Scholars Journal of Engineering and Technology (SJET) **ISSN 2347-9523 (Print)**
Abbreviated Key Title: Sch. J. Eng. Tech. **ISSN 2321-435X (Online)**

A Unit of Scholars Academic and Scientific Society, India
www.saspublishers.com

Immune Algorithm for Solving Large-Scale Job-Shop Scheduling Problems

Nengfa Hu[1]
Department of Computer Science and Engineering, Hanshan Normal University, Chaozhou 521041, China

Original Research Article

***Corresponding author**
Nengfa Hu

Article History
Received: 20.12.2017
Accepted: 28.12.2017
Published: 30.12.2017

DOI:
10.36347/sjet.2017.v05i12.007

Abstract: Job-shop scheduling is a combination explosion problem. Complexity of solving such problems grows exponentially with the increase of problem scope. Based on the ideology of human immune system, an algorithm suitable for solving larger-scale job-shop scheduling problem (JSSP) is designed by selecting operators using hyper-mutation antibody clonal selection algorithm. The test result shows that the algorithm exhibits high efficiency and favorable universality.
Keywords: job-shop scheduling problems, genetic algorithm, clone, mutation.

INTRODUCTION

Job-shop scheduling problem (JSSP), as a typical NP problem, is one of the most intractable problems among all combinational optimization problems till now. Over the years, people have put forward various methods to solve the problem including enumeration, construction and heuristic methods based on priority rules, relaxation method, moving bottleneck method, neural network, and ant colony system (ACS) method, simulated annealing (SA), genetic algorithm (GA), and tabu search method. For example, Giffler and Thompson put forward priority dispatching rule for production scheduling in 1960 and Gere W.S. proposed a heuristic algorithm based on priority dispatching rule for JSSPs in 1966[1,2]. In addition, Balas first solved scheduling problem of machines by using enumeration method based on disjunctive graphs in 1969[3]. In general, the technologies and methods for solving JSSPs are mainly divided into two types: approximate solution and optimization methods. Although solutions to the problem can be quickly obtained by using approximate solution method, they cannot be guaranteed to be optimal. By using optimization method, globally optimal solution can be acquired.

However, the method can only be used for solving small-scale problems at slow speed. From 1970s to mid-1980s, people paid more attention to verifying the complexity of JSSPs [4]. Through studying complexities of numerous extremely difficult problems, people found that only a very few special instances can be solved within polynomial time. Because of complexity of JSSPs and the above drawbacks of exact enumeration method, approximate solution method becomes a feasible choice. The method guarantees the calculation speed at the cost of abandoning the optimal solution. As the approximate method solves problems at quick speed, it can be used for solving large-scale scheduling problems.

Although GA has a few advantages in solving JSSPs, premature convergence and stopping phenomena tend to occur [5]. This is mainly because diversities of a population are damaged during evolution process. However, immune algorithm can sustain diversities of populations because of existence of evolution operators. Therefore, it exhibits certain superiority in solving JSSPs.

Basic ideas of the algorithm

Clone and selection of main operators applied in the immune algorithm are the important bases of biological immune system. That is, descendants are generated to gradually form a population through agamogenesis technology. Compared with common GA, clonal selection algorithm as an improved algorithm changes roulette-wheel selection based on probability into proportional selection based on antibody-antigen affinity and constructs mnemon. As s result, memorizing a single optimal individual of GA is changed into memorizing an optimal population. In addition, replacements of new-old antibodies in the clonal selection algorithm increase diversities of the population and clonal selection algorithm shows a better potential of solving problems than GA. The main steps of the algorithm are as follows:

- A candidate solution set P is produced, which is composed of a mnemon M and the preserved population Pr, namely, P=Pr+M.

- According to measurement of affinity, *n* individual Pn are selected.
- These *n* best individuals in the clonal population produce a temporary clonal population C in the size positively correlated with antibody-antigen affinity.
- High-frequency mutation is conducted on the temporary clonal population. Under this condition, high-frequency corresponds to antibody-antigen affinity, so a mutated antibody population C^* is obtained.
- Improved individuals are reselected from C^* to form a mnemon *M* and some individuals in P are replaced by other improved individuals in C^*.
- Newly generated antibodies replace *d* old antibodies.

Antibody clonal selection algorithm for solving JSSPs

Description of *n*×*m* JSSPs

There are *n* jobs (i=1, 2, …,n) and *m* machines (M_1, M_2, …, M_m). A JSSP is to obtain a feasible scheduling of the following conditions, so that *m* manufacturing procedures of *n* jobs take the least time from the beginning to the end of manufacturing process (n×m problems).

- Each job i contains m manufacturing procedures and corresponding processing time is t_{ij} (j=1, 2, …, *m*).
- Each manufacturing procedure is always processed on specified machines until the manufacturing procedure is finished.
- For all jobs, constraint of sequential order may occur among manufacturing procedures.
- At the same time, one machine can only process one procedure and each job can be processed on at most one machine at certain time.
- The processing sequence of each job is designed in advance for all manufacturing procedures. By conducting in specific flow modes, each job independent of other jobs is successively processed on the machines.

Hyper-mutation antibody clonal selection algorithm

Compared with GA, hyper-mutation antibody clonal selection algorithm significantly improves the performance in solving traveling salesman problems (TSPs). It emphasizes the dominant role of mutation operators in solving TSPs. During clone and clonal selection, the hyper-mutation operation of the algorithm is composed of two basic operators: inversion operator for clonal antibodies and mutation operation. The former is basically the same as GA and the differences lie in that probability of inversion is 1 and it is necessary to ensure different inversion positions of each clone individual. In addition, heuristic mutation operators based on triangle are applied aiming at the inversed antibodies. Hyper-mutation antibody clonal selection algorithm performs as follows:

- In the condition of k=0, after initializing antibody population and setting algorithm parameters, the affinity of the initialized population is calculated;
- Clone operations are performed according to designed antibody clonal scale;
- Hyper-mutation operations are conducted on clonal antibodies including inversion and mutation;
- Clonal selection;
- If k=k+1, calculation ends once satisfying stopping condition; otherwise, it turns to (2).

Solving JSSPs

Hyper-mutation antibody clonal selection algorithm is applied whose core content is to design mutation operators. Mutation operation is carried out to introduce new individuals, increasing diversities of populations, and conducting partial according to a certain procedure through randomly changing certain genes of chromosomes. Common mutation operations include three types: interexchange, reverse, and insert operations. Interexchange operation refers to randomly changing a certain number of different genes at different positions. Reverse operation is to reserve gene series between two random positions. As to insert operation, it means to insert genes at certain position (or gene series in a certain section) before or after another position, and if the two positions are close, the operation is also called drifting. If different coding schemes are used, mutation operators are designed in different methods. The design of mutation operators needs to enable coded antibodies after cloning to realize exchange of processing sequences of local procedures during coding process, so as to achieve local searching in the solution. For *n*×*m* problems, mutation operator is designed as follow. First, *n* positions are randomly selected in the coded string and then reversed operation is conducted on genes in the scope of less than m/2 in both sides of the *n* positions. One conflict procedure should be selected to process when activity scheduling occurs, while the other procedures are collected into the set of next alternative procedures. Different selections within m adjacent selections are equivalent to the selection process in a local search direction in the solution space. In addition, operator selection conducts inverse operations on segments with the length less than *m*. It is improper to have excessive inversed segments, which need to be within *n* due to the constraint of *n*×*m*.

CONCLUSION

In the study, different mutation operators are compared and tested (as shown in Table 1). The test result shows that in terms of small-scale simple

problems, performances of mutation operators are slightly better than those of other methods such as two-point inversion and insertion in most cases while there is little difference among them for certain small-scale problems. For large-scale test problems, mutation operators exhibit good performances, which are far better than those of GA. In terms of large-scale problems, the optimal solution cannot be solved by using traditional GA, while efficiency of the algorithm greatly improves by applying high-frequency mutation operators.

Table-1: Solutions of JSSPs

Problems	Scales	Clonal scales	Optimal solutions
La06	15×5	10	926
Ft10	10×10	15	940
La16	10×10	18	948
Ft20	20×5	20	1170
La35	30×10	50	1888

REFERENCES

1. Giffler B, Thompson GL. Algorithms for solving production-scheduling problems. Operations research. 1960 Aug;8(4):487-503.
2. Gere Jr WS. Heuristics in job shop scheduling. Management Science. 1966 Nov;13(3):167-90.
3. Balas E. Machine Scheduling via Disjunctive Graphs: An Implicit Enumeration Algorithm. Operations Research,1969; 17(1): 941-957.
4. Valls V, Perez MA, Quintanilla MS. A tabu search approach to machine scheduling. European Journal of Operational Research. 1998 Apr 16;106(2-3):277-300.
5. Holland JH. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press; 1992.