

# Optimizing The Distributed Production Scheduling Problems Using An Immune-Based Algorithm

Mohd Nor Akmal Khalid<sup>1</sup> and Umi Kalsom Yusof<sup>1</sup>

<sup>1</sup>School of Computer Sciences,  
Universiti Sains Malaysia, 11800 USM, Pulau Pinang  
mnak13\_com027@student.usm.my, umiyusof@cs.usm.my

**Abstract:** The challenges faced in the manufacturing industry involve problems that developed in each passing time, particularly in the semiconductor assembly areas. Satisfying customer demands and achieving higher profits, as well as maintaining high productivity have been the most important aspects which attract the expatriate's attentions in the semiconductor industry. The combined of productivity and flexibility provided by semiconductor assembly areas have elicited variety research efforts for several years. In addition, globalization trends in the manufacturing have encouraged a decentralized effort in the semiconductor assembly industry by implementing distributed production scheduling systems in the production line. With respect to the above mentioned problems, several approaches have been introduced which can be categorized based on the static and dynamic settings of the scheduling. An immune algorithm (IA), which is slightly modified to conform to the manufacturing constraints as well as solving the underlying problem, has been proposed and tested with public data set and industrial data set to establish its effectiveness. The proposed IA algorithm is selected due to its explorative powers of the hyper-mutation operator, solution diversity through its receptor editing operator, and incubation of the memory cell. The results from the experiments was achieved from the proposed IA is effective for the aforementioned problems where the best solution obtained from the public data is between 11% to 19% deviation, while production efficiency for the case studies had obtained within the range of 10% to 66%.

**Keywords:** Semiconductor industry, distributed production scheduling, immune algorithm.

## I. Introduction

The semiconductor industry is well-known for having one of the most complex manufacturing processes because of its adopted technologies and the intricacy of the manufacturing processes. Technologies that are rapidly growing and very competitive are the common criteria of the industry [1]. The semiconductor industry's primary challenge involves the struggle to meet on-going customer demands for reduced prices while simultaneously maintaining operating costs. The main interest for many players in this industry is to be able to equip better facilities while balancing the capital investment and maintaining a constant, happy customer base.

Instead of relying on good forecasting for product demands to sustain their competitive edge, the industry has resorted to initiate various enhancement to the planning, allocation, and control of the production line.

The production scheduling problems in the manufacturing industry have been the subject of research initiatives for several years. Rapid developments in computer technology have promoted new ways to solve problems in production scheduling domain. The newly developed methods have rendered exact approaches insufficient to handle complex and changing environments of production scheduling. In general, a production scheduling can be defined as an allocation of tasks to a set of resources within a time horizon. Resources in the semiconductor industry refers to the machines that are capable of producing different part types without the need for major retooling and able to handle multiple part types [2]. These resources composes of computer controlled, integrated configurations of centralized numerically controlled (CNC) machines with automated material handling systems [3]. These resources are also found to be suitable for mid-volume and mid variety productions because of the combined machines efficiency and flexibility of the factory production flow.

The predetermined production scheduling parameters involved in the semiconductor assembly areas includes the product mix, production levels, resource availability and due dates of jobs [4]. Various levels of commitment in the companies' management are required in order to develop a detailed plan for a particular product. Typically, the production schedule will be modified to fulfil the requirements based on the availability of resources and clients' orders at any single time. In addition, a particular company can benefits added advantages by having low labor cost, correct skill set, and demographic proximity to the customer. These benefits indirectly speeds up the production, reduces inventory, able to handle broad variety of products, introduce profitable investment opportunity, and enhances flexibility. However, the abundant options of feasible solutions with different task-resource assignments, the production scheduling problems are considered one of the *NP*-hard problems [5]. Given these issues, both knowledge of the practitioners and the academicians to solve the production scheduling problems are essentially needed.

Recent advancement in manufacturing industry and the effect of globalization trends have initiated the notion of the distributed scheduling (DS) system where a single factory production is gradually substituted by multi-factory productions where factories are physically located in distinct geographical locations and different factories to fulfill unique requirement of the product parts [6, 7]. However, implementing DS is much more complicated for a semiconductor industry compared to a single factory due to the following issues [6]: (1) allocating lots to a suitable factory, and (2) determination of the production scheduling in each factory. The main objective of this study is to optimize the allocation of assembly lots to a corresponding machines in a network of factories.

The immune algorithm (IA) is a meta-heuristic algorithm that inspired by several mechanisms that form the building block of a very complex natural immune systems defense against invading organisms. The explorative powers of the hyper-mutation operator, solution diversity through its receptor editing operator, and evolutionary capacity of memory cell, motivates this study to employ IA as a suitable approach for optimizing the production scheduling in DS settings. The reliability and ability of IA to achieve good results had been proven in [3, 8] in similar problems. Therefore, this study extends the prior work of IA in optimizing the distributed production scheduling problems by performing the proposed theoretical framework to the public data sets as well as applying it to the industrial data sets with respect to minimizing the maximum completion time of the production line.

## II. Background Study

In the semiconductor industry, production scheduling problems had been one of the most popular research focus for many years. Variety of approaches have been introduced in order to solve the production scheduling problems which can be further generalized into static (or predictive) scheduling and dynamic (or reactive) scheduling approaches. There have been a series of previously done researches regarding scheduling problems in manufacturing, which can be discretely classified based on the algorithm approaches as depicted in Table 1.

One of the earliest solutions proposed for production scheduling problems is the heuristic search algorithms. Several authors have proposed heuristic search or heuristic functions in solving the scheduling problems in production scheduling problems with respect to certain performance criteria [9, 2, 4]. Most of the solutions proposed failed to integrate the dynamic or reactive settings in their solutions which considered the materials availability [2], resource availability and its associated constraints [9, 11], and dynamic nature of the scheduling procedure [13]. However, consideration of dynamic and real-time scheduling environments had been increased in these recent years. Additional consideration scopes that composed of the stochastic and unexpected events that might occur in the real scheduling should also be addressed. Increased attentions on the production scheduling problems had also been addressed through the means of an artificial intelligence (AI) based algorithms. Cases of the classical AI approach had been conducted in effort of solving production scheduling problems [14]. However, the com-

plexity and various conflicting performance criteria had been a major challenge [10].

Most approaches in solving the production scheduling problems had been conducted through meta-heuristic algorithms which the solution is derived based on either nature-inspired, swarm intelligences, or phenomenon mimicking. Nature-inspired algorithms can be defined as algorithms that derived from the natural behaviors, scaling from behaviors or processes of the molecular reactions to the complex cortical maps of the biological organization [53]. Artificiality that represented in the biological processes had inspired researchers into various computing optimization algorithms, such as genetic algorithm (GA) [1, 45], simulated annealing (SA) [24], shuffled frog leaping algorithm (SFL) [22], and symbiotic evolutionary algorithm (SE) [23].

On the other hand, swarm intelligence or “collective” intelligence refers to the decentralized and self-organized problem solving behavior, derived from the interactions of individual agents between other agents, in reacting to the local environments. An example of such algorithms includes ant colony optimization (ACO) [25, 26, 27, 46, 47], particle swarm optimization (PSO) [48, 36, 37, 49], artificial immune system (AIS) [5, 31], artificial bee colony (ABC) [32, 33, 34], and the recently adopted, biogeography-based optimization (BBO) [35] and cuckoo search (CS) [30]. Another rare derivation of meta-heuristic algorithms is the algorithm that mimics a certain natural phenomenon. This phenomenon mimicking algorithm refers to the optimization processes conducted through the emulations of naturally occurred phenomenon, like the harmony search algorithm (HS) [50, 51] that mimics the improvisation process of a musical performance, and tabu search (TS) [38, 52, 7] that imitate the phenomena of accursed or “taboo” belief in the behavior of the search process.

Although certain limitation had been identified in conducting the meta-heuristics algorithms in solving the production scheduling problems [54], the effort had been continuously adopted in the last 25 years. Some of the initial trends of research conducted is more focused on static production scheduling environments with either single (i.e. [10]) or multiple [19, 27] performance criteria. Additionally, the complexity of implementing the meta-heuristics with respect to the problem domain tend to be arduous. However, attention towards dynamic scheduling environments had been increased due to the importance of reducing the scheduling time [42], difficulty in the schedule implementations and short validity [43], and enhancing the productivity by incorporating alternative scheduling plans or routing [25].

Works that have adopted IA have successfully solves problems in variety domains. IA features that are not limited to self-organizing, adaptivity, and uniqueness have the potential to be used in developing the computational models applied to business [3, 55, 56, 57], sciences and engineering [58, 59, 60, 61], and optimization domain [62, 3, 61, 63, 8]. In order to enhance the approach, the method introduced in the literatures include extension of generic IA with adoption of Gaussian and Cauchy mutation operators [3], incorporate Genetic Algorithm (GA) as the candidate solution pre-processors [61], and solution pool namely as Antigenic Clustering method [8]. AIS has outperformed other soft-

Table 1: Researches on production scheduling problems in manufacturing industry

Problem Environment	Category of AI Approach	Approaches	References	
Static/Predictive Scheduling	Heuristic	Local Search	[9]	
		Petri Nets	[10];[11];[12]	
		A* and Node Pruning	[13]	
	Classic AI Algorithm	Fuzzy Logic	[14]	
	Nature-Inspired	Genetic Algorithm (GA)		[15];[16];[17]; [18];[19];[20];[21]
			Shuffled Frog Leaping (SFL)	[22];
		Symbiotic Evolutionary Algorithm (SE)	[23]	
		Simulated Annealing (SA)	[24]	
		Swarm Intelligence	Ant Colony Optimization (ACO)	[25];[26];[27]; [28];[29]
			Cuckoo Search (CS)	[30]
			Artificial Immune System (AIS)	[5];[31]
	Artificial Bee Colony (ABC)		[32];[33];[34]	
	Phenomenon Mimicking	Biogeography-Based Optimization (BBO)	[35]	
		Particle Swarm Optimization (PSO)	[36];[37]	
Dynamic/Reactive Scheduling	Heuristic	Filtered-Beam-Search	[39];[40]	
		Fuzzy Rules	[41]	
	Nature-Inspired	Genetic Algorithm (GA)	[42];[43];[44];[45];	
	Swarm Intelligence	Ant Colony Optimization (ACO)	[46];[47]	
		Particle Swarm Optimization (PSO)	[48];[49]	
	Phenomenon Mimicking	Harmony Search (HS)	[50];[51]	
		Tabu Search (TS)	[52];[7]	

computing algorithms within their respective problems of the literatures as well as proved to be robust and capable of handling different types of optimization domains. In addition, the work in [8] had also given the insights on the applicability of the IA in handling the production scheduling domain, especially the dynamic natures of the semiconductor industry. As such, these reasons warrant the need to adopt this IA approach as the solution to address the production scheduling problems faced in the semiconductor industry.

### III. The Distributed Production Scheduling Problem

The production scheduling problem involves a number of lots or jobs ( $i$ ) which are expected to be received in the distributed network, and a suitable factory ( $f = 1, \dots, F$ ) will be assigned to the lot or job to generate corresponding production schedule. 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 lot or job has up to  $N_i$  operations, and every operation can be performed by more than one machine (not all), but must be in the same factory. The traveling time between factory  $f$  and lot or job  $i$  is denoted as  $D_{if}$ .

#### A. Decision Variables and Constraints

The decision variables are as follows:  $\chi_{ij}$  denoted true if lot or job  $i$  is allocated to factory  $f$ ; and  $\delta_{ijfkh}$  if operation  $j$  of lot or job  $i$  occupies time slot  $k$  on machine  $h$  in factory  $f$ . The production scheduling problem is subjected to the following

constraints:

- Completion of the preceding operation is required for the current operation to begin.

$$S_{ij} \geq E_{i(j-1)} \quad (i = 1, 2, \dots, I; j = 2, 3, \dots, N_i). \quad (1)$$

- Completion without interruption will be carried out once an operation starts.

$$E_{ij} - S_{ij} = \sum_{fh} \chi_{if} T_{ijfh} \quad (i = 1, 2, \dots, I; j = 1, 2, \dots, N_i). \quad (2)$$

- Each operation can be processed by one machine for each unit of time and vice-versa.

$$\sum_{fkh} \delta_{ijfkh} = 1 \quad (i = 1, 2, \dots, I; j = 1, 2, \dots, N_i). \quad (3)$$

- Each machine can only process a single operation for each unit of time.

$$\sum_{ij} \delta_{ijfkh} \leq 1 \quad (k = 1, 2, \dots, K; h = 1, 2, \dots, H_f; f = 1, 2, \dots, F). \quad (4)$$

- Each lot or job can only be assigned to a single factory.

$$\sum_f \chi_{if} = 1 \quad (i = 1, 2, \dots, I). \quad (5)$$

### B. Performance Measures

Once the value of  $\chi_{ij}$  and  $\delta_{ijfhhk}$  are obtained, the starting time value of operation  $j$  of lot or job  $i$  ( $S_{ij}$ ), ending time of operation  $j$  of lot or job  $i$  ( $E_{ij}$ ), and the completion time ( $C_i$ ) can be calculated. The performance measure considered is to minimize the total maximum completion time of the last lot or job operation which is defined in (6). Completion time ( $C_i$ ) as defined in (7) is the summation of the completion time of the last operation  $N_i$  of lot or job  $i$  and the delivery time between the factory  $f$  and the lot or job  $i$ .

$$\text{Objective } Z : \min(\max\{C_i\}). \quad (6)$$

$$C_i = E_{iN_i} + \sum D_{if}\chi_{if}. \quad (7)$$

However, in term of measuring and quantifying the production scheduling, specific for the semiconductor industry, parameters such as the total quantity of the package demands ( $Q_{demand}$ ), unit processing per hour of the individual machine produces ( $UPH_h$ ), and the size of lot or job ( $i_{size}$ ) must be considered. Thus, for the considered case study, the revised performance measures are:

$$C_t = \frac{Q_{demand}}{UPH_h} \cdot i_{size} \quad (8)$$

$$Ave_h = \frac{C_i}{H_f} \quad (9)$$

$$P_e = \frac{\sum N_i * \delta_{ijfhhk}}{N} * 100 \quad (\delta_{ijfhhk} = 1) \quad (10)$$

where  $C_t$  is the total completion time,  $Ave_h$  is the average output of an individual machine, and  $P_e$  is the production efficiency of the assembly areas, respectively.

## IV. Artificial Immune Algorithm for The Distributed Production Scheduling Problem

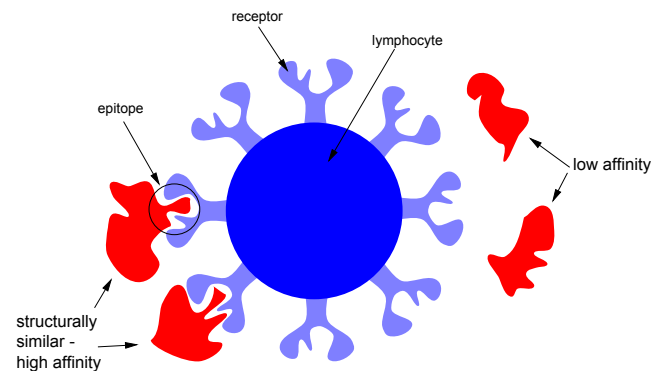
### A. Generic Artificial Immune System

In terms of biology, AIS is a complex pattern recognition system of a vertebrate which that defends the organism against diseases by detecting, identifying, and killing foreign entity such as pathogens [63]. The system can recognize or identify cells (or molecules) within the organism as either harmful (non-self-cell) or harmless (self-cell) [5] to allow the system to naturally evolve to recognize and neutralize threats. In a typical infection process, infestation and proliferation of a pathogen within the organism occurs. Pathogens and antigens correspond to specific foreign proteins. To understand the AIS processes, the following immunology terms need to be emphasized [3]:

- Immune cells: Consists of two major cells known as B-cells and T-cells, that identify antigenic patterns present in the human system.
- Antigens: Disease-causing elements in the immune system, categorized as non-self antigens (harmful) and self-antigens (harmless).

- Antibodies: molecules produced by B-cells which respond when stimulated by non-self antigens. Successive binding of B-cells and antigens induce the formation of antibodies to destroy the antigens.

When a harmful non-self-cell enters the body, the immune system responds through an innate immunity, providing immediate but nonspecific defense to protect the organism from any possibilities of an infection [63]. An antigen-presenting cell (known as phagocyte), will detect the presence of non-self cells and fight them by secreting T-cell-activating molecules. When innate immunity is penetrated by antigens or pathogens, the system initiates what is known as an adaptive immunity [5]. The activated T-cells select appropriate B-cells, which have receptors that closely resemble the antigenic signatures of the foreign proteins (clonal selection hypothesis). Next, these B-cells attach to the detected foreign protein's signature (binding site/epitope) which process known as the affinity. Affinity is the measure for evaluating the successful binding of foreign proteins and B-cells [64]. This scenario is illustrated in Fig. 1.



**Figure 1:** Illustration of B-cells attach themselves to the detected foreign protein's signature (binding site/epitope) adopted from [65].

The B-cells then undergo cellular reproduction via somatic hyper-mutation or receptor editing to attain better affinity against antigens by rapidly mutating or randomly changing their receptors' genetic orientation, respectively (affinity maturation). Afterwards, B-cells undergo the proliferate process to produce clones where a large number of identical B-cells are duplicated. Some of the mature B-cells will produce new plasma cells while others with high affinity threshold will be sustained as long-lasting memory cells [5]. The roles of the plasma cells are producing a large number of antibodies to be distributed throughout the blood and lymph systems randomly for recognition, killing, and detecting foreign proteins and malfunctioning self-cells. The roles of the long-lasting memory cells is to remain in the system to effectively accelerate the response of the immune system in future encounter. The other remaining clones of B-cells will die or be replaced by another new clone.

### B. The Proposed Immune Algorithm (IA) for The Distributed Production Scheduling Problem

The implementation of the proposed immune algorithm (I-A) is tailored towards the distributed production scheduling

problem. For clarity, the analogy of IA for this particular study is given in Table 2.

Table 2: Analogy of AIS

Immunology Terms	Scheduling Terms
Antigens	Machine Assignment
B-cells	Lots/Jobs
Antibodies	Schedule
Affinity	Evaluation Function
Hyper-mutation	Improvement Strategy
Generation	Iterations

The process of IA starts with the initialization of population size. The assignment of job to a machine is conducted to create the initial schedule with respect to the allowable machine for the respective job. The affinity is used as a measuring instrument for evaluating the successful binding of an antigen and an antibody. The parameter of the IA which is the termination criterion with the maximum number of generations, is also initialized. The clonal selection is a selection mechanism of the antibody-antigen bindings from the initialized population which will then, undergoes cloning, hyper-mutation, and receptor editing. Somatic hyper-mutation is a mutation operator that performs random flips of strings, decimals, or binary numbers, but mutate at a higher rate for an inferior antibody and vice versa. Slight changes obtained from the cloned solution after the hyper-mutation process is known as receptor editing. Both hyper-mutating and receptor editing act as exploratory and exploitation mechanisms of the search space in the optimization domain. The sample antibody is as shown in Fig. 2.

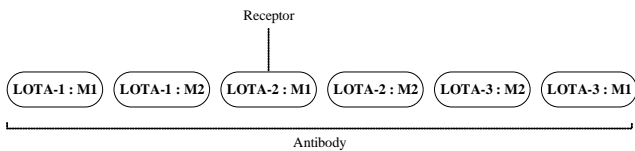


Figure 2: Sample antibody (candidate solution)

During the clonal selection, a set of candidate solutions from the current population are chosen to apply the next IA operators in order to produce high affinity memory cell(s) to include for the next generation. The clonal selection is dependent on the affinity (completion time) of the antibody. Therefore, the affinity or the maximum completion time of the candidate solution is computed and ranked by decreasing affinity (starting from the best to the worst affinity). The top best % ( $C_r$ ) of ranked population are cloned, in which each of these cloned cells undergo hyper-mutation process. The hyper-mutation is performed based on the objective value (completion time) of the current schedule. In this case, minimizing the maximum completion time is better which is inversely proportionate to the affinity. As such, the frequency or in immune algorithm context, the rate of the mutation is calculated as below:

$$Rate_{mutate} = \frac{1}{C_t * N_i} \quad (11)$$

The mutation process conducted selecting two random positions of the receptors (jobs or lots) within the candidate so-

lution (schedule) and exchanging their machine assignment. This basically will only affect the machine parameters of the candidate solution. This process can be demonstrated as shown in Fig. 3.

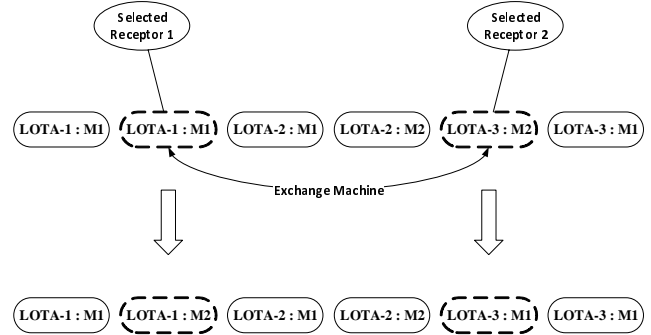


Figure 3: Hyper-mutation process

The process is repeated until the mutation rate is satisfied. After that, the process continues for the next generation of the populations until the termination criterion is equal to the maximum generation number. If the termination criterion is met, the solution is then compared with the last objective value (completion time) obtained.

## V. Experimental Results and Discussion

In this section, the application of the proposed IA approach is to demonstrate and evaluate its computational performance. The experimental procedures are twofold; application to the public data sets and industrial data sets. The first experiment conducted to get the best parameter calibrations of the proposed IA algorithm as well as its capability in the distributed production scheduling problems. In the second experiment, the proposed IA is applied, with respect to best parameter obtained from the first experiment, to a case study of the real semiconductor industry area to evaluate the its effectiveness in handling complex real-world settings. IA was implemented in C# compiler and run independently on a personal computer equipped with a 4.0 GHz Intel Core i7 processor and 4 GB RAM.

### A. Public Data Sets Testing

The performance of the IA is tested with several instances which are separated into two experiments. The first experiments (A1) was based on data set that was obtained from Chan *et al.* [66, 67, 68], while the second experiment (A2) was based on data sets that were obtained from Fisher and Thompson's benchmark data [69]. The first experiment involves comparison of IA with other algorithms designed for distributed production scheduling problems, in particular, ant colony optimization (ACO) [25], genetic algorithm with dominant gene (GADG) [66, 67, 68], modified genetic algorithm with dominant gene [70], and improved genetic algorithm (IGA) [71]. The second experiment compares IA with other algorithms that were used on the benchmark data set; these algorithms are modified genetic algorithm (MGA) [72] and IGA [71].

The data sets considered in this study are summarized in Table 3. Both experiments were run independently and the best

results among five test runs were recorded. IA parameters were calibrated for the preliminary test on all data sets described above. The details of four parameters for each data sets considered are given in Table 4.

Table 3: Experimental settings

Experiment A1					
Data labels	$F$	$H_f$	$i$	$N_i$	Reference
fjs	1	3	5	4	[66, 67, 68]
dfjs	2	3	10	4	[66, 70]
Experiment A2					
Data labels	$F$	$H_f$	$i$	$N_i$	Reference
Mt06	1	6	6	6	[69]
Mt10	1	10	10	10	[69]
Mt20	1	5	20	5	[69]

Table 4: IIA control parameters

Parameter	fjs	dfjs	Mt06,10,20	
Generation No.	500	100	5000	
Run No.	5	5	5	
Options No.	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	0.75

Results of the first and second experiments are given in Table 5. The first column reports the data set name of the testing instance, and the following column represents the compared algorithms consecutively with the relative deviation of the maximum completion time with respect to the proposed IA. The relative deviation is defined as in (12).

$$dev = [(C_{comp} - C_{IA})/C_{comp}] * 100\% \quad (12)$$

$C_{IA}$  is the maximum completion time obtained by IA, while  $C_{comp}$  is the maximum completion time obtained by other algorithm. From the summary of the result, IA outperforms other algorithms by obtaining optimal results for every data sets on both experiments considered in this study. The relative deviation obtained by IA compared with that of other algorithms for Experiment A1 are between  $12\% \leq dev \leq 19\%$ , whereas average relative deviation for Experiment A2 are between  $11\% \leq dev \leq 14\%$ . In total, results obtained by IA relatively deviate between  $11\% \leq dev \leq 19\%$ . Although only five runs conducted, IA shares relatively coherent convergence rate with GA [73], which generally requires a high number of generation numbers to converge but able to achieve optimum solution. Thus, few test runs can support the capabilities of our proposed IA against those of other algorithms.

In addition, to demonstrate IA convergence rate, Fig. 4 shows the decrease of the average and best maximum completion time over five runs for the Mt06 data set. The figure indicates that IA improved the average maximum completion time very rapidly where the best maximum completion time (52) was achieved around 25 generations. This is possible due to the hyper-mutation process, where lower affinity anti-

body is rapidly mutated compared to the higher affinity, increasing the probability of retaining better solution in each generation.

IA considers two different parameter combinations, which are the population size ( $pop_N$ ) and the clonal selection rate ( $C_r$ ). Determining the appropriate parameter effects the quality of the solutions and reduces the probability of avoiding premature convergence. Therefore, identifying the parameter combinations by analyzing the result obtained from combinations of  $C_r$  and  $pop_N$  values were investigated. The details of different parameter combination results are graphically shown in Fig. 5. The  $pop_N$  value used are 50, 75, 150, 300, whereas the  $C_r$  value used are 0.25, 0.45, 0.65, and 0.75. From these values of  $pop_N$  and  $C_r$ , the relative average deviation of maximum completion time was computed. Based on observation, fluctuation pattern occurs when  $pop_N$  value is other than 75. However, a steady pattern is observed when  $pop_N$  value is 75, where higher  $C_r$  value gives higher deviation values. Higher deviation values impose that the solution obtain inconsistent result and the best value may or may not be achieved. Thus, the suggested  $C_r$  values is 0.25 which is combined with  $pop_N$  values of 75, respectively.

#### B. A Case Study in The Semiconductor Assembly Area

For the case study, the data is obtained from the semiconductor industry involving the assembly area. However the scopes of the data comes with the assumption of infinite materials, none of new job or lot arrival, unexchangeable tool types, absent of the machine setup time and lead time, and non-existing machine downtime. The performance measure considered are the (8),(9), and (10). The properties of each machine is predetermined which includes its operational capabilities (the operation which they can perform) and number of lot or job units they can process each hour (UPH). The data set also includes six package types with their respective number of operations, package sizing for each lots, lot quantity, total demand quantity, and machines that operational capabilities. Package B and C are considered as small demand packages, package A and E as medium demand packages, and package D and F as large demand packages. Table 6 shows the data sets used for this case study.

Table 6: Testing data for the proposed system

Machine Properties				
	$h$	$f$	$N_i$	$UPH_h$
	1	1	2,3	100
	2	1	2	700
	3	1	1,3	300
Lots Operation Requirements				
Package Types	$N_i$	$i_{size}$	$Q_{demand}/i_{size}$	$Q_{demand}$
A	1,2	8700	34	290,872
B	1,3	6500	5	30,007
C	1,2	2200	30	65,899
D	1,2,3	9500	274	2,594,872
E	1,3	8600	34	132,334
F	1,3	7650	34	1,312,821

Table 7 shows the result summary of the production schedule before optimization and the best production schedule after optimization. Before optimization, each packages has

*Table 5: Result summary of the first and second experiments.*

Experiment A1									
Data Name	ACO	<i>dev</i> (%)	GADG 1,2,3	<i>dev</i> (%)	MGADG	<i>dev</i> (%)	IGA	<i>dev</i> (%)	IA
fjs	42	+33.33	36	+22.22	35	+20.00	35	+20.00	<b>28</b>
dfjs	n.a.	n.a.	42	+16.67	n.a.	n.a.	37	+5.41	<b>35</b>
Average improvement				+19.45			+12.71		
Experiment A2									
Data Name	MGA	<i>dev</i> (%)		IGA	<i>dev</i> (%)		IA		
Mt06	55	+9.09		55	+9.09		<b>50</b>		
Mt10	972	+8.64		930	+4.52		<b>888</b>		
Mt20	1207	+24.28		1172	+22.01		<b>914</b>		
Average improvement			+14.00			+11.87			

their respective operation  $N_i$ , lot size  $i_{size}$ , and lot quantity  $i$  ( $Q_{demand}/i_{size}$ ), where they were assigned randomly to the available and allowable machines  $h$ . For instance, package A has the  $Q_{demand}$  of 290872 and the total lots or jobs  $i$  of 34 where every lot or job size  $i_{size}$  contains of 8700 units. In addition, the  $C_t$  of the production schedule before optimization is 1218 unit of time,  $Ave_h$  is 531.33, and the current  $P_e$  is 30%. After optimization, however, the results of the  $C_t$  is reduces into 609 unit of time. This shows that the  $C_t$  after optimization had decreased about half of the one before optimization. Furthermore, changes to the  $Ave_h$  and  $P_e$  of the optimized production schedule is observed where it improves into 356.33 and 60%, respectively. Similar effect also applies to other package types as well. Nevertheless, all package types had improved regardless of small, medium, or large demand quantity.

Based on the result summary, the percentage of differences is computed and graphically depicted in Figure 6. As observed, the improvement obtained for the maximum completion time are within the range of  $23\% \leq C_t \leq 66\%$ , while the average machine output are within the range of  $10\% \leq Ave_h \leq 40\%$ . As mentioned earlier, package B and C are the small demand packages which showed that the proposed algorithm able to achieve significant improvement either for  $C_t$  and  $Ave_h$  performance measures ( $33\% \leq improve \leq 66\%$ ). In addition, package A and E, which are the medium demand packages, also able to improved about half from the original performance of  $C_t$  ( $50\% \leq improve \leq 60\%$ ) and about  $\frac{1}{3}$  of the  $Ave_h$  ( $32\% \leq improve \leq 39\%$ ) through the means of the proposed IA approach. Lastly, package D and F, which are the large demand packages, is observed to have minor improvement ( $10\% \leq improve \leq 25\%$ ). From the improvement of the  $C_t$ , the insights that can be elicited within the optimized production schedule is the ability to reduce the overall production schedule earlier than the original schedule, which directly related to the improvement of  $Ave_h$  where the lot or job allocation to machine is distributed to other allowable machines to process and increases the overall machine utilization.

## VI. Concluding Remarks

The production scheduling problem in semiconductor industry is motivated in reducing cost and increasing the overall productivity as well as processing a large variety of jobs while conforming to various constraints. This is possible because limited number of machines can operate at an optimum level and the possibility of under-utilized or over-utilized can be reduced. The main purpose of this study is to develop an efficient algorithm that has the capability to solve the real world problems in manufacturing. Production schedule generates input for the optimization algorithm, which in turn, generates lots or jobs to be distributed to the available machine and factory.

This paper proposes an IA approach to solve and optimize the production schedule in manufacturing industry. The proposed IA enhances the applicability of traditional clonal algorithms by making some modifications in the operators. Based on the results, IA is able to produce a relatively satisfactory solution than other meta-heuristic algorithms applied in a similar field. By minimizing the maximum completion

time and enhancing machine utilization, it also increases the production efficiency in the production line. Additionally, the proposed hyper-mutation process enhances the probability of achieving better solution, thereby increasing the convergence rate in each generation. Therefore, IA was found suitable and competitive in solving the scheduling problem in semiconductor assembly industry.

## Acknowledgment

The author wish to thank Universiti Sains Malaysia for the support extended for this research through grant number 304/PKOMP/6313026.

## References

- [1] U. K. Yusof, S. Deris, and R. Budiarto, "Machine allocation optimization in semiconductor assembly industry using a hybrid of bio-inspired and musical-harmony approach," *The 7th International Conference on Innovative Computing, Information and Control (ICICI-C2012)*, vol. 7, no. 3, pp. 839–845, 2013.
- [2] K.-J. Wang and S.-M. Wang, "Simultaneous resource portfolio planning under demand and technology uncertainty in the semiconductor testing industry," *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 5, pp. 278–287, 2013.
- [3] 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.
- [4] I. Harjunkski, C. T. Maravelias, P. Bongers, P. M. Castro, S. Engell, I. E. Grossmann, J. Hooker, C. Méndez, G. Sand, and J. Wassick, "Scope for industrial applications of production scheduling models and solution methods," *Computers & Chemical Engineering*, vol. 62, pp. 161–193, 2014.
- [5] 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.
- [6] H. K. Chan and S. H. Chung, "Optimisation approaches for distributed scheduling problems," *International Journal of Production Research*, vol. 51, no. 9, pp. 2571–2577, 2013.
- [7] F. T. Chan, A. Prakash, H. Ma, and C. Wong, "A hybrid tabu sample-sort simulated annealing approach for solving distributed scheduling problem," *International Journal of Production Research*, vol. 51, no. 9, pp. 2602–2619, 2013.
- [8] M. N. A. Khalid and U. K. Yusof, "An improved immune algorithms for solving flexible manufacturing system distributed production scheduling problem subjects to machine maintenance," in *Proceedings of the 2013 International Conference on Applied Mathematics and Computational Methods*, 2013, pp. 28–30.



*Table 7: Summary of the optimization result for the case study*

Package Types	$C_t$		$P_e$		$Ave_h$		$P_e$ differences (%)
	Before	After	Before (%)	After (%)	Before	After	
A	1218	609	30	60	531.33	356.33	30
B	195	65	30	80	71	42.33	50
C	308	126	100	240	128	85.67	140
D	12445	9310	20	30	5235.67	4488.67	10
E	430	172	40	90	215.67	130.33	50
F	4788	3648	40	50	2405.67	2150.67	10

- [9] L. Xing, Y. Chen, and K. Yang, "An efficient search method for multi-objective flexible job shop scheduling problems," *Journal of Intelligent Manufacturing*, vol. 20, no. 3, pp. 283–293, 2009.
- [10] A. W. Yao and Y. Pan, "A petri nets and genetic algorithm based optimal scheduling for job shop manufacturing systems," in *System Science and Engineering (ICSSE), 2013 International Conference on*. IEEE, 2013, pp. 99–104.
- [11] M. S. Anand and S. K. Sindhe, "Modeling and simulation of job shop scheduling using petri-nets," *International Journal of Engineering Research and Applications*, vol. 3, no. 2, pp. 492–496, 2013.
- [12] J. Eloundou, D. Baudry, A. Benshair, A. Louis, B. Mazari *et al.*, "Flexible job shop models for solving scheduling and layout problems using coloured petri nets," in *Applied Mechanics and Materials*, vol. 598. Trans Tech Publ, 2014, pp. 638–642.
- [13] G. Mejía and N. Odrey, "An approach using petri nets and improved heuristic search for manufacturing system scheduling," *Journal of Manufacturing Systems*, vol. 24, no. 2, pp. 79–92, 2006.
- [14] A. Seker, S. Erol, and R. Botsali, "A neuro-fuzzy model for a new hybrid integrated process planning and scheduling system," *Expert Systems with Applications*, vol. 40, no. 13, pp. 5341–5351, 2013.
- [15] G. Zhang, L. Gao, X. Li, and P. Li, "Variable neighborhood genetic algorithm for the flexible job shop scheduling problems," *Intelligent Robotics and Applications*, vol. 5315, pp. 503–512, 2008.
- [16] H. Qiu, W. Zhou, and H. Wang, "A genetic algorithm-based approach to flexible job-shop scheduling problem," in *Natural Computation, 2009. ICNC'09. Fifth International Conference on*, vol. 4. IEEE, 2009, pp. 81–85.
- [17] X. Wang, L. Gao, C. Zhang, and X. Shao, "A multi-objective genetic algorithm based on immune and entropy principle for flexible job-shop scheduling problem," *The International Journal of Advanced Manufacturing Technology*, vol. 51, no. 5, pp. 757–767, 2010.
- [18] L. Song and X. Xu, "Flexible job shop scheduling problem solving based on genetic algorithm with chaotic local search," in *Natural Computation (ICNC), 2010 Sixth International Conference on*, vol. 5. IEEE, 2010, pp. 2356–2360.
- [19] A. Prakash, F. Chan, and S. Deshmukh, "Fms scheduling with knowledge based genetic algorithm approach," *Expert Systems with Applications*, vol. 38, no. 4, pp. 3161–3171, 2011.
- [20] C. Gutiérrez and I. García-Magariño, "Modular design of a hybrid genetic algorithm for a flexible job-shop scheduling problem," *Knowledge-Based Systems*, vol. 24, no. 1, pp. 102–112, 2011.
- [21] B. Ak and E. Koc, "A guide for genetic algorithm based on parallel machine scheduling and flexible job-shop scheduling," *Procedia-Social and Behavioral Sciences*, vol. 62, pp. 817–823, 2012.
- [22] W. Teekeng and A. Thammano, "A combination of shuffled frog leaping and fuzzy logic for flexible job-shop scheduling problems," *Procedia Computer Science*, vol. 6, pp. 69–75, 2011.
- [23] Y. Kim, J. Kim, and K. Shin, "An asymmetric multi-levelled symbiotic evolutionary algorithm for integrated fms scheduling," *Journal of Intelligent Manufacturing*, vol. 18, no. 6, pp. 631–645, 2007.
- [24] M. Dai, D. Tang, A. Giret, M. A. Salido, and W. Li, "Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm," *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 5, pp. 418–429, 2013.
- [25] 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.
- [26] J. Jerald, P. Asokan, G. Prabaharan, and R. Saravanan, "Scheduling optimisation of flexible manufacturing systems using particle swarm optimisation algorithm," *The International Journal of Advanced Manufacturing Technology*, vol. 25, no. 9, pp. 964–971, 2005.
- [27] P. Udhayakumar and S. Kumanan, "Sequencing and scheduling of job and tool in a flexible manufacturing system using ant colony optimization algorithm," *The International Journal of Advanced Manufacturing Technology*, vol. 50, no. 9, pp. 1075–1084, 2010.
- [28] D. Luo, S. Wu, M. Li, and Z. Yang, "Ant colony optimization with local search applied to the flexible job shop scheduling problems," in *Communications, Circuits and Systems, 2008. ICCAS 2008. International Conference on*. IEEE, 2008, pp. 1015–1020.
- [29] L. Xing, Y. Chen, P. Wang, Q. Zhao, and J. Xiong, "A knowledge-based ant colony optimization for flexible job shop scheduling problems," *Applied Soft Computing*, vol. 10, no. 3, pp. 888–896, 2010.
- [30] S. Burnwal and S. Deb, "Scheduling optimization of flexible manufacturing system using cuckoo search-based approach," *The International Journal of Advanced Manufacturing Technology*, vol. 64, no. 5-8, pp. 951–959, 2013.
- [31] B. Naderi, M. Mousakhani, and M. Khalili, "Scheduling multi-objective open shop scheduling using a hybrid immune algorithm," *The International Journal of Advanced Manufacturing Technology*, vol. 66, no. 5-8, pp. 895–905, 2013.

- [32] J. Li, Q. Pan, and K. Gao, "Pareto-based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problems," *The International Journal of Advanced Manufacturing Technology*, vol. 55, no. 9, pp. 1159–1169, 2011.
- [33] L. Wang, G. Zhou, Y. Xu, S. Wang, and M. Liu, "An effective artificial bee colony algorithm for the flexible job-shop scheduling problem," *The International Journal of Advanced Manufacturing Technology*, vol. 60, no. 1, pp. 303–315, 2012.
- [34] L. Wang, G. Zhou, Y. Xu, and M. Liu, "An enhanced pareto-based artificial bee colony algorithm for the multi-objective flexible job-shop scheduling," *The International Journal of Advanced Manufacturing Technology*, vol. 60, no. 9–12, pp. 1111–1123, 2012.
- [35] S. Rahmati and M. Zandieh, "A new biogeography-based optimization (bbo) algorithm for the flexible job shop scheduling problem," *The International Journal of Advanced Manufacturing Technology*, vol. 58, no. 9, pp. 1115–1129, 2012.
- [36] J. Tang, G. Zhang, B. Lin, and B. Zhang, "A hybrid algorithm for flexible job-shop scheduling problem," *Procedia Engineering*, vol. 15, pp. 3678–3683, 2011.
- [37] G. Moslehi and M. Mahnam, "A pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search," *International Journal of Production Economics*, vol. 129, no. 1, pp. 14–22, 2011.
- [38] J. Li, Q. Pan, P. Suganthan, and T. Chua, "A hybrid tabu search algorithm with an efficient neighborhood structure for the flexible job shop scheduling problem," *The International Journal of Advanced Manufacturing Technology*, vol. 52, no. 5, pp. 683–697, 2011.
- [39] W. Shi-jin, X. Li-feng, and Z. Bing-hai, "Filtered-beam-search-based algorithm for dynamic rescheduling in fms," *Robotics and Computer-Integrated Manufacturing*, vol. 23, no. 4, pp. 457–468, 2007.
- [40] S. Wang, L. Xi, and B. Zhou, "Fbs-enhanced agent-based dynamic scheduling in fms," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 4, pp. 644–657, 2008.
- [41] K. Lee, "Fuzzy rule generation for adaptive scheduling in a dynamic manufacturing environment," *Applied Soft Computing*, vol. 8, no. 4, pp. 1295–1304, 2008.
- [42] Y. Honghong and W. Zhiming, "The application of adaptive genetic algorithms in fms dynamic rescheduling," *International Journal of Computer Integrated Manufacturing*, vol. 16, no. 6, pp. 382–397, 2003.
- [43] Y. Shiue and R. Guh, "Learning-based multi-pass adaptive scheduling for a dynamic manufacturing cell environment," *Robotics and Computer-Integrated Manufacturing*, vol. 22, no. 3, pp. 203–216, 2006.
- [44] 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.
- [45] Q. Zhang, H. Manier, and M. Manier, "A genetic algorithm with tabu search procedure for flexible job shop scheduling with transportation constraints and bounded processing times," *Computers & Operations Research*, vol. 39, pp. 1713–1723, 2011.
- [46] V. V. Panicker, R. Vanga, and R. Sridharan, "Ant colony optimisation algorithm for distribution-allocation problem in a two-stage supply chain with a fixed transportation charge," *International Journal of Production Research*, vol. 51, no. 3, pp. 698–717, 2013.
- [47] Y.-C. Chang, V. C. Li, and C.-J. Chiang, "An ant colony optimization heuristic for an integrated production and distribution scheduling problem," *Engineering Optimization*, vol. 46, no. 4, pp. 503–520, 2014.
- [48] J. Grobler, A. Engelbrecht, S. Kok, and S. Yadavalli, "Metaheuristics for the multi-objective fjsp with sequence-dependent set-up times, auxiliary resources and machine down time," *Annals of Operations Research*, vol. 180, no. 1, pp. 165–196, 2010.
- [49] S.-C. Wang and M.-F. Yeh, "A modified particle swarm optimization for aggregate production planning," *Expert Systems with Applications*, vol. 41, no. 6, pp. 3069–3077, 2014.
- [50] M. N. A. Khalid, U. K. Yusof, and M. Sabudin, "Solving flexible manufacturing system distributed scheduling problem subject to maintenance using harmony search algorithm," in *Data Mining and Optimization (DMO), 2012 4th Conference on*. IEEE, 2012, pp. 73–79.
- [51] K. Gao, P. Suganthan, Q. Pan, T. Chua, T. Cai, and C. Chong, "Pareto-based grouping discrete harmony search algorithm for multi-objective flexible job shop scheduling," *Information Sciences*, 2014.
- [52] M. Saidi-Mehrabad and P. Fattahi, "Flexible job shop scheduling with tabu search algorithms," *The International Journal of Advanced Manufacturing Technology*, vol. 32, no. 5, pp. 563–570, 2007.
- [53] N. Shadbolt, "Nature-inspired computing," *IEEE Intelligent Systems*, vol. 19, no. 1, pp. 2–3, 2004.
- [54] N. Jawahar, P. Aravindan, and S. Ponnambalam, "A genetic algorithm for scheduling flexible manufacturing systems," *The International Journal of Advanced Manufacturing Technology*, vol. 14, no. 8, pp. 588–607, 1998.
- [55] G. Wojtyla, K. Rzacca, and F. Seredynski, "Artificial immune systems applied to multiprocessor scheduling," *Parallel Processing and Applied Mathematics*, vol. 3911, pp. 904–911, 2006.

- [56] 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.
- [57] 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.
- [58] 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.
- [59] J. Al-Enezi, M. Abbod, and S. Alsharhan, "Artificial immune systems-models, algorithms and applications," *International Journal of Research and Reviews in Applied Sciences*, vol. 3, no. 2, pp. 118–131, 2010.
- [60] O. Kilic and Q. Nguyen, "Application of artificial immune system algorithm to electromagnetics problems," *Progress In Electromagnetics Research*, vol. 20, pp. 1–17, 2010.
- [61] 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.
- [62] C. Coello and N. Cort'es, "Solving multiobjective optimization problems using an artificial immune system," *Genetic Programming and Evolvable Machines*, vol. 6, no. 2, pp. 163–190, 2005.
- [63] 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.
- [64] 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.
- [65] S. Hofmeyr, *An interpretative introduction to the immune system*. New York: Oxford University Press, 2000, vol. 3.
- [66] F. T. 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.
- [67] 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.
- [68] F. T. S. 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.
- [69] H. Fisher and G. L. Thompson, "Probabilistic learning combinations of local job-shop scheduling rules," *Industrial scheduling*, pp. 225–251, 1963.
- [70] S. Chung, F. T. S. Chan, and H. Chan, "A modified genetic algorithm approach for scheduling of perfect maintenance in distributed production scheduling," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 7, pp. 1005–1014, 2009.
- [71] 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.
- [72] 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.
- [73] U. Ojha and M.-Y. Chow, "An analysis of artificial immune system and genetic algorithm in urban path planning," in *IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society*, 2010, pp. 1064–1069.

## Author Biographies

**Mohd Nor Akmal bin Khalid** is currently a master student in the School of Computer Sciences, Universiti Sains Malaysia (USM) main campus since 2013. He had received his bachelor degree in Computer Sciences (intelligent system) in the School of Computer Sciences, Universiti Sains Malaysia, Pulau Pinang, Malaysia in 2013. His specializations are artificial intelligence techniques, evolutionary computing and algorithms, and expert system. His research interests are optimization, scheduling, and control problems in the manufacturing.

**Umi Kalsom Yusof** is a senior lecturer in the School of Computer Sciences, Universiti Sains Malaysia since 2008. She also the school program chairman of software engineering since 2013. She received her bachelor's degree in information science (computer science) in Western Illinois University, USA, on 1986, master's degree of information technology in Universiti Sains Malaysia (USM) on 2004, and her doctorate degree in computer science in Universiti Teknologi Malaysia (UTM), Malaysia on 2013. She has previously worked in Petronas, Toyota, ASE Electronics, and Motorola before joining the academia. Her specializations are database design and management, web engineering and technologies, artificial intelligence, and evolutionary algorithms. Her research interests are resource (machine) optimization and line balancing in the area of machine loading problems and semiconductor industry capacity planning, scheduling and work-in-progress (WIP).