J Ambient Intell Human Comput
DOI 10.1007/s12652-016-0425-9

ORIGINAL RESEARCH

Flexible job shop scheduling based on improved hybrid immune algorithm

Xu Liang[1] · Ming Huang[1] · Tao Ning[1]

Received: 3 June 2016 / Accepted: 7 October 2016

Abstract An improved hybrid immune algorithm (HIA) with parallelism and adaptability is proposed to solve the flexible job shop scheduling problem. In order to represent the actual characteristics of the problem's solution, in the algorithm the author uses a hybrid encoding method of piece—machine. Firstly, adaptive crossover operator and mutation operator are designed based on the encoding antibody method and the affinity calculation based on group matching is adopted. Secondly, the algorithm uses adaptive crossover probability and mutation probability in the operation of immune for the antibody population. The new antibody after crossing can automatically meet the constraints of the problem. Next, a hybrid algorithm based on simulated annealing algorithm is introduced to avoid the local optimization in this paper. Finally, it is demonstrated the effectiveness of the proposed algorithm through the simulation and comparison with some existing algorithms.

Keywords Flexible job shop scheduling · Hybrid immune algorithm · Simulated annealing algorithm

1 Introduction

Flexible Job Shop Scheduling Problem (FJSP) is a kind of complex combinatorial optimization problem and its scheduling scheme is more flexible, however, it also increases the uncertainty of machine.

The intelligent algorithm is the hot spot of the research in solving FJSP at present. Li et al. (2011) proposed the algorithm of hybrid artificial bee colony, introducing partial update algorithm in observation stage and using the coefficient of exit temperature to update the size of neighborhood. Gao et al. (2014) proposed ant colony genetic algorithm to solve FJSP with the limit of work ability. Liu et al. (2015) fused the ant colony algorithm, genetic algorithm and particle swarm optimization (PSO) to solve multi-objective FJSP Mokhtari et al. (2011) introduced simulated annealing algorithm into genetic algorithm, and puts forward a hybrid local search algorithm. Geyik and Dosdogru (2013) acted the ant colony optimization algorithm as the main level and PSO as minor level to solve the scheduling group of FJSP. Li and Liang (2016) used an effective hybrid algorithm for the FJSP. Few domestic scholars have applied the immune algorithm to solve FJSP, but the method is still not flexible enough (Zhao et al. 2014; Rahmati et al. 2013). Immune algorithm is a new kind of intelligent algorithm, which uses group search strategy with parallelism. However, it is a probabilistic random search algorithm, easy to fall into the local optimal. Therefore, this paper proposes a novel hybrid method of combining the immune algorithm and simulated annealing algorithm to solve FJSP.

In Sect. 1, the objective function of the problem was established. In Sect. 2.1, the adaptive crossover operator and mutation operator were designed based on the encoding antibody method and the affinity calculation based on group matching was adopted. In Sect. 2.2, in order to avoid the unnecessary random search and add new information fragment, in the algorithm, it was used adaptive crossover probability and mutation probability in the operation of immune for the antibody population. The new antibody after crossing could automatically meet the constraints of

✉ Tao Ning
daliannt@126.com

[1] Institute of Software, Dalian Jiaotong University, 116045 Dalian, China

the problem. Next, a hybrid algorithm based on simulated annealing algorithm was introduced to avoid the local optimization in this paper. In Sect. 3, it was demonstrated through the simulation and comparison with other algorithms that the method was effective.

2 Model of FJSP

2.1 Description of FJSP

The FJSP can be described as follows: there are N work pieces to be processed on M machines in the workshop, and each work piece i $(i \in \{1, 2, \ldots, N\})$ consists of a sequence of n_i $(n_i \geq 1)$ working procedures, which should be processed in a certain route. R_{ij} is the jth $(j \in \{1, 2, \ldots, n_i\})$ procedure of workpiece i, M is the machine set which can process the above working procedures, and R_{ij} can be processed by any machine m $(m \in \{1, 2, \ldots, M\})$ with processing capability, besides the machine m has the ability to process $q \geq 1$ working procedures which belong to different work piece. The FJSP consists of the machine-selecting sub-problem and the sequencing sub-problem.

FJSP usually do the following assumptions:

1. One machine can not process more than one process at the same time;
2. One workpiece can not be processed on more than one machine;
3. Once the workpiece is started, it is not allowed to be interrupted;
4. The processing time of the workpiece in the machine can be determined;
5. All the workpiece can be processed at moment of 0.

2.2 Objective function

The model of FJSP is established in formula (1), including three objective functions.

$$f = \min\left[\max\left(\sum_{m=1}^{M} F_m\right)\right] \tag{1}$$

In Eq. (1), $\min\left[\max\left(\sum_{m=1}^{M} F_m\right)\right]$ is minimizing the makespan, F_m is the total completion time of machine m, $F_m = \sum_{i=1}^{N} \sum_{j=1}^{n_i} (S_{ijm} b_{ijm} + S_{ijm} t_{ijm})$, b_{ijm} is the starting time for R_{ij} on machine m, t_{ijm} is the processing time for R_{ij} on machine m and S_{ijm} is the state whether R_{ij} is processed on machine m or not, that is:

$$S_{ijm} = \begin{cases} 1, & R_{ij} \text{ is processed on the machine } m; \\ 0, & R_{ij} \text{ is not processed on the machine } m. \end{cases}$$

In actual production, minimizing the makespan is usually used as the main objective of job shop scheduling (Prakash and Vidyarthi 2014).

3 Improved hybrid immune algorithm

3.1 General structure of an immune algorithm

The immune system is a highly evolved and complex function system in organism. It can adaptively identify and eliminate the intruding antigenic foreign body, and it has the adaptive capacity of learning and memory (Ning et al. 2016a, b). Immune algorithm is a bionic algorithm imitating such function (Ning et al. 2016a, b). In which, the antigen corresponds to the objective function of the problem. The antibody corresponds to the solution. The affinity of antigen and antibody are decided according to the objective function, and the affinity between the antibodies depends on the similarity degree of the corresponding solutions. The alternative solutions is evaluated and selected using these two kinds of affinity, and the searching efficiency near the optimum is improved through the interaction between antibodies. At last, the local optimum is got rid through the inhibition of antibody memory cells to achieve the ultimate convergence to the global optimum.

The main steps are as follows.

3.1.1 Encoding of antibody

This paper adopted the chromosome encoding method in literature (Xue et al. 2014). The antibody was coded by ordinal number. A processing sequence in FJSP can form the antibody with the length of $l + m + 1$, l here is the number of workpieces and m here is the number of machine:

$$(0, i_{11}, i_{12}, \ldots, i_{1s}, 0, i_{21}, i_{22}, \ldots, i_{2t}, 0, \ldots, 0, i_{m1}, i_{m2}, \ldots, i_{mw}, 0)$$

In which, m is the number of machine, i_{mw} is the mw^{th} task, and the encoding of the antibody can be understood as: workpiece 1 starts to be processed from workpiece center 0 and completes the processing after the mission i_{11}, i_{12},…,i_{1s}. Workpiece 2 starts to be processed from workpiece center 0 and completes the processing after the mission i_{21}, i_{22},…,i_{2t}. For example, the antibody 0134025601780 represents the following processing sequence:

Sequence 1 Workpiece center→machine 1→machine 3→machine 4→workpiece center

Sequence 2 Workpiece center→machine 2→machine 5→machine 6→workpiece center

Sequence 3 Workpiece center→machine 1→machine 7→machine 8→workpiece center

In this encoding of antibody, the internal sequence is orderly. If the point 5 and 6 of sequence 2 exchange positions, then the objective function value will change. If sequence 1 and sequence 2 exchange positions, the objective function value will not change. Moreover, the positive order and reverse order are the same for the same sequence, i.e., if the sequence 2 is represented as 0652, then the objective function value will not change.

3.1.2 Calculation of affinity

The affinity of A_v between the antigen and antibody of v may be obtained by the transformation of the objective function, and the reciprocal of the objective function is taken in this paper, i.e.,

$$A_v = \frac{1}{f(v)} \tag{2}$$

In Eq. (2), $f(v)$is as follow:

$$f(v) = \sum_i \sum_j \sum_m c_{ij} x_{ijm} + P \times \sum_i \max\left[\sum_{i \in c} d_i \sum_{j \in N} x_{ijm} - q, 0\right] \tag{3}$$

The second term in the above equation is the penalty for violation of the constraint condition. P is set as a relatively large positive solution. The affinity between two antibodies reflects their degree of similarity.

In this paper, the affinity calculation based on group matching is adopted. Group matching is to divide one antibody into several groups according to the sequence, and compare each group with another antibody to get the matching number, and the affinity of the two antibodies can be obtained by dividing the sum of each group matching number by the antibody length. For example, if the two antibodies are 012034506780 and 035870214060, then the first antibody is divided as 12\345\678,the matching number between the first group of 12 and the second antibody is 2; the one of the second group of 345 is 0; the one of the third group of 678 is 2, and the sum of each matching number is 4, the length of antibody is 8, then the affinity of the two antibodies is 1/2.

3.1.3 Calculation of antibody concentration

The antibody concentration can be calculated as follow:

$$c_v = \frac{\sum_w S_{v,w}}{N} \tag{4}$$

In which, N is the number of antibody;

$$S_{v,w} = \begin{cases} 1, & B_{v,w} \geq T_{ac1} \\ 0, & B_{v,w} < T_{ac1} \end{cases}.$$

$B_{v,w}$ is the affinity between the antibody of v and w, T_{ac1} is the antibody affinity threshold.

The antibodies with high degree of similarity may be looked as the same antibody in the calculation of antibody concentration.

3.2 Improved algorithm

In order to improve the convergence speed of the algorithm, the initial population is produced by repeatedly being generated and certain number of antibody is extracted each time. Concentrations of the antibodies in the population are inhibited and encouragement with reference to the expectations of each antibody reproductive probability when the antibody population is cloned to maintain the diversity of the population. When the population is updated, the immune operation is made on antibody with the adaptive crossover and mutation probability (Ning et al. 2016a, b; Gutierrez and Garcia-Magario 2011), moreover, in order to break the local equilibrium state of the algorithm, the simulation annealing operation is acted as a local search algorithm to convergent in the global direction. In order to improve the efficiency of the implementation of the algorithm, this paper proposes the method of adding the elite reservation strategy in the use of immune memory library, which can record the optimum antibody in each iteration and improve the convergence speed of the algorithm. The specific algorithm flow is shown in Fig. 1:

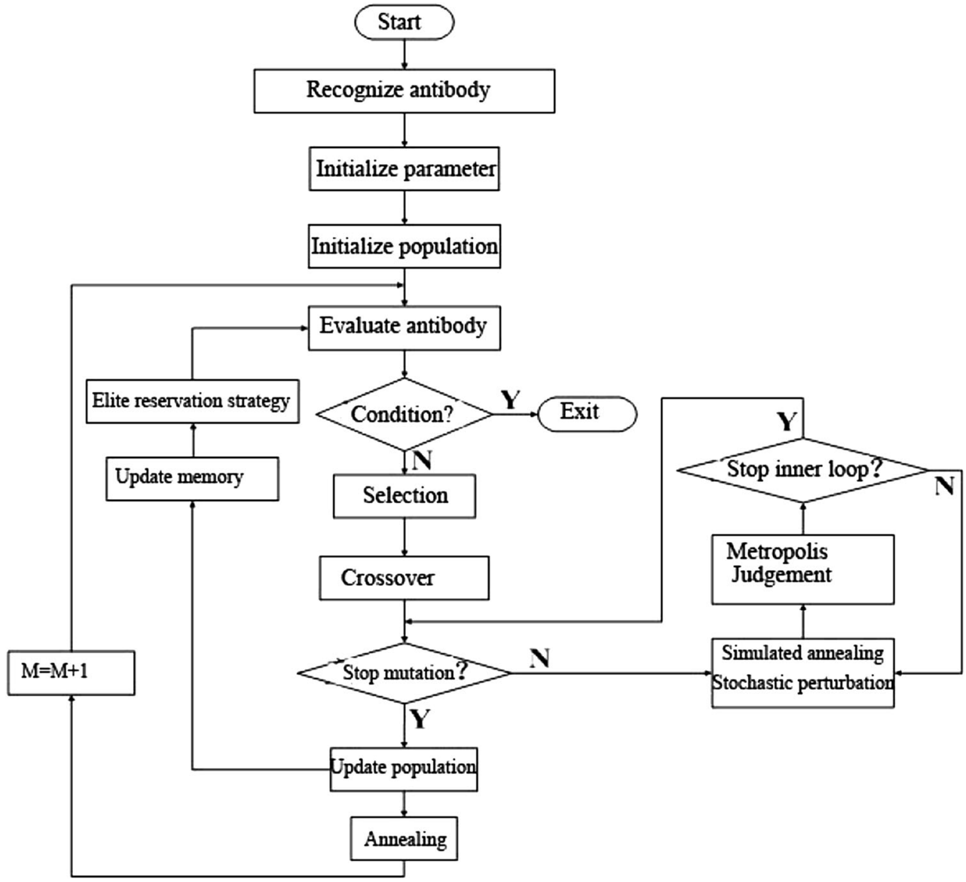
3.2.1 Update operation of population

The population needs to be continuously updated in the immune algorithm. However, updating too fast is not conducive to the retention of their own good antibody information, which may make the search process tend to a large range of random search (Bai et al. 2016).

In order to avoid the unnecessary random search and add new information fragment, the algorithm uses adaptive crossover probability of p_c and mutation probability of p_m in the operation of immune for the antibody population (Bai et al. 2016; Palacios et al. 2015).

$$P_c(v,w) = \begin{cases} \frac{k_1(A' - A_{avg})}{A' - A''}, & A' > A_{avg} \\ k_2, & A' \leq A_{avg} \end{cases} \tag{5}$$

Fig. 1 Improved hybrid immune algorithm



$$P_m(v) = \begin{cases} k_3 \sin(\frac{\pi}{2}\frac{A_{\max} - A_v}{A_{\max} - A_{avg}}), & A_v > A_{avg} \\ k_4, & A_v \le A_{avg} \end{cases} \quad (6)$$

Where, $A_{\max}$ is the maximum fitness of antibody for each iteration, A_{avg} is average fitness of antibody for each iteration, A_v is the fitness value of antibody of v, $A^{'}$is the larger fitness value between antibody of v and w, $A^{''}$ is the smaller one between them.

3.2.2 Combination of mutation operation and simulated annealing algorithm

Considering the immune algorithm is easy to convergence to the local optimal solution near the end and the probability of simulated annealing algorithm (SAA) is introduced so that it can avoid the local optimization, a hybrid immune algorithm introducing to SAA is proposed in this paper. The simulated annealing algorithm is an optimization algorithm based on Metropolis principle, and it simulates the physical annealing process. This algorithm begins to solve iteratively under the given initial temperature T_0 and determine whether or not to accept the search to another state, i.e., $T_{c+1} = \upsilon T_c$, where, υ represents the annealing rate, T_c represents the current temperature, c is the current iteration times. When the annealing temperature is reduced to the extreme point, it will stay at the optimal value with probability 1.

In view of the complexity of simulated annealing operation, a hybrid method combining mutation operation and simulated annealing operation is proposed in this paper. The annealing function (Ning 2013) is set as follows:

$$T = ceil\{T_0[1 - G'/(G+1)]\} \quad (7)$$

Where T_0 is the initial temperature, G' is the current execution generation, G is the maximum performing generation, $ceil(*)$ represents the content in the brackets will be rounded down. The back temperature function can avoid the negative influence to local algorithm because of cooling too fast or too slow.

4 Data analysis and verification

4.1 Simulation experiment

In order to verify the optimization of the algorithm, the problem of 6 workpieces and 10 machines in literature (Tavakkoli-Moghaddam et al. 2011) was solved by the proposed method and was compared with the existing SA and IA.

The testing parameters based on the proposed HIA are as follows: the times of iteration M is 100, the size of the population Sizepop is 50, the memory capacity Overbest is 15, the initial temperature T_0 is 1000 and the diversity evaluation parameters P_s is 0.9. The optimal solution obtained through the proposed method is 44. The results compared with the existing algorithms after being conducted for 50 times are shown in Table 1.

It can be seen from Table 1 that the HIA has the strongest ability to obtain the optimal solution. HIA can obtain the optimal solution for seven times in the 10 simulations and the optimal solutions are better that obtained by SA and IA. The comparison of population mean is shown in Fig. 2, and it also can be known from Fig. 2 that the method of random retention makes HIA have better initial population. The optimal solution obtained by SA is 48, that by IA is 50. However, it is obvious from the comparison of the calculation results that the HIA is superior to the single algorithm in terms of the search for the optimal solution and the efficiency of the algorithm, which proves its effectiveness.

4.2 Benchmark example test

In order to verify the effectiveness of the proposed method, 10 examples of the Brandimarte (1993) were tested. The ten examples were tested using the proposed algorithm above and some algorithms in the papers cited in the current paper. The results are shown in Table 2, in which "S_b" presents the known optimal solution, "t" is the time cost with HIA, "t_x" is the makespan.

In Table 2, HGA is a hybrid genetic algorithm for a flexible job-shop scheduling problem proposed by Gutierrez and Garcia-Magario (2011). GABFO is a hybrid genetic algorithm based on bacteria foraging optimization proposed by (Prakash and Vidyarthi 2014). IIA is an improved immune algorithm based on quantum theory proposed by Xue et al. (2014). SSPR is scatter search with path relinking for the scheduling problem proposed by Martí et al. (2011). It can be known from Table 2 that HIA can obtain the optimal solution for 6 times in Brandimarte set, moreover, except example of Mk04, the solution solved with HIA is better than that of the existing four methods in the other 9 examples. The average improvement rate comparing to the other four algorithms are 5.76, 4.12, 3.11 and 1.14 %.

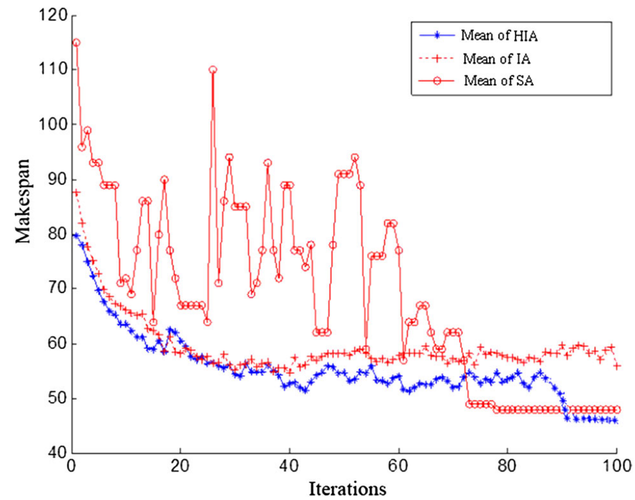


Fig. 2 Comparison of average optimal solutions

Table 2 Comparison with different algorithms of Brandimarte

Example	n × m	S_b	HIA t_x	HGA t_x	GABFO t_x	IIA t_x	SSPR t_x
Mk01	10 × 6	36	36	40	40	39	38
Mk02	10 × 6	24	24	26	26	25	24
Mk03	15 × 8	204	204	204	204	204	204
Mk04	15 × 8	48	52	62	58	56	50
Mk05	15 × 4	168	168	173	173	172	170
Mk06	10 × 15	33	44	62	59	54	48
Mk07	20 × 5	133	135	140	140	136	136
Mk08	20 × 10	523	523	523	523	523	523
Mk09	20 × 10	299	301	317	312	310	302
Mk10	20 × 15	165	165	206	188	186	176

In order to verify the convergence speed for the optimal solution with the proposed method, this paper compared it with the aforementioned methods of HGA, GABFO, IIA and SSPR. It can be seen from Fig. 3 that the optimal completion time is 51 min with HIA in 100 iterations, while the value is about 73 with GABFO, IIA or SSPR, the worst value is about 124 obtained by HGA. In addition, the optimal solution is obtained with HIA in iteration 25, the

Table 1 Comparison of four algorithms

	Optimal value	Average value	Average convergence rate (%)	Average convergence algebra	Average convergence time (s)
SA	48	50.7	33.3	78.7	21.8
IA	50	50.8	60	65.4	16.7
HIA	44	44.5	70	43.5	12.3

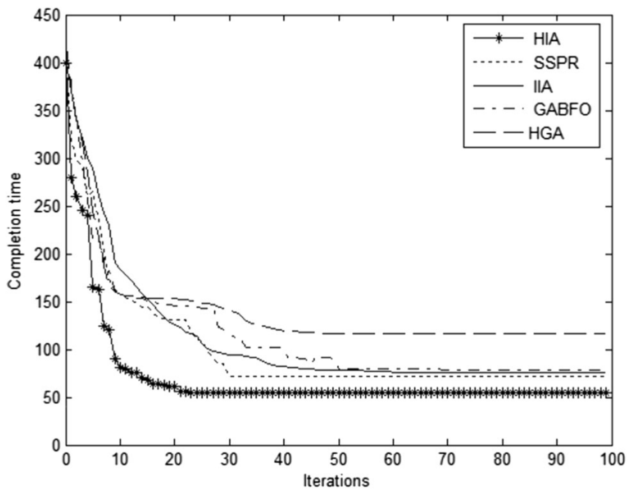


Fig. 3 Iterations of different algorithms

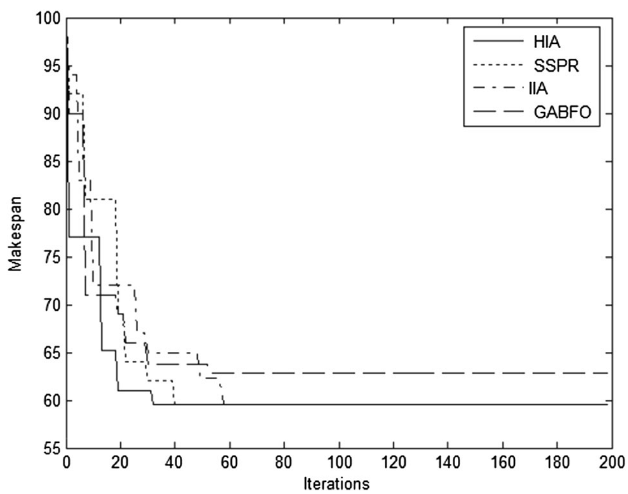


Fig. 4 Comparison of convergence efficiency

next is 32 with SSPR and the other two are worse obviously.

4.3 Verification of actual example

To solve the problem of 6 workpieces × 10 machines in the mold shop of some mechanical company using the GABFO, IIA, SSPR and HIA, and the comparison is shown in Fig. 4. It can be seen that IIA, SSPR and HIA can all search to the optimal solution of 59.6 in 200 iteration, while GABFO can only search to 62.8. However, the convergence speed of GABFO and IIA is significantly slower than HIA, i.e., IIA can convergence to the optimal solution in the 49th generation, SSPR does it in the 41st generation, and the HIA in the 33rd generation.

The Gantt chart of the optimal solution is shown in Fig. 5.

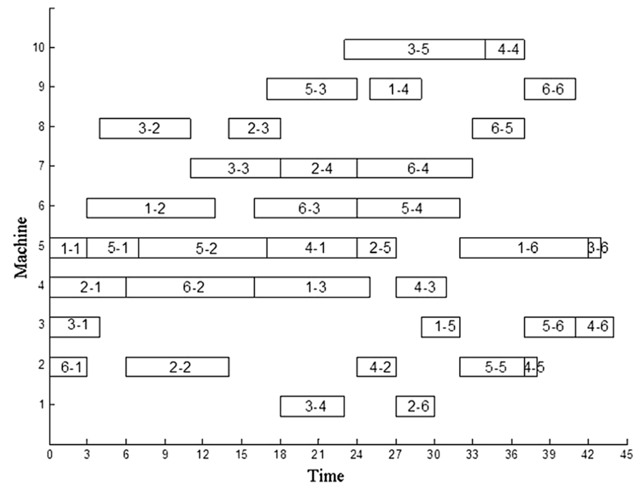


Fig. 5 Gantt chart of 6 workpieces × 10 machines

5 Conclusion

On the basis of immune algorithm and simulated annealing algorithm, an improved HIA was proposed to solve the FJSP in this paper. The typical example of FJSP was simulated to show the feasibility and effectiveness of HIA. The effectiveness was verified through comparing the proposed HIA with other existing methods. The objective function of makespan was mainly considered in this paper so the further research work should be to apply the algorithm to the multi-FJSP.

Acknowledgments This work is partially supported by the Talented Young Scholars Growth Plan of Liaoning Province Education Department, China (No. LJQ2013048), the Dr scientific research fund of Liaoning Province (No. 201601244) and the Project of Liaoning BaiQianWan Talents Program, China (No. 2014921062).

References

Bai DY, Zhang ZH, Zhang Q (2016) Flexible open shop scheduling problem to minimize makespan. Comput Oper Res 67:207–215

Brandimarte P (1993) Routing and scheduling in a flexible job shop by tabu search. Ann Oper Res 41:157–183

Gao KZ, Suganthan PN, Pan QK et al (2014) Pareto-based grouping discrete harmony search algorithm for multi-objective flexible job shop scheduling. Inf Sci 289(24):76–90

Geyik F, Dosdogru AT (2013) Process plan and part routing optimization in a dynamic flexible job shop scheduling environment an optimization via simulation approach. Neural Comput Appl 23(6):1631–1641

Gutierrez C, Garcia-Magario I (2011) Modular design of a hybrid genetic algorithm for a flexible job-shop scheduling problem. Knowl Based Syst 24:102–112

Li XY, Liang G (2016) An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem. Int J Prod Econ 174:93–110

Li XL, Lu JS, Chuai Guo-zhong GZ et al (2011) Hybrid bee colony algorithm for flexible Job Shop scheduling problem. Comput Integr Manuf Syst 17(7):1495–1500

Liu XB, JIAO X, Ning T et al (2015) Improved method of flexible Job Shop scheduling based on double chains quantum genetic algorithm. Comput Integr Manuf Syst 21(2):495–502

Martí Rafael, Pantrigo Juan-José, Duarte Abraham et al (2011) Scatter search and path relinking: a turorial on the linear arrangement problem. Int J Swarm Intell Res 2(2):1–21

Mokhtari H, Abadi INK, Cheraghalikhania A (2011) A multi-objective flow shop scheduling with resource-de-pendent processing times: trade-off between makespan and cost of resources. Int J Prod Res 49(19):5851–5875

Ning T (2013) Study of application of hybrid quantum algorithm in vehicle routing problem. Dalian Maritime University, Dalian

Ning T, Guo C, Chen R, JIN H (2016a) A novel hybrid method for solving flexible job-shop scheduling problem. Open Cyvernetics Syst J 10:13–19

Ning T, Huang M, Liang X, Jin H (2016b) A novel dynamic scheduling strategy for solving flexible job-shop problems. J Ambient Intell Humaniz Comput 7(5):721–729

Palacios JJ et al (2015) Genetic tabu search for the fuzzy flexible job shop problem. Comput Op Res 54:74–89

Prakash S, Vidyarthi DP (2014) A hybrid GABFO scheduling for optimal makespan in computational grid. Int J Appl Evol Comput 5(3):57–83

Rahmati SHM, Zandieh M, Yazdani M (2013) Developing two multi-objective evolutionary algorithms for the multi-objective flexible job shop scheduling problem. Int J Adv Manuf Technol 64(5):915–932

Tavakkoli-Moghaddam R, Azarkish M, Sadeghnejad-Barkousaraie A (2011) A new hybrid multi-objective pareto archive PSO algorithm for a bi-objective job shop scheduling problem. Expert Syst Appl 38(9):10812–10821

Xue HQ, Zhang P, Wei SM, Yang L (2014) An improved immune algorithm for multi-objective flexible job-shop scheduling. J Netw 9(10):2843–2850

Zhao SK, Fang SL, Gu XJ (2014) Machine selection and FJSP solution based on limit scheduling completion time minimization. Comput Integr Manuf Syst 20(4):854–865