

# An Improved Immune Algorithms for Solving Flexible Manufacturing System Distributed Production Scheduling Problem Subjects to Machine Maintenance

Mohd Nor Akmal Khalid  
School of Computer Sciences  
University Sains Malaysia  
11800 USM, Penang, Malaysia  
Email: mnak104041@student.usm.my

Umi Kalsom Yusof  
School of Computer Sciences  
University Sains Malaysia  
11800 USM, Penang, Malaysia  
Email: umiyusof@cs.usm.my

Received: February 20, 2021. Revised: March 12, 2021. Accepted: March 16, 2021. Published: March 26, 2021.

**Abstract**—Competitiveness and rapid expansion of flexible manufacturing system (FMS) as one of the industrial alternatives has attracted many practitioners' and academicians' interest. Recent globalization events have further encouraged FMS development into distributed, self-reliant units of production center. The flexible manufacturing system in distributed system (FMSDS) considers multi-factory environments, where jobs are processed by a system of FMSs. FMSDS problems deal with the allocation of jobs to factories, independent assignment of job operation to the machines, and operations sequencing on the machine. Additionally, in many previous studies, impact of maintenance as one of the core parts of production scheduling has been neglected. This significantly affects the overall performance of the production scheduling. As such, maintenance has been considered in this paper as part of the production scheduling. The objective of this paper is to minimize the global makespan over all the factories. This paper proposes an Improved Immune Algorithm (IIA) to solve the FMSDS problem. Antibody encoding adoption explicitly represents the information of factory, job, and maintenance, whilst a greedy decoding procedure exploits flexibility and determines the job routing. Rather than a traditional mutation operator, an improvised mutation operator is used to improve the solutions by refining the most promising individuals of each generation. The proposed approach has been compared with other algorithms and obtained satisfactory results, where the algorithm performance has been tested with several parameter tunings.

**Keywords**—flexible manufacturing system, distributed system, machine maintenance, immune algorithm.

## I. INTRODUCTION

The production scheduling problems in the manufacturing industry has for several years attracted many research initiatives. Advancement and development in computer technology has promoted the way of solving production scheduling problems to a whole new level. Exact approaches are insufficient to handle the complex and changing environment of production scheduling. Production scheduling involves allocating a limited amount of resources (i.e. machine) to a number of task over time. Significant numbers of feasible solutions available from different task-resource assignments makes the production scheduling problems one of the *NP*-hard problems [1]. As

such, this importance serves to encourage both practitioners and academicians to solve the production scheduling problems firsthand.

High market competition and a challenging manufacturing environment has evolved the way organization attain success and competitive advantage. Flexible manufacturing system (FMS) is the result of growing demand both in terms of quantity and quality. Combinations of both efficiency of high-production line, and the flexibility of job shops, correspond well with mid-volume batch production and mid-variety of products [2]. The high investment value in FMS has increased the importance of effective and well-rounded performance, while utilizing resources efficiently. Extensive studies regarding FMS has been thoroughly investigated from the 1980s, where the concern over FMS was mostly concentrated in the area of allocation, scheduling, loading and control problem.

Production scheduling of a single-factory is devoted to minimizing the total operating cost, completion time, and order fulfillment of an assigned machine to process the operations of a job. The recent trend of globalization has cultivated the emergence of the distributed system (DS) in production scheduling. Generally, DS can be defined as a multi-factory production where each factory is geographically distributed and owns the ability to process product parts independently. Every factory has a distinctive production line in terms of efficiency and constraints, which depends on machine availability, labor costs and skill, and transportation facilities. These indirectly yield different production lead times, operating costs, and completion times [3], [4]. An exact solution becomes insufficient to handle the production scheduling problem considering the different alternatives of process plan combination in DS. As such, some of the recent work that considers these features of FMS have been reported [3], [5], [6]. Concerning the Flexible Manufacturing System Distributed Scheduling (FMSDS) problem, optimization of the production schedule involves three hierarchical problems that need to be solved sequentially or simultaneously [7], [8]:

- 1) Allocation of the most suitable factory for the job (assignment problems).

- 2) Routing of the most suitable machine for each of the assigned operations of the job within the given factory (routing problem).
- 3) Sequencing the most suitable assignment of the operations to machines over the time span (sequencing problem).

In a real manufacturing environment, machine maintenance is unavoidable. Unexpected machine breakdown (stochastic unavailability) and scheduled preventive maintenance (deterministic unavailability) are the main reasons for a machine to be unavailable for a period of time [9]. The importance of machine maintenance has escalated the attention of most researcher because it directly affects the production rate, product quality, machine availability and utilization ratio [4]. Nonexistent machine maintenance also disrupts the predetermined planning or scheduling due to various process mismatching. As such, the maintenance policy in production scheduling plays a major role in perpetuating the machine availability and utilization ratio, while maximizing the facility with minimum cost and reducing unforeseen breakdown. To the best of our knowledge, the first work addressing all the features of FMSDS and maintenance consideration are [4], proposing Genetic Algorithm with Dominated Genes (GADG).

Development of Immune Algorithm (IA), inspired from the adaptive natural immune system of vertebra, has been implemented in various ranges of problems, among them, machine learning, pattern recognition and detection, scheduling, intrusion detection, data manipulation and analysis, evolutionary computation, and optimization [10], [11], [12], [13], [14], [15], [16]. IA features that are not limited to self-organizing, adaptivity, and uniqueness, proffer various developments of computational models applied in business [10], [12], [16], [17], [18], sciences and engineering [14], [15], [19], [20], and optimization domain [11], [20], [21], [22]. Generally, IA is renowned for its criteria of memory cells (reservation of good solutions), high rate somatic mutation or hyper-mutation (explorative and/or diversification mechanism), and receptor editing (escaping local optima, adaptivity) [23].

Despite the available IA literatures, inspiration to undertake this research arose from an encouraging yet challenging opportunity to employ a renowned IA as an ideal choice to address the underlying problems in FMSDS subject to maintenance. The objectives of this study includes proposing a feasible IA algorithms with guided initialization mechanism, yielding optimal makespan while considering the impact of maintenance inclusion.

The rest of this paper is organized as follows: the formulation and constraints of the problem are described in Section II. The algorithm overview and the proposed algorithm are presented in Section III. Computational study performed with the presented algorithm and its results are reported in Section IV. Section V concludes the paper and highlights future works.

## II. THE FMSDS PROBLEM

The FMSDS problem can be stated as follows: a numbers of jobs ( $i$ ) is expected to be received in the distributed network and a suitable factory ( $f = 1, \dots, F$ ) will be assigned to the job in order to generate corresponding production scheduling. Each individual factory has a number of machines ( $h = 1,$

$2, \dots, H_f$ ) with different efficiencies or operating lead times ( $T_{ijfh}$ ) in producing various product types. Each job has up to  $N_i$  operations, and every operation can be performed in more than one machine (not all), but must be in the same factory. The traveling time between factory  $f$  and job  $i$  is symbolized as  $D_{if}$ .

Each machine conforms to a maximum machine age ( $M$ ), where the machine age equals to the cumulated processing time of operations. A maintenance procedure has to be carried out right after the completion of the current operation when the machine age reaches the threshold denoted as  $M$ , outlined in [4]. After every maintenance, the machine age of the particular machine will be reset to 0.

The objective of the study is to minimize the total maximum makespan of the last job operation. As such, the objective function is defined in (1). Completion time ( $C_i$ ) is defined as the summation of the completion time of the last operation  $N_i$  of job  $i$  and the delivering time between the factory  $f$  and the job  $i$ , as defined in (2). The decision variables are:  $\chi_{ij}$  denoted true if job  $i$  is allocated to factory  $f$ ;  $\delta_{ijfhk}$  if operation  $j$  of job  $i$  occupies time slot  $k$  on machine  $h$  in factory  $f$ ; and  $\gamma_{ijfh}$  if machine  $h$  in factory  $f$  is maintained after operation  $j$  of job  $i$ ; Once obtained, the starting time value of operation  $j$  of job  $i$  ( $S_{ij}$ ), ending time of operation  $j$  of job  $i$  ( $E_{ij}$ ) and completion time ( $C_i$ ) can be calculated.

$$\text{Objective } Z : \min(\max\{C_i\}). \quad (1)$$

$$C_i = E_{iN_i} + \sum D_{if}\chi_{if}. \quad (2)$$

The problem is subject to the following constraints:

- 1) Every operation can only begin after the completion of the prior operation.
- 2) An operation will continue to commences until it finishes without any disruption.
- 3) Assigned time slot must be equal to the required operation time.
- 4) Each operation to be carried out on a single machine throughout the horizon.
- 5) each operation to be executed on a single machine at each unit of time.
- 6) Each machine to handle only a single operation at each unit of time.
- 7) Each job can only be assigned to a single factory.

## III. AN ARTIFICIAL IMMUNE ALGORITHM FOR FMSDS

### A. General IA

The IA is a “collection” of complex adaptive pattern recognition system that mimics the natural immune system which defends the organism’s body from foreign pathogens (bacteria or viruses). The system is capable of recognizing all cells (or molecules) within the organism as either harmful (non-self-cell) or harmless (self-cell) [24]. In typical infection process, infestation and proliferation of a pathogen within the organism occurs. Pathogens are the correspondence of specific proteins (antigens). The immune cells (antibodies) are randomly distributed throughout an immune system capable

of recognizing antigens and killing pathogens. Additionally, the immune system can also respond effectively to future infections from an earlier encountered pathogen (adaptive immunity) [1].

The clonal selection and affinity maturation hypothesis are used to elaborate the immune system responding to pathogens and enhance its capability of recognizing and eliminating pathogens [24]. Clonal selection can be defined as the immune system's reaction to pathogens that invade the organism in such a way that the immune cells are cloned and proliferated, where some of the cloned cells become effector cells while others with high affinity threshold will be sustained as memory cells. The effector cells secrete a large number of antibodies with an antigenic signature as their external receptors whereas the memory cells with long life spans will effectively drive the immune system to a much faster response in future exposure to the same or similar pathogen. During cellular reproduction (cloning and proliferate), the cells suffer high rates of somatic mutation (hyper-mutation). The process of somatic mutation and clonal selection is known as affinity maturation [1]. These aforementioned hypothesis that form the building block of a very complex natural immune system in order to defend against pathogenic organisms, act as a source of inspiration for solving optimization problems. As such, the Immune Algorithm (IA) is a meta-heuristic which is developed based on such a system. This paper aims to propose an immune algorithm for solving FMSDS.

**B. Proposed Improved Immune Algorithm (IIA)**

In order to illustrate the proposed IIA procedures, the overall flows of the proposed IIA are depicted in Fig. 1. The first procedures involve setting up the parameters where user-defined parameters such as population size ( $pop_N$ ), generation number, and clonal selection percentage ( $C_r$ ) are given with an individual value. The procedure is then followed by the initialization of the populations, population ranking, and clonal selection, which details are described in Subsection III-B2. A simple encoding during the initialization phase and greedy-based decoding scheme during evaluation phase is conducted where the detail regarding this encoding and decoding are discussed in the following Subsection III-B1. After the clonal selection phase, a set of individuals is selected from the total population size where cloning is performed first, followed by the somatic mutation (hyper-mutation) conducted on the cloned individual. At this point, only the local mutation operators are involved. Next, a receptor editing (global mutation) is performed on one or more individuals of the population based on a probabilistic scheme. Then, the best among cloned individuals will be retained as an immune memory for the remaining generation numbers (iterations). If the termination condition (maximum generation number) is met, the proposed IIA terminates.

1) *Antibody Encoding and Decoding:* Information encoded in the antibody of the IIA for FMSDS has to specify the allocation of each job to factory, the routing of every job through machine, and the sequence of the operations. Basically, this work reuses the simple operation-based encoding method proposed in [7] for the distributed scheduling problems without routing flexibility, where relevant extension that includes the flexibility issues of the FMSDS is considered. The size of

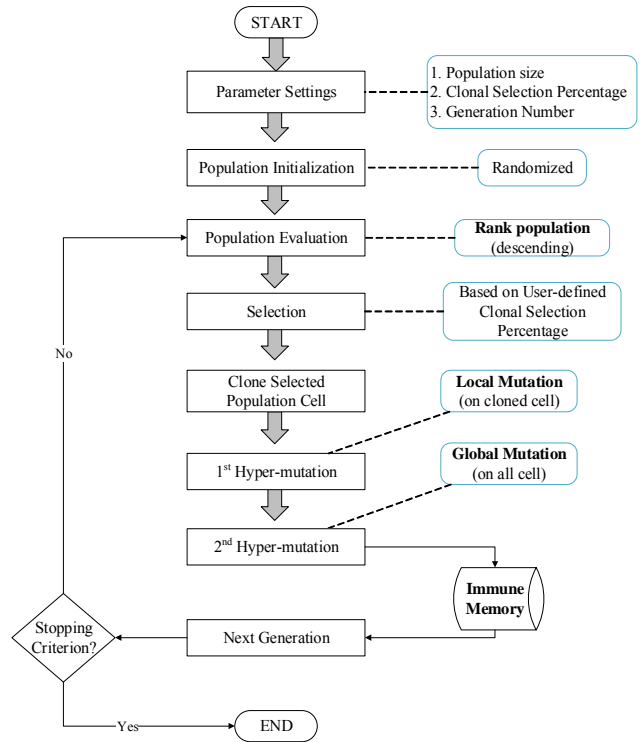


Fig. 1. Flowchart of proposed IIA

receptor ( $r_p$ ) in an antibody is equal to the total number of operations of all the jobs. Every receptor is represented with a triplet notation  $(f, i, p)$ , which denotes the factory ( $f$ ), the assigned job ( $i$ ), and the PM flag ( $p$ ). Note that all the operations of the same job are represented by different receptors within the same antibody, which interpret according to the order of the receptor occurrence on the antibody, given that the order for the operation of a job is fixed. Concerning the adoption of the simple representation as per [7], no information about alternative machine routes is explicitly encoded into the receptor. This information will be retrieved during the decoding phase. A sample individual is given in Fig. 2.

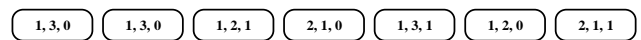


Fig. 2. A sample antibody encoding

Lets assume that job 1, job 2, and job 3 have two, two, and three operations respectively so that an antibody consists of 7 receptors. Each receptor consists of three types: “ $2,1,<p>$ ”, “ $1,2,<p>$ ”, and “ $1,3,<p>$ ”, meaning that jobs  $N_1$  and  $N_2$  are processed in factory  $F_1$  and  $N_3$  is processed in factory  $F_2$ .

The decoding process exploits the information provided by each antibody in order to generate a schedule plan where the affinity of each individual is evaluated. The objective of the FMSDS is to minimize the global makespan of the factory network so that the affinity of an individual is inversely related to the global makespan.

As previously discussed, antibodies explicitly represent information on job assignments to factories and the order of the antibody's receptor is relevant to determine the priority

of each operation, with no information on job routing considered. Rather than complicating gene encoding, the flexibility problem is considered in the decoding phase, where it can dispatch job operations to one of the alternative machines of the selected factory. The information on job routing is thus implicitly conducted in the decoding process. Based on the order determined by the antibody, operations are considered sequentially. When the respective operation is dispatched to a machine, the starting time equals the completion time of the last operation assigned to the machine. If the considered operation requires more than one machine, the decoding process selects the routing that guarantees the lowest current local makespan where the one giving the lowest completion time for the operations assigned so far is picked. However, if different routings lead to the same current makespan, the machine with the smallest processing time is chosen. If the available machines have the same smallest current makespan and processing time, any of them is selected at random, to give the optimization algorithm the opportunity to search different regions of the solution space. The decoding process is completed by adding the delivery time (according to the factory the job is assigned to) as soon as all the operations have been scheduled, thus obtaining the local makespans and the global one.

2) *Population Initialization and Clonal Selection:* The initial population is determined by three phases: the first phase randomly generates jobs until all the operations of the jobs are generated; the second phase randomly assigns jobs to factories in which related operations of the respective jobs will be amended to satisfy the factory allocation constraints; the third and last phase generates the maintenance flag at random. This process repeats until all individuals of the population ( $pop_N$ ) are initialized.

During the clonal selection phase, a set of individuals from the current population is chosen in order to apply IIA operators and generate high affinity memory cell(s) to include in the next generation. Since the clonal selection is dependent on the affinity (makespan) of the antibody, a ranking strategy is conducted by sorting the population by decreasing affinity (starting from the best to the worst individual). The  $C_r$  % of the best population will be considered for cloning, in which each of these cloned cells undergo affinity maturation process, whereas the rest of the population will be re-initialized with the three phase initialization mechanism previously described.

3) *Hyper-mutation Operators:* In this study, the mutation operators shares a coherent behaviours with Genetic Algorithm (GA) where both have a mutation operator that either randomly generates a string or a decimal, or randomly flips a binary digit of the individual. However, IIA mutation operator differs in such, dependent on individual affinity, inferior antibodies mutate at a higher rate compared to superior antibodies. This process is known as somatic mutation (hyper-mutation). To comply to this requirement, the mutation operator is conducted in a continuously loop in where the loop limit is calculated as below:

$$R_m = Round\{(1 - A_{pop_n}) * pop_N\} \quad (3)$$

$R_m$  is the mutation rate,  $A_{pop_n}$  is the affinity of the  $n$ th population, and  $pop_N$  is the total population size. The somatic

mutation employed can be categorized into two; local and global.

a) *Local Mutation:* The local mutation is involved in exchanging information of the antibody's receptor, which is conducted on a single antibody when only routing of the operations of jobs affected (local effects). This aims to enhance the algorithms to better examine the search space. Two types of local mutations operator are adopted, **uniform** and **exploration**. The uniform mutation operator is where the mutation is conducted repeatedly in the somatic mutation loop, whereas the exploration mutation operator mutates based on user-defined probabilities within the somatic mutation loop.

Simple swapping mechanism (SSM) is a uniform local mutation operator which randomly selects a pre-defined number of pair of receptors within a single antibody to permute their positions (Fig. 3). However, an end-to-end swapping mechanism (EESM) is employed as an exploration local mutation operator which exchanges first and last pairs of every receptors within a single antibody to permute their positions (Fig. 3). Note that every antibody explicitly encodes just the jobs. However, exchanging the antibody's receptor does not effect the feasibility of scheduled routing of jobs.

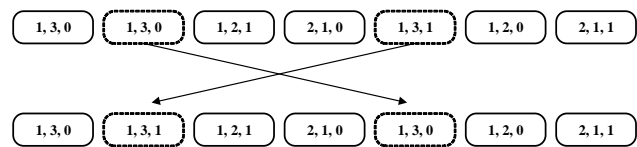


Fig. 3. A simple swapping mechanism (SSM)

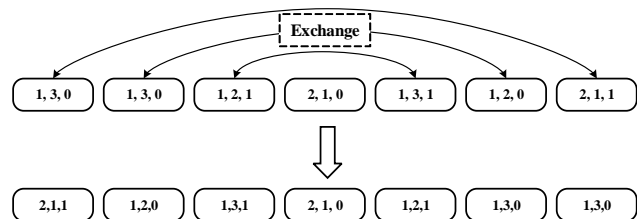


Fig. 4. An end-to-end swapping mechanism (EESM)

b) *Global Mutation:* The global mutation involve exchanging information of the antibody's receptor, which is conducted on a single antibody at a time involving the factory assignment of jobs and the maintenance flag. This aims to explore more solutions of the search space with different assignments of jobs to factories and varied scheduled maintenance. Note that, in order to maintain consistency with the antibody's remaining receptors and meet the factory constraint, all the receptors have to reflect the new job assignments in which all the receptors related to the selected job in the antibody have to be updated (global effects). The updating process is conducted on all antibodies after the last immune operator to maintain the antibody's feasibility (receptor editing).

Two types of global mutation considered (Fig. 5): random factory assignment (RFA) and random scheduled maintenance (RSM). Due to the significant impact global mutation has on operation scheduling, it is applied at some iteration based on certain probabilities, in order to let the algorithm explore

solutions with a given job assignment before changing it. As such, two additional parameters are defined based on 4 and 5, respectively: the probability of applying RFA mutation ( $R_{g1}$ ) and the probability of applying RSM mutation ( $R_{g2}$ ), both applied to every generation.

$$R_{g1} = (1 - R_m)/2 \quad (4)$$

$$R_{g2} = (1 - R_m)/3 \quad (5)$$

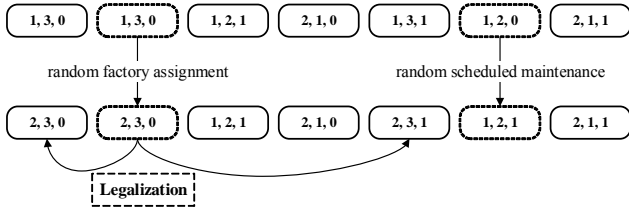


Fig. 5. Global mutation: random factory assignment (RFA) and random scheduled maintenance (RSM)

$R_{g1}$  is the mutation rate for RFA,  $R_{g2}$  is the mutation rate for RSM, and  $R_m$  is the mutation rate. As aforementioned before, global mutation has significant impact on the operation scheduling. Whenever a local mutation is not performed, the global mutation is conducted with a reduced probability. For RFA, the probability is reduced into half (divided by 2) while the RSM probability is reduced by one over three ( $\frac{1}{3}$ ). As such, the global mutation will be applied to at least “some” of the population’s individual. Note that  $R_m$  mutation rate is used because of the global mutation is inversely proportional to the individual’s affinity.

#### IV. COMPUTATIONAL RESULTS

The performance of the IIA has been tested on several instances. Four datasets were considered. The first, second, and third datasets were obtained from Chan *et al.* [4], [25], [26], whereas the fourth dataset obtained from Fisher and Thompson’s benchmark data [27]. Two separate tests were conducted. The first test used the first, second, and third datasets whereas the second test used the fourth dataset. The first test aims to compare IIA with other algorithms designed for FMSDS, in particular, Ant Colony Optimization (ACO) by Kumar *et al.* [28], Genetic Algorithm with Dominant Gene (GADG) by Chan *et al.* [4], [25], [26], Modified Genetic Algorithm with Dominant Gene (MGADG) by Chung *et al.* [29], and Improved Genetic Algorithm (IGA) by De Giovanni and Pezella [8]. The second test aims to compare IIA with other algorithms that were conducted on the same dataset; particularly, Modified Genetic Algorithm (MGA) by Jia *et al.* [7] and Improved Genetic Algorithm (IGA) by De Giovanni and Pezella [8]. IIA has been implemented in C# compiler and run independently on a personal computer equipped with a 2.0 GHz Intel Core i5 processor and 2GB RAM.

All datasets considered in this study are summarized in Table I. IIA parameters have been calibrated for the preliminary test on all datasets described above. The details of four parameter option’s setting for each datasets considered are

TABLE I. DATASETS PARAMETERS/PROPERTIES

Data labels	$F$	$H_f$	$i$	$N_i$	Reference
fjs01	1	3	5	4	[4], [25], [26]
fjs02	1	10	100	n.a.	[25]
dfjs01a	2	3	10	4	[4], [29]
dfjs01b	2	3	10	4	[4], [29]
Mt06	1	6	6	6	[27]
Mt10	1	10	10	10	[27]
mt20	1	5	20	5	[27]

\*a without maintenance integration, \*b with maintenance integration  
 \*n.a.: not available/no specific numbers of operation (flexible)

TABLE II. IIA CONTROL PARAMETERS

Parameter	fjs01,02	dfjs01(a)	dfjs01(b)	Mt06,10,20
Generation No.	500	100	5000	5000
Run No.	5	5	5	5
Hyper-mutation Rate ( $R_m$ )	0.05	0.1	0.15	0.3
Options No.	4	4	4	4
Based on Option:	1	2	3	4
Population Size ( $pop_N$ )	50	75	150	300
Clonal Selection Rate ( $C_r$ )	0.25	0.45	0.65	0.76

given in Table II. Results of the first and second test are given in Fig. 6(a) and (b), respectively.

The results of IIA dominate other algorithms designed for FMS, FMSDS, and even job shop (JS), by obtaining optimal results for every datasets on both test cases considered in this paper. Note that results of algorithms given as zero are to denote that the algorithm consideration of the datasets is unavailable. When comparing the iteration sizes (generations) of IIA and IGA [8], IIA requires more to converge which contradicts with IGA. However, the IIA results obtained are better than all test instances of IGA. Moreover, consideration of maintenance cases (included and excluded) of the FMSDS are also able to obtain superior results with maximum numbers of iteration (generations) as from Chan *et al.* [4], [25], [26].

Additionally, IIA considered various combinations of parameters. Determining the appropriate parameters give vital effects on the solutions and probability reduction of premature convergent. As such, identifying the appropriate parameter combinations, by analyzing different combinations of parameters, specifically the population size ( $pop_N$ ), clonal selection rate ( $C_r$ ), and hyper-mutation rate ( $R_m$ ), is investigated. The details of different parameter combination results are graphically shown in Fig. 6(c). The  $R_m$  value used are 0.05, 0.1, 0.15, and 0.3, while  $C_r$  used 0.25, 0.45, 0.65, and 0.75. From the overall view of Fig. 6(c), it can be concluded that higher  $C_r$  produces better results, in which simultaneously combined high number of  $pop_N$ , gives higher probability of achieving an optimal result with small deviations. The best possible parameter combinations suggested from Fig. 6(c) are  $R_m = 0.1$ ,  $pop_N = 300$ , and  $C_r = 0.75$ .

Referring to the obtained results, the IIA relies solely on the mutation operators as the evolutionary driver in order to obtain (near-) optimal solution. Compared to other algorithms (i.e. Genetic Algorithm), there is no crossover operator to maintain population diversity. The top best individual of the population are guaranteed being selected due to the ranking scheme, followed by cloning for the following mutation process. However, probable duplication is inevitable because of the high number of selections. As such, by performing the global mutation

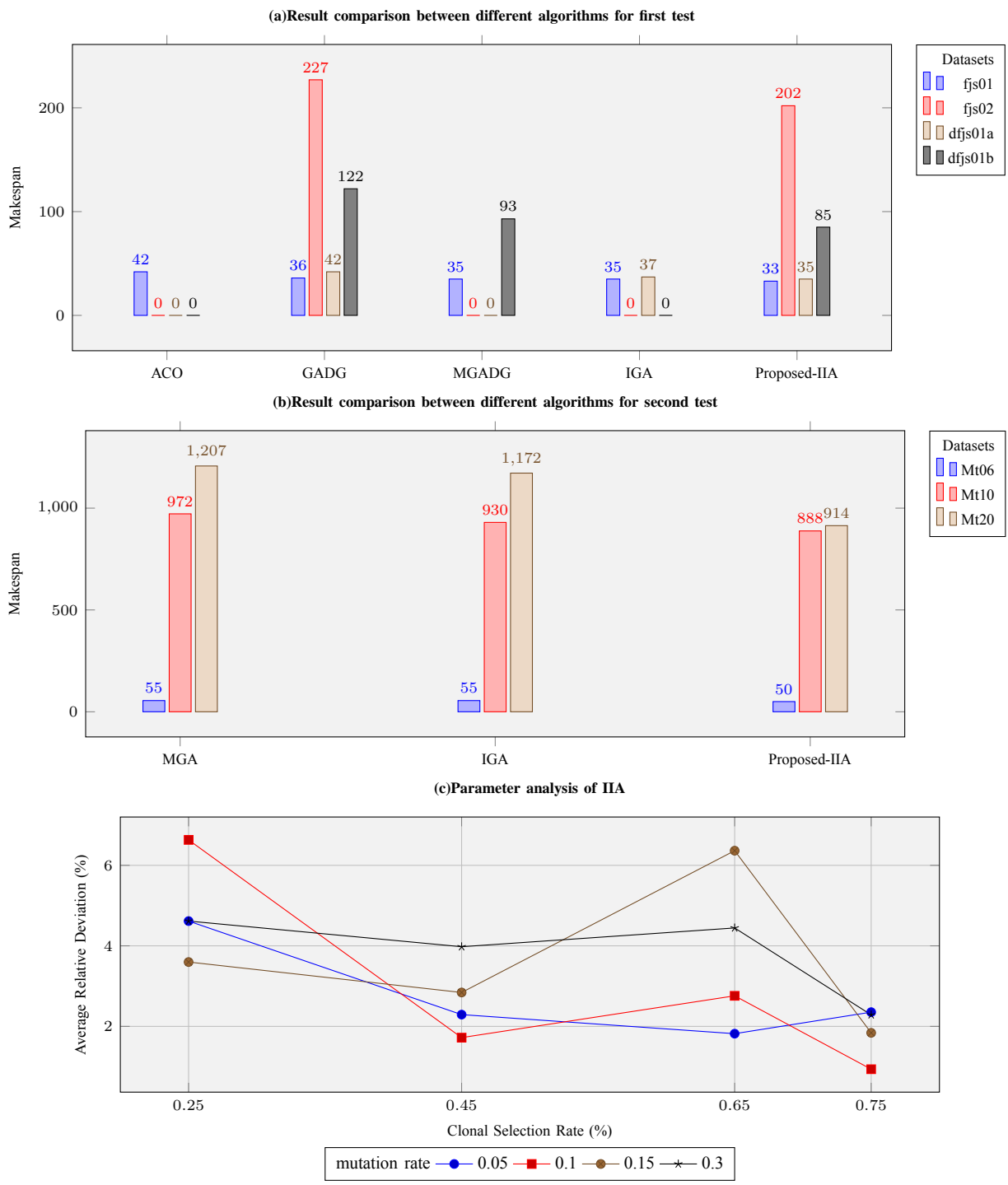


Fig. 6. Result comparison of frst test (a), second test (b) and parameter analysis (c)

operator, relative “tweaks” of individual population had been achieved to maintain overall population diversity. Dispersion effect is observed when the cloned individual undergoes global mutation, where larger search space is covered in every generation and reduces the chance of trapping in a local optimum by exploring neighboring areas of the search space. Every individual population can develop at a consistent pace without having competition between each others. In addition, decoding scheme of the individual population during an evaluation process had guaranteed that lower makespan will be chosen at all time. Indirectly, the individual population can accelerate their pace in achieving (near-) optimal solution, especially in small dataset.

## V. CONCLUSION AND FUTURE WORKS

In conclusion, this paper proposed the IIA approach in solving FMSDS problem subject to machine maintenance. The IIA parameters and operators have been presented whereas compared with other algorithms in similar venture have been conducted to justify IIA overall performance and optimization capabilities. We summarize that IIA as a suitable alternative for solving FMSDS problem subject to machine maintenance under the following reasons:

- 1) IIA requires less iteration numbers, indirectly promotes higher computational efficiency to achieve (near-) optimality.
- 2) The solution obtained is closer to other meta-heuristic algorithms and capable of providing global optimum.
- 3) IIA promotes solution diversity and faster solution evaluation due to a simplified solution modeling scheme.
- 4) Greedy decoding scheme always guarantees a superior solution selection rather than the inferior one.

Best results obtained prove to be an encouragement to further extend this work to a more complex and challenging environment. Nevertheless, the datasets are obtained from literature and benchmarks, merely serve as an abstraction of a real world manufacturing problem that is substantially more complex and difficult to apprehend. As such, achieving conceivable results which satisfy the actual manufacturing problem still far from reality and actual implementation. The applicability of IIA on FMSDS problem subject to machine maintenance has considerable potential with further refinement on a specific aspect which outlined as following:

- 1) Due to the stochastic nature of IIA, an extension with an artificial neural-network is possible in order to specify system-specific parameters or operating strategies in an FMSDS environment.
- 2) Considering rescheduling strategies incorporated in IIA to improve solution quality as well as the system state in a real-time operation, in order to enhance productivity.
- 3) Coupling IIA with an efficient machine maintenance strategy can improve solution reliability and quality.
- 4) Simulating the worst-case scenarios (i.e. machine breakdown) could be done to further test IIA capabilities in machine maintenance environment.
- 5) Developing a systematic methodology in order to add values on the main IIA operators specific to the scheduling problem, such as the somatic mutation.

- 6) Inclusion of other hardware elements of the manufacturing system to make the scheduling task as an integrated one.

## ACKNOWLEDGMENT

The main author wishes to thank Universiti Sains Malaysia for the support it has extended in the completion of this research through grant number 304/PKOMP/6313026..

## REFERENCES

- [1] A. Bagheri, M. Zandieh, I. Mahdavi, and M. Yazdani, “An artificial immune algorithm for the flexible job-shop scheduling problem,” *Future Generation Computer Systems*, vol. 26, no. 4, pp. 533–541, 2010.
- [2] F. T. S. Chan, H. Chan, and H. Lau, “The state of the art in simulation study on fms scheduling: a comprehensive survey,” *The International Journal of Advanced Manufacturing Technology*, vol. 19, no. 11, pp. 830–849, 2002.
- [3] F. Chan, S. Chung, and P. Chan, “An adaptive genetic algorithm with dominated genes for distributed scheduling problems,” *Expert Systems with Applications*, vol. 29, no. 2, pp. 364–371, 2005.
- [4] F. Chan, S. Chung, L. Chan, G. Finke, and M. Tiwari, “Solving distributed fms scheduling problems subject to maintenance: genetic algorithms approach,” *Robotics and Computer-Integrated Manufacturing*, vol. 22, no. 5, pp. 493–504, 2006.
- [5] W. Shen, “Distributed manufacturing scheduling using intelligent agents,” *Intelligent Systems, IEEE*, vol. 17, no. 1, pp. 88–94, 2002.
- [6] M. Aboutalebi, H. Shirgahi, and H. Motameni, “Distributed flexible manufacturing system (fms) scheduling using memetic algorithm, particle swarm optimization and timed petri net,” *International Journal of the Physical Sciences*, vol. 6, no. 14, p. 35573563, 2011.
- [7] H. Jia, A. Nee, J. Fuh, and Y. Zhang, “A modified genetic algorithm for distributed scheduling problems,” *Journal of Intelligent Manufacturing*, vol. 14, no. 3, pp. 351–362, 2003.
- [8] L. De Giovanni and F. Pezzella, “An improved genetic algorithm for the distributed and flexible job-shop scheduling problem,” *European Journal of Operational Research*, vol. 200, no. 2, pp. 395–408, 2010.
- [9] M. Gholami and M. Zandieh, “Integrating simulation and genetic algorithm to schedule a dynamic flexible job shop,” *Journal of Intelligent Manufacturing*, vol. 20, no. 4, pp. 481–498, 2009.
- [10] A. Prakash, N. Khilwani, M. Tiwari, and Y. Cohen, “Modified immune algorithm for job selection and operation allocation problem in flexible manufacturing systems,” *Advances in engineering software*, vol. 39, no. 3, pp. 219–232, 2008.
- [11] K. Woldemariam and G. Yen, “Vaccine-enhanced artificial immune system for multimodal function optimization,” *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 40, no. 1, pp. 218–228, 2010.
- [12] G. Wojtyła, K. Rządca, and F. Seredynski, “Artificial immune systems applied to multiprocessor scheduling,” *Parallel Processing and Applied Mathematics*, pp. 904–911, 2006.
- [13] L. De Castro and F. Von Zuben, “Learning and optimization using the clonal selection principle,” *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 3, pp. 239–251, 2002.
- [14] V. Cutello, G. Nicosia, M. Romeo, and P. Oliveto, “On the convergence of immune algorithms,” in *Foundations of Computational Intelligence, 2007. FOCI 2007. IEEE Symposium on*. IEEE, 2007, pp. 409–415.
- [15] J. Al-Enezi, M. Abbod, and S. Alsharhan, “Artificial immune systems-models, algorithms and applications,” *International Journal*, 2010.
- [16] D. Dasgupta and S. Forrest, “Artificial immune systems in industrial applications,” in *Intelligent Processing and Manufacturing of Materials, 1999. IPMM’99. Proceedings of the Second International Conference on*, vol. 1. IEEE, 1999, pp. 257–267.
- [17] H. Yu, “Optimizing task schedules using an artificial immune system approach,” in *Proceedings of the 10th annual conference on Genetic and evolutionary computation*. ACM, 2008, pp. 151–158.

- [18] S. Darmoul, H. Pierreval, and S. Gabouj, "Scheduling using artificial immune system metaphors: a review," in *Service Systems and Service Management, 2006 International Conference on*, vol. 2. IEEE, 2006, pp. 1150–1155.
- [19] O. Kilic and Q. Nguyen, "Application of artificial immune system algorithm to electromagnetics problems," *Progress In Electromagnetics Research*, vol. 20, pp. 1–17, 2010.
- [20] C. Chu, M. Lin, G. Liu, and Y. Sung, "Application of immune algorithms on solving minimum-cost problem of water distribution network," *Mathematical and Computer Modelling*, vol. 48, no. 11, pp. 1888–1900, 2008.
- [21] L. De Castro and F. Von Zuben, "The clonal selection algorithm with engineering applications," in *Proceedings of GECCO*, vol. 2000, 2000, pp. 36–39.
- [22] C. Coello and N. Cortés, "Solving multiobjective optimization problems using an artificial immune system," *Genetic Programming and Evolvable Machines*, vol. 6, no. 2, pp. 163–190, 2005.
- [23] N. Khilwani, A. Prakash, R. Shankar, and M. Tiwari, "Fast clonal algorithm," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 1, pp. 106–128, 2008.
- [24] U. Aickelin and D. Dasgupta, "Artificial immune systems," in *Search Methodologies*. Springer, 2005, pp. 375–399.
- [25] F. Chan, S. Chung, and P. Chan, "Application of genetic algorithms with dominant genes in a distributed scheduling problem in flexible manufacturing systems," *International Journal of Production Research*, vol. 44, no. 3, pp. 523–543, 2006.
- [26] F. Chan, S. Chung, and L. Chan, "An introduction of dominant genes in genetic algorithm for fms," *International Journal of Production Research*, vol. 46, no. 16, pp. 4369–4389, 2008.
- [27] H. Fisher and G. L. Thompson, "Probabilistic learning combinations of local job-shop scheduling rules," *Industrial scheduling*, pp. 225–251, 1963.
- [28] R. Kumar, M. Tiwari, and R. Shankar, "Scheduling of flexible manufacturing systems: an ant colony optimization approach," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 217, no. 10, pp. 1443–1453, 2003.
- [29] S. H. Chung, F. T.S.Chan, and H.K.Chan, "A modified genetic algorithm approach for scheduling of perfect maintenance in distributed production scheduling," *Engineering Applications of Artificial Intelligence*, vol. 22, pp. 1005 – 1014, 2009.

## **Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)**

This article is published under the terms of the Creative Commons Attribution License 4.0

[https://creativecommons.org/licenses/by/4.0/deed.en\\_US](https://creativecommons.org/licenses/by/4.0/deed.en_US)