2012 International Conference on Computer Technology and Science (ICCTS 2012)
IPCSIT vol. 47 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V47.70

A Combined Optimization Algorithm for Multi-objective Flexible Job Shop Scheduling Problem

Li Li[1], Wang Keqi[2+], and Yue Qi[3]

[1,3]Information and Computer Engineering College, Northeast Forestry University, Harbin, China
[2]Forestry Engineering Automation Discipline, Northeast Forestry University, Harbin, China

Abstract. In order to exert the advantage of ant colony algorithm and particle swarm optimization algorithm respectively, a method combined the two algorithms was designed for solving multi-objective flexible job shop scheduling problem in this paper. The proposed algorithm was composed by two phases. The first phase made use of the fast convergence of PSO to search the particles optimum position and made the position as the start point of ants. In the second phase, the traditional ant colony algorithm was improved and was used to search the global optimum scheduling according to its characters of positive feedback and structure of solution set. The combined algorithm was validated by practical instances. The results obtained have shown the proposed approach is feasible and effective for the multi-objective flexible job shop scheduling problem.

Keywords: multi-objective flexible job shop scheduling; particle swarm optimization algorithm; ant colony algorithm

1. Introduction

Job shop scheduling problem is a method of resource allocation for simple-objective or multi-objective optimization scheduling with constraints. Job shop scheduling problem is a sub-set of scheduling problem, and it is a NP hard problem for its large number of constraints. Classical job shop scheduling problem has been paid much attention to and has been studied widely. As an extension of classical job shop scheduling, flexible job shop scheduling problem (FJSSP) reduces the constraints on machine. Each operation is allowed to be executed on more than one machine from allowed machine set. Multi-objective flexible job shop scheduling problem (MFJSSP) is more complex and closer to real production than classical job shop scheduling problem. Bruker and Schile carried out the research earlier [1]. It has become an important research point for CIMS to obtain the optimization solution of MFJSSP.

ACA is proposed by Italian researcher Dorigo[2] in1991. This algorithm can find better solution quickly with characters of positive feedback and powerful capability of distributed transaction.Today, applying ACA in solving multi-objective combination optimization problem becomes a very worthy subject of study. PSO algorithm is proposed by Eberhart[3] in 1995. It has a wide range of global search capability. The convergence speed of PSO is fast. It has scalability and it is easy to integrate with other algorithms. But the local search for the latter part of PSO algorithm is poor and the use of feedback information is not sufficient.

In this paper, we improve traditional ACA and make full use of the fast convergence of PSO and the positive feedback of ACA to combine the two algorithms in solving MFJSSP.

2. Formulation of multi-objective flexible job shop scheduling problem

+ Corresponding author.
E-mail address: zdhwkq@163.com

For FJSSP, there are several objects. The first is based on production time, such as makespan, total workload, critical machine workload, etc.; The second is based on date of delivery, such as cost of advance and/or delay; The third is based on cost, such as cost of starting machines, changing line, processing, overtime pay, compensation for overdue, stock on line, scheduling management, etc.. These objections are always conflict with each other in real production. We can design different object and its weigh according to our need for real production.

In this paper, the objects for FJSSP include makespan ($F_1(C)$), total workload ($F_2(C)$) and critical machine workload ($F_3(C)$).

Formulation of FJSSP can be described as follows.

Machine set M= $\{M_1, M_2, \ldots, M_m\}$;

Job set J= $\{J_1, J_2, \ldots, J_j\}$;

The operations of each job and the processing time of each operation on machines are given;

The start time, finish time and processing time of operation J_i ($i \in [1,j]$) on machine m are noted $St_{jim,}$ Ft_{jim} and Δt_{jim};

$$Ft_{jim} = St_{jim} + \Delta t_{jim} \tag{1}$$

The object function is as (2).

$$F(C) = w_1 \times F_1(C) + w_2 \times F_2(C) + w_3 \times F_3(C) \tag{2}$$

Other constraints are as follows.

- Only one job can be carried on one machine at the same time.
- Order constraints only exist in operations of the same job.
- Once an operation starts, it can not be terminated before it finishes.
- Different job has the same priority.
- Start and operations changing time on machine are neglected.

The algorithm optimization objec is to obtain a solution to make the object function value minimum with satisfying the constraints.

3. PSO ALGORITHM

In PSO algorithm, each particle has its position x, velocity v and reasonable solutions. The best solution of particle and swarm is saved as p_{id} and p_{gd}. The update rule of v and x is as (3) and (4).

$$v_{id}(t+1) = w \times v_{id}(t) + c_1 r_1 (p_{id}(t) - x_{id}(t)) + c_2 r_2 (p_{gd}(t) - x_{id}(t)) \tag{3}$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \tag{4}$$

The flow diagram of PSO algorithm is described as figure 1.

4. Traditional ant colony algorithm

The theory of ant colony algorithm is described as follow: ants leave a kind of chemistry material named pheromone when they are looking for food. The shorter ways have more chances to be chosen. So the pheromone leaved by ant on these ways becomes thicker and thicker, and the chance of these ways being chosen by other ants becomes bigger and bigger, too. At last more and more ants choose the shortest way to look for food. ACA looks for the best solution according the information exchange between ants. This algorithm can find better solution quickly with characters of forward feedback and powerful capability of distributed transaction.

General structure of ant colony is described as follow:

- Initialize pheromone trail and parameters;
- While (terminated condition is not met) do the following steps:

Construct a solution and save the nodes which have been visited by ant into a tabu table;

Improve the solution;

Update the pheromone trail by the solution;

- Return the best solution having been found.

5. The improvement of ant colony algorithm for multi-objective flexible job shop scheduling problem

To get better effect, we improved traditional ACA to solve MFJSSP.

5.1. Structure of our improved ACA

According to the analysis of FJSSP, we improved ACA. The flow diagram of our improved algorithm is described as figure 2.

5.2. Description of the improved ACA rule

1) Allowed se: **W**e name each node P_{jim}. Taking the problem 4×5[4] as instance, if ants set out from the first job, the first node will be P_{11m} and the allowed set is { P_{12m}, P_{21m}, P_{31m}, P_{41m} } . A new allowed set will be constructed according to state transition rule and we can put each operation in an appropriate order.

2) Solution set: In our algorithm, the number of subsets is defined by the number of jobs. Our algorithm will choose the best optimum solution of each subset for the global optimum solution.

3) State transition rule[6]:q is a random number which satisfies a uniform distribution in the range [0, 1]. q_0 is a designed parameter and the value is as (5).

$$q_0 = \frac{\lg(g_c)}{\lg(g_m)} \tag{5}$$

g_c is the current iterative generation, and g_m is the predefined maximum iterative generation.

Rule of ant k moves from node x to y is as follow:

if $q<q_0$

$$y = \arg\max_{y \in J_k(x)} \left\{ t(x,y)^{\alpha} \cdot \eta(x,y)^{\beta} \right\} \tag{6}$$

else select y randomly by $P_k(x, y)$

where if $y \in J_k(x)$

$$P_k(x,y) = \frac{t(x,y)^{\alpha} \cdot \eta(x,y)^{\beta}}{\sum_{y \in J_k(x)} \left\{ t(x,y)^{\alpha} \cdot \eta(x,y)^{\beta} \right\}} \tag{7}$$

$$\text{Else } P_k(x,y) = 0 \tag{8}$$

From x to y, the pheromone trail is noted t(x, y) and the visibility is noted η(x, y). η(x, y) is the heuristic value achieved by operation processing time of node y. $\eta(x, y)=1/\Delta t_y$. When all operations are scheduled, ant finishes its work.

In this rule, α is the weigh of heuristic value achieved by pheromone trails; β is the weigh of heuristic value achieved by operation processing time.

4) Trail intensities update rule: To avoid ACA falling into local optimum, Thomas Stuetzle[5] brought out max-min ant system (MMAS).

In our algorithm, the pheromone trails are limited to an interval [t_{min}, t_{max}]. The insistence of pheromone is noted ρ (ρ is a parameter in the range [0, 1]). The pheromone increased on the way from x to y is noted Δt(x, y). The pheromone increased by ant k on this way is noted $\Delta t(x, y)^k$.

a) Local Update Rule:

$$t(x,y) = \rho \cdot t(x,y) + \Delta t(x,y) \tag{9}$$

$$\Delta t(x,y) = \sum_{k=1}^{K} \Delta t(x,y)^k \tag{10}$$

if ant k moves from x to y

$$\Delta t(x,y)^k = \frac{Q}{T_k} \tag{11}$$

$$\text{else } \Delta t(x,y)^k = 0 \tag{12}$$

Q is total pheromone; T_k is the object function of ant k.

b) Global Update Rule: In ACA, the ant with best object function from the very beginning to now has the ability to carry out global update. Global update rule is as (13)-(15):

$$t(x,y)=(1-\rho)\cdot t(x,y)+\rho\cdot\Delta t(x,y) \tag{13}$$

if ant with best value moves from x to y

$$\Delta t(x,y)=\frac{Q}{T_{best}} \tag{14}$$

$$\text{else } \Delta t(x,y)=0 \tag{15}$$

T_{best} is the best object function which has been found till now.

c) Trail Intensities Evaporating Rule: In this paper, t_{min} and t_{max} are designed as (16), (17) [7]:

$$t_{max}=\frac{1}{(1-\rho)\cdot Z_{best}} \tag{16}$$

$$t_{min}=\frac{t_{max}}{5} \tag{17}$$

Z_{best} is the best object function we have known till now. To avoid early stagnation state, we limit t(x, y) as follows:

if (1-ρ) t(x, y)> t_{max}

t(x, y)= t_{max}

else if (1-ρ) t(x, y) < t_{min}

t(x, y) = t_{min}.

5) Local search: The method of local search in this paper is described as follow:

Adjust each operation to the machine with the minimal processing time;

Adjust each operation to the machine with the second minimal processing time;

Adjust each operation to the machine with the minimal total processing time.

Select the best scheduling with the most excellent objective function as the scheduling obtained by this ant. Carry out the local update of pheromone trails according to this scheduling.

6) Choose suitable parameters for the improved ACA: In ACA, parameters affect the algorithm performance clearly. In order to make our algorithm obtain excellent performance, we design α=1, β=1.2, ρ=0.7, Q=80, n=10 for problem 4×5[4] and n=20 for problem 8×8[4].

6. Design of the combined algorithm

To strengthen the search capability for optimal solution and fast convergence of algorithm, we combine PSO with the improved ACA. PSO algorithm has the advantage of fast convergence. Our improved ACA has capability of positive feedback. The algorithm has two phases. The first phase makes use of the fast convergence of PSO to search the particles optimum position and makes this position as the start position of ants. The second phase makes use of the merit of positive feedback and solution set proposed by the improved ACA to search the global optimum scheduling.

The flow diagram of our algorithm is as Figure 3.

7. Data analysis

To evaluate the performance of our algorithm in solving multi-objective flexible job shop scheduling, we carry out tests with problem 4×5 with 12 operations [4] and problem 8×8 with 27 operations [4].

7.1. Problem 4×5 with 12 Operations

This is a total flexibility instance. Figure 4 is the Gantt chart of optimal solution achieved by our algorithm. The object function is 14.8 for this problem. The weigh of $F_1(C)$, $F_2(C)$ and $F_3(C)$ are w_1=0.5, w_2=0.2 , and w_3=0.3.

7.2. Problem 8×8 with 27 Operations

This is an instance of partial flexibility. Table Ⅰ shows the effectiveness comparison of our algorithm and other algorithms [4], [7]. w_1=0.4,w_2=0.2,w_3=0.4.

TABLE I. EFFECTIVENESS COMPARISON ON PROBLEM 8×8 WITH 27 OPERATIONS

Algorithm	Function			
	F(C)	***F_1(C)***	***F_2(C)***	***F_3 (C)***
Temporal Decomposition	33.4	19	91	19
Approach by Localization	26.6	16	75	13
Controlled GA	26.2	16	77	11
PSO+SA	25.8	15	75	12
AL+CGA	—	15	79	—
This paper	25.8	14	77	12

Table 1 shows that the algorithm in combination with PSO and the improved ACA proposed in this paper is effective in solving the problem 8×8 with 27 Operations.

8. Conclusion

In order to obtain optimal solution, we improve ACA and combine PSO with the improved ACA in solving multi-objective flexible job shop scheduling problem in this paper. In the part of our improved ACA, the number of subsets is defined by the number of jobs, and an effective local search method is applied for better solution. We apply trail intensities evaporating rule to avoid the state of early stagnation. Considering the scale of real problem, we design reasonable parameters for the balance of global searching capability and convergence. Our algorithm makes full use of the fast convergence of PSO and the positive feedback of ACA to strengthen the search capability for optimal solution and quick convergence of algorithm. Based on the results analysis obtained by tests, our algorithm is proved to be feasible and effective for multi-objective flexible job shop scheduling problem.

9. Acknowledgment

We thanks National Natural Science Foundation of China #71003020，Natural Science Foundation of Heilongjiang Province of China # F2009192 and the Fundamental Research Funds for the Central Universities # DL10AB02.

10. References

[1] P. Bruker, R.Schlie, “Job-shop scheduling with multi-purpose machines”, *Computing*, 1990, pp.45: 369–375.

[2] A. Colorni, M. Dorigo, V. Maniezzo, *Distributed optimization by ant colonies in European Conference on Artificial Life*, Elsevier Publishing, Amsterdam ,1991, pp.134-142.

[3] J. Kennedy, R C.Eberhert, “Particle swarm optimization”, *In proceeding of the IEEE International Conference on Neural networks, IV [C]*. Piscataway: IEEE Service Center, 1995, pp.1942-1948.

[4] I. Kacem, S. Hammadi and P. Borne, “Approach by localization and multi-objective evolutionary optimization for flexible job-shop scheduling problems”, *IEEE Trans. Syst. Man Cybern*, 2002, pp. 32(1):1–13.

[5] Stuetzle T., Hoos H H., “Max–min ant system”, *Future Generation Computer System*, 2000, pp.16: 889–914.

[6] M. Dorigo, T.Stutzle, *Ant colony optimization*, MIT Press, Cambridge, MA, 2004.

[7] W. J. Xia, Z. M. Wu, “An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems”, *Comput. Ind. Eng*, 2005,.pp. 48: 409–425.

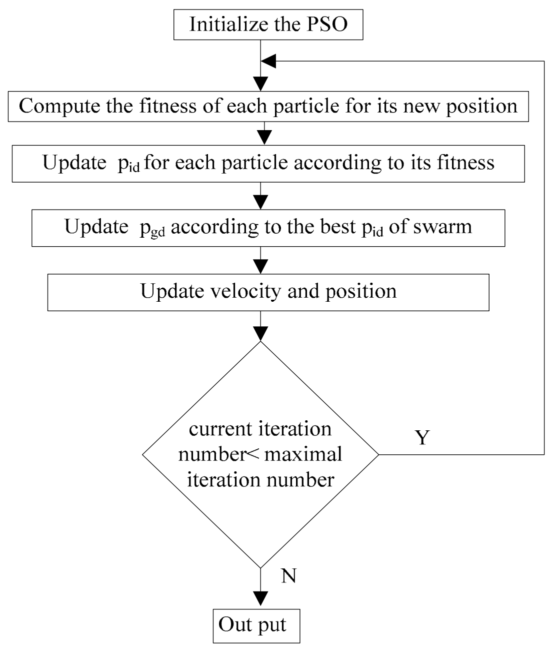

Fig.1: PSO algorithm flow diagram

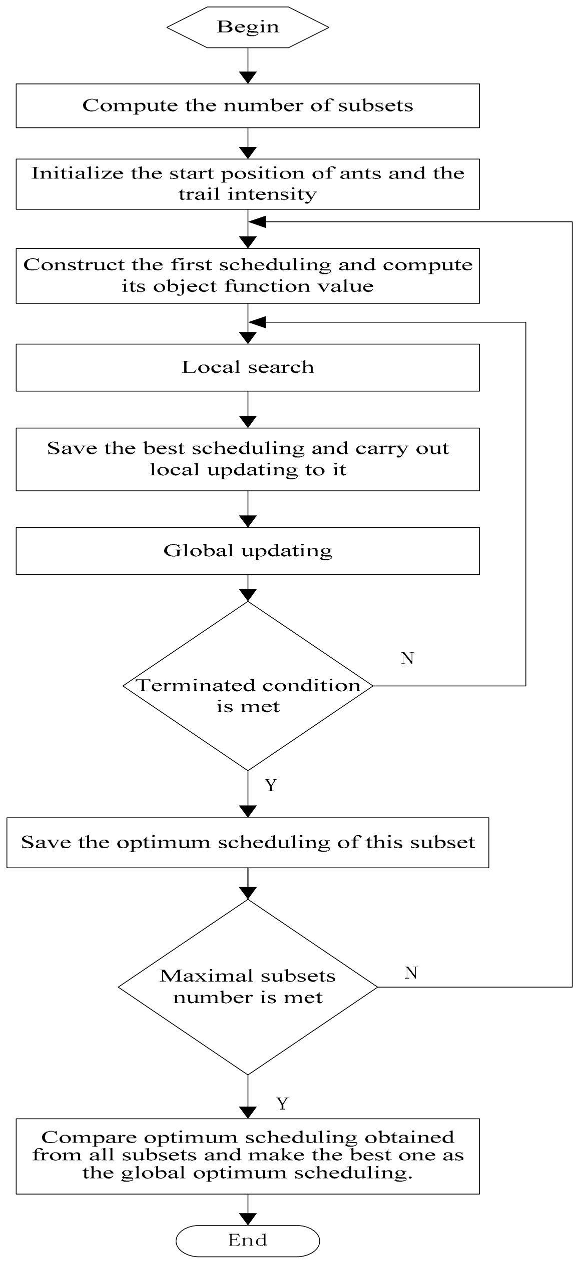

Fig.2: The improved ACA flow diagram

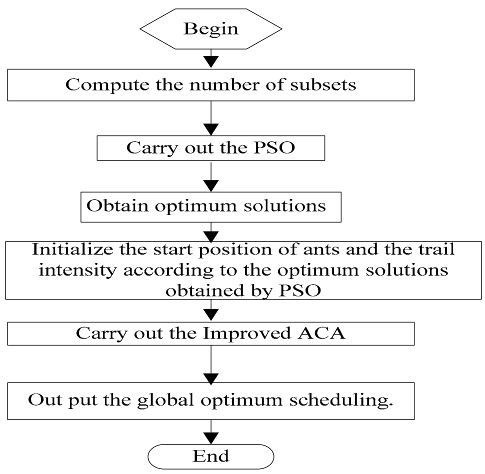

Fig.3: The flow diagram of combined algorithm

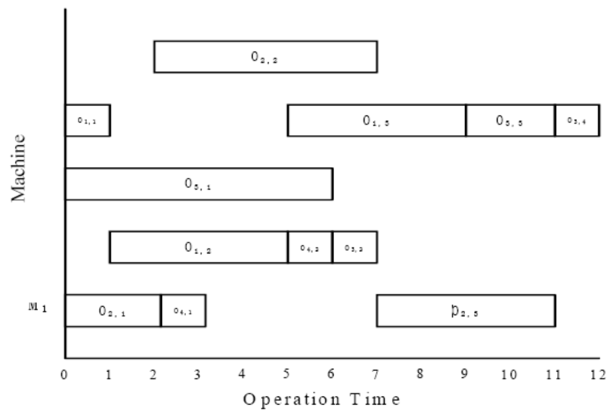

Fig.4: The gantt chart of optimal solution obtained by our algorithm