A Tabu-Genetic Hybrid Search Algorithm for Job-shop Scheduling Problem

Yan Ge[1], Aimin Wang[1] , Zijin Zhao[1] and Jieran Ye[1]

[1] *Department of Mechanical Engineering, Beijing Institute of Technology, 5 South Zhongguancun Street, Haidian District, Beijing, China*

Abstract. To deal with the job-shop scheduling problem (JSP), a tabu-genetic hybrid search algorithm is proposed. The algorithm generates several initial solutions distributed in the whole solution space for tabu search by genetic algorithm, which avoids the over-dependence on the initial solution of tabu search algorithm. With the mechanism mentioned above, the algorithm proposed has both global search performance of genetic algorithm and local search performance of tabu search algorithm. Finally, a program was developed with the actual data of FT (10×10). to verify the feasibility and effectiveness of the algorithm. The result shows that the algorithm achieves satisfactory results in all indexes mentioned above.

1 Introduction

Scheduling plays a very important role in many industrial systems [1]. As an important branch of production scheduling and combinatorial optimization problem [2], Job-shop scheduling problem (JSP) has been proved to be NP-hard problem and the number of feasible solutions increases exponentially with the increase of the scale of the problem.

Job shop scheduling problem has attracted a lot of researches in recent years. With the development of algorithms and the deepening of the research on the problem, group optimization algorithms represented by genetic algorithm and local search algorithms represented by tabu search algorithm, are widely used at present. Bierwirth C [3] improved the crossover operator in genetic algorithm and realized the solution of JSP. Zhang C Y [4] proposed hybrid genetic algorithm for the JSP. Aiming at minimizing the makespan, Nowicki E & Smutnicki C [5] proposed a fast and easily implementable approximation algorithm based on a taboo search, which realized well-known 10 times 10 hard benchmark problem and got the best solution at the time. Geyik F & Gedimoglu I H [6] also used tabu search algorithm to solve JSP. Zhang C Y et al. [7] applied a new enhanced neighborhood structure to tabu search algorithm and solved a series of standard JSPs.

The researches mentioned above show that both genetic algorithm and tabu search algorithm can solve job shop scheduling problems. Genetic algorithm has better global search ability but is prone to fall into premature [8]. Tabu search algorithm has strong local search ability, but its optimization results depend heavily on the initial solution [9]. Therefore, a tabu-genetic hybrid algorithm is designed for JSP, which combines the advantages of genetic algorithm and tabu search algorithm. Genetic algorithm is used to generate several initial solutions for tabu search algorithm to carry out the multi-point local optimization so that the final solution can be achieved.

2 Problem description and mathematical modelling

2.1 Problem description

The job shop scheduling problem involved in this paper can be described as: all processes involved are allocated to appropriate time period of m machines by reasonably planning the arrangement sequence of n jobs to minimize the maximum completion time (Cmax).

To solve the problem better, some hypotheses are put forward:

(1) There is no sequence constraint among any two processes of different jobs;

(2) Once a process starts, it cannot be interrupted.

2.1 Mathematical modelling

Parameters:

J_i : job i , $i = 1, 2, ...n$;

M_j : machine j , $j = 1, 2, ...m$;

P_i : count of processes of J_i ;

O_{ik} : the k-th process of J_i , $k = 1, 2, ...P_i$;

T_{ik} : processing time of O_{ik} ;

U_{ikj}: equals 1 if O_{ik} can be arranged on M_j, and 0 otherwise.

Decision variables:

ST_{ik} : the start time of O_{ik} ;

X_{ikj} : equals 1 if O_{ik} is arranged on M_j , and 0 otherwise.

Based on the aforementioned assumptions and notations, the model is presented as follows:

$$Min\ C_{\max} \quad (1)$$

Subject to:

$$ST_{ik}+T_{ik}\le C_{\max}\ ,\ \forall i,k \quad (2)$$

$$ST_{ik}+T_{ik}\le ST_{i(k+1)}\ ,\ \forall i,k \quad (3)$$

$$\sum_{i=1}^{n}\sum_{k=1}^{P_i}\sum_{j=1}^{m}U_{ikj}=1\ ,\ \forall i,k,j \quad (4)$$

$$\sum_{i=1}^{n}\sum_{k=1}^{P_i}\sum_{j=1}^{m}X_{ikj}=1\ ,\ \forall i,k,j \quad (5)$$

$$X_{ikj}\text{-}U_{ikj}=0\ ,\ \forall i,k,j \quad (6)$$

$$\sum_{k=1}^{Pi}U_{ikj}=1\ ,\ \forall i,k,j \quad (7)$$

$$\left(ST_{ik}-ST_{pq}-T_{pq}\right)\cdot\left(ST_{pq}-ST_{ik}-T_{ik}\right)\cdot\left(X_{ikj}\cdot X_{pqj}\right)\le 0\ ,\ \forall i,k,p,q,j \quad (8)$$

$$X_{ikj}\text{-}U_{ikj}=0\ ,\ \forall i,k,j \quad (6)$$

$$\sum_{k=1}^{Pi}U_{ikj}=1\ ,\ \forall i,k,j \quad (7)$$

$$\left(ST_{ik}-ST_{pq}-T_{pq}\right)\cdot\left(ST_{pq}-ST_{ik}-T_{ik}\right)\cdot\left(X_{ikj}\cdot X_{pqj}\right)\le 0\ ,\ \forall i,k,p,q,j \quad (8)$$

The scheduling objective of the job shop scheduling problem involved in this paper is minimizing the maximum completion time of all jobs, which is shown in Equation (1). The constraints of the problem are shown in Equation (2) to Equation (8). Equation (2) ensures that Cmax is the maximum completion time of all processes. Equation (3) is constraint on processing craft route, meaning that a single process cannot start until its former process belonging to the same job I s completed. Equation (4) - (6) are the constraints of processing machines, meaning that each process must have only one processing machine, and the processing machine must belongs to its optional machines. Equation (7) expresses that there will be no repetitive machines in the processing craft route of any job. Equation (8) describes that each machine can only process at most one process at the same time.

3 Tabu-genetic hybrid search algorithm

To solve the job shop scheduling problem involved, a tabu-genetic hybrid search algorithm is proposed in this paper. Fig. 1 shows the processing flow of the algorithm.

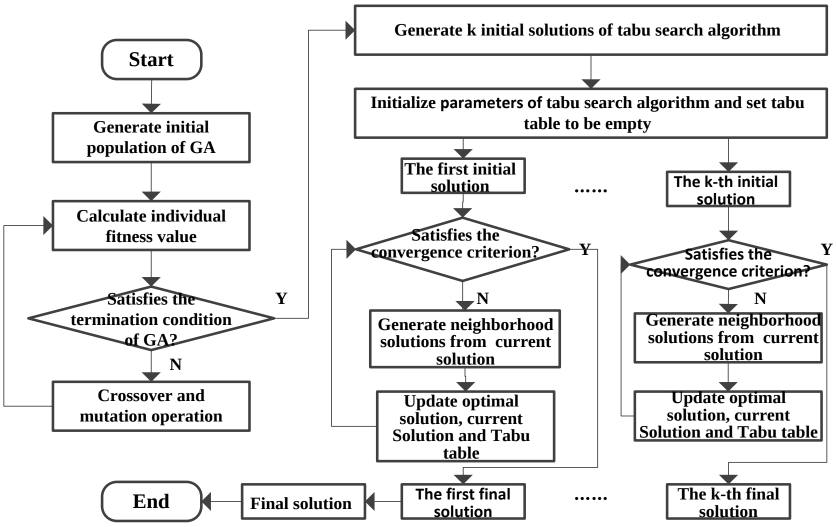
Start
Generate initial population of GA
Calculate individual fitness value
Satisfies the termination condition of GA?
Y
N
Crossover and mutation operation
Generate k initial solutions of tabu search algorithm
Initialize parameters of tabu search algorithm and set tabu table to be empty
The first initial solution
......
The k-th initial solution
Satisfies the convergence criterion?
Y
N
Generate neighborhood solutions from current solution
Update optimal solution, current Solution and Tabu table
Satisfies the convergence criterion?
Y
N
Generate neighborhood solutions from current solution
Update optimal solution, current Solution and Tabu table
End
Final solution
The first final solution
......
The k-th final solution

Figure 1 The processing flow of the algorithm proposed

The algorithm is the combination of genetic algorithm and tabu search algorithm：Firstly, genetic algorithm is used to generate a series of initial solutions for tabu search algorithm. Then tabu search algorithm is used to get multiple local optimal solutions according to the initial solutions. Finally, the final solution of the algorithm is generated from several local optimal solutions. The algorithm has both global search ability of genetic algorithm and local search ability of tabu search algorithm by avoiding the shortcomings of them, which are the characteristic of being prone to prematurity and the excessive dependence on initial solution.

3 Generating initial solutions for tabu search algorithm by genetic algorithm

3.1 Coding rules and the generation of initial population

The algorithm proposed in this paper uses process-based encoding method to generate initial population. A coding of three jobs is shown in Fig. 2, which expresses the sequence of each process is: the first process of Job 2→ the first process of Job 1→the second process of Job 1→ the first process of Job 3→the second process of Job 2→ the second process of Job 3→the third process of Job 2→ the third process of Job 3.

2	1	1	3	2	3	2	3

Figure 2. A coding of three jobs

3.2 Rules of crossover and mutation

Because the chromosomes are generated by process-based encoding method, the POX crossover operator as shown in Fig. 3 is selected.

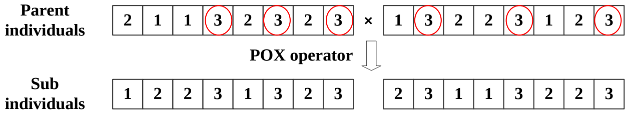

Figure 3. The POX crossover rule

In this section, a two-point mutation method is used to realize the mutation operation of a single chromosome, that is, two mutation points are randomly selected, and the new chromosome is got by exchanging positions of the corresponding two genes.

The processes of tabu search in the hybrid tabu-genetic search algorithm proposed in this paper are as follows:

Input: one of the initial solutions gotten by genetic algorithm;

Output: A local optimal solution gotten by tabu search process.

Step 1: In the initial state, set the current solution $C(x)$ and the optimal solution $C^*(x)$ to the initial solution, and the tabu table to empty. Get all the neighborhood solutions of $C(x)$.

Step 2: If there are no neighborhood solutions, turn to step 6. Otherwise, get the optimal solution of the neighborhood solutions. If the optimal solution is already in the tabu table, turn to step 4; otherwise, turn to step 3.

Step 3: Update the current solution $C(x)$ to the optimal solution achieved and press it into the top of the tabu table. If $C(x)$ is better than the historical optimal solution, $C^*(x)=C(x)$, otherwise $C^*(x)$ remains unchanged. Turn to Step 6;

Step 4: If the optimal solution in the current neighborhood solutions satisfies the desire criterion, which means that the solution is superior to the historical optimal solution, turn to step 3. Otherwise, turn to step 5;

Step 5: Remove the optimal solution from the neighborhood solutions, and turn to Step 2.

Step 6: If the stopping criterion of the algorithm is reached, end the algorithm and output $C^*(x)$. Otherwise, calculate all the neighborhoods of $C(x)$ and turn to Step 2.

In this algorithm, the neighborhood solutions of a solution are generated by a two-point exchange method. Fig. 4 shows the neighborhood solutions of the chromosome shown in Fig. 2.

3 Process of tabu search

1 2 1 3 2 3 2 3 Exchange genes of position 1 and position 2	2 3 1 1 2 3 2 3 Exchange genes of position 2 and position 4	2 1 3 1 2 3 2 3 Exchange genes of position 3 and position 4
1 1 2 3 2 3 2 3 Exchange genes of position 1 and position 3	2 2 1 3 1 3 2 3 Exchange genes of position 2 and position 5	2 1 2 3 1 3 2 3 Exchange genes of position 3 and position 5
3 1 1 2 2 3 2 3 Exchange genes of position 1 and position 4	2 3 1 3 2 1 2 3 Exchange genes of position 2 and position 6	2 1 3 3 2 1 2 3 Exchange genes of position 3 and position 6
3 1 1 3 2 2 2 3 Exchange genes of position 1 and position 6	2 2 1 3 2 3 1 3 Exchange genes of position 2 and position 7	2 1 2 3 2 3 1 3 Exchange genes of position 3 and position 7
3 1 1 3 2 3 2 2 Exchange genes of position 1 and position 8	2 3 1 3 2 3 2 1 Exchange genes of position 2 and position 8	2 1 3 3 2 3 2 1 Exchange genes of position 3 and position 8
2 1 1 2 3 3 2 3 Exchange genes of position 4 and position 5	2 1 1 3 3 2 2 3 Exchange genes of position 5 and position 6	2 1 1 3 2 2 3 3 Exchange genes of position 6 and position 7
2 1 1 2 2 3 3 3 Exchange genes of position 4 and position 7	2 1 1 3 3 3 2 2 Exchange genes of position 5 and position 8	2 1 1 3 2 3 3 2 Exchange genes of position 7 and position 8

Figure 4. The neighborhood solutions of the chromosome shown in Fig. 2.

4 Experimental study

According to the Tabu-genetic hybrid search algorithm proposed, a simulation example has been performed in VS2010. The data of jobs and machines are from the famous workbench FT(10×10). The information of the example is shown in Table 1. The selection probability of the genetic algorithm is 0.5, the crossover probability is 0.65, and the mutation probability is 0.2. The tabu table length of tabu search algorithm is 5.

https://doi.org/10.1051/e3sconf/20199504007

Table 1. Information of FT (10×10)

	1		2		3		4		5		6		7		8		9		10	
J1	1	29	2	78	3	9	4	36	5	49	6	11	7	62	8	56	9	44	10	21
J2	1	43	3	90	5	75	10	11	4	69	2	28	7	46	6	46	8	72	9	30
J3	2	91	1	85	4	39	3	74	9	90	6	10	8	12	7	89	10	45	5	33
J4	2	81	3	95	1	71	5	99	7	9	9	52	8	85	4	98	10	22	6	43
J5	3	14	1	6	2	22	6	61	4	26	5	69	9	21	8	49	10	72	7	53
J6	3	84	2	2	6	52	4	95	9	48	10	72	1	47	7	65	5	6	8	25
J7	2	46	1	37	4	61	3	13	7	32	6	21	10	32	9	89	8	30	5	55
J8	3	31	1	86	2	46	6	74	5	32	7	88	9	19	10	48	8	36	4	79
J9	1	76	2	69	4	76	6	51	3	85	10	11	7	40	8	89	5	26	9	74
J10	2	85	1	13	3	61	7	7	9	64	10	76	6	47	4	52	5	90	8	45

Table 2 and Fig.5 show the results of scheduling.

Table 2. The result of scheduling

The best solution	The worst solution	Average computing time	Historical best solution
930	1178	149.2s	930

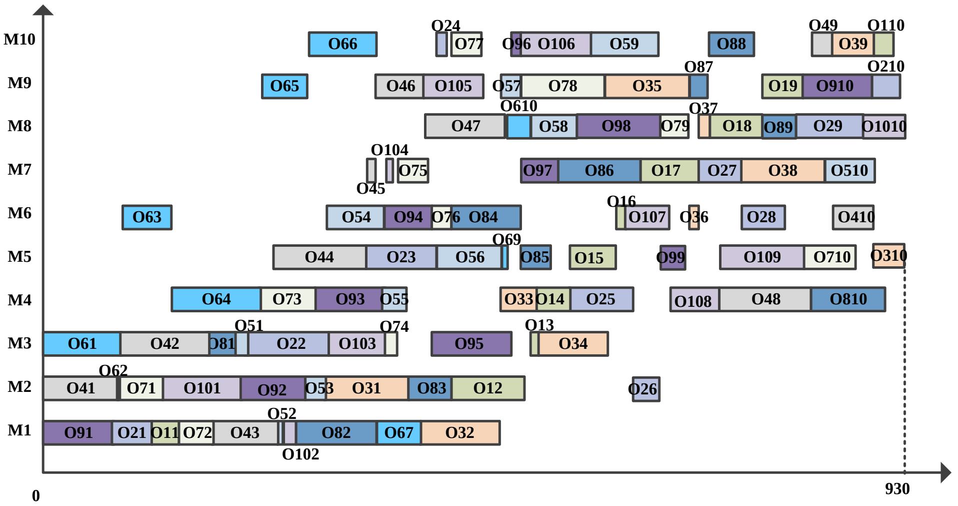

Figure 5 The results of scheduling

As can be seen from Table 2 and Figure 5, for the 10×10 case given above, the proposed algorithm can achieve the optimal solution which is equal to the historical best solution.

4 Conclusions

A tabu-genetic hybrid search algorithm is proposed to solve JSP with the objective of minimizing the maximum completion time. The algorithm generates several initial solutions distributed in the whole solution space for tabu search by genetic algorithm, which avoids the over-dependence on the initial solution of tabu search algorithm. With the mechanism mentioned above, the algorithm proposed has both global search performance of genetic algorithm and local search performance of tabu search algorithm.

The JSP can be solved efficiently with the algorithm proposed in this paper. In combination with the data of FT(10×10), the verification shows that the proposed method in this paper can effectively solve the JSP and reduce Cmax.

5 References

1. K. Vahit, Expert.Syst. Appl **45**,71–84 (2016)
2. I. Driss, K.N. Mouss, A. Laggoun, J.Mech.Sci.Technol **29**,1273–1281 (2015)
3. C. Bierwirth, OR Spektrum **17 ,** 87-92 (1995)
4. C.Y. Zhang, Y.Q. Rao, P.G. Li, Int.J.Adv.Manuf.Tech **39** ,965-974 (2008)
5. E. Nowicki, C. Smutnicki,Manage.Sci **42,** 797-813 (1996)
6. F. Geyik, I.H. Cedimoglu, J.Intell.Manuf **15**, 439-448 (2004)
7. C.Y. Zhang, P.G. Li, Z.L. Guan, et al, Comput.Oper.Res **34** ,3229-3242 (2007)
8. A. Kerr, K. Mullen,Comp.Mater.Sci(in press ,2018)
9. M. Qiu, Z. Fu, R. Eglese et al, Comput.Oper.Res **100**,102-116 (2018)