	6
			4	13, 14	2
			5	3, 4, 5	5
9	2	17	1	3, 4, 5	3
			2	2	5
			3	11, 12	2
10	1	25	1	13, 14	4
			2	2	2
			3	11, 12	6
			4	15	3
			5	3, 4, 5	3
			6	8, 9, 10	8

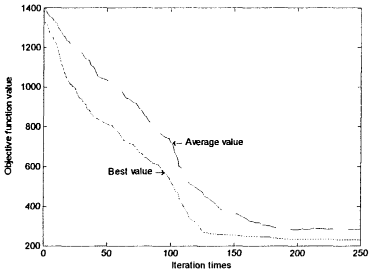

Figure 5. The evolution curve of the algorithm.

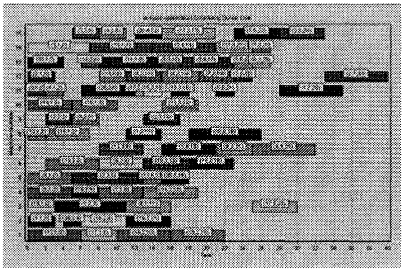

Figure 6. The scheduling Gantt chart.

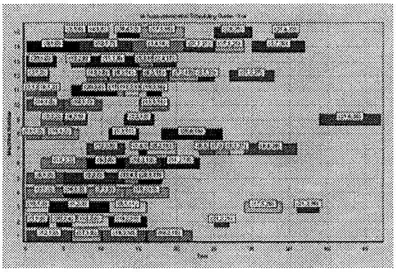

Figure 7. The rescheduling Gantt chart.

6. CONCLUSION

The complex dynamic scheduling of the flexible manufacturing cell is researched, which considers various disturbance factors. The operation has flexible technical routes, and multi-objective optimization is considered. The model of FMCDS is set up, and a new immune algorithm is put forward for FMCDS. The concepts of the rolling window technology and the scheduling mask pool are introduced and the rolling-disturbance hybrid rescheduling strategy is developed. The practice application of FMCDS in Xi'an Aviation Engine Manufacturing Corporation indicates that the algorithm, the strategy and the model can satisfy the requirement of FMCDS.

7. REFERENCES

1. Jain, A.S., Meeran, S.,(1998), A state-of-the-art review of job-shop scheduling techniques, Technical Report, Department of Physics, University of Dundee, Scotland, pp.130–140.
2. Tapan, P., Bagchi, (1999), *MultiObjective Scheduling by Genetic Algorithms*, Kluwer Academic Publishers, NewYork (ISBN 0-7923-8561-6).
3. Leandro, N. C., Jonathan, T., (2002), *Artificial Immnue System: A New Computational Intelligence Approach*, Springer Verlang, Great Britain (ISBN 1-8523-594-7).
4. JIAO, L. DU, H., (2003), Development and prospect of the artificial immune system, *Acta Electronica Sinica*, 31(10), pp.1540–1548.
5. Church, L., Uzsoy. R, (1992), Analy of periodic and event-driven rescheduling policies in dynamic shops, *International Journal of Computer Integrated Manufacturing*, 5(3), pp.153–163.