Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 15 (2011) 3678 – 3683

Procedia Engineering

www.elsevier.com/locate/procedia

Advanced in Control Engineeringand Information Science

A Hybrid Algorithm for Flexible Job-shop Scheduling Problem

Jianchao Tang[a,b,*], Guoji Zhang[a], Binbin Lin[a], Bixi Zhang[c]

[a]South China University of Technology, Guangzhou,Guangdong,China, 510641
[b]South China Normal University, Guangzhou, Guangdong, China, 510631
[c] Guangdong University of Technology Guangdong, Guangzhou, Guangdong,China, 510006

Abstract

By combining the chaos particle swarm optimization with genetic algorithm, a hybrid algorithm is proposed in this paper. A novel initialization method is proposed based on the improved Kacem assignments scheme. And according to the characteristic of flexible job-shop scheduling problem, genetic operators are presented. Finally, this method is validated on a series of benchmark datasets. Experimental results indicate that this method is efficient and competitive compared to some existing methods.

Keywords: Flexible job-shop scheduling problem; Chaos particle swarm optimization; Genetic algorithm

1. Introduction

The flexible job shop scheduling problem (FJSP) is a generalization of the classical job shop problem. Each operation can be processed on a given machine chosen among a finite subset of candidate machines. The aim is to find an allocation for each operation and to define the sequence of operations on each machine to minimize the makespan.

For its strongly NP-hard nature [1], many effecient heuristics and meta-heuristics methods are developed to get nearly optimal solutions. A knowledge-based ant Colony Optimization algorithm is developed by Li-Ning Xing (2009) [2].A double-layer evolution scheduling algorithm with frame population space and belief space to solve FJSP was proposed by Tie-Ke Li (2010)[3]. A new hybrid swarm intelligence algorithm to solve the job-shop scheduling problem is propose by Tsung-Lieh Lin(2010)[4] .And a novel hybrid tabu search algorithm with a fast public critical block neighborhood structure is presented by Jun-Qing Li (2011)[5] . In these heuristics methods, GA is popular in the field of production scheduling for its potential parallelism and robustness. However, Genetic algorithm convergence rate is slow, how to solve FJSP efficiently with GA is still regarded as a challenging task. Particle swarm is faster and cheaper but easy to fall into local extremum.

An improved Kacem assignments scheme is adopted here for better initialization. To overcome GA and PSO's defects, chaotic optimization method, which is a new systemic random searching method with ergodicity is applied. And self-adaptive parameter setting strategy is integrated to improve the solution's quality and efficiency.

This article is organized as follows. Section 2 introduces the FJSP. Section 3 presents the hybrid algorithm. In Section 4, a comparative evaluation of the effectiveness and efficiency of the proposed algorithm is given. Finally, Section 5 presents the conclusions.

1877-7058 © 2011 Published by Elsevier Ltd. Open access under CC BY-NC-ND license.
doi:10.1016/j.proeng.2011.08.689

2. Problem description

The FJSP could be described as follows: Let $J = \{J_1, J_2, ..., J_n\}$ be a set of n jobs, which are carried out by m machines $M = \{M_1, M_2, ..., M_m\}$. Each job $J_i (1 \le i \le n)$ consists of a sequence of n_i operations O_{ij} $(j = 1, 2, ..., n_i)$, each operation O_{ij} is to be processed on a machine selected from a set of available machines. The assignment of the operation O_{ij} to the machine $M_k (1 \le k \le m)$ entails the occupation of the latter one during a processing time, denoted t_{ijk}. If $t_{ijk} = 0$, the operation O_{ij} could not select the machine M_k. In this paper, the objective is to minimize the makespan, and the hypotheses considered in this paper are summarized as follows:

- Each machine can process only one operation at a time;
- Each operation can be processed without interruption on one of a set of available machines;
- Job are independent and no priorities are assigned to any job type;
- All job are released at time 0, and all machines are available at time 0 too;
- Breakdowns are not considered;
- The order of operations for each job is predefined and cannot be modified.

Table 1 shows an instance of the completely flexible job shop scheduling problem with 3 jobs, 4 machines and 8 operations.

Table 1. An instance with 3 jobs, 4 machines and 8 operations

	M_1	M_2	M_3	M_4		M_1	M_2	M_3	M_4		M_1	M_2	M_3	M_4
O_{11}	7	6	4	5	O_{21}	2	5	1	3	O_{31}	8	6	3	5
O_{12}	4	8	5	6	O_{22}	4	6	8	4	O_{32}	3	5	8	3
O_{13}	9	5	4	7	O_{23}	9	7	2	2					

The FJSP consists of two sub-problems. The first one is to assign each operation to a machine out of a set of capable machines, and the second one deals with sequencing the assigned operations on all machines.

3. A hybrid algorithm for FJSP

3.1. Coding

Considering that the FJSP comprises two sub-problems, the feasible coding of FJSP embodies two parts. Fig. 1 shows a feasible solution for the instance shown in Table 1. The first N=8 dimensional vector presents the operation sequence. And the second N=8 dimensional vector shows that the machine allocation for the corresponding operation.

3.2. Initialization

Most recent researches adopt random initialization, which generates the inferior population so that the iteration number or population number needs to be enlarged to obtain the approximated optimization. Kacem et. represented an approach by localization [6-7].The Kacem assignments scheme assigns each operation to the machine with minimum processing time, and updates the corresponding machine workload. However, this approach is strongly dependent on the order in which operations and machines are given in the table. We modify it with a mix of random initialization strategy (RIS).

Here, 40% of initial population is generated by Method (1) and 60% by Method (2) as follows:

(1) Random initialization strategy (RIS): Generate operation order and machine order randomly.
(2) Improve on Kacem assignments scheme. Randomly generate operation's sequence, and determine the machine sequence according to machines' workloads, which can be detailed as following 4 steps.

Operation sequence:	2	2	1	3	1	2	3	1
Machine sequence:	1	4	3	3	1	3	4	3

Fig. 1. An instance of coding

- Step1: Generate an operation sequence;
- Step2: According to the operation sequence, converts Table 1;
- Step3: Assign the feasible machine with shortest process time to each operation, and renew the machines' load;
- Step4: Repeat Step3 until all operations are assigned.

The advantage in using RIS mix is to generate different initial assignment in different runs of the algorithm, and to explore the search space better than solely using Kacem assignments scheme [16].

Suppose RIS generates operation sequence as in the first line, Table 2 shows us the first two iterations. The last four columns represent the final assignment obtained. As shown in Table 2, bold values report the workload updates.

Table 2. Improvement on Kacem assignments scheme

	M_1	M_2	M_3	M_4	M_1	M_2	M_3	M_4	M_1	M_2	M_3	M_4		M_1	M_2	M_3	M_4
O_{21}	2	5	1	3	2	5	1	3	2	5	1	3		2	5	1	3
O_{22}	4	6	8	4	4	6	**9**	4	4	6	**9**	4		4	6	**9**	4
O_{11}	7	6	4	5	7	6	**5**	5	7	6	**5**	**9**	…	7	6	**5**	**9**
O_{31}	8	6	3	5	8	6	**4**	5	8	6	**4**	**9**		8	6	**9**	**9**
O_{12}	4	8	5	6	4	8	**6**	6	4	8	**6**	**10**		4	**14**	**11**	**10**
O_{23}	9	7	2	2	9	7	**3**	2	9	7	**3**	**6**		**13**	**13**	**8**	**6**
O_{32}	3	5	8	3	3	5	**9**	3	3	5	**9**	**7**		**7**	**11**	**14**	**13**
O_{13}	9	5	4	7	9	5	**5**	7	9	5	**5**	**11**		**20**	**11**	**10**	**17**

The machine sequence can be obtained as Fig.2 according to the coding mechanism.

3.3. Chaotic particle swarm optimization method based on logistic map

The particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling.

Defining that the search space of PSO is d-dimensional and the i th particle of the swarm is $X_i = (x_{i1}, x_{i2}, ..., x_{id})$, which velocity is $V_i = (v_{i1}, v_{i2}, ..., v_{id})$. The local best particle is denoted as $pbest_i = (p_{i1}, p_{i2}, ..., p_{id})$, the global best particle is denoted as $gbest = (g_{i1}, g_{i2}, ..., g_{id})$.

The particle updates as follows:

$$V_i(t+1) = \omega \bullet V_i(t) + c_1 \bullet r_1 \bullet (pbest_i(t) - X_i(t)) + c_2 \bullet r_2 \bullet (gbest(t) - X_i(t)) \quad (2)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (3)$$

Where ω is the inertia weight, t is the number of iterations; c_1, c_2 are given positive constants called accelerating factors; r_1, r_2 are two independent random numbers that is uniformly distributed between M and N.

In conventional PSO algorithm, r_1 and r_2 are set to be random numbers as in equation (2), which can't necessarily lead to complete coverage of the states space. Considering the ergodicity of chaotic system, the logistic mapping technique is adopted here. Logistic mapping is a dynamical system for population prediction [8]. Given x's value of the t-th iteration, that of the (t+1)-th can be update as follows:

$$r(t+1) = \mu \bullet r(t) \bullet (1 - r(t)) \quad r(t) \in (0,1) \quad (4)$$

Where, μ is the control parameter, when $\mu = 4$ and $r(0) \neq \{0.25, 0.50, 0.75\}$, the system is in chaos state. Here, the chaotic mechanism is performed on r_1 and r_2 as equation (5) to improve the global convergence.

$$r_i(t+1) = 4.0 \bullet r_i(t) \bullet (1 - r_i(t)) \quad r_i(t) \in (0,1) \quad i = 1,2 \quad (5)$$

Machine sequence:	3	4	3	2	1	4	1	3

Fig. 2. Machine sequence got by Kacem assignments scheme

3.4. Self-adaptive parameter setting

PSO algorithm's convergence is sensitive to parameter ω.When ω is large, it is capable of large-scaled search, however the precision is limited. On the other hand, when ω is small, precision is improved with higher probability to fall into local extremum [8]-[9]. To solve this problem, ω is self-adapted to reduce gradually as in Equation (6).

$$\omega(t) = \omega_{\max} - \frac{(\omega_{\max} - \omega_{\min}) \bullet gen}{M} \quad (6)$$

Where $\omega_{\max}$ and $\omega_{\min}$ respectively denotes the maximum and the minimum value of ω(generally, $\omega_{\max} = 1.2, \omega_{\min} = 0.4$), gen is the current iteration number, and M is the maximum iteration number.

3.5. Genetic Operator

(1) Fitness Function

The objective is to minimize the makespan, so the following fitness function is applied:

$$f(x) = \frac{1}{makespan(x)} \quad (7)$$

(2) Crossover Operator

The first part of solution is performed with the POX crossover operator develop by Shi [10], and the second part with random multi-point crossover operator. In this way, feasible offspring can be constructed.

(3) Mutation operator

For the operation sequence, select an operation by random, and find its immediate precedence activity and immediate successor activity, which positions are a and b, respectively. Then inserts the selected operation into interval (a, b). This Mutation operator preserves the constraints of the priorities, and ensures the feasibility.

For the machine sequence, Balance Load Mutate [11] is operated here. In the machine with the maximum workload, search the operations which can be performed on the machine with minimum workload. And assign one of these operations to the machine with minimum workload.

3.6. The flowchart of the Hybrid PSO algorithm

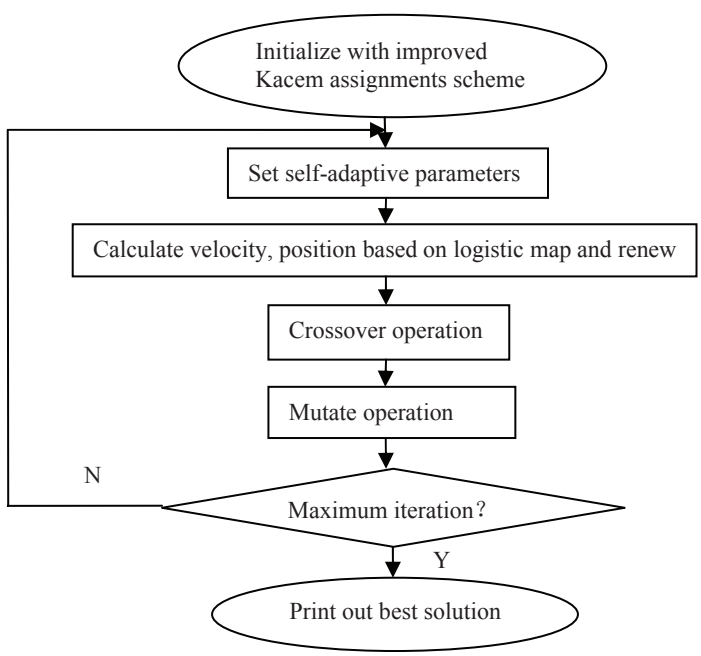

Fig. 3. Flowchart of the Hybrid PSO algorithm

4. Experimental results

To illustrate the performance of the proposed algorithm, tests on problem instances from Kacem [7] and Brandimarte[12] were performed. The population size is set to 500, the maximum iterative generation is set to 1000, the crossover rate is 0.85, and the mutation rate is 0.15.

In Table 3 and Table 4, column "n" reports the number of operation, column "m" reports the machine number, column "C_{max}" reports the makespan, column "Dev" reports the relative deviation with respect to the result of this method. The column "LB" reports the best known lower bound [13]. The line "AI" reports the average improvement, and is the average of the front ten relative deviations.

$$Dev = \frac{C_i - C}{C_i} \times 100\% \quad i = 1,2,3 \qquad (8)$$

Where $C_i\,(i = 1,2,3)$ denotes the makespan of other three algorithms we compared with, and C denotes the makespan obtained by this method.

As can be seen from Table 3 that, this method is compared with the method "Kacem" [7], the bilevel genetic algorithm [14] (by C. Zhang et al., 2007), and the improved genetic algorithm [15] (by G. Zhang et al., 2009). For all instances from Kacem, this method outperforms in terms of solution quality and the running time

Ten benchmark cases from Brandimarte were tested. Table 4 compares this method with the genetic algorithm integrated different strategies [16] (by Pezzella et al., 2008), the improved algorithm [17] (by Q. Liu et al., 2007), and the integrated genetic algorithm [11] (by W. Wu et al., 2009).

The bold values show the best results of the given algorithms. In Table 4, the values within parenthesis are optimal. For 9 problems from Brandimarte, this algorithm is superior than or equal to the cited three algorithms. For Mk07, the solution is slightly poor. The computational results validate this algorithm's effectiveness.

Table 3. Results of Tests on instances from Kacem

n,m	Kacem [7]	C. Zhang et al. [14]		G. Zhang et al. [15]		This method	
	C_{max}	C_{max}	Time（s）	C_{max}	Time（s）	C_{max}	Time（s）
8,8	15	14	17	14	2.7	**14**	**1.4**
10,10	7	7	16.9	7	4.1	**7**	**2.0**

Table 4. Results of Tests on instances from Brandimarte

	n	m	LB [18]	This method	Pezzella et al.[16]		Q. Liu et al. [17]		W. Wu et al. [11]	
					C_{max}	Dev(%)	C_{max}	Dev(%)	C_{max}	Dev(%)
Mk01	10	6	36	**40**	40	0.00	40	0.00	40	0.00
Mk02	10	6	24	**26**	26	0.00	26	0.00	27	3.70
Mk03	15	8	204	**(204)**	204	0.00	204	0.00	204	0.00
Mk04	15	8	48	**60**	60	0.00	60	0.00	60	0.00
Mk05	15	4	168	**173**	173	0.00	173	0.00	173	0.00
Mk06	10	15	33	**60**	63	4.76	62	3.23	62	3.23
Mk07	20	5	133	140	139	-0.72	140	0.00	139	-0.72
Mk08	20	10	523	**(523)**	523	0.00	523	0.00	523	0.00
Mk09	20	10	299	**307**	311	1.29	307	0.00	309	0.65
Mk10	20	15	165	**205**	212	3.30	215	4.65	206	0.49
AI						0.86		0.79		0.74

5. Conclusion

In this work, according to the characteristics of FJSP, a hybrid algorithm combining the chaos particle swarm optimization and GA is proposed to solve the FJSP. With the induction of the improved Kacem assignments scheme, an initialization mechanism is presented. And the coverage of the states space is improved based on logistic map. Experimental results on various benchmark instances verify the effectiveness of this algorithm, and show that this method outperforms some existing methods.

Acknowledgment

The authors are grateful to the support of National Natural Science Foundation of China under Grant No. 70971026, and Natural Science Foundation of Guangdong Province under Grant No. 9151009001000040.

References

[1] Garey E L. Johnson, D S. Sethi R. The Complexity of Flow-shop and Job-shop Scheduling [J]. Mathematics of Operations Research, 1976, 1: 117-129.

[2] Li-Ning Xing, Ying-Wu Chen, Peng Wang, Qing-Song Zhao, Jian Xiong. A Knowledge-Based Ant Colony Optimization for Flexible Job Shop Scheduling Problems. Applied Soft Computing. 2010; 10(3): 888-896.

[3] Tie-Ke Li, Wei-Ling Wang, Wen-Xue Zhang. Solving flexible Job Shop scheduling problem based on cultural genetic algorithm. Computer Integrated Manufacturing Systems. 2010;16(4): 861-866.

[4] Tsung-Lieh Lin, Shi-Jinn Horng, Tzong-Wann Kao, Yuan-Hsin Chen, Ray-Shine Run, Rong-Jian Chen, Jui-Lin Lai, I-Hong Kuo. An efficient job-shop scheduling algorithm based on particle swarm optimization. Expert Systems with Applications. 2010;37(3): 2629-2636.

[5] Jun-Qing Li, Quan-Ke Pan, P. N. Suganthan and T. J. Chua. A hybrid tabu search algorithm with an efficient neighborhood structure for the flexible job shop scheduling problem. THE INTERNATIONAL JOURNAL OF ADVANCED MANUFACTURING TECHNOLOGY. 2010; 52: 683-697.

[6] Kacem I., Hammadi S., Borne P.. Approach by Localization and Multi-objective Evolutionary Optimization for Flexible Job-shop Scheduling Problems [R]. IEEE Transaction on Systems, Man, and Cybernetics-Part C, 2002, 32(2): 2599-2604.

[7] Kacem I., Hammadi S., Borne P.. Approach by Localization and Multi-objective Evolutionary Optimization for Flexible Job-shop Scheduling Problems [R]. IEEE Transaction on Systems, Man and Cybernetics-Part C, 2002, 32(1): 1-13.

[8] Daohua Liu, Sicong Yuan, Yang Lan, Xinjian Ma. Method of particle swarm optimization based on the chaos map [J]. JOURNAL OF XIDIAN UNIVERSITY(NATURAL SCIENCE). 2010, 37(4): 764-769.

[8] Maurice C., Kennedy J.. The Particle Swarm-explosion, Stability and Convergence in a Multi-dimensional Complex Space [J]. IEEE Transactions on Evolutionary Computation, 2002, 6(1): 58-73.

[9] Eberhart R..C., Shi Y.. Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization [C]. Proceedings of the IEEE Congress on Evolutionary Computation, San Diego, 2000: 84-88.

[10] Shi GY. A Genetic Algorithm Applied to a Classic Job-shop Scheduling Problem [J]. International Journal of Systems Science, 1997, 28(1): 25-32.

[11] Wenyao Wu, Hongming Cai, Lihong Jiang. Integrated genetic algorithm for flexible job-shop scheduling problem [J]. Computer Engineering and Applications. 2009, 45（22）, 183-186.

[12] Brandimarte P. Routing and Scheduling in a Flexible Job-shop by Tabu Search [J]. Annals of Operations Research, 1993, 22(2): 157-183.

[13] Mastrolilli M, Gambardella LM. Effective neighbourhood functions for the flexible job shop problem. Journal of Scheduling 1996;3:3–20.

[14] Chaoyong Zhang, Yunqing Rao, Peigen Li, Xinyu Shao. BILEVEL GENETIC ALGORITHM FOR THE FLEXIBLE JOB-SHOP SCHEDULING PROBLEM [J]. Chinese Journal of Mechanical Engineering, 2007, 43(4): 120-124.

[15] Guohui Zhang, Liang Gao, Peigen Li, Chaoyong Zhang. Improved Genetic Algorithm for the Flexible Job-shop Scheduling Problem [J]. Chinese Journal of Mechanical Engineering. 2009(45), 145-151.

[16] Pezzella F, Morganti G, Ciaschetti G. A genetic algorithm for the Flexible Job-shop Scheduling Problem [J]. Computers & Operations Research, 2008, 35: 3202-3212.

[17] Qiong Liu, Chaoyong Zhang, Yunqing Rao, Xinyu Shao. Flexible Job-Shop Scheduling Problem with Improved Genetic Algorithm [J]. Industrial Engineering and Management. 2009(14), 59-66.