International Journal of Engineering & Technology, 7 (4.31) (2018) 220-227

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper



An Artificial Immune System Approach for Type E Assembly Line Balancing Problem

Mohd Nor Akmal Khalid [1], Umi Kalsom Yusof [2]*, Zarul Fitri Zaaba[3]

[1] *School of Computer Sciences, University of Science Malaysia, 11800, Georgetown, Pulau Pinang, Malaysia*
**Corresponding author E-mail:*

Abstract

The manufacturing industry has evolved in the past decades, due to the competition of the global economy, where the market demands high quality and customized products, and meeting lowest possible costs are paramount. One of the many processes in manufacturing is the assembly line operation, namely the assembly line balancing (ALB) problem. ALB problem is dependent on optimum resource utilization in order to improve production output, reduces costs, and shortens production lead times. In these recent years, various approaches have been proposed to solve the complexity of assembly line balancing operations, which composed of exact, heuristic, and meta-heuristic approaches. However, little work had been done to solve the type E assembly line balancing problem. This paper proposed an approach using artificial immune system (AIS) algorithm, namely as the artificial immune cell (AIC) approach, for solving type E assembly line balancing problem. An initialization mechanism through the bone marrow model and probabilistic clonal selection mechanism had encouraged efficient exploration and exploitation of the solution space. The computational results over 242 instances of 24 datasets had demonstrated the efficiency of the proposed AIC approach by achieving high-quality solutions (up to 85.94% optimum solutions was obtained). Also, the results were statistically justified by comparing with multi-rule multi-objective simulated annealing (MRMOSA), priority-based genetic approach (PriGA), two-phased genetic approach (2P-GA), and assignment genetic approach (MA-GA).

Keywords: *Artificial immune system; Assembly line balancing problem; Clonal selection; Job sequencing; Manufacturing system; Optimization; Type E;*

1. Introduction

Manufacturing industry has faced considerable changes in the past few years, relatively from the local economy to a highly competitive global economy. These circumstances invoke demand for high quality customizable products at the lowest possible cost with the shortest life cycles. In manufacturing, one of the core production activity is the assembly; it encompasses both the production time and cost. Consequently, the assembly is accountable between 20% and 50% on both cost and lead-time of a product; it can also reach up to 90% in newly emerged areas of micro-technologies and electronics [15]. As such, balancing the resources (machines or workstations) in term of utilization and performance during the assembly is crucial.

The assembly line balancing (ALB) is one of the problem that has been the subject of large body of the literature with wide-range of applications, including the automotive industry, consumer electronics, and household items [16]. When manufacturing high-demand products, the ALB problem arises when a firm introduces a new assembly line or redesigns an existing one. The new balanced system is expected to save capital expenditure and reduce cycle time into a value (actual cycle time) less than the value predefined by the desired production rate of the firms (theoretical cycle time). The importance of the current subject in the production research had been shown by vast numbers of studies made to solve the ALB problem.

Depending on the number of product models considered in the manufacturing production [26], the ALB problem can be further classified into single-model, mixed-model, and multi-model ALB problems. The most popular ALB problem is called the single-model or simple assembly line balancing (SALB) problem. The mixed-model ALB problem deals with several models simultaneously with negligible setup cost, while the SALB problem is characterized by mass production of the single standardized product. Meanwhile, a multi-model ALB problem involves different items which are performed in small batches, where division of labour, specialization, and standardization are still being benefited [20]. Compared to the single-model ALB problem, the mixed-model and multi-model ALB problems are generally focused on specialized markets and tend to be application-based. Meanwhile, the SALB problem has been utilized for decades to provide a basis for testing different approaches to varying characteristics of the problem with respect to the known optimality. As such, this motivates the adoption of the SALB problem as the main focus area of this study.

The SALB problems are known to be NP-hard, where j tasks and r ordering constraints result in a j!/2^r number of possible task sequences [17]. Assessing the performance in SALB problem, the most sought objective measure is the Type 1 (minimize station number, K), followed by Type 2 (minimize cycle time, C) while the least sought one is the Type E (minimize both station and cycle time number or, maximizing efficiency E). Although several approaches had been proposed in optimizing different performances of the ALB problem, Type E performance of the ALB problem had been rarely emphasized. In addition, directly solving the Type-E ALB problem (or SALB-E problem) is challenging due to difficulty of directing the search procedure to promising regions when neither machine number (K) nor the cycle time (C)

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

is fixed or known beforehand. As such, the aim of this study is to propose an approach to effectively solve the challenging combinatorial nature of the ALB problem with respect to the Type E problem.

The remainder of the paper is organized as follows. First, Section 2 presents previously published literatures and related procedures in solving the SALB domain problem. Section 3 describes in details on the SALB problem, constraints, and its performance measure. The proposed AIS approach is presented in details on Section 4. In Section 5 the proposed approach is tested on a well-known ALB datasets where the result of the proposed approach is analyzed and compared with other approaches, with respect to the SALBP-E problem. Lastly, Section 6 concludes this study.

2. Related Works

Exact or mathematical approaches always guaranteed optimal result, but have been unpopular due to its significant computational capacity. However, due to recent advances in computer and information science, exact approach is revived to solve the prominent ALB problems. Exact approach often modelled and formalized the respective ALB problem mathematically. A recently adopted exact approach is the tree-based search with partial solution [16]. Other exact approaches are the simulation approach where assignment or sequencing decisions is a sets of feasible solutions [11], and decision support approach utilizing rigid rule structure (e.g., assignment or sequencing, constraint satisfaction, etc.) to arrive for conclusion or sets of possible or probable conclusion (e.g., optimal solution) [18].

Heuristic approaches is another approach that manipulates "rule of thumb" in searching, where it guaranteed near-optimal solution (partially greedy-based procedure) that is obtained within reasonable amount of computational time. Some researchers adopted heuristic by combining it with exact approaches, such greedy-based procedure [24]. The derivation of the rules usually from the domain problems (e.g., constraints, restrictions, etc.) and working elements of the problem (e.g., workstations assignment, cycle time calculation, etc.).

Meta-heuristic approaches is an approach that derive solutions by utilizing approximation methodology to computationally challenging tasks for which an exact solution cannot achieve in polynomial time. Some authors have improvised a well-known genetic algorithm (GA) by incorporating multiple stages of GA operation [28], priority-based GA [14], and multiple assignment GA [1]. Other meta-heuristic approaches for solving the assembly line balancing problem which includes bacterial foraging optimization (BFO) [2], simulated annealing (SA) [4], ant colony optimization (ACO) [13], particle swarm optimization (PSO) [10], and artificial bee colony (ABC) [23].

Among the available meta-heuristic approaches, the artificial immune system (AIS) approach have been adopted to solve a different variety of problems in manufacturing, such as scheduling with preventive maintenance [25], engineering design [27], scheduling [6], assembly sequencing [3] and hybridized AIS approach in complex manufacturing problems [19]. An AIS approach was naturally enriched within its meta-heuristic framework with features that encompasses the ability for adaptation, self-organization, scalability, robust, and decentralization. Additionally, AIS algorithm is capable of solving problems for a wide range of areas such as optimization, data mining, computer security, and robotics [9]. AIS optimization procedure, namely the clonal selection (CS) approach, is capable of rapid exploitation and exploration of the problem's search space, embedding a diversity preservation mechanism of the solutions, and composing of elitists within its elementary structures (no backward effect during iterations). As such, the motivation of this paper is to propose AIS algorithm, more specifically the CS algorithm, for solving SALB-E problem.

To the best of our knowledge, AIS has only been adopted in solving SALB-1 problem [29]. Based on the computational experiment conducted by [29], encouraging results had been achieved where the proposed approach outperforms the previously published approaches. Besides the scarcity of AIS approach in addressing the SALB-E problem, the encouraging results obtained by AIS approach in solving SALB-1 problem acts as the motivation of adopting AIS as the proposed approach in this study. In addition, the core elements in the SALB problem (such as tasks and machines) behaved analogously to the core elements (antibody and antigen) of the AIS approach, where attempting to represent their interactions is expected to influence the performance of the SALB-E problem.

Compared to the generic AIS approach, the proposed AIS approach in this study is tailored to the SALB-E problem while generalized enough to be adapted to similar types of problem. In addition, the generic AIS approach utilizes binary representation scheme which requires modification to properly address the discrete combinatorial optimization problem of the SALB-E domain. As such, encoding the problem is not straightforward and overcoming the constraint is challenging. Therefore, the main contribution of this study is to propose a modified version of an artificial immune system (AIS) approach that can effectively address the challenging combinatorial nature of the SALB-E problem.

3. Problem Description

Table 1: Problem notations

Indices	
i	index of product in a cycle ($i=1,\ldots,I$)
j	index of task ($j=1,\ldots,J$)
k	index of station ($k=1,\ldots,K$)
Decision variables	
The decision variables possessing a value of 0 and 1 integers are as follow:	
$x_{jk} = \begin{cases} 1 \text{ if task } j \text{ is assigned to machine/station } k \\ 0 \text{ otherwise,} \end{cases}$ $j = 1,\ldots,J; k = 1,\ldots,K$	
$y_k = \begin{cases} 1 \text{ if a task is assigned to machine/station } k \\ 0 \text{ otherwise,} \end{cases}$ $k = 1,\ldots,K$	
Parameters	
I	total product produced in a cycle
J	total tasks
K	total stations/machines
C_t	the existing cycle time
t_{sum}	summation of all task processing time, $\sum t_j$
S_k	set of tasks assigned to station k
$Suc(j)$	set of direct successor of task j
$Pre(j)$	set of direct predecessor of task j
$t(S_k)$	cumulated task time at station k
t_j	processing time of task j
C	cycle time, $max\{t(S_k)\}$

Manufacturing a product on a simple or straight assembly line requires delegating the total amount of work into a set of elementary operations named *tasks j=1,…,J*. Performing a task *j* consume a *task time* t_j, and requires certain equipment of machines and/or skills of workers [14]. Due to technological and organizational requirements, several constraints have to be addressed. The precedence constraints (Figure 1) state that all predecessor of task *j* must be assigned to a machine, which is in front ($l = k-1$) of or the same as the machine that task *j* is assigned in. Assignment constraint ensures that task *j* must be assigned to only one machine. The cycle time constraint calculates the total machine time $t(S_{jk})$ on machine *k* and guarantees that the total machine time $t(S_{jk})$ is not greater than the upper bound (C_t).

The S_k set of tasks are assigned to a station $k=1,\ldots,K$, constituting its station load, in which the cumulated task time $t(S_k) = \sum_{j \in Sk} t_j$ is called station time. When a fixed common cycle time C_T is given, a line balance is feasible only if the station time of neither station

exceeds C_T. In case of $t(S_k) < C_T$, the station k has an idle time of $C_t - t(S_k)$ time units in each cycle. Finding a feasible line balance is common in any type of ALB problem. For Type-E simple assembly line balancing (SALB-E) problem, the objective is to maximize the line efficiency E:

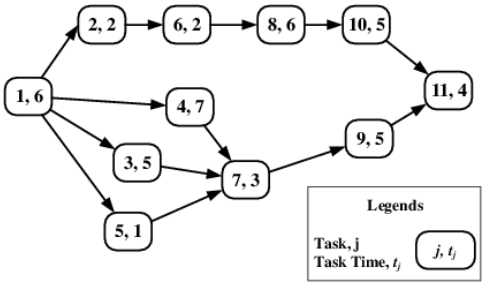

Fig. 1: An example of precedence diagram with N=11 tasks. It contains node weights for the task times and arcs for the precedence constraints. The precedence constraints of task 7 requires tasks 3, 4 and 5 (direct predecessors) and 1 (indirect predecessor) to be completed. On the other hand, task 7 must be completed before its (direct and indirect) successors, task 9 and 11 can be started.

$$E = \frac{\sum_{j=1}^{J} t_j}{K \times C}. \quad (1)$$

$$\sum_{k=1}^{K} x_{jk} y_k \geq \sum_{k=1}^{K} x_{lk} y_k, \quad \forall\, l \in Pre(j) \quad \forall\, j. \quad (2)$$

$$\sum_{k=1}^{K} x_{jk} = 1, \quad \forall\, j. \quad (3)$$

$$t(S_{jk}) = \sum_{j=1}^{J} t_j x_{jk} \leq C_t \quad \forall\, k. \quad (4)$$

$$x_{jk} \in \{0,1\} \quad \forall\, j, k. \quad (5)$$

$$y_k \in \{0,1\} \quad \forall\, k. \quad (6)$$

The precedence constraints in (2) state that all predecessor of task j must be assigned to a machine, which is in front ($l=k-1$) of or the same as the machine that task j is assigned in. Assignment constraint in (3) ensures that task j must be assigned to only one machine. The cycle time constraint in (4) calculates the total machine time $t(S_{jk})$ on machine k and guarantees that the total machine time $t(S_{jk})$ is not greater than the upper bound (C_t). The integrity constraints in (5) and (6) ensure the correct binary value of the decision variables.

4. The Proposed Swarm of Artificial Immune Cells Approach

Vertebrates (organisms having internal bones), have developed a highly complex and effective natural immune system composing vast number of cells, molecules, and organs that work as an identification mechanism capable of perceiving and combating dysfunction from its own cells (infections self) and from the action of exogenous infectious microorganisms (infectious non-selves) such as viruses, bacteria and other parasites (so-called invading pathogens). A clonal selection (CS) principle is used to elaborate the *adaptive immune system* which involves responding on any recognized antigens (the correspondence of specific proteins of the pathogen) and enhance its capability of recognizing and eliminating future encounters [12]. Variety mechanisms which formed 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 the problems in the optimization domains.

The artificial immune cells (AIC) approach which is based on CS algorithm, is composed of five major components; (1) Gene archiving (2) Solution representation and initialization; (3) Clonal selection and cloning process; (4) Hyper-mutation; and (5) Memory cell generation. The generalized overall flow of the proposed AIC algorithm starts by initializing the gene archives (task and machine sequences) where each binding of genes in both archives will be evaluated. The binding of these gene archives form a solution, which is part of population that being initialized. The population will undergo clonal selection operator. The selected population will be cloned and undergo somatic hyper-mutation. Then, the receptor editing is considered where some portion of the population is replaced by randomly generated solutions. The last component is the memory cell generation where best solution from the population will be copied and disassembled into genes for archiving in the gene archives.

4.1 Solution Archiving (Bone Marrow Model), Solution Encoding, and Initialization

The initial production of solution (or population of solutions) of the proposed AIC algorithm is based on the bone marrow model where specific segment from each gene library are combined to form the initial solution. At early stage, the initial antigen (task assignment sequence) is considered as the *self-antigen* (body's own tissues) which bind with the antibody (machine allocation sequence). This is necessary for the antibody to recognize the *non-self-antigen* presented by pathogen (foreign molecules) during infection [8]. The antibody gene library involves randomly generate machine allocation ($1 \leq k \leq K$) based on the length of the task (J). Conversely, the antigen gene library involves randomly generate task assignment sequence with respect to the precedence constraint.

The antigen gene (task assignment sequence) is generated with respect to the precedence constraint by employing a direct acyclic graph (DAG) model, where the graph is traverse starting from node(s) without predecessor followed to its successor node(s). The selection of the node is randomized where only node without unassigned predecessor is assigned directly while selected node with unassigned predecessor(s) will be skipped. If all the predecessor(s) of the selected node is assigned, then the currently selected node is assigned. This process repeats until all nodes were assigned into a list of task sequence. The size of the generated gene archives are based on the number of solutions N_{sol} (user-defined).

The initially generated gene archives (both antigen and antibody genes) are unique and no duplication allowed in order to provide the most distinct gene archives as possible. This also applies when memory cell (best solution) decomposed into its genetic materials (antigen and antibody genes) and to be stored as new gene archive. The stored gene archives also being used as the genetic components to randomly generate new solution during the initialization process as well as during the receptor editing process (see Section 4.4). Since no duplication of gene archive allowed, only unique sequence of the best solution discovered during hyper-mutation will be stored into each gene library. Note that the number of solution is flexible and may change over time (add and remove solutions) to encourage diversity among the population of the solutions (at most $N_{sol} \times 10$).

The encoding of the proposed AIC involves the representation of three distinct cells; antibody cells (machine allocation sequence), antigen cells (task assignment sequence), and memory cells. The memory cell correspond with the high performance or optimal solution, where its composition (machine and task sequences) is stored as gene archives for future generation. The binding of an antigen (tasks) with an antibody (machines) would form a feasible solution without violation of the task precedence's constraint. The binding also defines the "strength" of their receptors matching with one another, called *affinity* measure (p). The better the matching strength, the higher the affinity of the antibody-antigen bind-

ing; and vice versa. The p is directly proportional to the performance measure (Equation 1) considered for ALB domain problem.

4.2 Clonal Selection and Selection Process

A probabilistic clonal selection (PCS) is proposed by randomly select solution that is less than the pre-calculated *threshold* (p_{th}). This threshold is the average affinity of all solutions ($p_{th} = (\sum p_n)/N$). This method normalize the selection probability where low affinity solution (worst solution) will share the same chance of improving their affinity alongside high affinity solution (good solution). As such, diversity among the solutions can be retained while potential discovery of a new feasible solution can be attained.

After a solution is selected, the selected solution is cloned in fixed number, where the number of the resulting cloning for the selected solution is determined by multiplying the affinity of a single solution with 20% of the total solutions ($p_n \times \{N/5\}$). This method will increase the likeliness of unique clones and enable flexible amount of clones to be produced without compromising the diversity of the available solutions while potentially explores more areas of the search space.

4.3 Somatic Hyper-mutation

After the selected solution is cloned, the cloned solution will undergo somatic hyper-mutation (rapid mutation rate) in order to change its affinity. Typical somatic hyper-mutation would involves the affinity *maturation*, where lower affinity solution will have high mutation rate, and vice versa. However, the process will not guarantee the resultant mutated cell to have higher affinity [5]. A two-stage hyper-mutation mechanisms are proposed. By adopting the two-stage mechanism, limited dependency towards the domain problem can be fulfilled as well as sufficient control on the exploratory search. The rate of mutation (repetition of the mutation process) is defined by quantifying the affinity of the cloned solution, $(1 - p_{clone}) \times 1000$. As such, the lower the affinity, the higher the mutation rate will be. This quantifying method was chosen to provide enough maturity of the mutated solution.

The RSf mutation mechanism corresponds to random reassignment of all tasks or random reallocation of all machines. Both IpmSR and IptSR mutation mechanism imitates vaccination concept, where vaccine is used to stimulate the adaptive immune response by introducing familiar genetic materials [22]. These three mutation mechanisms (Figure 2(a), (b), (c)) are intended to provide large "jump" in the affinity landscape. In addition, these three mechanisms are focusing on significant leaps on the affinity landscape of the target solution, thus provides broader and rapid search. Meanwhile, the CPS and SRM mutation mechanism may leads new unexplored feasible solutions. These two mutation mechanisms (Figure 2(d), (e)) are important for escaping local optima and maintaining solution diversity. However, SRM mutation could cause increases in the expected machine workloads. Although the expected outcome of the CPS and SRM mutation mechanisms are contradictory, they provide their own benefits in term of exploiting the search space.

4.4 Solution Archiving (Bone Marrow Model), Solution Encoding, and Initialization

Receptor editing is process of adding new elements (or solutions) to the total population. This is necessary to enable exploration of new undiscovered feasible solution by introducing new random solution (regardless of their affinity) into the solution population in each generation cycle [21]. The main purpose of receptor editing is to preserve diversity of the population of solution, which encourages the algorithm to escape from local optimal. The proposed receptor editing process involves replacing the worst affinity solution (solution with the worst E) with new solution generated with randomly selected gene materials from the gene archives (similar to the initialization of first generation population).

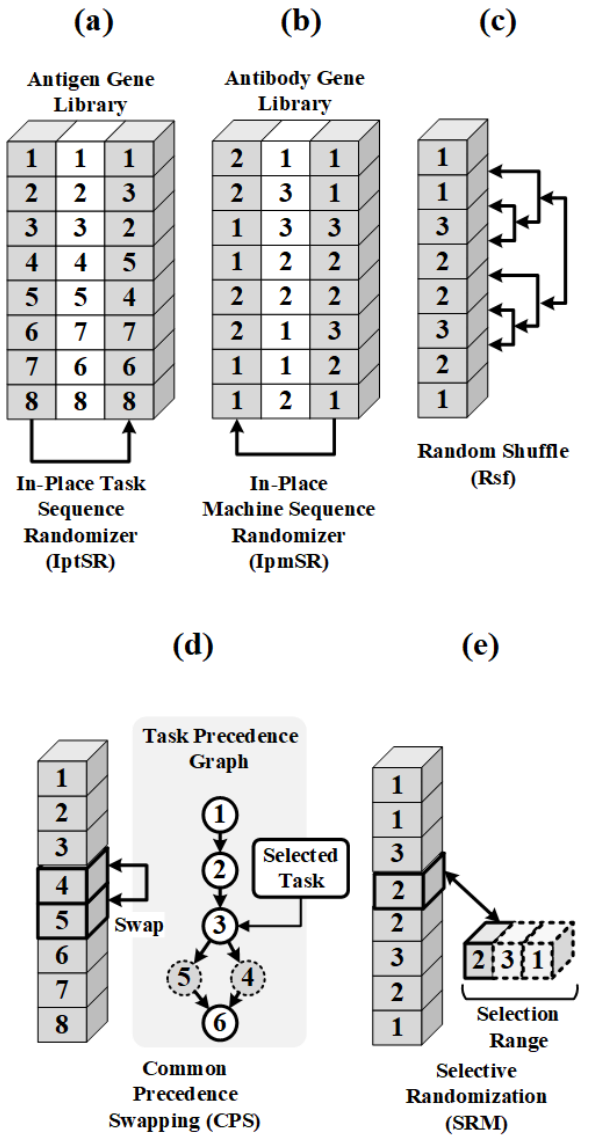

Fig. 2: An example of first stage mutation processes: (a) RSf, (b) IpmSR, and (c) IptSR, and an example of second stage) mutation processes: (d) CPS and (e) SRM. Both IpmSR and IptSR mutation mechanisms involves randomly selects a random sequence of machines or tasks. The CPS mutation involves selecting a random task from two or more task successors. The SRM focuses on reallocating one randomly selected task to another machine.

Subsequently, a certain number of solution with high affinity are typically stored as memory cells after the clonal selection and somatic hyper-mutation process [21]. In this paper, the proposed memory cell generation is based upon the cell proliferation of highest affinity solution; which involves copying the antibody gene (machine allocation sequence) or antigen gene (task assignment sequence) as a new gene archives. This memory cell will be used to form new random solution when receptor editing is triggered.

5. Experimental Result and Discussion

In this section, the experimental results over the SALB dataset and comparative results of the proposed AIC approach, against other approaches from the literature, are presented (Section 5.1). Then, detailed discussion and insights on the obtained results over the SALB data sets and comparative analysis of the proposed AIC

approach, compared to other approaches from the literature, were given (Section 5.2).

5.1 Computational Result

The proposed AIC approach was applied to the benchmark datasets for the ALB problem. The well-known instances of the SALB problem can be found on the website: http://assembly-line-balancing.mansci.de/. The characteristic of datasets of SALB-E consists of 24 precedence graphs that accumulate into a total of 242 test instances. For each dataset, the required processing time of each task and the precedence relations between the tasks are given. For each of the SALB-E data instances, a pair of numbers (k ,c_t) is defined, with k being the known number of the machines and c_t the known minimal cycle time of the assembly line. Additionally, as indicated by [7], the datasets were also categorized as a small datasets ($7 \leq J \leq 45$), medium datasets ($45 < J \leq 111$), and large datasets ($J \geq 112$).

The experiment was run 20 times where the best solution obtained from the run is recorded. Two parameters were involved for the proposed AIC approach, which is the maximum generation number (G_{max}) and the initial population size (p_{init}). After extensive experimentation with multiple parameter settings, it was found that the parameter combination of $p_{init} = 30$ and $G_{max} = 50 \times 10^3$ are the best parameter combination. Using these parameter combinations, the effectiveness of the proposed AIC approach can be measured by computing the relative deviation (dev: $\{(E^* - E)/E^*\} \times 100$) of the actual assembly line efficiency (E) from the known optimal assembly line efficiency (E^*).

The full results obtained by the proposed AIC approach is given in Table A1 (Appendix). The table contains the following information: (1) name refers to the SALB dataset; (2) K_t refers to the known optimal machine number; and (3) C_t refers to the known optimal cycle time; (4) K refers to the obtained machine number by the AIC approach; (5) C refers to the obtained cycle time by the AIC approach; (6) $\%E$ is the computed efficiency of the obtained K and C with respect to the objective given in Equation 1. In addition, a summary of the results obtained over the three dataset sizes (small, medium, and large) in regard to this performance measure is provided in Table 2. The table contains the following information:

- **avg%dev:** the average relative deviation of E from optimal E^* in percentage.
- **max%dev:** the maximum relative deviation of E from optimal E^* in percentage.
- **%opt:** percentage of the instances in the respective dataset sizes for which the optimal solution was found.

For small dataset size, the quality of the generated solutions is high, where the proposed AIC approach found the known optimal solutions in 85.94% of the dataset instances. Although the performance of the proposed AIC approach gets inferior in the medium and large dataset sizes, it is still relatively satisfactory giving solutions with average percent deviation from optimum equal to 5.29 and 6.85, respectively. Also, the worst solutions obtained by the proposed AIC approach in both medium and large dataset instances deviated from optimum no more than 32% and 23%, respectively. Given the complexity of the SALB-E problem, this indicates a competitive performance of the proposed AIC approach.

Table 2: Summary of results of the proposed AIC approach and comparison against other approaches from literatures

Result summary of the proposed AIC approach				
Dataset Size		**avg%dev**	**max%dev**	**%opt**
Small		0.23	6.67	85.94
Medium		5.29	31.34	22.02
Large		6.85	22.86	5.80
Comparison of AIC approach against other approaches				
Data No.	10	7	18	7
Instances	27	15	134	15
Efficiency (Average)	**96.52**[a]	**98.78**[a]	93.51[a]	**98.78**[a]
	91.40[b]	93.21[c]	**96.10**[d]	98.04[e]
***p*-value**	**0.0091**	**0.006**	**0.8871**	**0.2214**

[a]Proposed AIC; [b]MRMOSA; [c]PriGA; [d]2P-GA; [e]MA-GA;

Additionally, Table 2 also summarizes the comparison result of the proposed AIC approach against four approaches from the literature. The four approaches from the literature are the multi-rule multi-objective simulated annealing (MRMOSA) [4], priority-based genetic approach (PriGA) [14], two-phased genetic approach (2P-GA) [28], and multiple assignment genetic approach (MA-GA) [1]. However, each of the four approaches only applied on a certain portion of the SALB-E datasets. As such, each of the four approaches was compared individually only on the instances that were adopted. On average, the results of the AIC approach is higher the one obtained by the MRMOSA, PriGA, and MA-GA approaches, while the average performance of 2P-GA outperformed the AIC approach. A one-tailed Wilcoxon rank-sum test of the AIC approach with the significant level of $\alpha = .05$ is considered. The E values obtained by the AIC approach is significantly different when statistically compared to the one obtained by the MRMOSA and PriGA approaches (95% confidence interval). However, the difference between the E values obtained by the 2P-GA and MA-GA compared to AIC approach is not significant enough to justify statistically, but different nonetheless.

5.2 Solution Archiving (Bone Marrow Model), Solution Encoding, and Initialization

The solution behaviour in the population of the AIC approach had been influenced by the average performance values of the population (p_{th}), where most of the solutions have similar performance values where the difference was small compared to the p_{th}. This is mainly because of the proposed probabilistic clonal selection (PCS) where solution with performance less than the p_{th} will likely to be improvised in order to improve its performance or vice versa. However, since the improvised solution would not guarantee solution greater than p_{th}, potentially good solution will likely become worst. In addition, the performance of the population will likely stagnate during a higher number of the generation which causes premature convergence because of trapping in the local optimum. Also, exploitation on the currently good solutions is randomized when the value of p_{th} increases in later generation, which may also contribute to premature convergence as good-enough solution unable to escape local optima as well as becoming much worst. The PCS also have limited option since improvisation only conducted to the one with performance less than the p_{th}.

Certain insights can also be deducted from evaluating the performance and solution quality of the proposed AIC approach against other approaches from the literature. From the local optimization perspectives, direct interactions with the information dynamics of the domain problem was necessary (such as the task precedence constraints) while being generalized enough to be adopted in related areas of the domain problem. For example, various rules that were adopted by MRMOSA approach for assigning tasks to the machines does not always lead to the desired result. This is especially true when only certain rules played significant role in improving solution quality [4].

On the other hand, PriGA approach used information of its current solutions to improve their performance. Similarly, the 2P-GA approach utilizes two-phased generational improvement where the first phase seeded the second phase with best-so-far solutions to lead the overall population into better search regions after the second phase. These proved to be interesting since its search procedure is partially guided based on known information of the domain problem. Also, MA-GA approach utilizes task assignment procedures which improved candidate solution by directing the task assignment in multiple directions. These involves the exploi-

tation of the domain problem or the problem model to make critical decisions on the key areas of the problem.

As opposed to the AIS approaches, a randomized matching of the possible machine and task sequences were offered, while occasionally exchanging certain sequence information. Although these procedures do not guarantee better solutions (due to the logical paradox between task and machine sequence), it does rapidly exploit the solution where the discovery of new unexplored solution as well as advancing the more promising one is possible. The need for redundant procedures (such as the 2P-GA approach) is not needed where the search focus can be stressed in a single generational run.

However, compared to the 2P-GA and MA-GA approaches, the proposed AIS approaches have performed similarly in most instances, especially WCH approach. Even though some instances favor the proposed AIS approaches in term of balancing efficiency against the 2P-GA and MA-GA approaches, the differences were not statistically significant to either be practical or to outperform the compared approaches. Nevertheless, the comparative study were limited to the results obtained based on SALB instances adopted by each approaches, as opposed to the complete application of the SALB data set instances.

6. Conclusion

This paper proposed an AIS algorithm, which named as the AIC approach, in order to solve type E ALB problem. The proposed AIC algorithm was applied to a discrete combinatorial optimization problem, where encoding the problem is not straightforward and overcoming the constraint is challenging. These issues has been solved by adopting gene archiving for both task and machine sequences, where solution generation is independent of the domain problem. Also, high quality solution was achieved by adopting the probabilistic clonal selection (PCS) method and rapid solution exploitation was possible by utilizing the gene archiving.

However, the proposed AIC algorithm can be further improved by applying to larger problem and different ALB problems (such as machine restrictions, etc.). In addition, introducing stochastic task times in the proposed AIC algorithm and exploring multi-objective functions are possible. Lastly, the proposed AIC algorithm can also be tested and applied to investigate its performance in other type of ALB problems (such as mixed-model ALB problems).

Acknowledgement

The authors wish to thank the Universiti Sains Malaysia (USM) for the support it has extended in the completion of the present research through the Fundamental Research Grant Scheme (203/PKOMP/6711463).

References

[1] Tarek Al-Hawari, Marwan Ali, Omar Al-Araidah, and Ahmad Mumani. Development of a genetic algorithm for multi-objective assembly line balancing using multiple assignment approach. *The International Journal of Advanced Manufacturing Technology*, 77(5-8):1419–1432, 2015.

[2] Yakup Atasagun and Yakup Kara. Bacterial foraging optimization algorithm for assembly line balancing. *Neural Computing and Applications*, 25(1):237–250, 2014.

[3] MVA Raju Bahubalendruni, MVA Raju Bahubalendruni, BBVL Deepak, BBVL Deepak, Bibhuti Bhusan Biswal, and Bibhuti Bhusan Biswal. An advanced immune based strategy to obtain an optimal feasible assembly sequence. *Assembly Automation*, 36(2):127–137, 2016.

[4] Adil Baykasoglu. Multi-rule multi-objective simulated annealing algorithm for straight and u type assembly line balancing problems. *Journal of Intelligent Manufacturing*, 17(2):217–232, 2006.

[5] Ali Berrichi and Farouk Yalaoui. Bi objective artificial immune algorithms to the joint production scheduling and maintenance planning. In *Control, Decision and Information Technologies (CoDIT), 2013 International Conference on*, pages 810–814. IEEE, 2013.

[6] Arindam Chakravorty and Dipak Laha. A heuristically directed immune algorithm to minimize makespan and total flow time in permutation flow shops. *The International Journal of Advanced Manufacturing Technology*, 93(9-12):3759–3776, 2017.

[7] Albert Corominas, Alberto García-Villoria, and Rafael Pastor. Improving the resolution of the simple assembly line balancing problem type e. *SORT-Statistics and Operations Research Transactions*, 1(2):227–242, 2016.

[8] Leandro Nunes De Castro and JI Timmis. Artificial immune systems as a novel soft computing paradigm. *Soft computing*, 7(8):526–544, 2003.

[9] Leandro Nune De Castro and Fernando J Von Zuben. Learning and optimization using the clonal selection principle. *IEEE transactions on evolutionary computation*, 6(3):239–251, 2002.

[10] Jianping Dou, Jun Li, and Chun Su. A novel feasible task sequence-oriented discrete particle swarm algorithm for simple assembly line balancing problem of type 1. *The International Journal of Advanced Manufacturing Technology*, 69(9-12):2445–2457, 2013.

[11] Anoop Kumar Elia and D Choudhary. Balancing the mixed model assembly line for minimizing number of stations, cycle time and idle time by work load adjustment. *International Journal of Current Engineering and Technology*, 6(2):483–491, 2016.

[12] MY El-Sharkh. Clonal selection algorithm for power generators maintenance scheduling. *International Journal of Electrical Power & Energy Systems*, 57:73–78, 2014.

[13] Mohamed Essafi, Xavier Delorme, and Alexandre Dolgui. Balancing lines with cnc machines: A multi-start ant based heuristic. *CIRP Journal of Manufacturing Science and Technology*, 2(3):176–182, 2010.

[14] ReaKook Hwang and Hiroshi Katayama. Integrated procedure of balancing and sequencing for mixed-model assembly lines: a multi-objective evolutionary approach. *International Journal of Production Research*, 48(21):6417–6441, 2010.

[15] Romeo M Marian, Lee HS Luong, and Kazem Abhary. A genetic algorithm for the optimisation of assembly sequences. *Computers & Industrial Engineering*, 50(4):503–527, 2006.

[16] David R Morrison, Edward C Sewell, and Sheldon H Jacobson. An application of the branch, bound, and remember algorithm to a new simple assembly line balancing dataset. *European Journal of Operational Research*, 236(2):403–409, 2014.

[17] Ashkan Mozdgir, Iraj Mahdavi, Iman Seyedi Badeleh, and Maghsud Solimanpur. Using the taguchi method to optimize the differential evolution algorithm parameters for minimizing the workload smoothness index in simple assembly line balancing. *Mathematical and Computer Modelling*, 57(1):137–151, 2013.

[18] Alena Otto and Christian Otto. How to design effective priority rules: Example of simple assembly line balancing. *Computers & Industrial Engineering*, 69:43–52, 2014.

[19] Iwona Paprocka and Bożena Skołud. A hybrid multi-objective immune algorithm for predictive and reactive scheduling. *Journal of Scheduling*, 20(2):165–182, 2017.

[20] Jordi Pereira. Modelling and solving a cost-oriented resource-constrained multi-model assembly line balancing problem. *International Journal of Production Research*, pages 1–23, 2018.

[21] Xueni Qiu and Henry YK Lau. An ais-based hybrid algorithm for static job shop scheduling problem. *Journal of Intelligent Manufacturing*, 25(3):489–503, 2014.

[22] Huizhi Ren, Han Xu, and Shenshen Sun. Immune genetic algorithm for multi-objective flexible job-shop scheduling problem. In *Control and Decision Conference (CCDC), 2016 Chinese*, pages 2167–2171. IEEE, 2016.

[23] Ullah Saif, Zailin Guan, Weiqi Liu, Chaoyong Zhang, and Baoxi Wang. Pareto based artificial bee colony algorithm for multi objective single model assembly line balancing with uncertain task times. *Computers & Industrial Engineering*, 76:1–15, 2014.

[24] Armin Scholl, Nils Boysen, and Malte Fliedner. The assembly line balancing and scheduling problem with sequence-dependent setup times: problem extension, model formulation and efficient heuristics. *OR spectrum*, pages 1–30, 2013.

[25] Fatima Benbouzid-Si Tayeb and Wahiba Belkaaloul. Towards an artificial immune system for scheduling jobs and preventive main-

tenance operations in flowshop problems. In *Industrial Electronics (ISIE), 2014 IEEE 23rd International Symposium on*, pages 1065–1070. IEEE, 2014.

[26] Caijun Yang and Jie Gao. Balancing mixed-model assembly lines using adjacent cross-training in a demand variation environment. *Computers & Operations Research*, 65:139–148, 2016.

[27] MW Yaw, SP Koh, and KH Chong. The effect of mutation rate to artificial immune system algorithm optimization in solving engineering gear train problem. In *Open Systems (ICOS), 2011 IEEE Conference on*, pages 384–388. IEEE, 2011.

[28] P Th Zacharia and Andreas C Nearchou. A meta-heuristic algorithm for the fuzzy assembly line balancing type-e problem. *Computers & Operations Research*, 40(12):3033–3044, 2013.

[29] Han-ye Zhang. An improved immune algorithm for simple assembly line balancing problem of type 1. *Journal of Algorithms & Computational Technology*, pages 1–7, 2017.

Appendix

Table A1: Detailed results obtained by the proposed AIC algorithm based on small, medium, and large SALB-E datasets

Dataset Info			**Proposed AIC**		
Name	K_t	C_t	K	C	$\%E$
Small Datasets					
Heskiakoff	2	512	**2**	**512**	**100.00**
	3	324	**3**	**324**	**99.81**
	4	256	**4**	**256**	**100.00**
	5	205	**5**	**205**	**99.90**
	6	171	**6**	**171**	**99.81**
	7	147	7	150	97.52
	8	129	8	132	96.97
	9	116	9	120	94.81
Kilbridge	3	184	**3**	**184**	**100.00**
	4	138	**4**	**138**	**100.00**
	9	62	**9**	**62**	**98.92**
	10	56	10	60	92.00
Sawyer	3	108	**3**	**108**	**100.00**
	4	81	**4**	**81**	**100.00**
	5	65	**5**	**65**	**99.69**
	8	41	**8**	**41**	**98.78**
	7	47	**7**	**47**	**98.48**
	9	37	**9**	**37**	**97.30**
	10	34	**10**	**34**	**95.29**
	12	28	12	30	90.00
Buxey	3	108	**3**	**108**	**100.00**
	5	65	**5**	**65**	**99.69**
	8	41	**8**	**41**	**98.78**
	7	47	**7**	**47**	**98.48**
	9	37	**9**	**37**	**97.30**
	10	34	**10**	**34**	**95.29**
	12	28	12	30	90.00
Mertens	3	10	**3**	**10**	**96.67**
	5	7	**5**	**7**	**82.86**
Jackson	3	16	**3**	**16**	**95.83**
	4	12	**4**	**12**	**95.83**
	5	10	**5**	**10**	**92.00**
Gunther	3	161	**3**	**161**	**100.00**
	4	121	**4**	**121**	**99.79**
	5	97	**5**	**97**	**99.59**
	9	54	**9**	**54**	**99.38**
	6	84	**6**	**84**	**95.83**
	10	50	**10**	**50**	**96.60**
	11	48	**11**	**48**	**91.48**
	12	44	**12**	**44**	**91.48**
Mansoor	3	62	**3**	**62**	**99.46**
	4	48	**4**	**48**	**96.35**
Mitchell	3	35	**3**	**35**	**100.00**
	5	21	**5**	**21**	**100.00**
	6	18	**6**	**18**	**97.22**
	8	14	**8**	**14**	**93.75**
Roszieg	3	42	**3**	**42**	**99.21**
	4	32	**4**	**32**	**97.66**
	6	21	**6**	**21**	**99.21**
	8	16	**8**	**16**	**97.66**
	10	14	**10**	**14**	**89.29**
Bowman	3	28	**3**	**25**	**100.00**
	5	17	**5**	**17**	**88.24**
Jaeschke	3	13	**3**	**13**	**94.87**
	4	10	**4**	**10**	**92.50**
	5	9	**5**	**9**	**82.22**
	6	8	**6**	**8**	**77.08**
	7	7	**7**	**7**	**75.51**
Lutz1	4	3574	**4**	**3542**	**99.80**
	5	2827	**5**	**2827**	**98.47**
	6	2396	**6**	**2396**	**98.36**
	7	2096	**7**	**2096**	**96.37**
	9	1638	**9**	**1638**	**95.92**
	10	1526	**10**	**1526**	**92.66**
Medium Datasets					
Wee-Mag	3	500	**3**	**500**	**99.93**
	4	375	**4**	**375**	**99.93**
	5	300	**5**	**300**	**99.93**
	6	250	**6**	**250**	**99.93**
	10	150	10	152	98.62
	9	167	9	168	99.14
	12	125	12	128	97.59
	13	116	13	120	96.09
	15	100	15	103	97.02
	16	94	16	98	95.60
	21	72	21	75	95.17
	22	69	22	73	93.34
	26	65	26	68	84.79
	31	52	31	69	70.08
	32	48	32	69	67.89
	27	65	27	67	82.86
	35	46	35	67	63.92
	36	46	36	55	75.71
Arcus2	3	50133	**3**	**50133**	**100.00**
	9	16711	9	16717	99.96
	4	37600	4	37601	99.99
	5	30080	5	30082	99.99
	6	25067	6	25070	99.99
	8	18800	8	18806	99.97
	10	15040	10	15050	99.93
	11	13673	11	13693	99.85
	12	12534	12	12560	99.79
	13	11570	13	11595	99.78
	14	10747	14	10779	99.66
	15	10035	15	10116	99.12
	19	7928	19	8166	96.94
	20	7526	20	7770	96.78
	22	6859	22	7250	94.29
	26	5856	26	6322	91.50
Arcus1	3	25236	**3**	**25236**	**100.00**
	4	18927	4	18931	99.98
	5	15142	5	15157	99.90
	6	12620	6	12642	99.81
	7	10826	7	10830	99.86
	10	7580	10	7642	99.07
	8	9554	**8**	**9554**	**98.35**
	12	6412	12	6415	98.39
	13	5864	13	5928	98.24
	14	5441	14	5503	98.27
	15	5104	15	5180	97.44
	17	4516	17	4600	96.81
	19	4068	19	4210	94.65
Hahn	3	4787	**3**	**4787**	**97.67**
	5	2823	**5**	**2823**	**99.37**
	6	2400	**6**	**2400**	**97.40**
	8	1907	**8**	**1907**	**91.94**
Warnecke	3	516	**3**	**516**	**100.00**
	4	387	4	388	99.74
	6	258	6	261	98.85
	9	172	9	185	92.97
	10	155	10	172	90.00
	13	120	13	140	85.05
	14	111	14	132	83.77
	15	104	15	125	82.56

	17	92	17	113	80.58
	20	79	20	98	78.98
	24	66	24	88	73.30
	25	64	25	85	72.85
	27	60	27	79	72.57
	29	56	29	77	69.32
Tonge	3	1170	**3**	**1170**	**100.00**
	5	702	**5**	**702**	**100.00**
	6	585	6	586	99.83
	8	439	8	441	99.49
	14	251	14	258	97.18
	11	320	11	324	98.48
	17	208	17	214	96.48
	18	196	18	205	95.12
	19	186	19	196	94.25
	20	177	20	194	90.46
	22	162	22	182	87.66
Lutz2	3	162	**3**	**162**	**99.79**
	6	81	**6**	**81**	**99.79**
	9	54	**9**	**54**	**99.79**
	10	49	**10**	**49**	**98.98**
	12	41	**12**	**41**	**98.58**
	14	35	14	36	96.23
	15	33	15	34	95.10
	17	29	17	31	92.03
	19	26	19	27	94.54
	29	17	29	20	83.62
	20	25	20	26	93.27
	26	19	26	21	88.83
	21	24	21	26	88.83
	25	20	25	22	88.18
	24	21	24	23	87.86
	31	16	31	19	82.34
	34	15	34	17	83.91
	37	14	37	15	87.39
	44	12	44	14	78.73
	46	12	46	15	70.29
Lutz3	3	548	**3**	**548**	**100.00**
	4	411	**4**	**411**	**100.00**
	5	329	**5**	**329**	**99.94**
	10	165	10	166	99.04
	7	236	**7**	**236**	**99.52**
	9	184	**9**	**184**	**99.28**
	15	110	15	113	96.99
	14	118	14	121	97.05
	17	98	17	101	95.75
	18	93	18	96	95.14
	22	76	22	84	88.96
	19	89	19	92	94.05
	21	80	21	86	91.03
Large Datasets					
Barthold	3	1878	**3**	**1878**	**100.00**
	6	939	6	940	99.89
	5	1127	**5**	**1127**	**99.98**
	9	626	9	627	99.84
	7	805	**7**	**805**	**99.98**
	10	564	10	565	99.72
	12	470	12	472	99.47
	13	434	13	437	99.17
	14	403	14	407	98.88
Barthol2	5	847	**5**	**847**	**99.98**
	29	146	29	162	90.12
	7	605	7	606	99.81
	9	471	9	472	99.67
	11	358	11	386	99.72
	12	353	12	354	99.67
	13	326	13	329	98.99
	19	223	19	230	96.89
	16	265	16	270	98.01
	20	212	20	221	95.79
	26	163	26	173	94.13
	21	202	21	209	96.47
	22	193	22	203	94.81
	24	177	24	185	95.36
	27	157	27	179	87.61
	31	137	31	155	88.12
	35	121	35	137	88.30
	32	133	32	154	85.92
	34	125	34	137	90.90
	36	118	36	133	88.43
	40	106	40	126	84.01
	39	109	39	126	86.16
	42	101	42	128	78.76
	43	99	43	128	76.93
	50	85	50	102	83.02
	44	97	44	117	82.25
	49	87	49	92	93.92
	45	95	45	122	77.12
	48	89	48	90	98.01
	51	84	51	87	95.42
Scholl	4	17414	4	17416	99.98
	5	13931	5	13934	99.98
	8	8707	8	8711	99.95
	7	9951	7	9955	99.96
	9	7740	9	7751	99.85
	21	3317	21	3427	96.79
	10	6966	10	6974	99.88
	12	5805	12	5826	99.63
	14	4976	14	5020	99.11
	15	4644	15	4688	99.05
	16	4354	16	4409	98.74
	18	3870	18	3961	97.70
	20	3483	20	3586	97.12
	23	3029	23	3151	96.11
	27	2580	27	2805	91.97
	29	2402	29	2639	91.02
	31	2247	31	2491	90.20
	24	2903	24	3060	94.85
	25	2787	25	2967	93.91
	33	2111	33	2359	89.48
	36	1935	36	2245	86.19
	34	2049	34	2346	87.33
	37	1883	37	2147	87.68
	38	1834	38	2077	88.25
	40	1742	40	2066	84.29
	42	1659	42	1951	85.01
	44	1584	44	1915	82.67
	46	1515	46	1838	82.39
	48	1452	48	1806	80.35
	50	1394	50	1807	77.09