Application of Improved Simulated Annealing Particle Swarm Optimization in Displacement Flow Shop Scheduling

Dianna Song

School of Shandong Normal University, Jinan 250000, China

568875103@qq.com

Abstract

This paper proposes an improved simulated annealing particle swarm optimization algorithm for particle swarm optimization, and apply it in solving the scheduling problem of permutation flow shop and no-wait flow shop. In the process of solving the problem of permutation flow shop, given that the simulated annealing algorithm has a strong local Expand the search ability, can jump out of the local optimum, the annealing strategy is embedded in the population particle update process, constitute the simulated annealing particle swarm optimization, by optimizing the population of the optimal solution, the particle swarm optimization is limited by a local optimal In the process of optimization, three kinds of neighborhood search mechanisms of exchange, insertion and reverse order are adopted. The resulting solutions are selected according to the Metropolis acceptance criterion. The improved algorithm is used to solve the mathematical model of the replacement flow shop.

Keywords

Simulated annealing algorithm; Particle swarm optimization algorithm; Flow shop scheduling.

1. Introduction

Particle swarm optimization is a population-based intelligent optimization algorithm, which adopts multi-point simultaneous search. The particles in different populations interact with each other and the convergence speed is fast. It is easy to implement and does not need to adjust too many parameters and is widely used in the adjustment of production activities. Particle swarm optimization is easily trapped in the local optimal predicament. The Simulated Annealing Algorithm (SA) was proposed by Metropolis in 1953. After continuous improvement research, Kirkpatric firstly introduced the simulated annealing algorithm to solve the optimal combinatorial problem in the 1980s. The algorithm solves the problem of optimal combination by changing the temperature To control the progress of the algorithm. According to the Metropolis acceptance criterion update, mainly to search the local range, researchers have been NP-hard, artificial intelligence, pattern and image processing, optimization combination, signal processing and other fields .

Simulated annealing Particle swarm optimization (PSO) is to introduce simulated annealing algorithm into the iterative process of particle swarm position and velocity. The iterative process is changed regularly so that the algorithm departs from the original process. By using the local search ability of simulated annealing algorithm and getting rid of the local The optimal characteristics make up for the shortcomings of particle swarm optimization, and make the improved algorithm have strong ability to solve the problem. How to effectively combine the two has become a research hotspot. The main research in this paper is how to combine the two effectively to solve the scheduling problem of displacement flow shop.

2. An Overview of Displacement Flowshop Scheduling Problem

Displacement flow shop scheduling is one of the most widely studied problems in assembly line production scheduling. It is a simplified model for many assembly line manufacturing enterprises , which has high industrial application value. Although the process of replacing process flow shop is relatively simple, When the number of machines is more than 3, it becomes a typical NP-hard problem. Therefore, the research of the replacement flow shop scheduling problem has been a hot topic in the field of scheduling. For the size of a flow shop which is $n\times m$, the number of all feasible solutions is $(n!)^m$. However, for a replacement flow shop of the same size, since all the processes must be sorted on all machines in the same order, the number of all feasible solutions becomes $n!$, for the flow shop scheduling Just a condition will make the problem a lot less complex.

Displacement flow shop scheduling model is a simplified model of flow shop scheduling, requiring all processes to be sorted on all machines in the same order. It can be described as follows: n workpieces $\{J_1,J_2,J_3 \dots J_n\}$are machined on the machine in the same order and the workpieces must pass through the m machines $\{m_1 m_2 \dots m_m\}$. The processing time $t(i,\ J_j)$ of the workpiece i on the machine is definite, and the processing order of the workpieces on all the machines must be the same, and the following constraints are met: Only one workpiece can be processed on any one machine. Two workpieces can not be machined at the same time. Can follow the production process on a machine for processing can not be processed on both machines at the same time; to ensure the performance of the machine running well without failure; only consider the processing time of the workpiece on the machine, ignore the other time; the workpiece must be completed a process can start the next process. For the scheduling objectives can be intuitively represented by Gantt chart must ensure that all parts of the workpiece in all machines must be processed in the same order, for the scale 4×3 of the replacement flow shop scheduling goals as shown in Fig.1.

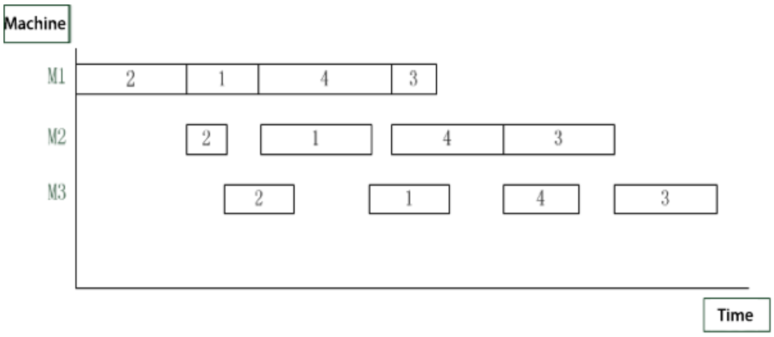

Fig.1 Replacement Gantt chart of the flow shop

The goal of scheduling is to determine the order in which the workpieces are machined so that the final completion time is minimized, usually number of workpieces, number of machines, replacement flow and completion time are expressed in quaternary terms $n/m/F/C_{max}$, which is described mathematically as follows:

$$C(1,J_1) = t(1,J_1)\ ; \tag{1}$$

$$C(i,J_1) = C(i\text{-}1,J_1) + t(1,J_1),\ (i = 2,3,4 \dots, n) \tag{2}$$

$$C(1,J_j) = C(i,J_{j\text{-}1}) + t(1,J_j),\ (j = 2,3,4 \dots, m) \tag{3}$$

$$C(i,J_j) = max[C(i\text{-}1,J_j), C(i,J_{j\text{-}1})] + t(i,\ J_j)(i = 2,3,4 \dots, n,\ j = 2,3,4 \dots, m) \tag{4}$$

Scheduling goals are:

$$C_{max} = C(n, J_m) \tag{5}$$

Which $C(i, J_j)$ represents the completion of the workpiece i on the machine J_j. $C(n, J_m)$ indicates that the completion time is the completion time of the last job on the last machine.

The main objective of this article is based on minimizing the maximum completion time, which is one of the most common scheduling objectives, the shorter the maximum completion time, the higher the utilization of equipment, the short production cycle and the production of more products in a certain period of time , Optimize the production structure of enterprises and enhance the profits of enterprises.

3. Improved Simulated Annealing Particle Swarm Optimization

3.1 The Basic Process

The basic principle of simulated annealing particle swarm algorithm is to combine the characteristics of particle swarm optimization and simulated annealing algorithm. The specific algorithm steps can be described as follows:

Step1: Initialize the control parameters. Population Initialization: In order to control the effect of the initial population on the algorithm, a small part of the population initialization process will be adjusted. The specific process will be introduced below. The parameters of inertial acceleration factor, self-awareness and social cognition are set, simulated annealing Cooling schedule.

Step2: Obtained the fitness of each particle according to the objective function.

Step3: Select the population of the optimal particle swarm as the annealing solution of the initial solution.

Step4:Local search of new solutions, selection according to Metropolis acceptance criteria, repeated L times.

Step5: Use the cooling schedule to cool the temperature to determine whether to meet the termination conditions, if satisfied, skip to Step6, otherwise continue to Step4.

Step6: The updated particle is brought into an updated formula for particle velocity and position to get new particle position and velocity information.

Step7: Observe if the termination condition is satisfied, and if yes, output the optimal solution; otherwise, return to Step2 to recalculate.

3.2 Local Search Strategy

It is found through practice that although particle swarm optimization has a strong capability of expanding search space, the algorithm can easily fall into the local optimum in the late iteration, the evolution speed obviously drops, and even the algorithm evolution ceases. Therefore, the simulated annealing algorithm Better local search ability to overcome this flaw. Simulated annealing algorithm usually adopts the way of real number coding, which is a discrete intelligent optimization algorithm.

Particle swarm optimization is based on particle swarm optimization and particle swarm optimization algorithm. Particle swarm optimization algorithm is based on particle swarm optimization (PSO) algorithm. In this paper, the new solution to the simulated annealing algorithm is generated by using three kinds of local search mechanisms: exchange method, interpolation method and reverse order method. When the local search mechanism is used, the position of the particles is directly changed locally to generate a new scheduling solution. For example, Particles can be expressed as $X_i = (x_{i1}, x_{i2}, \dots x_{id})$,a direct search of the local search mechanism for each dimension of the vector, the change is mapped to the scheduling scheme. Although the ultimate goal is to search the scheduling scheme locally, if the local search of the scheduling scheme is adopted, the adjustment process needs to be transformed into the population by the anti-coding method, and the local searching mechanism of the process relative to the particle location is relatively complicated. Therefore, This article is not suitable for the use of local search results scheduling.

4. Simulation

4.1 Lab Environment

In order to facilitate the programming, display the results of the algorithm, all the experiments in this paper are running in the version of matlab7.0, using matlab language, the same HP machine running under the same environment, the machine is configured for Win7 operating system, The processor is Intel i3, 2.10GHZ, video card size is 1G, memory size is 2G.

4.2 Experimental parameters

In the simulation experiment, we use the improved algorithms of insert, exchange and reverse order respectively to solve the car class and Taillard class with different scales, and compare them with the NEH algorithm and the unmodified PSO algorithm.

The setting of the parameters in the algorithm has great influence on the effect of the algorithm. By consulting a large amount of literature and carrying out a simple experimental test, the parameters in the simulated annealing particle swarm optimization are set as follows: the initial temperature of the simulated annealing is set to be $T_0 = 1000$. Take the cooling method $\alpha = 0.8$, the number of iterations of the algorithm at each temperature $L = 20$; the termination temperature is $T_{end} = 10$;the size of the population in the PSO $ps = 50$, in order to make the algorithm have the ability to expand the search range in the initial stage, the inertia factor is updated by the following formula $\omega(t+1) = \beta\omega(t)$.The initial value of ω is 0.9 and β is 0.8, the acceleration parameter C_1, $C_2$is set to the same number$C_1 = C_2 = 2$; the iteration number of particle swarm algorithm is 300; each algorithm runs independently 10 times in order to ensure the fairness of results.

4.3 Initialization of experimental values

For the first step of large-scale simulated annealing particle swarm optimization algorithm, we need to generate the population first, and the selection of the population will determine the convergence speed of the particle swarm optimization algorithm. In the initial population, Heuristic algorithms are usually used to initialize the set population. The heuristic algorithms include the Palmer method, the Gupta method, the CamPleu-Dudek-Smith (CDS) method, the Rapid Access (RA) method and the Nawaz-Enscore-Ham Compared with other heuristics, NEH algorithm is superior to other heuristics in solving the scheduling problem of permutation flow shop. The NEH algorithm mainly determines the production priority through the total processing time of the workpiece, and has better performance. In this paper, we use NEH to realize one particle in the population, and the remaining particles will be randomly generated within the set range.

4.4 Experimental results and analysis

The results obtained after the algorithm is implemented are shown in Table 1. SPSO-SWAP represents the exchange-based simulated annealing particle swarm optimization algorithm, SPSO-INSERT, which represents the insertion-based simulated annealing particle swarm optimization algorithm. SPSO-INVERSE represents the reverse order based simulated annealing particle swarm optimization algorithm. MAX represents the upper limit of the algorithm, and MIN represents the best fitness after the algorithm is executed, that is, the minimum time required to complete the maximum completion of the algorithm, indicating the average of the algorithm, STD indicating the standard deviation of the algorithm, C * is the minimum.

Table 1 NEH Algorithm, PSO Algorithm and Three Improved Algorithm comparing

problem	Size	C^*	NEH		MAX	MIN	AVG	STD
TA001	20×5	1278	1381					
				PSO	1378	1297	1302.3	7.51
				SPSO-SWAP	1370	**1278**	1287.7	26.84
				SPSO-INSERT	1351	**1278**	1285.5	10.06
				SPSO-INVERSE	1371	**1278**	1289.5	14.83
TA002	20×5	1359	1458					
				PSO	1435	1366	1370.7	10.11
				SPSO-SWAP	1420	**1359**	1366.9	7.42
				SPSO-INSERT	1389	**1359**	1362	5.35
				SPSO-INVERSE	1432	**1359**	1365.5	6.96
TA003	20×5	1081	1213					
				PSO	1213	1088	1116	23.27
				SPSO-SWAP	1213	1086	1099.9	33.90
				SPSO-INSERT	1213	**1081**	1098.7	22.31
				SPSO-INVERSE	1213	1088	1109.2	.2167
TA004	20×5	1293	1488					
				PSO	1488	1308	1329	22.79
				SPSO-SWAP	1477	**1293**	1318.4	31.73
				SPSO-INSERT	1480	1297	1313.7	24.04
				SPSO-INVERSE	1488	1296	1319.6	28.17
TA005	20×5	1235	1381					
				PSO	1381	1250	1295.5	18.09
				SPSO-SWAP	1376	**1235**	1242.8	18.8
				SPSO-INSERT	1359	**1235**	1241	16.39
				SPSO-INVERSE	1338	**1235**	1247.9	18
TA006	20×5	1195	1385					
				PSO	1364	**1195**	1214.2	21.85
				SPSO-SWAP	1314	**1195**	1209.3	15.81
				SPSO-INSERT	1379	**1195**	1200.1	14.89
				SPSO-INVERSE	1336	**1195**	1200.4	22.65
TA007	20×5	1234	1360					
				PSO	1360	1239	1258.9	12.81
				SPSO-SWAP	1360	1239	1251.6	13.92
				SPSO-INSERT	1352	**1234**	1243.7	11.33
				SPSO-INVERSE	1355	**1234**	1252.3	12.69
TA008	20×5	1206	1401					
				PSO	1401	1211	1255	18.52
				SPSO-SWAP	1389	**1206**	1217.9	24.17
				SPSO-INSERT	1330	**1206**	1220.9	18.57
				SPSO-INVERSE	1355	**1206**	1227.2	22.67

TA009	20×5	1230	1366					
				PSO	1366	**1230**	1256.2	36.16
				SPSO-SWAP	1350	1233	1253.8	16.14
				SPSO-INSERT	1366	**1230**	1256	24.63
				SPSO-INVERSE	1366	1233	1243.2	21.33
TA010	20×5	1108	1264					
				PSO	1264	1111	1125.3	17.94
				SPSO-SWAP	1264	**1108**	1121.4	26.47
				SPSO-INSERT	1264	**1108**	1122.9	22.46
				SPSO-INVERSE	1255	1111	1121	22.46
TA011	20×10	1582	1798					
				PSO	1798	1610	1625.7	34.68
				SPSO-SWAP	1783	1593	1605.4	33
				SPSO-INSERT	1798	**1583**	1609.1	30.55
				SPSO-INVERSE	1798	1586	1608.9	33.96
TA012	20×10	1659	1907					
				PSO	1907	1691	1711.1	36.26
				SPSO-SWAP	1907	**1659**	1709.6	28.25
				SPSO-INSERT	1907	**1659**	1700	27.22
				SPSO-INVERSE	1807	**1659**	1689	29.63
Car1	20×10	7038	8243					
				PSO	7898	**7038**	7042.1	53.5
				SPSO-SWAP	7472	**7038**	7043	26.9
				SPSO-INSERT	7757	**7038**	7040	25.6
				SPSO-INVERSE	7756	**7038**	7046	27.8
Car2	13×4	7166	8458					
				PSO	8230	**7166**	7181.2	97.2
				SPSO-SWAP	7867	**7166**	7171	46.9
				SPSO-INSERT	8142	**7166**	7171	68.8
				SPSO-INVERSE	7968	**7166**	7176.6	69.9
Car3	12×5	7312	8126					
				PSO	8126	**7312**	7340.5.	55.9
				SPSO-SWAP	7747	**7312**	7328.3	65.6
				SPSO-INSERT	8126	**7312**	7318.7	44.6
				SPSO-INVERSE	7590	**7312**	7333.6	54.6

From the above table, we can see that the improved algorithm of the three strategies and the original particle swarm optimization algorithm can solve the optimal solution to the small scale problem of car, while for the slightly larger TA class, three improvements Particle Swarm Optimization (PSO) shows better performance than the original Particle Swarm Optimization (PSO). For all problems, the optimal solution can be obtained from the known literature, which shows that the improvement of this paper is still more effective. In addition, Among the three kinds of local search strategies, the performance based on the inserted local search strategy shows the best performance, and the better optimal solution can be obtained every time.

5. Conclusion

This paper first describes the problem of permuted flow shop scheduling, mathematically modeling it, followed by a brief introduction to the simulated annealing algorithm used to improve the particle swarm, and describes the process of improvement, using three kinds of local search mechanism to produce The results show that the performance of the simulated annealing particle swarm optimization algorithm is superior to the original particle swarm optimization algorithm, and it can solve the scheduling problem of the replacement flow shop well.

Acknowledgements

This work was supported by the Natural Science Foundation of China (No.61502283 and No.61472231).

References

[1] Rahman H F, Sarker R, Essam D. A genetic algorithm for permutation flow shop scheduling under make to stock production system [J]. Computers & Industrial Engineering, 2015, 90: 12-24.
[2] Garey M R, Johnson D S, Sethi R. The complexity of flowshop and job shop scheduling[J]. Mathematics of operations research, 1976, 1(2): 117-129.
[3] Kulkarni M N K, Patekar M S, Bhoskar M T, et al. Particle Swarm Optimization Applications to Mechanical Engineering-A Review[J]. Materials Today: Proceedings, 2015, 2(4-5): 2631-2639.
[4] Hanafi I, Cabrera F M, Dimane F, et al. Application of Particle Swarm Optimization for Optimizing the Process Parameters in Turning of PEEK CF30 Composites[J]. Procedia Technology, 2016, 22: 195-202.
[5] Kennedy J, Eberhart R. Particle swarm optimization, proceedings of IEEE International Conference on neural networks[J]. 1995. 1942-1948.
[6] Shi Y, Eberhart R. A modified particle swarm optimizer[C]//Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Conference on. IEEE, 1998: 69-73.
[7] Clerc M, Kennedy J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space[J]. IEEE transactions on Evolutionary Computation, 2002, 6(1): 58-73.
[8] Turgut O E. Hybrid chaotic quantum behaved particle swarm optimization algorithm for thermal design of plate fin heat exchangers[J]. Applied Mathematical Modelling, 2016, 40(1): 50-69.
[9] Zhang L, Tang Y, Hua C, et al. A new particle swarm optimization algorithm with adaptive inertia weight based on Bayesian techniques[J]. Applied Soft Computing, 2015, 28: 138-149.
[10] Kundu R, Das S, Mukherjee R, et al. An improved particle swarm optimizer with difference mean based perturbation[J]. Neurocomputing, 2014, 129: 315-333.
[11] Wang Y, Li B, Weise T, et al. Self-adaptive learning based particle swarm optimization[J]. Information Sciences, 2011, 181(20): 4515-4538.
[12] Ali A F, Tawhid M A. A hybrid particle swarm optimization and genetic algorithm with population partitioning for large scale optimization problems[J]. Ain Shams Engineering Journal, 2016.
[13] Cheng C Y, Chen Y Y, Chen T L, et al. Using a hybrid approach based on the particle swarm optimization and ant colony optimization to solve a joint order batching and picker routing problem[J]. International Journal of Production Economics, 2015, 170: 805-814.
[14] Jiang S, Ji Z, Shen Y. A novel hybrid particle swarm optimization and gravitational search algorithm for solving economic emission load dispatch problems with various practical constraints[J]. International Journal of Electrical Power & Energy Systems, 2014, 55: 628-644.
[15] Yang K, Liu Y, Yang G. An improved hybrid particle swarm optimization algorithm for fuzzy p-hub center problem[J]. Computers & Industrial Engineering, 2013, 64(1): 133-142.

[16] Xia X, Liu J, Hu Z. An improved particle swarm optimizer based on tabu detecting and local learning strategy in a shrunk search space[J]. Applied Soft Computing, 2014, 23: 76-90.
[17] Xu L, Wang J, Li Y, et al. Resource allocation algorithm based on hybrid particle swarm optimization for multiuser cognitive OFDM network[J]. Expert Systems with Applications, 2015, 42(20): 7186-7194.
[18] Kaveh A, Bakhshpoori T, Afshari E. An efficient hybrid particle swarm and swallow swarm optimization algorithm[J]. Computers & Structures, 2014, 143: 40-59.
[19] Samarghandi H. A particle swarm optimization for the no-wait flow shop problem with due date constraints[J]. International Journal of Production Research, 2015, 53(9): 2853-2870.
[20] Chen C L, Huang S Y, Tzeng Y R, et al. A revised discrete particle swarm optimization algorithm for permutation flow-shop scheduling problem[J]. Soft Computing, 2014, 18(11): 2271-2282.
[21] Zhao F, Tang J, Wang J, et al. An improved particle swarm optimization with a linearly decreasing disturbance term for flow shop scheduling with limited buffers[J]. International Journal of Computer Integrated Manufacturing, 2014, 27(5): 488-499.
[22] Moslehi G, Mahnam M. A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search[J]. International Journal of Production Economics, 2011, 129(1): 14-22.
[23] Nourali S, Imanipour N. A particle swarm optimization-based algorithm for flexible assembly job shop scheduling problem with sequence dependent setup times[J]. ScientiaIranica. Transaction E, Industrial Engineering, 2014, 21(3): 1021.
[24] Liu B,WangL,Jin Y.H. An effective hybrid PSO-based algorithm for flow shop scheduling with limited buffers[J]. Computers Operations Research,2008,35(9):2791-2806.
[25] Lian Z, Gu X, Jiao B. A novel particle swarm optimization algorithm for permutation flow-shop scheduling to minimize makespan[J]. Chaos, Solitons& Fractals, 2008, 35(5): 851-861.
[26] Gandomi A H, Yun G J, Yang X S, et al. Chaos-enhanced accelerated particle swarm optimization[J]. Communications in Nonlinear Science and Numerical Simulation, 2013, 18(2): 327-340.
[27] Yin P Y. A discrete particle swarm algorithm for optimal polygonal approximation of digital curves[J]. Journal of Visual Communication & Image Representation, 2004, 15(2):241-260.
[28] Rameshkumar K, Suresh R K, Mohanasundaram K M. Discrete Particle Swarm Optimization (DPSO) Algorithm for Permutation Flowshop Scheduling to Minimize Makespan[J]. Lecture Notes in Computer Science, 2005, 3612:572-581.
[29] Aminbakhsh S, Sonmez R. Discrete particle swarm optimization method for the large-scale discrete time–cost trade-off problem[J]. Expert Systems with Applications, 2016, 51: 177-185.
[30] Lin Q, Gao L, Li X, et al. A hybrid backtracking search algorithm for permutation flow-shop scheduling problem[J]. Computers & Industrial Engineering, 2015, 85: 437-446.
[31] Liu B, Wang L, Jin Y H. An effective PSO-based memetic algorithm for flow shop scheduling[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2007, 37(1): 18-27.
[32] Low, C. ‘‘Simulated annealing heuristic for flow shop scheduling problems with unrelated parallel machines’’, Comput. Oper.2005, Res., 32(8): 2013–2025.
[33] Mousavi S M, Tavakkoli-Moghaddam R. A hybrid simulated annealing algorithm for location and routing scheduling problems with cross-docking in the supply chain[J]. Journal of Manufacturing Systems, 2013, 32(2): 335-347.
[34] Akram K, Kamal K, Zeb A. Fast simulated annealing hybridized with quenching for solving job shop scheduling problem[J]. Applied Soft Computing, 2016, 49: 510-523.
[35] Vincent F Y, Lin S W, Lee W, et al. A simulated annealing heuristic for the capacitated location routing problem[J]. Computers & Industrial Engineering, 2010, 58(2): 288-299.
[36] Dai M, Tang D, Giret A, et al. Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm[J]. Robotics and Computer-Integrated Manufacturing, 2013, 29(5): 418-429.

[37] Jamili A, Shafia M A, Tavakkoli-Moghaddam R. A hybrid algorithm based on particle swarm optimization and simulated annealing for a periodic job shop scheduling problem[J]. The International Journal of Advanced Manufacturing Technology, 2011, 54(1-4): 309-322.
[38] HamedSamarghandi, Tarek Y. ElMekkawy. A meta-heuristic approach for solving the no-wait flow-shop problem[J]. International Journal of Production Research, 2012, volume 50(24):1-14.
[39] Nagano M S, Da Silva AA, Lorena L A N. A new evolutionary clustering search for a no-wait flow shop problem with set-up times[J]. Engineering Applications of Artificial Intelligence, 2012, 25(6): 1114-1120.
[40] Meng A, Li Z, Yin H, et al. Accelerating particle swarm optimization using crisscross search[J]. Information Sciences, 2016, 329: 52-72.