An improved bacterial foraging optimization for multi-objective flexible job-shop scheduling problem

Bo Li[1,2], Chen Guo[1,*], Tao Ning[2]

1. Institute of Information, Dalian Maritime University, Dalian 116023, China

2. Institute of Software, Dalian Jiaotong University, Dalian 116045, China

daliannt@126.com

ABSTRACT. The purpose of this study is to solve the multi-objective flexible job shop scheduling problem (FJSP). An improved bacterial foraging optimization algorithm (IBFOA) based on adaptive step is proposed, which sets maximum makespan, the total goods as the optimization objectives. The results obtained in this study include the algorithm encodes based on the operation to make IBFOA be applicable to FJSP. A chemotaxis based on the crowding distance selection and adaptive step is put forward to improve the local search ability of BFOA. The impacts of the obtained results are the optimal bacterial individuals can preserve curing position to additional turning and make the common ones swim to the direction of them to absorb location information.

RÉSUMÉ. Le but de cette étude est de résoudre le problème de planification d'ateliers flexibles multi-objectifs (FJSP). Un algorithme d'optimisation améliorée de l'alimentation bactérienne (IBFOA) basé sur une étape adaptative est proposé, qui définit le makespan maximum, le total des biens comme objectifs d'optimisation. Les résultats obtenus dans cette étude incluent les algorithmes de codage basés sur l'opération visant à rendre l'IBFOA applicable aux FJSP. Une chimiotaxie basée sur la sélection de la distance d'encombrement et sur l'étape d'adaptation est proposée pour améliorer la capacité de recherche locale de BFOA. Les résultats obtenus ont pour impact que les individus bactériens optimaux peuvent conserver leur position de durcissement au retournement supplémentaire et permettre aux individus les plus communs de nager dans leur direction pour absorber les informations de localisation.

KEYWORDS: multi-objective flexible scheduling, bacteria foraging optimization algorithm, additional turning, multi-attribute grey target decision.

MOTS-CLÉS: planification flexible multi-objectifs, algorithme d'optimisation de l'alimentation des bactéries, retournement supplémentaire, décision de cible grise multi-attributs.

DOI:10.3166/JESA.51.323-332

1. Introduction

There are two sub problems in multi-objective FJSP such as machine assignment and the process sorting. The solution of the problem consists of two stages: the stage of optimal scheduling scheme and the decision stage of the scheduling scheme. It is known that the genetic algorithm (GA) does not rely on the specific area of the problems. Its coding and evolution process is so simple that it can search the parallel solution in the global solution space. For the above reasons, it has become the powerful tool for the job shop scheduling combinatorial optimization problems. However, this algorithm also has some shortcomings such as premature convergence and low searching efficiency in later. Ju and Zhu (2007) proposed a hybrid genetic algorithm combining the particle swarm optimization, which acted the production cycle and cost as the aim. Wu *et al.* (2006) (Vijaychakaravarthy *et al.,* 2014) proposed a hybrid genetic algorithm integrating the weight coefficient and niche technique. However, this method can't guarantee the optimality of non-dominated solution. The particle swarm optimization (PSO) is based on the information sharing of single particle and excellent individual. In which, the excellent individual can guide the evolution direction of the next generation. It has several advantages such as small population, simple calculation, high efficiency and strong robustness, and has achieved good application effect on solving job shop scheduling problem and multi-objective optimization Hao *et al.* (2013) proposed a machine selection approach to select the shorter working hour considering the load balance of machines Liu *et al.* (2015) proposed the double chains quantum coding, however, this approach is lack of considering the influence on process sequence because of the machine selection Geyik and Dosdoğru (2013) presented an optimization via simulation approach to solve dynamic flexible job shop scheduling problem (Teekeng and Thammano, 2012). The study deals with both determining the best process plan for each part and then finding the best machine for each operation in a dynamic flexible job shop scheduling environment. In this respect, a genetic algorithm approach is adapted to determine best part processing plan for each part and then select appropriate machines for each operation of each part according to the determined part processing plan. W. Teekeng proposed a modified version of the genetic algorithm for flexible job-shop scheduling problems (Ning *et al.*, 2016) J. Peng proposed a cloud model evolution algorithm for FJSP based on non-dominated sorting by introducing the cloud evolutionary strategy (Mahmood *et al.,* 2017). Literature (Moslehi and Mahnam, 2011; Demir and İşleyen, 2014; Ning *et al.,* 2016) expressed the particle as the priority level of the available machines and combined the Particle Swarm Optimization (PSO) with simulated annealing (SA) to solve the FJSP. But it adopted the weighted coefficient method to convert three objectives of FJSP into one, and can only get one optimal solution once not reflect the practical multi-objective problems. Literature (Prakash and Vidyarthi, 2014) used the equipment allocation rules and process scheduling strategy based on the priority to obtain the best local guidance with PSO.

At present, it is still in the exploratory stage of using BFOA to solve FJSP. Therefore, this paper proposes the IBFOA for multi-objective FJSP. The remaining section of the manuscript is explained as follows: the related work is delineated in

Section 2. Section 3 described the System investigated and the Result. Section 4 explains about the Discussion and Conclusion.

2. Experimental part

2.1. Description for FJSP

N: the total number of workpieces to be processed, M the total number of machines, i: the index of workpieces , $i \in \{1, 2, \ldots, N\}$; O_i :the i^{th} the workpiece, n_i the index of process, R_{ij} : the j^{th} process of workpiece O_i, M_{ij}: the machine set for R_{ij} , $M_{ij} \subseteq \{1, 2, \ldots, M\}$, m: the index of machine, $m \in \{1, 2, \ldots, M_{ij}\}$, S_{ijm}: R_{ij} can be processed on the machine m or not (1 or 0), t_{ijm}: the spent time for R_{ij} to be processed on the machine m, b_{ijm}: the start time for R_{ij} to be processed on the machine m, C_{ijm}: the completion time for $R_{ij.}$

2.2. Description of problem

The FJSP can be described as follow: there are N workpieces to be processed on M machines in the workshop, and each workpiece O_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 j^{th} ($j \in \{1, 2, \ldots, n_i\}$) procedure of workpiece O_i, M_{ij} ($M_{ij} \subseteq \{1, 2, \ldots, M\}$) is the machine set which can process the above working procedures, and R_{ij} can be processed by any machine m ($m \in \{1, 2, \ldots, M_{ij}\}$) with processing capability, besides the machine m has the ability to process $q \geq 1$ working procedures which belong to different workpiece .

2.3. Objective function

The primary goal for the production enterprises is to complete the production tasks timely and efficiently, therefore the primary scheduling goal for FJSP is to minimize makespan through selecting suitable machine for each process and arranging the optimal processing sequence. In this paper and several new workpieces are added after the initial scheduling, and the objective function is established as follows:

(1) Minimize the makespan $f1$

$$f1 = \min(C) = \min\{\max(C_i \mid i = 1, 2, \cdots, N)\}$$

$$C_{ijm} = S_{ijm} b_{ijm} + S_{ijm} t_{ijm} \tag{1}$$

In equation (1), C_i is the completion time of workpiece O_i, C_{iim} is the completion time for R_{ij} processed on machine m, b_{ijm} is the initial moment for R_{ij} to be processed on machine m, b_{ij}: is the start time for R_{ij}.

(2) Minimize the total load of all the machines $f2$

$$f2 = \min(\sum_{i=1}^{N}\sum_{j=1}^{n_i}\sum_{m=1}^{M} t_{ijm}S_{ijm}) \tag{2}$$

(3) Minimize the maximum load of single machine $f3$

$$f3 = \min[\max \sum_{i=1}^{N}\sum_{j=1}^{n_i} t_{ijm}S_{ijm}] \tag{3}$$

$$m = 1, 2, \cdots, M$$

2.4. Constraint conditions

(1) Constraint of sequence

Each process R_{ij} should start after the last one $R_{i(j-1)}$ has been completed, and the mathematical description is as follow:

$$\sum_{m=1}^{M} b_{ijm}S_{ijm} \geq \sum_{m=1}^{M}[(b_{i(j-1)m}t_{i(j-1)m})]S_{i(j-1)m} \tag{4}$$

In equation (4), S_{ijm}=$S_{i(j-1)m}$=1.

(2) Constraint of machine

Only one process can be processed on the same machine at the same moment, that is there is R_{ij} at moment t (t>0), if $\exists S_{ijm} = 1$, then S_{xym}=1will not set up ($i = x, j \neq y$).

(3) Constraint of continuity

R_{ij} cannot be interrupted after being processed:

$$C_{ijm} = \begin{cases} \max\{C_{i(j-1)m}, b_{ijm}\} + t_{ijm}, & j > 1; \\ b_{ijm} + t_{ijm}, & j = 1。 \end{cases} \tag{5}$$

2.5. The improved algorithm of IBFOA

FJSP is one kind of strong NP hard combinatorial optimization problems. The high dimension variables and complex constraints expand the solution space. According to the characteristics of FJSP, this paper proposes an improved differential evolution algorithm based on bacterial foraging optimization.

(1) IBFOA

$\theta^i(k,j,l)$: the position of the i^{th} bacteria in the k^{th} chemokine, the j^{th} replication and the l^{th} elimination $\lambda(i)$ the step of chemotaxis, $\varphi(i)$: the direction vector of unit length, $\Delta(i)$: the random vector, $\delta \in (0,1)$: the crowding distance factor, $f_\lambda f_\lambda f_\lambda$: the step adjustment factor, N_c :the number of chemotaxis, P_s: population size, d: the crowding distance, d=(N_c/P_s), I: the length of the search interval.

The innovation of IBFOA in this paper is as follows: (1) In order to avoid limiting the convergence, the variable step length strategy based on crowding distance is proposed in IBFOA. (2) In order to find out the local optimal position fully the multiple chemotaxis is implemented, that is, on the basis of the common individual turn, the optimal individual will carry out additional tumbling. (3) Normal individuals swim to the direction of the optimal ones to improve the search efficiency of the algorithm. The specific steps of the improved algorithm are as follows:

1) Initialization

Determine the parameter of individual bacteria, such as position, position size, the number of chemotaxis, reproductive and elimination. Build the mapping relationship between FJSP and IBFOA and use the encoding way based on working procedure. The problem of 3 workpiece ×4 machine is shown in Table 1, and the number is t_{ijm}.

Table 1. Example for 3×4

workpiece	process	machine			
		M_1	M_2	M_3	M_4
O_1	R_{11}	4	-	3	3
	R_{12}	5	4	-	6
	R_{13}	3	4	3	5
O_2	R_{21}	-	8	8	9
	R_{22}	3	3	-	-
O_3	R_{31}	4	-	3	3
	R_{32}	5	8	8	6
	R_{33}	9	8	-	26

2) Chemotaxis

a. Firstly, the unit vector is generated and the individual bacteria start to turn and swim according to the step size. In which, the position of the i^{th} bacteria is updated as Equation (6):

$$\theta^i(k+1,j,l) = \theta^i(k,j,l) + \lambda(i)\varphi(i) \tag{6}$$

$$\varphi(i) = \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{7}$$

The notations in equation (6) and equation (7) are as mentioned above.

In this paper, $\lambda(i)$ is improved adaptively. The equation of adjustable step is as follow:

$$\lambda(i) = f_\lambda[\frac{\delta(\delta+1)}{\delta+d} - \delta]I \tag{8}$$

In equation (8), if d is smaller, the individual will optimize in larger step; otherwise, the individual will optimize in a smaller step. That can ensure the algorithm has strong global search ability in early and strong local search ability in the latter.

b. Next, the bacterial community turns to update its position using approach of local searching. If the bacterial individual mapping to the multi-objective FJSP in Table 1 is “3 1 3 2 1 1 2 3” and select two point of $p1$ and $p2$ randomly, then the region between $p1$ and $p2$ of “3 2 1 1” will be stable and the information out of it will generate randomly, that is “2 3 3 2 1 1 3 1”. Compare the adaptive value of bacterial individuals before and after turning, if the fitness value appears to be backward, it is a non rational behavior.

After turning, the bacterial individuals with high adaptive value will obtain additional turning opportunities to search for more solutions in the neighborhood. The additional turning of the optimal individual is as follow: if the processing sequence is “3 1 2” before turning, the sequence will be retained as “3 1 2”.

After the local search of the optimal individual is completed, the common one may swim in the direction of the optimal one to absorb its positional information. If its value is “1”, the corresponding position will become “1”, and the original information of “3” will replace to the first position of “1”.

Offspring is selected through non-dominated sorting based on Pareto and crowding distance.

The non-dominated sorting achieves the classification through calculating the parameter X_N and X_n of population P_s, the specific steps are as follows:

Step 1: Initialize the parameter X_N as Ø;

Step 2: Initialize the variable X_n, where X_n is the number of individuals to dominate X;

Step 3: Calculate the dominance relationship between X and Y, X, $Y \in P_s$. If Y is dominated by X, then $X_N = X_N \cup \{Y\}$, otherwise, X_n= X_n +1. When X_n =0, X is considered to be non-dominated individuals, it is marked as X_r=1, then $R_1 = R_1 U\{X\}$;

Step 4: If $R_i \neq \emptyset$ then Q=Ø and set i=1. If $Y \in X_N$, set y_n= y_n -1 until y_n =0, then set Y_r=i+1, $Q = Q \cup \{Y\}$, R_i=Q;

Step 5: Calculation will stop if R_i is empty, otherwise, turn to Step 4.

3) Reproduction

If one bacterial individual absorbs enough nutrients to reproduce in the process of swimming, it will reach reproduction threshold. On the contrary, if the bacterial individual does not absorb enough nutrients, it will reach death threshold. Then the nutrient absorbed by the bacterial individual is calculated, and the excellent individuals in reproduction threshold will be copied until it reaches the predetermined times, go to (4), otherwise go to (2). Then the bacteria may be sequenced according to its adaptive value, which maps to the objective function in FJSP.

4) Elimination

Stop the algorithm if the bacteria colony has reach the predetermined times for elimination,

3. Results and discussion

3.1. Testing on kacem instance

Table 2. The result of Kacem instance with three algorithms

workpiece×machine	objective value	BFOA		HBCA			IBFOA			
		Su_1	Su_2	Su_1	Su_2	Su_3	Su_1	Su_2	Su_3	Su_4
4×5	T_x	11		11	12	13	11	11	12	11
	M_t	31		31	31	32	31	30	31	31
	M_x	9		10	8	7	9	8	8	8
8×8	T_x	15	15	14	15	15	14	15	14	14
	M_t	76	75	76	75	73	75	75	73	74
	M_x	12	12	12	12	13	12	12	12	11
10×7	T_x			12	11	12	12	11	11	
	M_t			61	62	60	60	61	60	
	M_x			11	11	12	11	10	12	
10×10	T_x	7		7	6	7	7	6	7	6
	M_t	43		41	42	41	41	42	41	41
	M_x	6		6	5	5	6	5	5	6

In order to verify the efficiency of the proposed method, this paper solved the classic four standard problems of Kacem (4 workpieces×5 machines, 8 workpieces×8 machines, 10 workpieces×7 machines, 10 workpieces×10 machines) by IBFOA, in which the objective was f_1, f_2 and f_3. The comparison with BFOA, Hybrid bee colony algorithm (HBCA) is shown in Table 2. It can be seen that IBFOA can not only obtain more non-dominated solutions but obtain the current optimal solution in Table 2. For case 10×7, although both HBCA and IBFOA have obtained three non-dominated solutions, the solution (12, 61, 11) obtained by HBCA is dominated by both (12, 60, 11) and (11, 61, 10) obtained by IBFOA, the solution (12, 60, 12) obtained by HBCA is dominated by (11, 61, 10) obtained by IBFOA, the solution (12, 60, 12) obtained by HBCA is dominated by both (12, 60, 11) and (11, 60, 12) obtained by IBFOA. The above can indicate that IBFOA is more effective than the existing algorithms.

In Table 2, Su_n (n=1, 2, 3, 4) is the different solution; T_x is the makespan of a certain process; M_t is the total load of machines; M_x is the maximum load of single machine.

3.2. Testing on benchmark

Table 3. The comparison of different algorithms for Benchmark

example	workpiece × machine	Sol_b	IBFOA	HBCA		BBEA		MSIM	
			T	Tx	C_{mr}(%)	Tx	C_{mr}(%)	Tx	C_{mr}(%)
Mk01	10×6	36	37	40	8.1	39	5.4	38	2.7
Mk02	10×6	24	24	26	8.3	25	4.2	24	0
Mk03	15×8	204	204	204	0	204	0	204	0
Mk04	15×8	48	55	60	9.1	60	9.1	54	-1.8
Mk05	15×4	168	169	173	2.4	172	1.8	171	1.2
Mk06	10×15	33	50	63	26	58	16.0	56	12.0
Mk07	20×5	133	133	140	5.3	138	3.8	136	2.3
Mk08	20×10	523	523	523	0	523	0	523	0
Mk09	20×10	299	306	312	2.0	310	1.3	308	0.65
Mk10	20×15	165	169	194	14.8	198	17.2	175	3.6

In order to further verify the performance of IBFOA, this paper applied it to the Benchmark case and complied it with the HBCA algorithm, BBEA (bi-population based estimation algorithm) and MSIM (machine selection initialization method) [9]. Sol_b in Table 3 is the known optimal solution. It can be seen from Table 3 that the optimal solution has been obtained for four times in ten cases with the IBFOA. Except example Mk04, it has obtained a better or equal solution than other three algorithms in the other nine cases. "C_{mr}" is the improvement rate of IBFOA comparing with other algorithms, T and T_x are shown in Table 3.

$$C_{mr} = \frac{T_x - T}{T} \times 100\% \tag{11}$$

The average of C_{mr} is 7.6%, 5.9% and 2.05% respectively.

4. Conclusions

IBFOA based on differential evolution is proposed in this paper, and the algorithm includes chemotaxis, reproduction and elimination. The MAGTD is introduced when the adaptive variable step adjustment strategy is adopted to select the optimal solution, the following conclusion may be got:

(1) The IBFOA has goodly global and local ability and can obtain more non-dominated solutions than other algorithms;

(2) The IBFOA can obtain the optimal solution more quickly than other algorithms and the efficiency of algorithm is improved;

(3) The feasibility of the algorithm and its strong ability of solving can be verified through the benchmark;

(4) The introduction of MAGTD guarantees the selection of the most satisfactory scheduling scheme of FJSP in actual production.

However, the proposed research has not considered the influence of the subjective preference of decision makers on the final scheduling results, which may make the following research have more practical value. Therefore, the future development paths should be adding the analysis of subjective factor with some advanced approach such as cloud computing technology.

Acknowledgements

This work is financially supported by Dr scientific research fund of Liaoning Province (No. 20170520229, 20180550499), and Social Science planning fund of Liaoning Province (L18BGL018, 2018lslktyb-016).

References

Demir Y., İşleyen S. K. (2014). An effective genetic algorithm for flexible job-shop scheduling with overlapping in operations. *International Journal of Production Research*, Vol. 52, No. 13, pp. 3905. https://doi.org/10.1080/00207543.2014.889328

Geyik F., Dosdoğru A. T. (2013). Process plan and part routing optimization in a dynamic flexible job shop scheduling environment: An optimization via simulation approach. *Neural Computing & Applications*, Vol. 23, No. 6, pp. 1631. https://doi.org/10.1007/s00521-012-1119-1127

Hao X., Lin L., Gen M., Ohno K. (2013). Effective estimation of distribution algorithm for stochastic job shop scheduling problem. *Procedia Computer Science*, Vol. 20, No. 1, pp. 102. https://doi.org/10.1016/j.procs.2013.09.246

Ju Q. Y., Zhu J. Y. (2007). Multi-objective flexible Job Shop scheduling of batch production. *Chinese Journal of Mechanical Engineering*, Vol. 43, No. 8, pp. 148-154. https://doi.org/10.21474/IJAR01/6308

Liu X. B., Jiao X., Ning T. (2015). Improved method of flexible job shop scheduling based on double chains quantum genetic algorithm. *Computer Integrated Manufacturing Systems*, Vol. 21, No. 2, pp. 495-502. https://doi.org/10.16356/j.1005- 2615.2017.06.004

Mahmood M., Mostafa T., Najafi R. M. (2017). Kinematic analysis and design of a 3-DOF translational parallel robot. *International Journal of Automation and Computing*, Vol. 14, No. 4, pp. 432-441. https://doi.org/10.1142/S0217984918401127

Moslehi G., Mahnam M. (2011). A pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search. *International Journal of Production Economics*, Vol. 129, No. 1, pp. 14-22. https://doi.org/10.1016/j.ijpe.2010.08.004

Ning T., An S. Y., Li X. X. (2016). Triaxial models description of the low-lying properties in 192Os. *International Journal of Modern Physics E*, Vol. 25, pp. 237. https://doi.org/10.1142/S021830131650083X

Ning T., Huang M., Liang X. (2016). A novel dynamic scheduling strategy for solving flexible job-shop problems. *Journal of Ambient Intelligence and Humanized Computing*, Vol. 7, pp. 721-729. https://doi.org/10.1007/s12652-016-0370-7

Prakash S., Vidyarthi D. P. (2014). A hybrid GABFO scheduling for optimal makespan in computational grid. *International Journal of Applied Evolutionary Computation,* Vol. 5, pp. 167-174. https://doi.org/10.4018/ijaec.2014070104

Teekeng W., Thammano A. (2012). Modified genetic algorithm for flexible job-shop scheduling problems. *Procedia Computer Science*, Vol. 12, pp. 122. https://doi.org/10.1016/j.procs.2012.09.041

Vijaychakaravarthy G., Marimuthu S., Naveen S. A. (2014). Comparison of improved sheep flock heredity algorithm and artificial bee colony algorithm for lot streaming in m-machine flow shop scheduling. *Arabian Journal for Science and Engineering*, Vol. 39, pp. 4285-4300. https://doi.org/10.1007/s13369-014-0994-x

Wu X. L., Sun S. D., Yu J. J. (2006). Research on multi-objective optimization for flexible Job Shop scheduling. *Computer Integrated Manufacturing System*, Vol. 12, No. 5, pp. 731-736. https://doi.org/10.1016/S1872-2040(06)60029-7