A multi-modal immune algorithm for the job-shop scheduling problem

Guan-Chun Luh [a,*], Chung-Huei Chueh [b]

[a] Department of Mechanical Engineering, Tatung University, 40 Chungshan N. Rd., Sec. 3, 104 Taipei, Taiwan, ROC
[b] MIM Department, Chenming Mold Ind. Corp., Taipei, Taiwan, ROC

ARTICLE INFO

ABSTRACT

This paper describes the application of an artificial immune system to a scheduling application. A novel approach multi-modal immune algorithm is proposed for finding optimal solutions to job-shop scheduling problems emulating the features of a biological immune system. Inter-relationships within the proposed algorithm resemble antibody molecule structure, antibody–antigen relationships in terms of specificity, clonal proliferation, germinal center, and the memory characteristics of adaptive immune responses. Gene fragment recombination and several antibody diversification schemes including somatic recombination, somatic mutation, gene conversion, gene reversion, gene drift, and nucleotide addition were incorporated into the algorithm in order to improve the balance between exploitation and exploration. In addition, niche antibody was employed to discover multi-modal solutions. Numerous well-studied benchmark examples in job-shop scheduling problems were utilized to evaluate the proposed approach. The results indicate the effectiveness and flexibility of the immune algorithm.

© 2008 Elsevier Inc. All rights reserved.

1. Introduction

Scheduling problems exist almost ubiquitously in real-world applications including distribution, transportation, management, construction, engineering and manufacturing, especially in the industrial engineering world. Many scheduling problems on manufacturing industries are quite complex and very difficult to solve using conventional optimization techniques. It has been the subject of extensive research and captured the interest of researchers from different research communities such as operation research, artificial intelligence, management science, as well as industrial engineering since the early 1950s. Its main focus is concerned with the allocation of finite resources to tasks to improve the resource utilization. It is well known that the job-shop scheduling problem (JSSP) is one of the most complicated and typical production scheduling problems. Scheduling for job shops is an important topic in production management. It is concerned with finding the operations and times of a set of jobs on the relevant machines subject to the processing constraints. The purpose is to improve the production efficiency and reduce the processing duration so as to gain profits as high as possible. The JSSP may be described as follows: given n jobs, each composed of several operations that must be processed on m machines. Each operation uses one of the m machines with a deterministic processing time and each machine can process at most one operation at a time. Once an operation initiates processing on a given machine it must complete processing on that machine without interruption. Each job consists of a specific set of operations, which have to be processed according to a given technical precedence order. The operation sequences on the machines are usually scheduled to minimize makespan, the total time required to complete all jobs. The total number of all possible schedules including feasible and infeasible solutions is $(n!)^m$. Apparently,

---

* Corresponding author. Tel.: +886 2 25925252x3410x806; fax: +886 2 25997142.
   E-mail address: gluh@ttu.edu.tw (G.-C. Luh).

it is impossible to exhaust all the alternatives for finding the optimal solution even though very small n and m values. A comprehensive survey of job-shop scheduling techniques can be found in [21].

The JSSP is one of the well-known hardest combinatorial optimization problems whose complexity grows very fast as the problem size increases. It has been illustrated [17,41] that job-shop scheduling is usually an NP-hard combinatorial problem and is therefore unlikely to be solvable in polynomial time. Brucker et al. [4] have proposed a branch-and-bound approach to find optimal schedules, but only for small size problems since it required considerable computation time. In the last decade, numerous techniques and methodologies imitating physical processes or natural phenomena especially biological-inspired methods have been employed to solve practical size problems. Among these includes simulated annealing [23,39], tabu search [32,36], neural networks [38,47], ant colony optimization [7,50], bacterial evolutionary algorithm [29], genetic algorithms [1,8,33,42], and immune algorithm [13,44,45,51].

Genetic algorithms (GAs) – powerful tools based on biological evolution mechanisms and natural selection theory [18] – have received considerable attention as the optimal design efforts. Methods based on GAs have become a popular and effective way for solving large-scale combinational optimization problems, including job-shop scheduling. Genetic algorithms are considered powerful in terms of global optimization; nevertheless, they have essential drawbacks regarding local searches. Tazawa et al. [40] have identified two of them as lack of local search ability, and premature convergence. Consequently a number of researchers have experimented with biological immunity-based optimization approaches to overcome these particular drawbacks implicit in genetic algorithm. Based on these research efforts, Bersini and Varela [3] offered a genetic immune recruitment mechanism to improve the local search ability of GAs, but failed to add preventive measures against premature convergence. Mori et al. [31] then suggested an immune algorithm using a sharing-like GA approach to prevent premature convergence, but without any control mechanism to create a balance between local and global searches. Later, Jiao and Wang [22] proposed an immune genetic algorithm with improved and faster global convergence. Yoo and Hajela [48] employed the immune system to solve multi-objective optimization problems. The best designs according to fitness value were defined as antigens while the rest of the population as a pool of antibodies. Antibodies were evolved through crossover and mutation operations. Fukuda et al. [15] used the somatic theory and network hypothesis of immune system while de Castro and Von Zuben [11] employed the clonal selection principle to illustrate the abilities of immune algorithm for multi-modal optimization problems. Nonetheless it is important to emphasize that genetic algorithms still serve as the framework for all the hybrid approaches mentioned above. The basic role of immunity is simply to support diversity via different levels of inter-antibody affinity, even though natural immune systems have a powerful capacity to diversify, to learn, memorize, and process information, and to discriminate between self and non-self when reacting to foreign pathogens [10,12]. Until recently, immune algorithm emulating the entire features of biological immune system was proposed for solving multi-objective [27] and multi-modal [28] optimization problems.

Several immune algorithms have been proposed for solving Scheduling problems [9]. An artificial immune system (AIS) was utilized for a dynamic job-shop scheduling problem in which jobs arrive continually, and the environment is subject to change due to practical reasons [20]. Concepts emulating natural immune system including somatic hyper-mutation, antibody niches, simple recombination, somatic recombination, and gene library were mentioned to improve the performance and robustness of the AIS. However, genetic algorithm was used to evolve the antibody population even though the antibody has its own evolution procedure. Then, Engin and Döyen [13] proposed a computation method based on clonal selection principle and affinity maturation mechanism for solving the hybrid flow shop scheduling problems. Cloning, mutations, and receptor editing are utilized in the evolutionary procedure. Coello Coello et al. [6] used clonal selection, hypermutations, and a library of antibodies to construct solution of JSSP. A local selection mechanism was used to improve solutions produced. Thirty one test problems from the OR-Library [2] were applied to evaluate the proposed algorithm and AIS was able to find the best known solutions in 38.7% of the problems. Later, Zuo and Fan [51] proposed an immune algorithm for JSSP. In addition to antibody crossover, antibody clone, hypermutation, receptor editing, niche technology was used to keep the diversity of the population. A benchmark problem FT06 was employed to verify the effectiveness of the method. More recently, Chandrasekaran et al. [5] proposed AIS algorithm for finding optimal makespan values of different size JSSPs. The algorithm was based on clonal selection and affinity maturation principles. A two-phase mutation procedure and receptor editing were used to generate new antibodies. The algorithm was tested with 130 benchmark problems. The results show that the proposed algorithm gives optimum bound value 96 out of the 130 problems.

To highlight the significant features and completeness of the immune system, MMIA [28] is utilized in this study to solve the job-shop scheduling problem. Inter-relationships within the proposed immune algorithm resemble antibody–antigen relationships in terms of specificity, heavy-chain/light-chain structure of antibody molecule, germinal center, clonal proliferation and the memory characteristics of adaptive immune responses. Moreover, gene fragment recombination and six antibody diversification schemes including somatic recombination, somatic mutation, gene conversion, gene reversion, gene drift, and nucleotide addition are incorporated to improve the balance between exploitation and exploration. In addition, it is intuitive that there exist many different operations with equal makespan as the number of jobs and machines increase in JSSP. This suggests that with the increase in sizes in the job shop, the resulting JSSP becomes multi-modal and hence a niche scheme is essential to find all the possible solutions. However, no research has mentioned this previously. Finally, several well-studied benchmark examples in JSSP were subsequently used to evaluate the performance and effectiveness of the proposed immune algorithm.