Job-shop Scheduling Problems Based on Immune Ant Colony Optimization

Ming-lei SUI
Engineering Institute of Engineering Corps
PLA Univ. of Science and Technology
Nanjing 210007, Jiang Su China

Yong-li ZHANG
Engineering Institute of Engineering Corps
PLA Univ. of Science and Technology
Nanjing 210007, Jiang Su China

Email:good823@163.com; zyl19810723@yahoo.com.cn

Abstract

Mathematic model of Job-shop scheduling problem was established, and a kind of immune ant colony algorithm is introduced to dispose Job-shop scheduling problem. Through introducing the mechanism of immunity into the operations of genetic algorithm, the vaccines is obtained and updated in those operations. Then, the immune operation is used on the evolution of populations. And the problems on easy appeared precocity, low searching efficiency can be avoided when immune operation takes effect. The simulations show that the algorithm is feasible and efficient.

Keywords: job-shop scheduling; Ant colony algorithm; artificial immune algorithm

1 Introduction

The job shop scheduling problem (JSP) is a central NP-hard problem in Operations Research and Computer Science [1] that has been studied extensively from a variety of perspectives in the last thirty years. Genetic algorithm has been heavily employed in JSP [2]. Aiming to GA's inefficiency and pre-mature, this paper combines the ideals of the Ant colony algorithm and Immune algorithm together, and applies the proposed algorithm into JSP. The main ideal is to distill and inject the vaccines from and in the evolving colony, in order that the evolution will be more steady-going and quickly.

2 Description for the problem and its math model

Job-shop scheduling problem's math model is a typical integer programming model. To this problem, there are two subjects: machine subject, which is processing order, another is time subject, which is the subject that process's time wasting.

$$\min \sum_{i=1}^{n} \max\{c_{ih}\} \quad (1)$$

$$s.t.\ c_{ik}\text{-}t_{ik}+M_0(1\text{-}a_{ihk})\geq c_{ih} \quad (2)$$

$$i,j=1,2,\cdots,n;\quad h,k=1,2,\cdots,m$$

$$c_{jk}\text{-}c_{ik}+M_0(1\text{-}x_{ijk})\geq t_{jk} \quad (3)$$

$$i,j=1,2,\ \cdots,n;\ h=1,2,\ \cdots,m$$

$$c_{ik}\geq 0;\quad i,\quad i,j=1,2,\ \cdots,n;\ h=1,2,\ \cdots,m$$

Difine a_{ihk} and x_{ijk} as follows:

$$a_{ihk}=\begin{cases}1 & \text{machine}i\text{,if machine}i\text{ before }k\\ 0 & \text{else}\end{cases} \quad (4)$$

$$x_{ijk}=\begin{cases}1 & \text{workpiece}i\text{ before }j\text{ to reach machine}k\\ 0 & \text{else}\end{cases} \quad (5)$$

M_0 is a maximum, which is used as a punishment for the subjects.

3 Ant colony optimization and Immune algorithm

Ant algorithm was first proposed by Dorigo and colleagues [3, 4]. Its essential idea of ACA is that: the individual searching behavior which is directed by the different probabilities to the next node forms colony intelligent. It is *versatile* and *robust* in solving integer programming such as NP problem[5].

However, due to its inherent *defects* it needs more time to find the global optimal, its complexity can reflects that. It is likely to be trapped in local optimal and stop evolving, and can't exploit new search space. This phenomenon can be seen in many simulations.

Immune algorithm (IA) is a bionic algorithm as a simulation of immune system, which combines the certain and random together and has the capacity to reconnaissance and exploit. It's a developmental random search algorithm. Immune algorithm is inspired of cell theory and net theory, and it realized the function of self-adjust and creation of different antibody. It's a new algorithm which is characteristic of searching in multi-peak value and the capacity of global searching [6]. Through the combination of ACA and IA, we can diversify the colony and the colony's evolution will not only be directed by the pheromone, but also by the immune system. We combined the random factor and certain factor together, so we can improve the search efficiency and quality, enrich the diversity and resist the pre-mature.

4 Job shop scheduling based on immune ACA

4.1 Initialize the antibodies

The antibodies are initialized as a series of random permutations of the integers from 1 to $m \times n$, which stand for the ants' routing order. We obtain the antibodies as Ref.7, after the steps as follows.

Step1: Create a random permutation of the integers from 1 to 9: such as [5 2 8 4 3 9 7 6 1].

Step2: array=array-1: [5 2 8 4 3 9 7 6 1]-1＝[4 1 7 3 2 8 5 0].

Step3: divide matrix by *m*, and then add 1: int([4 1 7 3 2 8 5 0]/3)+1=[2 1 3 2 1 3 3 2 1].

Then we obtain an antibody as Ref.7, where: 1 stands for job 1, and the same to 2, 3. The first 2 stands for stage 1 of job 2, the same to 1, 3.

4.2 Decoding scheme

The crucial of the procedure is to guarantee two steps:

(1) The same stage can't be done on two different machines;

(2) The same machine can't do two job at the same time.

In programming, two tags are set: job time tag (JobTime_Tag) and machine time tag (Mac_Time_Tag). Every stage's start time is *max* (MacTime_Tag , JobTime_Tag).

The steps to operate are as follows:

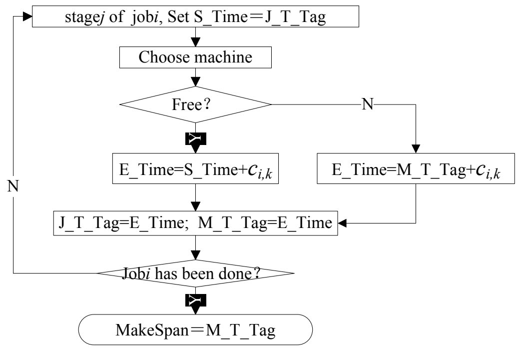

Fig. 1. sketch map for decoding

4.3 Steps for immune ACA

The procedure of the ACO algorithm manages the scheduling of three activities: ants' generation, actions and pheromone update. The steps of Immune Ant Colony Algorithm in this are follows:

1) Select list replace tabu list

The way in tabu list results in that it's hard to select next node. If some nodes have been selected, there will be some vacancies. It's more complex to handle the data, so change the tabu list to select list. The operations are explained as follows:

If the node has been selected, erase it from the select list, and the following nodes move forward. For example: if node.1 has been selected, change the list to: {2→3→4→5→6→7→8→0}, then the node.7: {2→3→4→5→6→8→0→0}, until the last: {5→0→0→0→0→0→0→0}.

2) Pick-up vaccines from memory storeroom

Memory storage is used for storing the best ant, the better ant and the poor ant. Vaccines are picked–up from the memory storage through the follow formulas at a certain probability.

The antibodies of higher thickness and lower appetency will be restrained, and the antibodies of lower thickness and higher appetency will be promoted. So the ant colony will be more diversiform. The *i*th antibody will be selected as [8]:

$$D(X_i) = \frac{1}{\sum_{j=1}^{N_0} \left| f(x_i) - f(x_j) \right|} \qquad i = 1, 2, \cdots, N_0 \tag{6}$$

$$p(x_i) = \frac{\frac{1}{D(X_i)}}{\sum_{i=1}^{N_0} \frac{1}{D(x_i)}} = \frac{\sum_{j=1}^{N_0} \left| f(x_i) - f(x_j) \right|}{\sum_{i=1}^{N_0} \sum_{j=1}^{N_0} \left| f(x_i) - f(x_j) \right|}, i = 1, 2, \cdots, N_0 \tag{7}$$

3) Decide the next one to be processed

Take advantage of Roulette select scheme in GAs: select operation is based on the individual's adaptive value. The most well-known scheme is Roulette select scheme proposed by Holland [9].

$$p_{si} = f_i \Big/ \sum_{i=1}^{N} f_i \tag{8}$$

N is the number of nodes that are can be selected, f_i = $[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}$. The steps are as follows: Compute the value according to f_i; accumulate every element; the array divides the sum of every element.

4) Injection vaccine

Select a piece of information segment from the selected antibody at a certain probability, and inject the vaccine to the ant. Just as fig.2 shows:

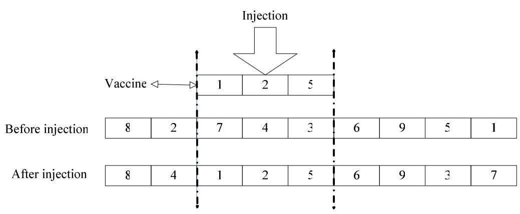

Fig. 2. sketch map for vaccinating

5) Introduce mutation, crossover

In order to avoid the pre-maturity of the ACA, mutation, crossover[10] are introduced into this way which can guarantee the diversity of solutions, exploit new search space, and then the phenomena trapped in the local optimal can be avoid to a certain extent.

Mutation is operated when the route has been done by an ant. Randomly select two nodes in its path, swap them, and then replace them to the rout.

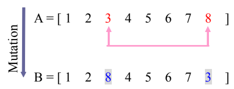

Fig. 3. Sketch map for mutation

After mutation, compute the new ants' f_i, select the better to save.

Crossover is operated after all the ants finished their routs. Randomly select two ants and operate as Fig.4 illustrates:

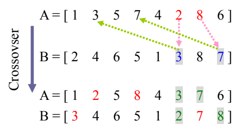

Fig. 4. Sketch map for crossover

After crossover, compute the new individuals' f_i, and the better individuals survive to the next repetition.

5 Instances simulation

Literature [12] gives the study conclusion on how to select α, β, ρ. In this paper, after some tries, the parameters are set as follow: α=1, β=5, ρ=0.6,Q=1000,and set iterations=500.

5.1 Instance 1

This is a Job-shop scheduling problem select from Ref.13: LA01(10×5). All of the basic ACA and IACA obtain the optimum: 666, the comparisons between them are showed as follows:

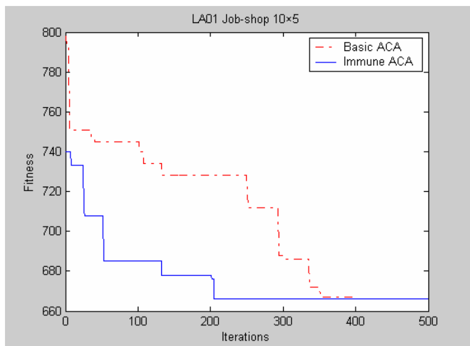

Fig. 5. Comparison in solving LA01

5.2 Instance 2

This is a Job-shop scheduling problem select from Ref.13: LA06(10×10). All of the basic ACA and IACA haven't obtained the optimum; basic ACA: 990, IACA: 945. The comparisons between them are showed as follows:

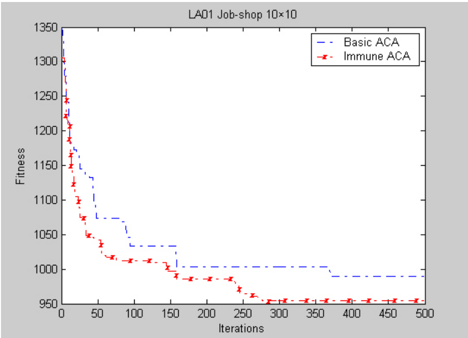

Fig. 6. Comparison in solving LA06

6 Conclusions

This paper applies ant colony algorithm to JSP, and then introduces immune algorithm to the ACA, and designed the selection and the injection criterions for vaccines. From the simulations we know that the search efficiency and quality are improved, and the diversity is enriched, the pre-mature is refrained.

References

[1] Dimitris Bertsimas, Jay Sethuraman (2002), From fluid relaxations to practical algorithms for job shop scheduling: the makespan objective. Math. Program., Ser. A 92: 61–102

[2] WANG Ling, ZHENG Da-zhong (2001), Advances in Job Shop Scheduling Based on Genetic Algorithm. Control and Decision, 11(16):641-646.

[3] M Dorigo,V Maniezzo and A Colorni (1991). The Ant System：An Autocalytic Optimizing Process[R]. Technical Report No.91-016Revised,Politecnicodi Milano,Italy.

[4] Marco Dorigo (1996). The Ant System：Optimization by a colony of cooperating agents[J]. IEEE Transactionson System,Man,and Cybernetics-part B, 26(1)：pp29-41.

[5] A.Colorni, M.Dorigo, V.Maniezzo, M.Trubian (1994), Ant system for Job–Shop Scheduling, Belgian Journal of Operations Research, Statistics and Computer Science.

[6] GE Hong (2002). Immune Algorithm: A Survey, JOURNAL OF SOUTH CHINA NORMAL UNIVERSITY (NATURAL SCIENCE EDITION), 3:12-126

[7] CHEN Xiong, LI Hai-gang, WU Qi-di (2002). Study on Job - shop Scheduling Problem Based on Genetic Algorithm. JOURNAL OF TONG JI UNIVERSITY, 30(1):88-91.

[8] Lovbjerg M, Rasmussen T K, Krink T (2001), Hybrid particle swarm optimization with breeding and subpopulations[C]. In: Proc of third Gentic and Evolutionary computation conference, San Francisco, USA.

[9] D. Goldberg and R. Lingle (1985), Alleles, loci and the traveling salesman problem, The first International Conference on Genetic Algorithm, pp. 154-159.

[10] Bierwirth C, Mattfeld D C (1999). Production Scheduling and Rescheduling with Genetic Algorithms. Evolutionary Computation , 7(1) :1-17

[11] Zne-Jung Lee, Chou-Yuan Lee and Shun-Feng Su (2002), An immunity-based ant colony optimization algorithm for solving weapon-target assignment problem. Applied Soft Computing. (2) pp:39-47.

[12] YE Zhi-wei, ZHENG Zhao-bao (2004), Configuration of Parameters α、β、ρ in Ant Algorithm. Geomatics and Information Science of Wuhan University. 07 pp:597-601. (In Chinese)

[13] WANG Ling (2001), Intelligence Optimal Algorithm & Application. Beijing: Tsinghua press.