Journal of Theoretical and Applied Information Technology
20th January 2013. Vol. 47 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

# RESEARCH ON JOB-SHOP SCHEDULING PROBLEM BASED ON IMPROVED PARTICLE SWARM OPTIMIZATION

[1]ZUFENG ZHONG

[1] School of Business, ZhanJiang Normal University,Zhanjiang,524048,China

E-mail: zhongzufeng@163.com

## ABSTRACT

Considering the standard particle swarm optimization (PSO) has the shortcomings of low convergence precision in job shop scheduling problems, the job shop scheduling solution is presented based on improved particle swarm optimization (A-PSO). In this paper, the basic theory of A-PSO is described. Also, the coding and the selection of parameters as well as the decoding of A-PSO are studied. It uses the maximum flow time which is minimized to evaluate the performance of the algorithm, and applies it to solve a typical scheduling problem. A large number of simulation results show that this algorithm has good feasibility and effectiveness in job-shop scheduling problem.

**Keywords:** *Particle Swarm Optimization; Coding; Job-Shop Scheduling Problem; Intelligence Optimization Algorithms*

## 1. INTRODUCTION

Job-shop scheduling aims to meet some performance indices by optimizing the allocation of shared resources and the arrangement of productive jobs within a certain period. Job-shop scheduling problem as one of the typical production scheduling problem requires strong engineering background, and transforms with many real engineering problems. Effective job-shop scheduling technologies are very important in production because they can intervene or adjust production and operation, and improve efficiency by rationalizing resource allocation. Job-shop scheduling problem is also a NP-hard problem, so efficient algorithms are very important in its scheduling and optimization.

## 2. LITERATURE REVIEW

So far, many algorithms have been used in job-shop scheduling problem. Sarin and Potts used mathematical programming [1, 2], which seeks the most accurate solution by using an enumerative technique called branch and bound, based on branch rules, bound mechanism and upper bound production. Wang and Luh used Lagrangian relaxation algorithm [3, 4] which quantitatively evaluates the suboptimality of the solutions, but is restricted by low efficiency in solving dual problems. Chen et al. employed genetic algorithm [5] which finds the minimized makespan of flexible job-shop scheduling problem, and simulates chromosomes by graph theory, where chromosome coding consists of the definition of route strategy and the procedures on machine. Hon et al. [6] held that special design was necessary for solving GA chromosome of flexible job-shop scheduling problem, and presented two standards to evaluate the performance of chromosome. Najid et al. [7] solved minimized makespan by integrating simulated annealing of nearby functions. Mohammad and Parviz [8] presented a two-step tabu search which targets at the sequence with minimized makespan and depends on the setting.

## 3. INTRODUCTION INTO JOB-SHOP SCHEDULING PROBLEM

In practical job-shop, there is one flexible job-shop scheduling problem, which refers to the job-shop scheduling where machines can be selected [9, 10]. In comparison with traditional job-shop scheduling problem, it is superior in breaking the uniqueness of resources, and allowing selecting several machines to complete each procedure. Therefore, flexible job-shop scheduling is more suitable for real working environment and has theoretical and practical meanings.

Flexible job-shop scheduling problem is extended from traditional job-shop scheduling problem, where each procedure can only be processed on a specified machine. In flexible job-shop scheduling problem, each procedure can be

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

processed on several machines, and each machine takes specific processing time. Flexible job-shop scheduling problem reduces the constraint of machinery, extends the search coverage of feasible solutions, and increases complexity.

Flexible job-shop scheduling problem can be described as: [11,12] with n independent jobs, $J_j$ (j=1,2,…,n) refers to the jth job; each job consists of $n_j$ procedures, $O_{ij}$ indicates the ith procedure for job $J_j$; M is a machine set composed of m machines, where $M_k$ is the kth machine; each procedure $O_{ij}$ can be processed by a machine set, which can be expressed as $M_{ij}$; $M_{ij} \in \{1,2,...m\}$; the processing time $t_{ijk}$ of procedure $O_{ij}$ on machine k is predetermined.

Each job contains one or more procedures, which are predetermined; each procedure can be processed on several machines, and its processing time varies with the performance of machines.

Moreover, the processing should meet the following constraint conditions:

(1) Only one part can be processed on one machine at one moment;

(2) One job can only be processed on one machine at one moment, and an operation cannot be stopped midway.

(3) Procedure constraints exist within one job, but not among different jobs;

(4) Different jobs have the same priority.

Scheduling aims to determine the best machine for each procedure, find the best sequence and starting time for each job on one machine, and to optimize some performance indices.

## 4. STANDARD PSO

PSO was proposed in 1995 and its modeling and simulation were inspired by bird foraging. Gird foraging was used to replace the principle of selection in intelligent algorithms, and swarm cooperation was used to solve optimizing problems.

In PSO, a foraging bird swarm was considered as an optimization problem to be solved; each particle sought by PSO iteration was treated as a foraging individual, and the searched food was the solution.

Let each individual in an optimization problem be a point with two independent variables: position and speed. Supposing a swarm has m particles, where m is called swarm size and large m will reduce operating and searching speeds. The two independent variables of one particle change with the positions of itself and other particles. Let $z_i = (z_{i1}, z_{i2},..., z_{iD})^T$ be the D-dimension position vector of the ith particle (i =1,2,…,m). According to the predetermined adaptive value function which is related to the problem to be solved, the current adaptive value of i can be calculated and evaluated. $v_i = (v_{i1}, v_{i2},..., v_{iD})^T$ is the flying speed of particle i, which is the drift when the particle changes its position. Any particle is aware of its current best position $p_i = (p_{i1}, p_{i2},..., p_{iD})^T$, or $p_{Best}$, as well as the best position $p_g = (p_{g1}, p_{g2},..., p_{gD})^T$, or $g_{Best}$ of the whole swarm. In each iteration, a particle changes its position and speed by locking these two extremums and based on the following two updating formulae:

$$v_{id}(k+1) = wv_{id}(k) + c_1 r_1 (p_{aid} - z_{id}(k)) + c_2 r_2 (p_{bgd} - z_{id}(k)) \quad (1)$$

$$z_{id}(k+1) = z_{id}(k) + v_{id}(k+1) \quad (2)$$

where i =1,2,…,m, d =1,2….D; k is the number of iterations, $r_1$ and $r_2$ change randomly within [0, 1], so it maintains the diversity of the swarm; $c_1$ and $c_2$ are called the accelerating factors, which allow the particle to learn from itself or other better particles, so it gets closer to a better position it or the swarm has passed [13, 14]. $z_{id}(k))$ refers to the current position of the searching particle; $z_{id}(k+1))$ is the updated position after iteration; $v_{id}(k)$ is the current speed of the searching particle; $v_{id}(k+1)$ is the updated speed after iteration; $p_{id}$ and $p_{gd}$ are the best speeds of the particle $p_{Best}$ and the sward $g_{Best}$, respectively; w is inertia weight, which balances the searching in small range or the whole range. w is calculated as $w = w_{\max} - k * \frac{w_{\max} - w_{\min}}{k_{\max}}$, where $w_{\max}$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

and $w_{\min}$ are the maximum and minimum weighting coefficients, respectively, k is the current number of iterations, $k_{\max}$ is the predetermined upper limit of iterations.

## 5. IMPROVED PARTICLE SWARM OPTIMIZATION AND SOLUTION FOR FLEXIBLE JOB-SHOP SCHEDULING

### 5.1 Improved PSO

In solving job-shop scheduling problem, though standard PSO has some effect, it still has defects of low convergence speed, and trapping into local extremum. In practice, a swarm clusters by some rules as flying around the "head bird" to the position where food is. Meanwhile, regarding the properties of a species-based evolutionary algorithm, when at low diversity, the swarm traps into local extremum, thus losing the ability to explore new regions. In premature or global convergence, particles in a swarm will "cluster", indicating that lack of diversity will lead to premature convergence. Therefore, it is important to increase operational efficiency by searching a strategy to increase diversity. Therefore, a new algorithm based on excellent particles clustering subgroup and k-means algorithm was proposed to construct optimum point. Experiments prove that this algorithm has higher optimizing ability than standard PSO. By introducing the dynamic nonlinear ω, the position updating formula is the same, but the speed updating formula is:

$$v_{id}(k+1) = wv_{id}(k) + c_1 r_1 (p_{id} - z_{id}(k)) + c_2 r_2 (\bar{p}_{bgd} - z_{id}(k)) \quad (3)$$

where $\bar{p}_{bgd} = (p_{1gd} + p_{2gd} \ldots + p_{bgd})/b$ (4) is the order of all the particles in the swarm, is the average position of the b particles before the optimum position.

### 5.2 Coding

Swarm algorithms were first used to optimize continuous functions. Job-shop scheduling problems are scattering, dynamic and multivariate. PSO coding has difficulty in job-shop sorting, and in finding a natural expression.

Based on the features of FJSP, a sequence-based coding pattern was designed to code the particles. By considering sorting and coding as the sequence of procedures, each particle presents a sorting plan. All procedures of the same job are assigned a job label, then each element in a particle corresponds to the job label. The procedure of this job can be determined according to the order of the job label in this procedure. For instance, a particle (121322313) in a 3×3 machine scheduling problem, each job contains 3 procedures, so this job label reappears 3 times. Here the 1st gene indicates the 1st procedure of job 1, because it's the 1st time "1" appears. The 3rd gene indicates the 2nd procedure of job 1, and so on.

### 5.3 Steps for the solution of job-shop scheduling

Based on the ordered operation table of scheduling plans, the target is to minimize the maximum makespan of all machines, and the steps are:

1) After analysis of the problem and understanding the properties of the solution, predetermine the particle swarmsize N, the maximum of iterations $T_{\max}$, the current number of iterations t.

2) Initialize particle swarm by random function, so the position $X_i$ and speed $V_i$ are generated randomly.

3) Sort the components in the position vector. Code each particle with the method proposed in this study, and produce the ordered operation table by combining the constraint conditions of job processing.

4) Decode the ordered operation table, find a scheduling plan, calculate the adaptive value of each particle, and record the average of b particles $\bar{G}_{b.best}$ with the largest global optimal value, as well as individual optimal value $P_{best}$.

5) Update the speed and position of the particles according to Eqs. (2-4).

6) Decide whether t is the maximum of iterations $T_{\max}$, if not, t=t+1, return to Step 3, otherwise go to Step 7.

7) Output the process sequence of the particle corresponding to global optimal value $\bar{G}_{b.best}$. This is the optimal sorting result.

## 6. SIMULATED RESULTS AND ANALYSIS

### 6.1 Test case and parameter setting

To validate the algorithm in this study, the standard testing cases LA (LA01-20) and FT10 for job-shop scheduling were used. The operation

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

parameters of PSO are set as: The number of original population is 50, the maximum evolution number is 200, c1= c2= 2, the original inertia weight ws= 1, the terminal inertia weight we= 0.4; All experiments were conducted on a CPU of 3.0 GHz, memory of 2G, hard disk of 80G, Windows XP, and VC++.

### 6.2 Comparison of calculation results with other algorithms

To validate its validity and superiority, genetic algorithm (GA) and taboo search (TS) were used for comparative experiments. The results are listed in Table 1. From the 20 FJSP standard testing data of the three algorithms (Table 1), the number of optimal solutions is 20 for this algorithm, is 16 for TS, and only 8 for GA. This indicates that the algorithm in this paper increases searching speed and avoids local optimal solution.

*Table 1. Comparison Of Calculation Results Between Different Algorithms*

| Case | Size | Real optimal value | GA | TS | A-PSO |
|---|---|---|---|---|---|
| FT10 | 10 × 10 | 930 | 930 | 930 | 930 |
| LA01 | 10 × 5 | 666 | 666 | 666 | 666 |
| LA02 | 10 × 5 | 655 | — | — | 655 |
| LA04 | 10 × 5 | 590 | 590 | 590 | 590 |
| LA05 | 10 × 5 | 593 | 593 | 593 | 593 |
| LA06 | 15 × 5 | 926 | 926 | — | 926 |
| LA07 | 15 × 5 | 890 | — | 890 | 890 |
| LA08 | 15 × 5 | 863 | 863 | — | 863 |
| LA09 | 15 × 5 | 951 | — | 951 | 951 |
| LA10 | 15 × 5 | 958 | — | 958 | 958 |
| LA11 | 20 × 5 | 1222 | — | 1222 | 1222 |
| LA12 | 20 × 5 | 1039 | — | — | 1039 |
| LA13 | 20 × 5 | 1150 | — | 1150 | 1150 |
| LA14 | 20 × 5 | 1292 | 1292 | 1292 | 1292 |
| LA15 | 20 × 5 | 1207 | — | 1207 | 1207 |
| LA16 | 10 × 10 | 945 | — | 945 | 945 |
| LA17 | 10 × 10 | 784 | — | 784 | 784 |
| LA18 | 10 × 10 | 848 | 848 | 848 | 848 |
| LA19 | 10 × 10 | 842 | — | 842 | 842 |

## 7. CONCLUSIONS

Job-shop scheduling problem is a hot issue in manufacturing system and combinatorial optimization, and needs to be immediately solved in practice. Targeting at the difficulties in the current job-shop scheduling problem, a solution plan based on A-PSO was proposed. It solves the sorting of complex job-shop working by using the advantages of PSO, and its searching efficiency is higher than other algorithms. Simulation cases apply PSO into the solution of typical FJSP, and the scheduling results indicate that it has better scheduling results and higher validity.

## REFERENCES

[1] Sarin S C, Ahn S. An improved branching scheme for the branch and bound procedure of Scheduling n jobs on m machine to minimize total weighted flow time. Internal Production Res,1988,26:1183-1191

[2] Potts C N. A branch and bound algorithm for the total weighted tardiness Problem. Operation Research, 1985, 33: 363-377

[3] Wang J, Luh P B. Scheduling job shops with batch machines using the lagrangian relaxation technique. European Journal of Control, 1997, 3: 268-279

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[4] Luh P B, Hoitomt D J. Scheduling of manufacturing systems using the lagrangian relaxation technique. IEEE Transactions on Automatic Control, 1993, 38(7): 1066-1080

[5] Chen H., thlow J., Lehlnann C. May 1999.A genetic algorithm for flexible job-shop scheduling [C]. Proceedings of the 1999 IEEE International Conference on Robotics & Automation.

[6] HON.B., Tay J.C.2004.GENACE:an efficient cultural algorithm solving the flexible job shop Problem [C]. Proceedings of Congress on Evolutionary Computation, 1759-1766.

[7] NajidN.M., Dauzere-peres5.,zaidat A. 2002. A modified simulated annealing method for flexible job shop scheduling Problem[C]. Proceedings of the IEEE International Conference on Systems Man and Cybemeties. NJ, USA, IEEE, PP. 89-94

[8] Mohammed S. M., Parviz F. May 2006. Flexble job shop scheduling with tabu search algorithms [J].International Journal of Advanced Manufacturing Technology,32(5-6):563-570.

[9] G.K. Deb, R.B. Agrawal. Simulated Binary Crossover for Continuous Search Space, Complex Systems. 1995: 115-148

[10] M. Gen, R. Cheng. Genetic Algorithms and Engineering Optimization. NY: John Wiley & Sons. 2000

[11]Brandimarte P. Routing and scheduling in a flexible job shop by tabu search. Annals of Operations Research,1993,22:158~183

[12] Gu Feng, Chen Pinghua, Lu Bingyuan, et al, Particle Swarm Optimization For Flexible Job Shop Scheduling, Systems Engineering, 2005, 23(9): 20-23 (in Chinese).

[13] Kennedy,J．,Eberhart,R.C．A Discrete Binary Version of the Particle Swarm Algorithm．On Systems,Man,andCybernetics ． Piscataway ． IEEE Service Center．1997:4104~4109

[14]Wang Ling, Liu Bo, Particle Swarm Optimization and Scheduling Algorithm, Tsinghua University Press, 2008: 95-101 (in Chinese).