Available online at www.sciencedirect.com
ScienceDirect
Journal of Bionic Engineering 5 (2008) 111−119

Clonal Selection Based Memetic Algorithm for Job Shop Scheduling Problems

Jin-hui Yang[1], Liang Sun[1], Heow Pueh Lee[2,3], Yun Qian[4], Yan-chun Liang[1]

1. *College of Computer Science and Technology*, *Jilin University*, *Key Laboratory of Symbol Computation and Knowledge Engineering of Ministry of Education*, *Changchun* 130012, *P. R. China*

2. *Institute of High Performance Computing*, *Singapore* 117528, *Singapore*

3. *Department of Mechanical Engineering*, *National University of Singapore*, *Singapore* 117576, *Singapore*

4. *Institute of Electric and Information Engineering*, *Beihua University*, *Jilin* 132021, *P. R. China*

Abstract

A clonal selection based memetic algorithm is proposed for solving job shop scheduling problems in this paper. In the proposed algorithm, the clonal selection and the local search mechanism are designed to enhance exploration and exploitation. In the clonal selection mechanism, clonal selection, hypermutation and receptor edit theories are presented to construct an evolutionary searching mechanism which is used for exploration. In the local search mechanism, a simulated annealing local search algorithm based on Nowicki and Smutnicki's neighborhood is presented to exploit local optima. The proposed algorithm is examined using some well-known benchmark problems. Numerical results validate the effectiveness of the proposed algorithm.

Keywords: job shop scheduling problem, clonal selection algorithm, simulated annealing, global search, local search

1 Introduction

A typical Job Shop Scheduling Problem (JSSP) can be described as follows: Given n jobs which must be processed on m machines, each job consists of a chain of operations to be processed in a specific sequence, on specified machines, and during an uninterrupted time period of a given length. Each machine can handle at most one job at a time. The problem consists of finding a schedule of the operations on each machine that minimizes the finishing time of the last operation in the schedule. The finishing time of the last operation is termed as makespan. To explain the problem more specifically, we define the set of jobs as $J = \{1,\ 2,\ \cdots,\ n\}$, the set of machines as $M = \{1,\ 2,\ \cdots,\ m\}$, and the set of operations to be scheduled as $O = \{0,\ 1,\ \cdots,\ n\times m,\ n\times m+1\}$, where 0 and $n\times m+1$ represent the dummy initial and final operations, respectively. The operations are interrelated by two constraints. The first one is the precedence constraints, that is, each operation j should be scheduled after all predecessor operations are finished. The second constraint is that operation j can be scheduled only if the required machine is idle. Further, let T_j and F_j denote the fixed processing time and the finishing time of operation j respectively, let P_j be the set of predecessor operations of job j, let $A(t)$ be the set of operations being processed at time t, and let $e_{jm}=1$ if operation j is required to process on machine m and $e_{jm}=0$ otherwise.

The conceptual model of the JSSP can be described as follows:

$$\text{minimize} \qquad F_{n\times m+1}, \tag{1}$$

subject to

$$F_k \le F_j - T_j, \quad (j=1,2,\cdots,n\times m+1;\quad k\in P_j) \tag{2}$$

Corresponding author: Yan-chun Liang
E-mail: ycliang@jlu.edu.cn

$$\sum_{j \in A(t)} e_{jm} \leq 1, \qquad (m \in M; t \geq 0) \qquad (3)$$

$$F_j \geq 0, \qquad (j = 1, 2, \cdots, n \times m + 1) \qquad (4)$$

The Objective function (1) minimizes the finishing time of the last operation, and therefore minimizes the makespan. Inequality (2) imposes the precedence constraints on operations. Inequality (3) means that one machine can process only one operation at a time. Finally, Inequality (4) forces the finish time of each operation to be non-negative.

The solution to the JSSP can be represented as the operational permutation of the jobs on each machine. The total number of all possible schedules (both feasible and infeasible) is $(n!)^m$ for the problems with n jobs and m machines. Obviously, it is impossible to exhaust all the alternatives for finding the optimal solution even if the values of n and m are small. For example, the search space size of the 10-by-10 Fisher-Thompson benchmark problem is about 3.96×10^{65}. So the JSSP is a hard combinatorial optimization problem and computationally challenging. The JSSP is also an important practical problem in the fields of production management and manufacturing engineering. Scheduling problems occur wherever a number of tasks have to be performed with limited resources. The applications of JSSP can be found in production planning, project resource management, distributed or parallel computing, and many other related fields. Efficient methods for arranging production and scheduling are very important for increasing production efficiency, reducing cost and improving product quality.

Due to the computational and practical significance of JSSP, it has drawn the attention of researchers for the last several decades. Existing approaches for JSSP include exact methods such as branch-and-bound[1] and dynamic programming[2], approximate and heuristic methods such as dispatching priority rules[3], shifting bottleneck approach[4], Lagrangian relaxation[5] and tabu search[6]. With the development of artificial intelligence techniques, many meta-heuristic methods, such as genetic algorithm[7,8], simulated annealing[9], ant colony optimization[10], particle swarm optimization[11,12] and artificial immune system[13], have been applied to JSSP. Among the above methods, the population based evolutionary algorithms such as genetic algorithm and particle swarm optimization can be regarded as problem independent approaches and are well suited to solving complex problems. Moreover, they are capable of producing high quality solutions with a reasonable computation effort. However, because they are used with increasing frequency, there may not remain much room for further improvement. On the other hand, it assumes that combining some problem dependent local search methods might improve the performance of evolutionary algorithms. Motivated by these perspectives, we present a clonal selection based memetic algorithm for the JSSP. The clonal selection is a population based evolutionary algorithm which is used for searching different regions of the search space, and the local search operation tries to make sure the individuals are local optimum among their neighbors.

The remaining part of this paper is organized as follows. Section **2** describes the clonal selection algorithm for optimization problem. Section **3** gives a detailed description of the memetic algorithm scheme and introduces the necessary operations for the proposed memetic algorithm. Section **4** presents and discusses some experimental results of the algorithm. Concluding remarks are given in Section **5**.

2 Clonal selection algorithm

According to the theory of clonal selection, when an animal is exposed to an antigen, some of its bone marrow derived cells (B cells) can recognize the antigen with a certain affinity. These B cells will be stimulated to proliferate and mature into terminal antibody secreting cells. Proliferation of B cells is an amitotic process which creates a set of clones identical to the parent B cells. The proliferation rate is in direct proportion to the affinity, i.e. the higher the affinity levels, the higher the proliferation rate. During proliferation, the clones undergo a hypermutation which diversifies the repertoire of the antigen-activated B cells. The receptor edit guides the process of proliferation hypermutation which results in B cells with a high affinity survive.

Motivated by clonal selection theory, we summarize the clonal selection algorithm for the optimization