SOLVING FLEXIBLE MANUFACTURING SYSTEM DISTRIBUTED SCHEDULING PROBLEM SUBJECT TO MAINTENANCE: AN ARTIFICIAL IMMUNE SYSTEM APPROACH

MOHD NOR AKMAL KHALID, UMI KAL SOM YUSOF AND AHAMAD TAJUDIN KHADER
School of Computer Sciences
University Sains Malaysia
11800 USM, Penang, Malaysia
mnak104041@student.usm.my; {umiyusof; tajudin}@cs.usm.my

Received May 2012; accepted July 2012

ABSTRACT. Flexible manufacturing system (FMS) in a distributed factory environment has attracted researchers’ attention in these recent years. Various introductory approaches were proposed to address the distributed production scheduling dilemma of FMS, where a significant part of the problem lies with preventive maintenance machine downtime. This affects overall production schedules, rendering previously proposed approaches ineffective. This paper introduces the Artificial Immune System (AIS) approach to tackle this underlying problem. Using the antigenic clustering features inherent in AIS, the problem will be replicated and tested with several parameter tweaks to demonstrate its effectiveness in mitigating the issue.

Keywords: Distributed manufacturing, Flexible manufacturing system, Artificial immune system, Preventive maintenance

1. Introduction. FMS is a manufacturing system that has the advantage of producing multiple product types in the absence of production line reconfiguration. Rapid turn-around, higher quality output, lower inventory costs, and lower labor costs make it the most desirable manufacturing system in today’s industrial sectors [1]. Distributed scheduling (DS) is a production system that follows a globalization model where geographically distributed multi-factory production plants are capable of processing their own productive parts [2]. Every factory conforms to different production efficiencies and constraints depending on the machine, labor cost, labor skill, and transportation facilities available. From this observation, different production lead time, operating costs, and makespans can be identified within the individual factory to determine the production schedule [3]. In the real world, downtime of machines is inevitable after hours of operation, rendering them unavailable for some time. This consumes more time and cost in the overall production schedule. The production rate, quality, machine availability, and utilization ratio, are greatly affected. Additionally, predetermined plans will be interrupted due to the mismatching processes [4].

This paper aims to propose an optimal solution for distributed FMS scheduling, while taking preventive maintenance into account, by using AIS. The remainder of this paper is organized as follows. Section 2 discusses on the background study. Section 3 highlights problem of DS subject to maintenance. Section 4 emphasizes on the proposed solution. Section 5 elaborates on the results obtained. Finally, Section 6 concludes the paper.

2. Background Study. FMS production scheduling problem in single manufacturing system has generated great interest from researchers in years past. Lee and DiCesare [5] and Reyes et al. [6] had formulated a Petri net model in an attempt to provide a solution to this dilemma. A different approach was proposed by Kumar et al. [7], utilizing an ant colony optimization model. Production scheduling in distributed manufacturing was
explored by Jia et al. [8] where a hybrid of genetic algorithm and Gantt chart to optimize job shop scheduling was introduced. Similarly, Shen [9] proposed a hybrid of agent-based algorithm and genetic algorithm in the distributed scheduling system.

Preventive maintenance in distributed FMS scheduling has also been studied by some researchers in recent years but information is scarcely available. Chan et al. [2] introduced genetic algorithm with the idea of dominant genes. A very large number of generations were needed as algorithm proposed is unable to escape the local optima. In overcoming the research setback, Chan et al. [3] enhanced the algorithm with premature and local search strategies. Chung et al. [4] conducted three separate studies of the machine maintenance model and its influences by using the approach proposed by Chan et al. [2,3]. Separate studies conducted by Yadollahi and Rahmani [10] introduced a mimetic approach where a combination of genetic algorithm and tabu search was proposed to solve the distributed FMS scheduling problem.


The inspiration to undertake this research arose from seeing a promising, yet challenging, opportunity to employ a well-known algorithm, AIS, as a suitable solution to tackle a significant issue faced in FMS distributed scheduling subject to maintenance. The presence of recent comparative literature pertaining to this subject also encouraged this.

3. Problem Description. Several numbers of jobs (i) are expected to receive in the distributed network and the job assigned with a suitable factory (f = 1, . . ., F) in order to generate corresponding production scheduling. Various product types are produced in every factory which has several machines (h = 1, 2, . . ., Hf) with different efficiency or operating lead time (Tijfh). Each job has up to Ni numbers of operations, and each operation can be performed in more than one machine (not all), but operating in the same factory. Dif symbolized the traveling time between the factory f and the job i.

Each machine is assigned a maximum machine age (M), where the machine age is equal to the cumulated processing time of operations. Maintenance has to be carried out right after the completion of the current operation when the machine age reaches M. The machine age of the particular machine will be reset to 0 when the maintenance occurred.

The objective of the study is to minimize the total maximum makespan of the jobs. As such, the objective function is defined as in Equation (1). Completion time (Ci) is defined as the summation of the completion time of the last operation (Ni) of the job i which obtained from completion time of job Eij, and the delivering time between the factory f and the job i considering χif parameter as 1 if job i is allocated to factory f. This is defined in Equation (2).

\[
\text{Objective } Z : \min (\max \{C_i\}). \tag{1}
\]

\[
C_i = E_{iN_i} + \sum D_{if} \chi_{if}. \tag{2}
\]

The problem is subject to the following constraints:
- Completion of the preceding operation required for every current operation to begin.
- Completion without interruption will be carried out once an operation starts.
- Time slot allocated is equal to required operation time.
- Each operation can only be carried out by one machine throughout the horizon.
• Each operation can be processed by one machine at each unit time.
• Each machine can only process a single operation at a unit time.
• Each job can only be assigned to a single factory.

4. Solving Distributed Production Scheduling Subject to Main tenances Using Artificial Immune System (AIS).

4.1. Artificial immune system implementation. The immune-inspired Algorithm (IA) or Artificial Immune System (AIS) is an emulation of the immune system of the human body which was inspired into an evolutionary technique [11,16]. To understand how AIS works, the following immunology terms need to be emphasized:

• Immune Cells: Two major groups of immune cells are known as B-cells and T-cells. These cells help to identify almost every antigenic pattern present in the body. These cells are the individual candidate population and complete solution representation of the algorithm.

• Antigens: Antigens are a disease causing element in the immune system and can be categorized as self and non-self-elements. Self-antigens are the harmless elements and non-self-antigens are the harmful elements in the immune system. These represent the available optimal or worse solution of the problem it represented.

• Antibodies: Antibodies are the molecules produced by B-cells which respond to stimulation from a non-self-antigen. Every individual antibody has the property of binding to an antigen, which will induce the formation of this antibody. This antibody molecule acts as a single schedule which builds the complete candidate solution.

The whole process of AIS starts with the initialization of the population size, generation number and clonal selection percentage. The initialization mechanism and population generation details will be given in Subsection 4.2. Next, the clonal selection is performed by first evaluating individual population affinity value and arranging them in decreasing order. Then, \( N \) percentage of the total population is selected. This is represented by user-defined clonal selection percentage. Affinity is defined as a successful binding of antigen and antibody which is used as a measuring instrument for evaluation purposes of the solution candidate. When the suitable affinity threshold is achieved, it will differentiate into both memory cells and effector cells [11]. The clonal selection, breeding and affinity maturity can be demonstrated in Figure 1.

The clonal selection principle is a selection mechanism of the population which undergoes hypermutation and receptor editing. Hypermutation shares a similar behavior pattern with Genetic Algorithms. Both have a mutation operator that either randomly generates a string or decimal, or randomly flips a binary digit of the candidate solution. AIS differs to Genetic Algorithm because inferior antibodies hypermutate at a higher rate compared with the antibody with higher antigenic affinity. This process is known as receptor editing. Both hypermutation and receptor editing act as an exploratory and exploitation mechanism of the search space in the optimization domain. Figure 2 shows the overall flowchart of the AIS process.

4.2. Candidate solution/antibody and antigenic cluster. The candidate solution is a solution representation of the problem. The complete production scheduling represented by several component variables formed within an antibody. Each antibody consists of \( \sum N_i \) numbers of schedules for every operation of the jobs and each schedule is represented by five parameters, (FMJOS), which are adopted from [2-4]. The sample antibody is shown in Figure 3. \( F \) represents factory, \( M \) represents a machine assigned, \( J \) is the processing job, \( O \) represents the job current operation, and \( S \) represents machine scheduled for maintenance after the operation finished. If maintenance is scheduled, the \( S \) parameter will be denoted as 1, otherwise 0. The maintenance is scheduled randomly.
Figure 1. Clonal selection, breeding, affinity maturity process adopted from [16]

Figure 2. Overall flowchart of the proposed AIS

Component Variable

<table>
<thead>
<tr>
<th>F</th>
<th>M</th>
<th>J</th>
<th>O</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Sample Antibody/Candidate Solution

<table>
<thead>
<tr>
<th>F</th>
<th>M</th>
<th>J</th>
<th>O</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 3. Sample antibody/candidate solution representation

throughout the corresponding production scheduling. The priority between the schedules is from left to right. As an example, job 4 of operation 1 is assigned to machine 2, then job 1 of operation 1 assigned to the same machine.

The random generation of an antibody (production scheduling) may result in an illegal schedule which conflicts with the constraints described in Section 3. As such, the antigenic cluster concept is introduced to initialize an antibody with a controlled antigen (solution) which stimulates the antibody to differentiate and mature into a high-affinity antibody (optimal solution). The high-matching antigen will be clustered together and random selection will be performed on the clustered antigen. The antigen represents a single machine-job-operation option which adheres to the constraints. After the binding process of antigen and antibody, the factory parameter will be randomly allocated with respect to the constraints. Maintenance scheduling will also generate randomly on each component variable of the antibody.

5. Results and Discussion. The proposed AIS approach was tested with the distributed problem adopted from [2,3,5,7], which has 2 factories, 10 jobs, 4 operations, and 3 machines in each factory. Implemented using C# compiler, the proposed AIS approach ran 5 times independently with 100 generations. Different parameters were tested including the population size of the candidate solution and the user defined $N$ percentage clonal selections. Option 1 with 0.25, Option 2 with 0.45, Option 3 with 0.65, and Option
with 0.75 of $N$ percentages clonal selection. Each option was tested with sample population sizes of 50, 75, 150 and 300. For every population size, the best makespan is taken from the best of the 5 run trials. The simulation results obtained can be referred in Figure 4. The maintenance model used is a linear relationship between the maintenance time required and the machine age, where maintenance time equals three times the machine age. For example, if the machine age is 40 units time, then the maintenance time is 120 units time.

In all cases of population sizes, the average and minimum makespan obtained in option 4 remain at the lowest compared with option 1. High number of $N$ percentage clonal selection value in option 4 involves more population being hypermutated and cloned to achieve the optimal solution. As for option 1, $N$ percentage of the clonal selection value resulted in a lower population coverage, which narrowed the search space. More generation numbers can be used to achieve the optimal solution. On the other hand, the average and minimum makespan obtained in option 2 and option 3 yielded slightly higher or lower compared with both option 1 and option 4. This irregular in behavior and unpredictable happened because $N$ percentage clonal selection value of option 2 and option 3 is approaching the middle point, where the probability of achieving an optimal solution is halved. As such, the chances of obtaining optimal solutions are much better for option 4 compared with other options. Additionally, an increase in population size affects the overall makespan obtained, as observed from the diffraction pattern shown in option 1 and 4 between different population sizes. Thus, it can be concluded that higher population sizes may result in lower makespan values. This is possible as the search space of the candidate solution expands with higher population numbers, resulting in an optimal solution achievable through fewer generations.

The results obtained were also compared against GADG and MGADG1 approaches proposed by Chan et al. [2,3] and MGADG2 by Chung et al. [4], as shown in Figure 5. From the results, it can be observed that the proposed approach performed better than most approaches available in the literature. The approach managed to achieve the best minimum and average makespan value of 797 and 853 respectively.

6. **Conclusion.** The main objective of the study is to develop an efficient AIS approach to solve the FMS distributed scheduling problem subject to maintenance. Improvisation includes parameters tuning and comparisons have been made to justify the performance and optimization capabilities of the proposed approach. An optimal or near-optimal result obtained poses an encouragement for future improvement in more sophisticated and challenging environments. However, the problem instance is obtained from literature and does not consider real world manufacturing problems. Hence, designing a real world solution that effectively addresses this issue has yet to be accomplished. Further research can be done to improve the AIS approach by considering larger problem instances, employing
different maintenance models, utilizing larger job and factory data, and by adopting a more comprehensive analysis with more extensive parameter considerations.

**Acknowledgment.** The main author wishes to thank Universiti Sains Malaysia for the support it has extended in the completion of the present research through grant number (1001/PKOMP/817047).

**REFERENCES**


