Received July 9, 2021, accepted August 17, 2021, date of publication August 20, 2021, date of current version August 31, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3106321

Leveraging Bio-Inspired Knowledge-Intensive Optimization Algorithm in the Assembly Line Balancing Problem

MOHD NOR AKMAL KHALID[1,2], **(Member, IEEE), AND UMI KALSOM YUSOF**[2], **(Member, IEEE)**

[1]School of Information Science, Japan Advanced Institute of Science and Technology, Nomi 923-1292, Japan
[2]School of Computer Sciences, Universiti Sains Malaysia, Gelugor 11800, Malaysia

Corresponding author: Umi Kalsom Yusof (umiyusof@usm.my)

This work was supported by the collaborative work between Japan Advanced Institute of Science and Technology (JAIST) and Universiti Sains Malaysia (USM) through Research University under Grant (RUI) 1001/PKOMP/8014084.

ABSTRACT With the increasing pressure from the market and the surge of "Industry 4.0," staying competitive and relevant is becoming more and more difficult. The assembly line, which represents a long-term investment of the manufacturing industry, needs to be efficiently utilized. While assembly line balancing (ALB) problem had been studied for decades, oversights on the bottleneck resources could significantly impede its efficiency. In leveraging such information as part of the optimization problem, a contagious artificial immune network (CAIN) approach is proposed to simultaneously address ALB efficiency and bottleneck resources while achieving a truly balanced production line. A computational experiment conducted on benchmark data sets has demonstrated a proof-of-concept, where leveraging knowledge-intensive optimization approach had successfully produced high-quality solutions up to 100% improvement with statistically significant justification. Such findings may play an essential determinant in the manufacturing industry, whether being relevant or left out in the era of increasingly being information-driven.

INDEX TERMS Manufacturing system, assembly line balancing, artificial immune system, bone marrow model, clonal selection, shifting bottleneck, Type E.

I. INTRODUCTION

The manufacturing industry had recently undergone a significant surge of "Industry 4.0" which offers a plethora of technological innovations that leverage several key enabling technologies such as internet-of-things (IoT), cyber-physical system (CPS), collaborative robots (Cobots), machine learning (ML), Big data, augmented reality, cloud computing, and additive manufacturing [1]. Such surge was partly due to the increasing pressure of market demands with mass customization (highly personalized product with high demand volumes). The assembly line balancing (ALB) problems are one such system that is defined as distributing the work tasks among the machines, satisfying both precedence and cycle time constraints [2]. The ALB problem underwent a great deal of research, which had been shown by the vast numbers of approaches through a wide range of applications (i.e., automotive industry, consumer electronics, and household items [3], [4]).

The associate editor coordinating the review of this manuscript and approving it for publication was Danilo Pelusi.

A knowledge-intensive optimization algorithm involves taking advantage of the available expert knowledge while using it to guide the optimization [5]. Compared to the knowledge-independent optimization algorithm, a knowledge-intensive optimization algorithm can achieve a high-quality solution with fewer function evaluations. Such benefits have been observed to improve the search performance in several application areas, which includes causal learning on medical application [6], engineering design [5], and energy system [7]. Nevertheless, such benefits in performance are slowly gaining momentum in the manufacturing industry.

One of the crucial parts of the ALB problem is the objective measures that represent long-term and significant investment [3]. The simple or straight assembly line balancing (SALB) problem has been utilized for decades to provide a basis for testing different approaches concerning the known optimality. This reason motivates the adoption of the SALB problem as the main focus of this study. Three primary objectives of the SALB problem are either minimization of machine number (Type-1), cycle time (Type-2), or both (Type-E).

Some of the recent studies focused on improving the performance of one objective, such as either the minimization of the machine number (Type-1) [8] or the cycle time (Type-2) [9], [10]. Meanwhile, other recent studies focused on multi-objective that were not directly related to the physical assembly line performance, such as loss of balance [11], smoothness index [11], total costs (including the processing cost and the fixed cost induced by employing workers) [12], [13], and energy consumption [12]. Another recent study focused on adopting problem-specific information, such as the consideration of preventive maintenance [14], [15], while become increasingly prominent in the garment [9], [11] [10] and automotive industries [8]. In addition, among the recently proposed approaches can be categorized into two major groups: (1) enumerative techniques such as variant of branch-and-bound [8], ranked positional weight [11], and variable neighborhood search [9], [13]; or (2) intelligence search techniques such as whale optimization [14], simulated annealing [10], [13], genetic algorithm [12], [13], and grey wolf optimizer [15].

Measuring the Type-E SALB problem (SALB-E) would determine the total productive portion of the assembly line, where significant capital costs (number of machines) and minimal demand lead time (cycle time) were addressed simultaneously. However, this objective is scarcely adopted due to its non-linearity,[1] making it more challenging to deal with [2], [3]. To the best of knowledge, the existing methods addressing the SALB-E problem to date are the multi-rule multi-objective simulated annealing (MRMOSA) [16], multi-objective hybrid improvement heuristic (MOHIH) [17], priority-based genetic approach (PriGA) [18], iterative heuristic procedure (WCH) [19], rule-based modeling and constraint programming (RMCP) [20], two-phased genetic approach (2P-GA) [21], mixed integer linear programming (MILP) [22], and multiple assignment genetic approach (MA-GA) [2].

Since the first mathematical formulation introduction by [23], various SALB approaches were introduced. Several well-known exact methods were mixed-integer linear programming [22], branch and bound algorithm [24], and assignment rules [25]. Other works focused on a more refined techniques and methods, such as meta-heuristics (see works by [16], [21], [26], [27]) and hybrids (see works by [28], [29]). It can be observed that the focus of most previous approaches were either to improve the optimization approach itself or the optimization problem to be solved. Two research directions are necessary in order to address the complex SALB problem. One direction focuses on improving the optimization approach itself since an efficient search procedure would increase the possibility of finding best known solutions. In contrast, incorporating a problem-specific knowledge will enhance the capabilities of those optimization approaches.

One such problem-specific knowledge that may enable a more exceptional production line control involves the distribution of limited machines efficiently, rather than evenly. A production line performance is usually indicated by their production rate (i.e., throughput) where one or more machines could constrain the whole system [30]. These machines are referred to as ‘‘bottleneck’’ machines [30], [31]. The primary importance of identifying the bottleneck machines is because it negatively impedes productivity and strongly impacts the production performance, especially when those machines are limited. As such, improvement efforts should concentrated on identifying the bottleneck [32].

This study is interested in problem-specific knowledge, called the bottleneck identification, which is the process of identifying the manufacturing resources (or machine) that significantly impact the performance of the production system [31]. Bottleneck that occurred at the time of the current operation of an assembly line is known as *static* bottleneck (also known as *sole* bottleneck). In contrast, a bottleneck that occurred in the future operations is called the *dynamic* bottleneck.[2] Study on *shifting* bottleneck identification had been scarce [31] and identifying it is not a straightforward procedures [30].

Some studies had addressed the identification of this shifting bottleneck via time series analysis [33], analytical model [32], discrete-event simulation [34], and augmented reality technology [35]. Bottleneck identification through simulation model demonstrates the possibility of evaluating the production system and realistically describe events in multiple time frames before realization and investment of capital [34], [36]. However, most bottleneck identification methods identify bottlenecks without emphasizing the improvements of the machine(s) afterward [37]. Also, wrong improvement [32] and a lopsided complexity of optimization and simulation [34] could degrade the performance further.

By considering the identified shifting bottleneck as part of an optimization problem, improvement action on the identified bottleneck can be formulated as a combinatorial optimization problem and part of the ALB problem. On the other hand, expanding the ALB problem with bottleneck identification provide grounds for investigation across various production problems [38]. Consequently, a knowledge-intensive optimization algorithm is suited for addressing the SALB-E through problem-specific knowledge of the shifting bottleneck identification.

An example of such an approach is the artificial immune system (AIS) algorithm, where its potentials have been

[1]Increase in machine number decreases the required cycle time, and vice versa.

[2]Also known as *shifting* bottleneck due to their shifting positions in the production system caused by operator's learning or machine's downtime

widely utilized in a variety of domain problems. Besides, AIS has previously been proven to be competitive and robust in solving a complex manufacturing problem in the past [39]. The contribution of this study is twofold. Firstly, this study introduces appropriate problem encoding, and effective improvisation utilizing the problem-specific knowledge to address the complex nature of the SALB-E problem. Secondly, on top of identifying the shifting bottleneck, this study introduces a shifting bottleneck improvement method subjected to the context and constraints of the SALB-E problem.

The remaining paper is organised as follows. First, Section II describes the SALB-E problem as well as the shifting bottleneck identification. The motivation and the proposed computational approach with three variants are presented in Section III. In Section IV the proposed approach is tested on a well-known benchmark set of instances; the approach is compared and the results are analyzed with respect to the SALB-E problem. Lastly, Section V concludes this study.

II. PROBLEM DESCRIPTION

Manufacturing a product on a simple or straight assembly line requires distributing the work into a set of elementary operations called *tasks* $j = 1, \ldots, J$. Performing a task j consumes a *task time* t_j. The task's precedence constraints have to be satisfied due to technological and organizational requirements. Another restriction includes assigning each task to exactly one machine. The S_k set of tasks are assigned to a machine $k = 1, \ldots, K$, constituting its *work load*, in which the cumulative task time $t(S_k) = \sum_{j \in S_k} t_j$ is called *machine processing time*. When a fixed common *cycle time* C_t is given, a line balance is feasible only if none of the machine processing time exceed C_t. In the case of $t(S_k) < C_t$, the machine k has an *idle time* of $C_t - t(S_k)$ time units in each cycle.

A precedence diagram represents the network of tasks needed to assemble a product that contains a node for each task, node weights for the task times, and arcs for the precedence constraints. Figure 1 shows an example of a precedence diagram with $N \equiv 11$ tasks, each having task times between 1 and 7 (time units). The precedence constraints of task 7 refer to its processing requirement where tasks 3, 4, and 5 (direct predecessors) and 1 (indirect predecessor) need to be completed. On the other hand, task 7 must be completed before its (direct and indirect) successors, task 9, and task 11 can be started.

A. BOTTLENECK IDENTIFICATION

The shifting bottleneck was identified by simulating the continuous assignment of a sequence of tasks (S_{jk}) of multiple products on the available machines K, with the violation of the cycle time (C_t). The percentage of the sole bottleneck (b^k_{total}) is then computed based on the active period of a group of assigned tasks to the machine (called momentary bottleneck [30]). The overlap of the active period of one machine with the previous or the subsequent active period of another machine represents the percentage of shifting bottleneck (b^k_{shift}), where one bottleneck may shift from one machine to the other.

In this study, the simulation is conducted by utilizing the bottleneck indicator matrix, adopted from [30], consists of an element of machine index (row = k) and time transition (column = t_{sum}) of the task processing on each machine. The time transition represents the same size as the sum of the total task time ($\sum_{j=0}^{J} t_j$). As such, the matrix size would be $k \times t_{\text{sum}}$. There are several matrices involved in producing the final bottleneck indicator variable where the primary bottleneck machine is identified (see Figure 1 in Section III-A) based on the following matrices:

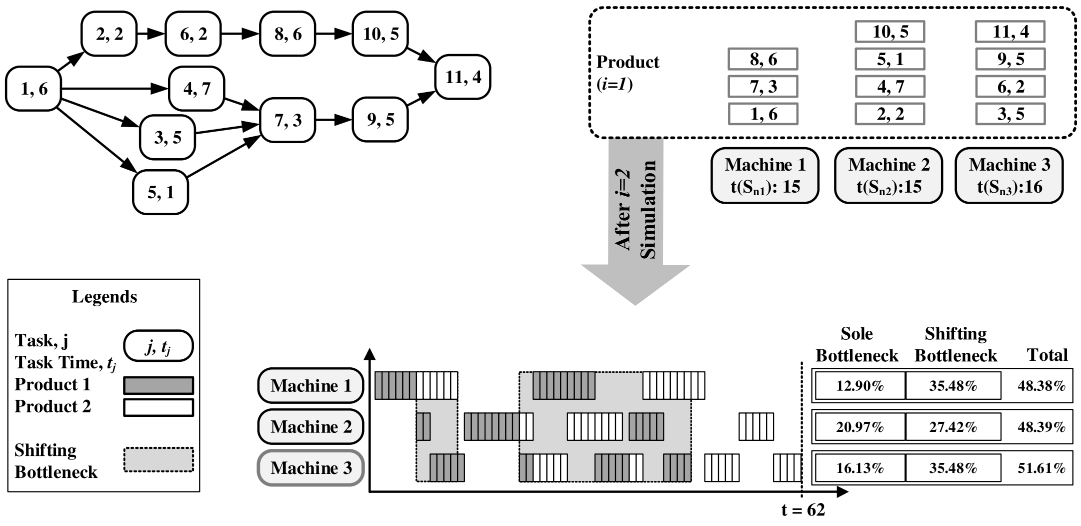
- Matrix A is the state transition matrix where the task time is expanded into individual task time onto each matrix cell for each machine
- Matrix B is the state accumulation matrix where each of the expanded task times is accumulated onto each time transition
- Matrix C is the shifting bottleneck matrix induced based on the overlap of the task time between the machine in Matrix B. The shifting bottleneck percentage is calculated by dividing the overlapping columns over the total column size, which is stored in a supplementary bottleneck variable (b^k_{shift})
- Matrix D is the sole bottleneck matrix (can alternatively be obtained by $D = B - C$) that calculated similar to Matrix C and stored in another supplementary bottleneck variable (b^k_{sole})
- Matrix E is the primary and secondary bottleneck matrix which summed up the supplementary bottleneck variables of Matrix C and Matrix D as the total computed bottleneck percentage (b^k_{total}), where its highest value represents the primary bottleneck.

B. PERFORMANCE MEASURES AND CONSTRAINTS

In the SALB problem, the cycle time is the time between consecutive releases of the assemblies at the end of the line or the total time (maximum time) allocated to each machine in the assembly line. All machines have the same cycle time. In addition, the cycle time and the number of machines are expected to be inversely proportional [40]. If the cycle time is more, the number of machines is expected to be less and vice versa.

The production lead time with minimal time window is an important issue to be demand-responsive which addressed by obtaining the minimum cycle time (C). Meanwhile, constructing a production line requires significant capital cost, which requires minimizing the number of machine (K). This study considers simultaneous minimization of C and K, defined by the maximization of the assembly efficiency (E) in (1). According to [41], maximizing E can also minimizes the horizontal, vertical balancing, total idle time, and balancing delay.

$$\max E = \left\{ \frac{t_{\text{sum}}}{(K \times C)} \right\} \times 100 \tag{1}$$



	Sole Bottleneck	Shifting Bottleneck	Total
Machine 1	12.90%	35.48%	48.38%
Machine 2	20.97%	27.42%	48.39%
Machine 3	16.13%	35.48%	51.61%

FIGURE 1. Example of a shifting and sole bottleneck situation simulated using the example precedence diagram with 11 tasks. There are several products, labeled as product I, where $i = 1, \ldots, I$, and three available machines of the assembly line will process these products, given that the task sequence (S_{jk}) is 1-3-2-4-5-7-6-8-10-9-11 and the resultant machine allocation ($S(x_{jk})$) is 1-3-2-2-2-1-3-1-2-3-3. Based on the maximum $t(S_{jk})$, the identified bottleneck machine is machine 3. The bottleneck occurred when the processing of tasks for the next products arrived at the assembly line, but machine 3 was still in progress of finishing its tasks. A different machine allocation or task assignment sequence can be planned; thus, mitigating long-term performance hindrances of the assembly line caused by the bottleneck machines.

The complexity of SALB-E problem can also be interpreted mathematically. A general combinatorial optimization problem mainly consists of a finite ground set $U = 1, 2, \ldots, n$, a subset of feasible solutions $F \subseteq U$ and a cost function f computing the cost of feasible solutions [42]. The goal of combinatorial optimization is to find the best known solution in all feasible solutions which has the minimum cost. By considering combinations (canonical to the U) of different number of machines $K \in [K_{\min}, K_{\max}]$ and all possible number of cycle times $C \in [C_{\min}, C_{\max}]$ that induces the highest efficiency measures E (canonical to the f) of an assembly line, the possible solutions would be $K \times C$ where the SALB-E problem becomes a complex NP-hard combinatorial optimization problems [43], [44].

The precedence constraints in (2) state that all predecessors of task j ($Pre(j)$) 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. The 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 machine number constraint in (5) ensures that machine k is within an allowable bound ($K_{\min} \leq K \leq K_{\max}$). The integrity constraint in (6) ensure the correct binary value of the decision variables.

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

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

$$T_j = \sum_{j=1}^{J} t_j x_{jk} \leq C_t \quad \forall k. \tag{4}$$

$$K \in \{K_{\min}, K_{\max}\}. \tag{5}$$

$$x_{jk} \in \{0, 1\} \quad \forall j, k. \tag{6}$$

III. THE ARTIFICIAL IMMUNE SYSTEM APPROACH

The artificial immune system (AIS) is a biologically-inspired algorithm inspired by the principles and processes of the vertebrate immune system, divided into four main branches [45]: immune network, clonal selection, bone marrow model, and negative selection. In this study, only the first three are discussed (see Figure 2). A shared feature among the three models is the antibody cell. The degree of similarity (in a given metric space) of the antibody is called *affinity*. The immune system's relevant events revolve around the antibody cells, their activities, and their interactions.

The artificial immune network (AIN) model, initially proposed by [46], suggests that the immune system is composed of a regulated associative network of antibody cells.[3] The AIN model can change the essential components of its network to dynamically adapt its structure without external intervention, making it proficient for noise tolerance, self-organized behaviors, and unsupervised learning [47], [48]. On each iteration, AIN suppresses cells similar to one another (similar solution) and recruit new cells (from bone marrow) to replace the worst cells. In the end, an optimized network is gradually formed for a target problem [45] (see Figure 2(a)).

Meanwhile, the clonal selection (CS) model (see Figure 2(b)) suggests antibody cells undergo clonal expansion (cellular reproduction based on cloning) and proliferation (subjected to a certain degree of differentiation or *hyper-mutation*) [45]. The proliferation rate of immune cells is directly proportional to their affinity; higher affinity

[3]Recognizes one another, as well as maintain an internal image of all existing invaders (or pathogens) exposed during its lifetime

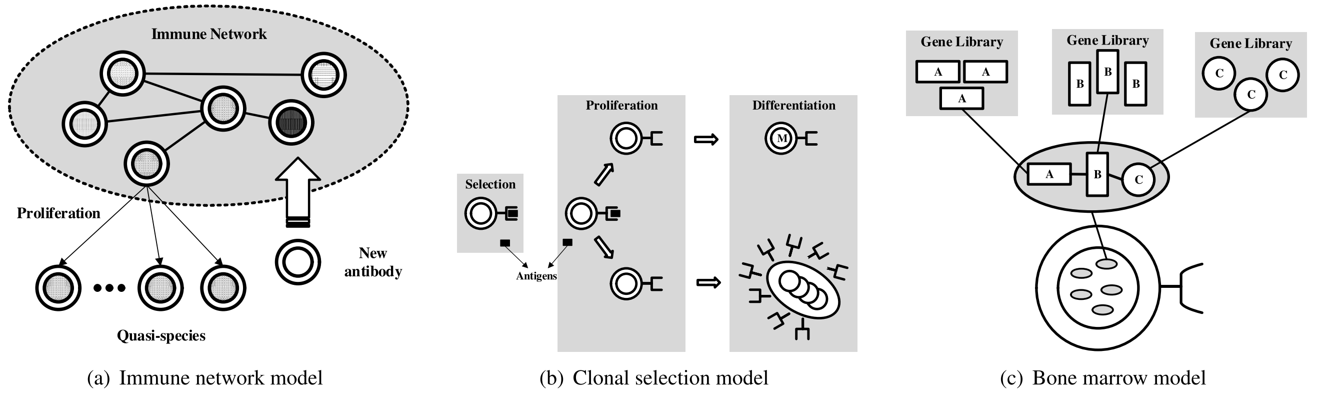


FIGURE 2. Three AIS theories adopted in this study.

produces a higher number of clones and vice-versa. Meanwhile, the hyper-mutation rate is inversely proportional to the cell affinity; low affinity requires a high rate of mutation to raise the affinity values quickly [49].

The bone marrow (BM) model is a simplification of the actual mechanism that generates sufficiently diverse repertoire of antibody cells.[4] The cells undergone a pseudo-random process, in which recombination of DNA results in different lymphocyte genes, which then produces different receptors; hence, diverse antibody cells. The bone marrow model can be utilized for solving basic combinatorial problems (such as the sequencing problem) [51]. However, the full extent of the true capabilities of the BM model is currently unexplored. The features offered by the BM model includes elimination of redundancy[5] and scalable generation[6] of possible combinations of solutions.

To this end, the complex interaction of the immune cells and its networking act as the source of inspiration for an adaptive and robust optimization approach with regards to the ALB problem. This study focuses on enhancements of the combination of BM, AIN, and CSA models (regarded as the bare-bone AIS approach), to address the SALB-E problem with shifting bottleneck. Relative to other much popular approaches, such as the trending artificial neural networks (ANN) and genetic algorithms (GA) in machine learning and computational intelligence, the CAIN approach introduces a unique take on an operational research study where problem-dependent knowledge is modeled in the problem-independent mechanism to provide a proof-of-concept approach. Such an approach could potentially be a general-purpose solver for other problems in computational intelligence domains (e.g., scheduling, planning, strategic decision-making). In addition, CAIN represents a combination of population-based search and agent-like technique, which provide the foundation for achieving such a general-purpose solver.

[4]For research concerning receptor diversity, see original work by [50]

[5]Not all genes existent in the genotype (total collection of genes) are expressed in the phenotype (generated antibody molecules).

[6]A relatively small number of genes c in the libraries l can generate a large number of different receptor molecules (c^l).

A. CONTAGIOUS ARTIFICIAL IMMUNE NETWORK (CAIN) APPROACH

The "contagious" nature of the proposed CAIN approach was inspired by the work in [45] and [52], which was achieved in twofold. Firstly, an individual of the population would quickly infect the neighboring individuals for competitive improvement (exploration) [45], simulating a "mass psychogenic illness" phenomenon.[7] Secondly, each individual undergoes repetitive mutation within the contagious region of a "hotspot" site, which accelerates its affinity maturation (exploitation) [52]. These were reflected through the clonal competition mechanism (CCM) operator and contagious mutation operator.

The process starts by initializing the solutions through the bone marrow model (Figure 3). To ensure the precedence constraint is preserved, the task sequence is generated by a direct acyclic graph (DAG) model, where the graph traverse from the node(s) without predecessor followed to its successor node(s). The node selection is randomized, where the only node without unassigned predecessor is assigned directly, while the selected node with unassigned predecessor(s) will be skipped (until there is none left). The process repeats until all nodes are assigned to a complete task sequence (S_k). Similar to the model proposed by [51], unique sequences are stored to avoid redundancy, computational time, and complexity. The proposed bone marrow model differs on two terms. Firstly, the proposed bone marrow model consists of only two sequence information (task and machine) instead of similar parts sequences. Secondly, the proposed bone marrow model produces solution sequences that preserves the formulation described in Section II-B.

The solution is generated by randomly match the task assignment and machine allocation from the bone marrow model into the two-dimensional solution encoding (Figure 4). The encoding scheme is partly inspired in the work by [54]. This solution encoding incorporates the most amount of information where the row represents the task number $(1, \ldots, J)$ while the column represents the machine number $(1, \ldots, K)$.

[7]A phenomenon caused by someone illness that happened in isolation, effecting others people psychologically rather than actual illness [53].

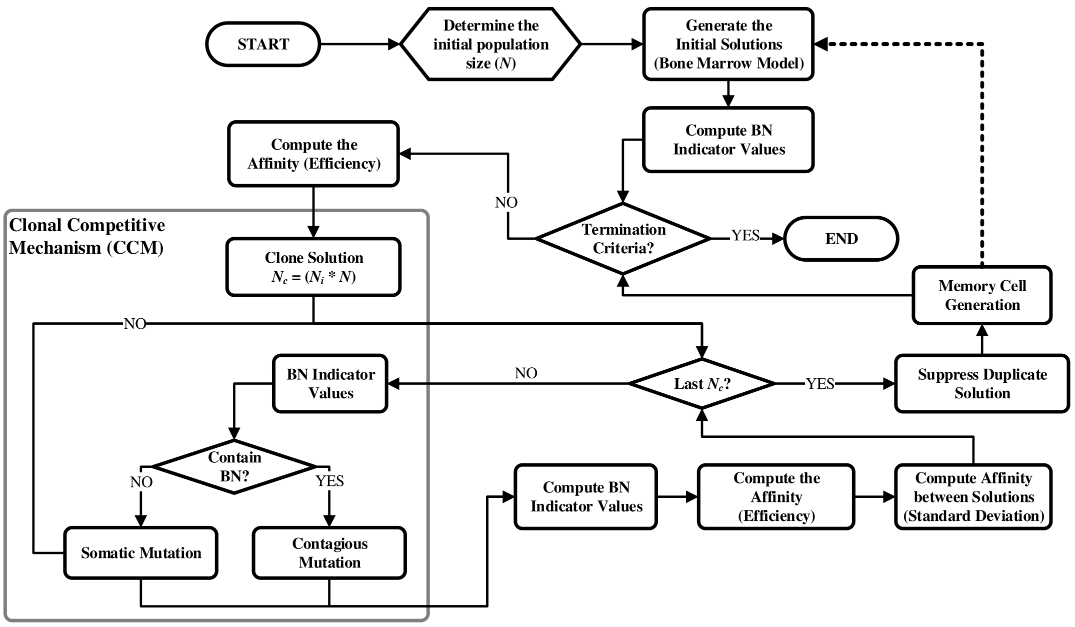

START
Determine the initial population size (N)
Generate the Initial Solutions (Bone Marrow Model)
Compute BN Indicator Values
Termination Criteria?
YES
END
NO
Compute the Affinity (Efficiency)
Clonal Competitive Mechanism (CCM)
Clone Solution $N_c = (N_i * N)$
NO
Memory Cell Generation
Last N_c?
YES
Suppress Duplicate Solution
BN Indicator Values
Contain BN?
NO
YES
Somatic Mutation
Contagious Mutation
Compute BN Indicator Values
Compute the Affinity (Efficiency)
Compute Affinity between Solutions (Standard Deviation)

FIGURE 3. Flowchart of the CAIN approach. The main components of the CAIN approach were the clonal competition mechanism and affinity (performance) manipulation of within and in-between solutions. In addition, the bottleneck (or BN) indicator values and the contagious mutation operator contribute to the proposed CAIN approach's knowledge-intensive framework.

Then, the solution's content consists of a positive number representing the task assignment $(1, \ldots, N)$ where its column position represents the machine allocated conformed with the precedence constraint. Incorporating the proposed solution encoding simplifies the evaluation mechanism, albeit more difficult to initialize. The encoding scheme provides a significant boost in reducing the combinatorial complexity of the problem search. An encoding of task $J = 5$ with machine $K = 2$ of order strength $R = 0.01$ $(\{J!/(2^R)\}^K \approx 14,201)$ is reduced to $\{J!/2^R\} \times K \approx 238$ of possible solutions. Additionally, the encoding works in tandem with other CAIN operators (e.g., mutation and archiving of the solutions) to ensure a high level of population diversity.

Each solution is evaluated based on three measures: (1) affinity value, p (the assembly line efficiency, E), (2) standard deviation of the workload of machines, and (3) bottleneck indicator, b. These measures constitute the "knowledge" of the proposed CAIN approach that can effectively be used to act upon the solution. In a way, this knowledge enables the CAIN approach to propagate, differentiate, and improve the population of solutions towards a better region of the search space effectively.

Typical affinity measure between solutions was calculated through distance computation and user-defined thresholds [45]. In this study, affinity measure between solutions was compared based on task assignment, the sequence of the tasks on each machine, and the standard deviation of the machines' cycle time of each solution. As such, similar affinity solutions can be distinctive, and suppressing duplicate and low-affinity solution can be conducted with ease. The bottleneck indicator is calculated using the bottleneck indicator matrices to represent bottleneck identification

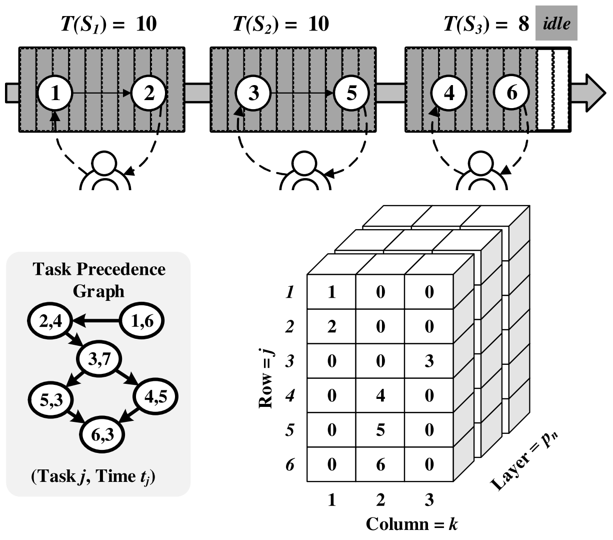


FIGURE 4. Example of a solution representation based on the example assembly line assignment with three resources (machine, stations, operators, etc.) and six tasks (top), assigned conforming to the precedence constraint given by the precedence diagram (bottom left). The solution encoding is generated by mapping the row and column to its equivalent tasks and resource designation.

mechanisms (see details in Section II-A), which constitute CAIN's knowledge-intensive component.

The population size is based on the number of task sizes $(N = J)$ to enable the CAIN approach to scale with the problem sizes. Also, the generation size is readjusted "on-the-fly" throughout the generations (based on the subsequent operators). Next, each solution will undergo a clonal competition mechanism (CCM) in each generation, which selects a set of solutions for cloning and hyper-mutate (see

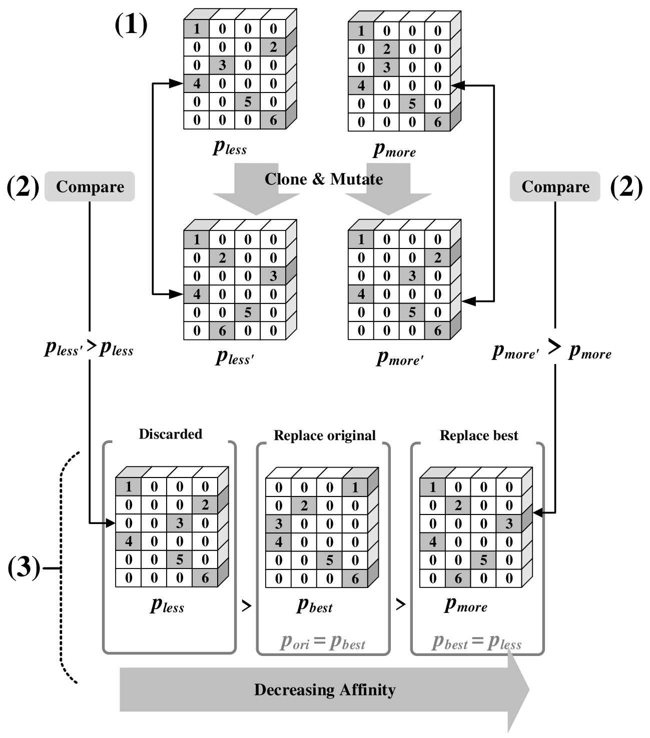


FIGURE 5. Clonal competition mechanism.

Figure 5). The clone number is determined based on the solution affinity multiplied by the total solutions ($p_n \times N$). The procedure starts by randomly selects $0.05 \times N$ solutions and comparing their affinities with a threshold, $p_h = \frac{\sum_{n=1}^{N} p_n}{N}$. The benefit of the proposed CCM is demonstrated by the competition between solutions for the whole range of performance (p_{more}, p_{less}, and p_{best}). This situation encourages fast convergence where both the best and worst solutions are directed towards a better region of the search space. Also, diversity in the population is retained, which helps in escaping local optimum.

The cloned solution will undergo hyper-mutation according to the bottleneck indicator, the somatic mutation for a solution without a unique bottleneck indicator (MS-1 and TS-1 mutation procedures), and contagious mutation for a solution with a unique bottleneck indicator (RS mutation procedures). The MS-1 and TS-1 perform a simple swapping mechanism[8] focusing on workload changes of the machine allocation and sequence exchanges of the task assignment, respectively. Since the trade-off between exploitation and exploration plays a crucial role in improvisation [55], the MS-1 explores the possibility of new solution discovery based on current situation while TS-1 exploits known information of the task (such as task time) to improve the current solution. The rationale of the proposed mutation procedures includes a reliable improvisation of the solutions, low computational overhead, and rapid solution exploration of the domain search space.

[8]Selecting two random variable/cell/data and swap them.

The contagious mutation operator is focused on resolving the primary bottleneck machines of the respective solution (so-called "hotspot"). The procedure, known as repetitive swapping (RS), leverage the knowledge of bottleneck identification in the solution by swapping out repeatedly and randomly the tasks of the bottleneck machine into other machines. The likelihood of trapping in local optima was also reduced since the procedures *force* the solution to rapidly explore more than the immediate neighborhood of the solution. As such, the mutation rate (M) for both somatic and contagious mutations is determined based on the affinity of the cloned solution as well as user-defined mutation steps ω (Equation 7).

$$M = N \times \exp\left\{-\omega \times \frac{p_n}{100}\right\} \tag{7}$$

Elitism strategy is adopted for accelerating the CAIN approach, where the best solution from the population (memory cell) is copied and stored as an archive for the creation of a new potentially high-performance solution (e.g., during suppression). This strategy is used to encourage the propagation of the population towards optimal performance while offering population diversity. Such a strategy also boosts the discovery of a new unexplored area of the search space (non-dominated solution) and advancing the high-affinity solutions. The initial generation size is $G_n = 1$. If the affinity of the best solution is $p_{\text{best}} < 100\%$, then the generation size $G = G + 1$. The approach terminates based on the current best-performed solution ($p_n \geq 100\%$) or the maximum time limits of 900 seconds. Both best known solution and near-best known solution can potentially be discovered.

B. APPROACH COMPARISON

Evaluating the performance and solution quality of the proposed CAIN approach, its differences compared to the approaches presented in the literature need to be described. These should provide insights to associate such approaches in what perspective the problem is being addressed compared to the proposed CAIN approach. Three perspectives of the problems formed the foundations of solving the complexity of the SALB-E problems: global, local, and problem-dependent.

From the global optimization perspective, the SALB-E was addressed by exploiting the approach structures without any direct influence on the information of the problem domain. For example, 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 [21]. Meanwhile, the PriGA approach used information of its current solutions to improve their performance [18]. This is interesting technique since its search procedure is partially guided based on general information of the domain problem.

From the local optimization perspectives, direct interactions with the information dynamics of the domain problem were necessary (e.g., assembly task assignment and task precedence constraints) while being generalized enough to be adopted in related areas of the domain problem. MRMOSA approach adopted various rules for assigning tasks to the machines, and only specific rules played a significant role in improving solution quality [16]. Meanwhile, the RMCP approach had adopted an alternative rule-based model, where direct modeling of the precedence constraint can be formulated to reduce the computational burden and evaluation number [20]. In addition, multiple alternatives of the problem model can be produced, which directly influences the cycle time and machine number.

Finally, exploiting the domain problem was necessary from a problem-dependent perspective, where knowledge is utilized for critical decisions. For example, the MA-GA approach utilizes task assignment procedures which improved candidate solutions by directing the task assignment in multiple directions [2]. In another perspective, the WCH approach introduces solution procedures by iterating through type-1 objective followed by type-2 [19]. This method is then translated to a solution procedure meant for the SALB-E problem. Meanwhile, the MILP approach introduces a generalized cutting plane method with valid inequalities based on the precedence relations to solve the task bounded model of the SALB-E problem and reduce computational time [22]. Likewise, the MOHIH approach had hybridized an adaptive learning strategy that adjusts the priority weight metric of the tasks and employs simulated annealing (SA) algorithm to escape local optimum and provide effective neighborhood search [17]. However, the MOHIH approach is dependent on station-oriented representation, which "learn" over time the best priority of the tasks where consideration of the adaptive learning procedure also requires additional control.

IV. COMPUTATIONAL EXPERIMENT AND RESULT ANALYSIS

The performance evaluation of the proposed CAIN algorithm is conducted using a computational experiment on benchmark data sets consisting of 24 precedence graphs and 242 data instances. This benchmark data sets for the SALB-E problem were taken from both open literature [56] and websites.[9] For each data set, the processing time of each task and the precedence relations between the tasks are given. Moreover, each data instances defines a pair of numbers: the known minimal number of the machines K_t and minimal cycle time C_t. The proposed CAIN algorithm was implemented using a C# compiler and run independently on a personal computer equipped with a 2.0 GHz Intel Core i5 processor and 2Gbs of RAM.

There are three categories of data sets as indicated by [57]; small ($7 \leq J \leq 45$), medium ($45 < J \leq 111$) and large ($J \geq 112$) data sets. As outlined by [56], the order strength ($OS \in [0, 1]$) is defined as the relative number of precedence relations in an acyclic graph with J nodes (number of precedence relation $/(J * (J - 1))$). The OS value indicates the complexity of solving the problem. The

[9]https://assembly-line-balancing.de/salbp/benchmark-data-sets-1993/

TABLE 1. The parameter settings of the CAIN approach.

Parameters	Settings
Initial population size (N)	J (adaptive)
Generation number (G)	$1, \ldots$ (adaptive)
Maximum population (N)	$N \times 10$
Affinity threshold (h)	$(\sum_{n=1}^{N} p_n)/N$
Solution archives (N_{archive})	N
Clone number (N_{clone})	$p_n \times N$
Mutation rate (M)	$\lceil N \times \exp\{-\omega \times \frac{p_n}{100}\} \rceil, \; n \in N$
Mutation step (ω)	$[0.1, 1.0]$

precedence constraints are permissive, and many sequences of tasks are feasible when OS value is small [4]. Hence, $OS \approx 0.5$ could be more difficult to solve since it is neither permissive nor restrictive.

In the subsequent sections, the adopted control parameters and settings used for the proposed CAIN approach are discussed (Section IV-A). The complete results of the proposed CAIN approach obtained from the Type E ALB data sets are presented (Section IV-B). An illustrative example of the improvement gained in the ALB problem when considering the shifting bottlenecks is presented in Section IV-C. Finally, Section IV-D discusses the comparative analysis of the CAIN approach compared to other approaches from the literature.

A. CONTROL PARAMETER CALIBRATION OF THE PROPOSED CAIN APPROACH

As mentioned by [58], a large population size would likely induce a more feasible solution close to the best known solution. However, the initial population size (N) of the proposed CAIN approach is actively changed (from the CCM and suppression operators) when generation number (G) increases, in which the initial values are $N = J$ and $G = 1$. Also, the only tunable parameter is the mutation step (ω). A full factorial design of the experiment was adopted to calibrate such a parameter where different level combinations were explored to obtain the best-expected results [59], [60]. All settings of *omega* $\in [0.1, 1.0]$, with the step size of 0.01, were considered in an experiment that was designed with 20 runs (Table 1).

The instance solved (in percentage) of SALB-E data sets given by the proposed CAIN approach for different levels of ω is depicted in Figure 6. The maximum percentage solved, and its confidence intervals[10] are recorded. It can be observed that the change in ω levels fluctuates around 90 to 100 %. Also, the minimum instance solved was obtained when $\omega = 0.61$ (about 90.17%). The average confidence interval for the percentage of instance that was solved for all *omega* levels is ± 0.63089 (about $\approx$ 2 instances), which showed the reliability of the CAIN approach in solving the greatest number of instances of the SALB-E data sets. Hitherto, it was found that $\omega = 0.5$ is the best parameter where the SALB-E data set was fully solved.

[10]Taking the run number as the sample and confidence levels of 99%, the confidence interval for each ω levels can be calculated.

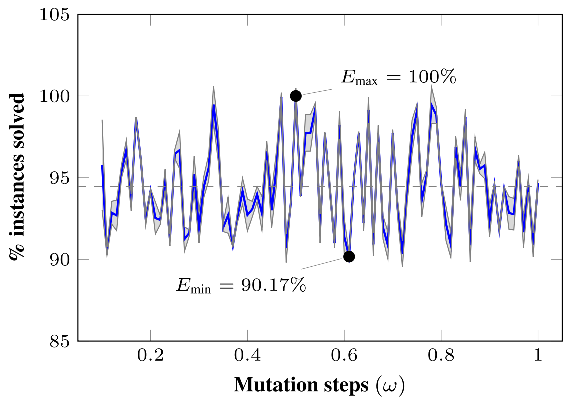


FIGURE 6. Experimental results summary of the total percentage optimal obtained for determining the optimal parameter of all range of possible ω parameter of the proposed CAIN approach. The dashed line is the average % optimal obtained of 94.45% over all SALB-E data sets. The average confidence intervals (99% confidence level) for 20 runs of all ω levels is ± 0.63089.

The calibration of the CAIN approach involves only a single parameter (mutation steps, ω), while the other parameters are not fixed or user-defined. Such a setting enables a fine-tuned performance of the search procedure [61], [62]. The population size and cloning size was adapted dynamically throughout the generation size. This situation had been empirically proven to be an impetus driving force in other approaches closely resembles the AIS approach (e.g., genetic algorithm) [21]. This situation makes the CAIN approach deal less parameter and, at the same time, operates on the whole range of population sizes. Also, the number of generations was no longer affecting the CAIN approach's capability, which was reflected in the best known solutions achieved with a fast convergence rate.

B. NUMERICAL RESULTS OF THE PROPOSED CAIN APPROACH

The full details of the obtained results from the proposed CAIN approach for all data sets are depicted in Figure 7, where the quality of the obtained results is measured by computing the relative deviation[11] of the obtained E from the best known solution reported from previous studies ($E^{\star}$). The CAIN approach discovered a better solution candidate for all instances of the Bowman, Lutz1 data sets (small-sized data set), and Hahn data set (medium-sized data set). Besides, the proposed CAIN approach had successfully found all the best known solutions for all other instances of the SALB-E data sets. Given the complexity of the SALB-E problem, this indicates a high-quality performance of the proposed CAIN approach.

Figure 8 depicted the average T spent by the proposed CAIN approach with its confidence intervals (99% confidence level) ranging about ± 6.16103, ± 239.54908, and ± 299.92014 for small-, medium-, and large-sized instances of SALB-E data sets, respectively.[12] The best solution was

[11]dev: $\frac{(E^{\star}-E)}{E^{\star}} \times 100$

[12]Taking the run number as the sample and confidence levels of 99%, the confidence interval of T for each data set instances can be calculated.

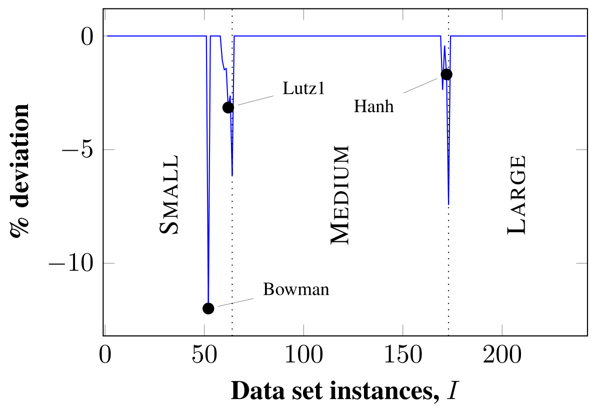


FIGURE 7. Numerical results obtained by the CAIN approach based on instances of small, medium, and large data sets ordered by increasing task sizes (J).

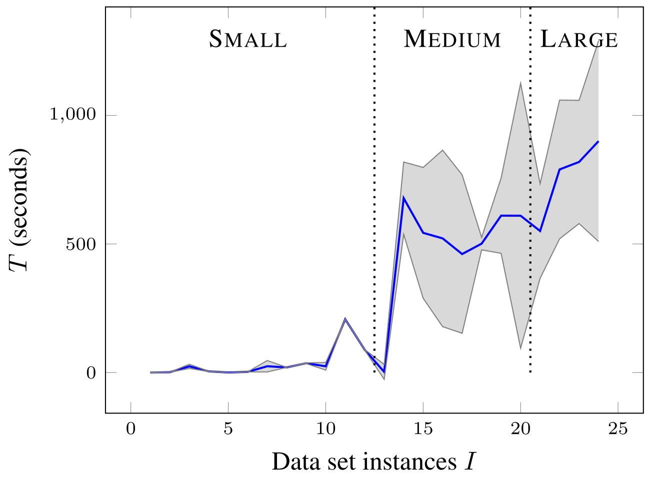


FIGURE 8. The computational time (T) used by the CAIN approach based on three types of SALB-E data set instances (small, medium, and large) ordered by increasing task size (J). The average confidence intervals (99% confidence level) for 20 runs of the SALB-E data sets instances are ± 6.16103, ± 239.54908, and ± 299.92014.

discovered within T values up to $\approx$4 minutes, $\approx$14 minutes, and $\approx$22 minutes for small-, medium-, and large-sized data sets, respectively. The T spent on the proposed CAIN approach is propositional to the problem size of the SALB-E data sets. Hence, the reported T is considered to be sufficiently fast, given the complexity of the SALB-E problem.

C. ILLUSTRATION OF THE BOTTLENECK IMPROVEMENT OF SOLUTION

The application of the proposed CAIN approach on a single small-sized SALB-E dataset was considered to demonstrate the significance of leveraging knowledge in the improvement process of the shifting bottleneck (Table 2). The first instance of the Lutz1 data set (32-tasks) was adopted as an example, where a new solution candidate is discovered for machine and cycle time of 4 and 3574, respectively. The task time ranges from 100 to 1400, whereas the order strength is 83.5 (the highest for small SALB-E data sets). The first column represents the available machines allotted, while the second column represents the tasks assigned to the allotted machines.

TABLE 2. Example ALB solutions for the first instance of Lutz1 dataset.

	Solution A	$t(S_k)$
$k=1$	1,3,4,5,6,11,14	3536^a
$k=2$	2,7,8,9,10,12,13,19,21,32	3534^a
$k=3$	15,16,17,18,20,24,31	3528
$k=4$	22,23,25,26,27,28,29,30	3542^a
	%E	99.8024
	Solution B	$t(S_k)$
$k=1$	5*,8,11*,20,22,26,28,30,32	3536^a
$k=2$	1,2*,6,10*,12*,14,21*,24,25,27,31	3534
$k=3$	3,7,13,15*,16*,18*,29	$3536^{a,b}$
$k=4$	4,9,17,19,23*	3534^b
	%E	99.9717

*unchanged tasks; ^{a}machine bottleneck;
^{b}changes to $t(S_k)$ of machine k;

The third column represents the accumulated task time $t(S_k)$ on the respective machine k. The first solution, designated as solution A, is evaluated with $\%E = 99.8024$.

Meanwhile, the second solution, designated as solution B, is obtained with a bottleneck indicator variable. The resulting solution B was based on the identified two bottleneck machines (labeled as 1 and 3). Because of the bottleneck indicator variable, solution B is evaluated with $\%E = 99.9717$ (0.1696 % improvement). Also, the bottleneck on machine 2 and machine 4 of solution A is mitigated as machine 1, and machine 3 became the new bottleneck machines. Although $t(S_1)$ and $t(S_2)$ did not change, most task assignments on all machines are changed to address the bottleneck machines, with machine 1 or 3 being the primary and secondary bottlenecks (similar $t(S_k)$ values). While machine 1 or 3 would be the target for improvement, further changes fail to improve the E and discarded. Nevertheless, the maximum $t(S_k)$ was reduced; thus, improving the $\%E$ of solution B.

D. COMPARATIVE ANALYSIS OF THE PROPOSED CAIN APPROACH AGAINST OTHER APPROACHES FROM THE LITERATURES

The proposed CAIN approach was compared against eight approaches identified from the literature only on the efficiency objective given by (1); MRMOSA [16], MOHIH [17], PriGA [18], WCH [19], RMCP [20], 2P-GA [21], MILP [22], and MA-GA [2]. Since each of these approaches was only tested over a portion of the ALB data sets, individually comparison was conducted where a paired one-tailed Wilcoxon Signed-Rank (WSR) tests[13] was used as the statistical analysis (Table 3). However, such analysis was not conducted on WCH due to result similarity. The null hypothesis is defined as $h_0 : \mu_{\text{CAIN}} \ngtr \mu_{\text{Others}}$, where μ_{Others} is the compared approaches. The alternative hypothesis (h_a) is assumed otherwise ($h_a : \mu_{\text{CAIN}} > \mu_{\text{Others}}$).

[13]In contrast to the paired t-test, the WSR test is a non-parametric statistical hypothesis test where two related samples, matched samples, or repeated measurements were compared to assess differences of their population mean ranks. The test does not require any particular distribution, robust against outliers, and heavy tail distributions [63].

It can be observed that the CAIN approach had outperformed other approaches over their respective instances of the SALB-E data sets, except for the WCH and RMCP approaches. The improvement achieved of the solved instances were at least ≈100% or more. Based on the p-value ($\alpha < 0.05$), the proposed CAIN approach had solved significantly more when compared against MRMOSA, MOHIH, PriGA, 2P-GA, and MA-GA statistically. However, CAIN performed similarly to WCH while under-performed compared to RMCP, due to superior modeling of the SALB-E problem. Nevertheless, WCH is applied to small test instances while RMCP is three times slower, albeit its superior performance. Also, there is not enough evidence for CAIN to be statistically significant against RMCP and MILP for $\alpha < 0.05$.

E. GENERAL DISCUSSION

The CAIN approach incorporates two-dimensional representation, which reduces the possible search space for a best known solution—inspired by the 3-dimensional representation by [54], the proposed two-dimensional representation is not limited to its original counterpart where the problem of this study is combinatorics in nature. The benefit of having multi-dimension representation involves promoting both exploration and exploitation of the improvisation operators [64], [65]. Also, problem representation is an essential criterion for any meta-heuristic approaches to effectively reduce the possible search space [64], where the possible search space had been reduced by half.

In another perspective, considering the knowledge-intensive approach at the algorithmic levels, specifically the bottleneck indicator matrices, provides active solution exploitation, where a better value of E was found by improving on some specific machine workload. An augmented matrix representation that approximates both current and shifting bottlenecks is achieved in a cost-effective measure. Hence, the complexity of the approach is reduced, and the best-known solutions were achieved effectively.

From a practical standpoint, the proposed approach provides the manager and planner a clear picture of the assembly line experience, where tasks can be simulated ahead of time, minimizing critical resource losses and avoiding the halting of the assembly line. This situation also serves to cushion the impact of bottlenecks before it occurs in the actual execution of the assembly line. In essence, having such visibility may become the determining factor for the manufacturing industry to have a competitive edge and being relevant in an information-driven era.

Intuitively, the observed performance of the CAIN approach, compared to other approaches, produces better results than MRMOSA without imposing explicit rules on the domain problem, although some domain rules adopted by MRMOSA played significant roles in improving its solution quality. Also, exploiting the information of current solutions, on top of rapid mutation and controlled solution initialization,

TABLE 3. Details and results of the compared approaches against the proposed CAIN approach.

Compared Approaches						CAIN		Improv.	p-value
Name	**Data (Inst)**	**Environment***	**TC**	**T**	**Dev.**	**T**	**Dev.**		($\alpha < 0.05$)
MRMOSA^{a}	10 (27)	[2, 2, N.A.]	t_c	N.A.	5.47	[1,20]	**2.81E-09**	-100.0000	**0.00657**
MOHIH^{b}	10 (27)	[1, 3, 0.512]	i = 10000	[1,150]	5.47	[1,150]	**2.81E-09**	-100.0000	**0.00657**
PriGA	7 (15)	[1, 3.2, 1]	i = 1000, t_c	N.A.	6.02	[3,231]	**-2.04E-04**	-100.0034	**0.00368**
WCH^{b,c}	7 (13)	N.A.	N.A.	[1,4]	0.00	[2,25]	0.00	N.A.	N.A.
RMCP^{d}	5 (18)	[2, 1.8, 3.5]	t = 2000	[1,243]	-1.14	[2,88]	-8.05E-02	92.9312	0.20045
2P-GA^{a}	18 (134)	[2, 2, N.A.]	i = 250, 500, 1000	[1,500]	1.35	[1,678]	**-2.08E-01**	-115.4055	**0.00368**
MA-GA^{e}	7 (15)	[1, 0.450, 0.256]	i = 1000, t_c	[30,240]	1.17	[3,550]	**-4.58E-04**	-100.0393	**< 0.00001**
MILPF^{d}	20 (157)	[1, 2.5, 4]	t = 1800	[1,20]	3.61E-03	[1,34]	**-1.48E-01**	-4189.1649	0.07078

Dev.: Total result deviation from best known solution (in %) Improv.: Total result deviation improvement TC: Termination condition(s)
T: Computational time (seconds) Inst: Data instances t_c: Constant performance for specific period/iteration i: Maximum iteration number
t: Maximum time (seconds) N.A.: Information not available *: [CPU Core Number, CPU Core Speed, Memory Size]
^{a}: MATLAB (https://www.mathworks.com/products/matlab.html) ^{b}: Visual Basic for Application ^{c}: LINGO (https://www.lindo.com/)
^{d}: IBM ILOG CPLEX Software ^{e}: Visual C programming

improved the CAIN approach's performance compared to the PriGA and MOHIH approach. Instead of relying on priority-rules and learning strategy, leveraging proper information exploits of the search space leads to a more prominent search mechanism.

Also, the CAIN approach was able to outperform high-quality solutions obtained from the bi-directional task assignment offered by the MA-GA approach. The solution encoding of CAIN had enabled fast evaluation while independent from the problem constraint. Compared to 2P-GA, the CAIN approach focused more on competition within the population throughout generations (or iterations), whereas 2P-GA focused more on competition between generations of the population. While this situation produces slightly higher computational costs, best known solutions have been achieved effectively.

Although some instances favor the proposed CAIN approach in terms of balancing efficiency, the differences were not statistically significant to either be practical or to outperform the compared approaches in some instances (e.g., MILP and RMCP approach). The MILP and RMCP approach introduce a non-conventional precedence graph utilizing a generalized cutting plane method with generated valid inequalities and rule-based models of the precedence relations, respectively. Such models address the task bounded model of the SALB-E problem and reduce computational time. Also, such models lead to an early advantage in solution exploitation.

Nevertheless, the CAIN approach ultimately leverages knowledge of a specific problem while its procedures being independent of the problem domain. This situation creates a balance between complete dependency and black-box technique while being robust to overcome the complexity of the domain problem. The comparative study conducted was also limited to the SALB problem instances acquired from the existing literature. Finally, the high-quality performance of the CAIN approach was achieved with the sacrifice on the overall computational time, which was a factor of about $1\frac{1}{2}$ times greater (or longer) compared to some of the approaches in the literature.

V. CONCLUDING REMARK

This study has proposed an AIS approach, namely the contagious artificial immune network (CAIN) approach, to solve a discrete combinatorial optimization of the SALB-E problem. A knowledge-intensive approach had been employed to achieve high-quality solutions while overcoming the complex challenge of the SALB-E problem. These issues have been solved by adopting a bone marrow model for rapid solution generation and two-dimensional solution encoding for fast evaluation and complexity reduction.

In general, the approach provided a new perspective in modeling the SALB-E problem. The emulated shifting bottlenecks are incorporated to provide insights on their adverse impacts on the efficiency of the assembly line performance and serve as the cushion on the reliability of assembly line resources. A high-quality solution was achieved by adopting a clonal competition mechanism (CCM) and contagious mutations. Incorporating the bottleneck indicator matrices also guides solution improvement and leads to discovering new solution candidates in some instances of the SALB-E data sets

Possible improvement of the solution was expected at least 100% or more than the conventional assembly line for small and medium-sized SALB problems. Illustrative examples and the efficacy of the CAIN approach was demonstrated through result improvements that were statistically significant at least in five out of eight approaches that were compared. Furthermore, from another perspective, simultaneous emphasis on the ALB problem and the bottleneck identification provides a synergistic effect on the production line where redistributing tasks of bottleneck machines tend to improve assembly line efficiency. Such a situation implies fewer disruptions in delivering timely market demands and potentially guiding better assembly resource management.

Based on the findings of the study, the potential future works are as follows:

- The CAIN approach can be applied to similar constraint and combinatorial optimization problems, particularly in allocation, loading, scheduling, planning, and sequencing. Such a condition can include an objective measure subjected to resource availability, allocation, product arrangement, and mixture considering its related constraints. Since only certain constraints with the assumption of sufficient resources available in the production line at all times (such as operators, subassemblies, types of equipment, fixtures, and pallets), modeling these resources would be worth venturing. Additionally, resource-related constraints (such as sequence-dependent setup time, operator skill level, and ergonomic restrictions), task-related information (such as fuzzy and stochastic task times), and multiple operation stages (such as inspections, testing facilities, and supply-chain-based production) could potentially be incorporated as well.

- The CAIN approach can incorporate data mining techniques (such as support vector machine, k-nearest neighbor) to improve, for instance, the selection operator, the hyper-mutation operator, and the suppression operator of the solution. Similarly, the bottleneck identification method could automate the potential bottlenecks' data extraction and classification (or clustering). Additionally, probabilistic machine learning techniques (e.g., Bayesian optimization) or general unsupervised learning (e.g., reinforcement learning) can be considered for solution initialization, improvisation operators, and solution maintenance.
- Integrating an intelligent search technique (such as uninformed or informed search techniques) with an embedded efficient and reliable heuristic evaluation function would potentially improve the overall performance of the CAIN approach, both space-wise and temporal-wise. In addition, adopting transfer learning and surrogate modeling could also enhance the CAIN approach towards a more generalized problem solver (or towards the frontier evolutionary computation called evolutionary transfer optimization) and significantly mitigate the computational complexity and requirement of the SALB-E problem with the bottleneck identification.

REFERENCES

[1] M. Bortolini, E. Ferrari, M. Gamberi, F. Pilati, and M. Faccio, ''Assembly system design in the industry 4.0 era: A general framework,'' *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 5700–5705, Jul. 2017.

[2] T. Al-Hawari, M. Ali, O. Al-Araidah, and A. Mumani, ''Development of a genetic algorithm for multi-objective assembly line balancing using multiple assignment approach,'' *Int. J. Adv. Manuf. Technol.*, vol. 77, nos. 5–8, pp. 1419–1432, Mar. 2015.

[3] O. Battaïa and A. Dolgui, ''A taxonomy of line balancing problems and their solution approaches,'' *Int. J. Prod. Econ.*, vol. 142, no. 2, pp. 259–277, 2013.

[4] D. R. Morrison, E. C. Sewell, and S. H. Jacobson, ''An application of the branch, bound, and remember algorithm to a new simple assembly line balancing dataset,'' *Eur. J. Oper. Res.*, vol. 236, no. 2, pp. 403–409, Jul. 2014.

[5] N. Hitomi and D. Selva, ''Incorporating expert knowledge into evolutionary algorithms with operators and constraints to design satellite systems,'' *Appl. Soft Comput.*, vol. 66, pp. 330–345, May 2018.

[6] M. J. Flores, A. E. Nicholson, A. Brunskill, K. B. Korb, and S. Mascaro, ''Incorporating expert knowledge when learning Bayesian network structure: A medical case study,'' *Artif. Intell. Med.*, vol. 53, no. 3, pp. 181–204, 2011.

[7] M. S. Mahbub, M. Wagner, and L. Crema, ''Incorporating domain knowledge into the optimization of energy systems,'' *Appl. Soft Comput.*, vol. 47, pp. 483–493, Oct. 2016.

[8] Q. Yin and X. Luo, ''A three-stage optimization method for assembly line balancing problem,'' *IEEE Access*, vol. 8, pp. 143607–143621, 2020.

[9] G. Jirasirilerd, R. Pitakaso, K. Sethanan, S. Kaewman, W. Sirirak, and M. Kosacka-Olejnik, ''Simple assembly line balancing problem type 2 by variable neighborhood strategy adaptive search: A case study garment industry,'' *J. Open Innov., Technol., Market, Complex.*, vol. 6, no. 1, p. 21, Mar. 2020.

[10] H. M. Dinh, D. V. Nguyen, L. V. Truong, T. P. Do, T. T. Phan, and N. D. Nguyen, ''Cycle time enhancement by simulated annealing for a practical assembly line balancing problem,'' *Informatica*, vol. 44, no. 2, pp. 127–138, Jun. 2020.

[11] O. Bongomin, J. I. Mwasiagi, E. O. Nganyi, and I. Nibikora, ''Improvement of garment assembly line efficiency using line balancing technique,'' *Eng. Rep.*, vol. 2, no. 4, Apr. 2020, e12157.

[12] R. Liu, M. Liu, F. Chu, F. Zheng, and C. Chu, ''Eco-friendly multi-skilled worker assignment and assembly line balancing problem,'' *Comput. Ind. Eng.*, vol. 151, Jan. 2021, Art. no. 106944.

[13] M. Salehi, H. R. Maleki, and S. Niroomand, ''Solving a new cost-oriented assembly line balancing problem by classical and hybrid meta-heuristic algorithms,'' *Neural Comput. Appl.*, vol. 32, no. 12, pp. 8217–8243, Jun. 2020.

[14] K. Meng, Q. Tang, Z. Zhang, and X. Qian, ''An improved lexicographical whale optimization algorithm for the type-II assembly line balancing problem considering preventive maintenance scenarios,'' *IEEE Access*, vol. 8, pp. 30421–30435, 2020.

[15] K. Meng, Q. Tang, Z. Zhang, and C. Yu, ''Solving multi-objective model of assembly line balancing considering preventive maintenance scenarios using heuristic and grey wolf optimizer algorithm,'' *Eng. Appl. Artif. Intell.*, vol. 100, Apr. 2021, Art. no. 104183.

[16] A. Baykasoglu, ''Multi-rule multi-objective simulated annealing algorithm for straight and U type assembly line balancing problems,'' *J. Intell. Manuf.*, vol. 17, no. 2, pp. 217–232, Apr. 2006.

[17] U. Özcan and B. Toklu, ''A new hybrid improvement heuristic approach to simple straight and U-type assembly line balancing problems,'' *J. Intell. Manuf.*, vol. 20, no. 1, pp. 123–136, Feb. 2009.

[18] R. Hwang and H. Katayama, ''Uniform workload assignments for assembly line by GA-based amelioration approach,'' *Int. J. Prod. Res.*, vol. 48, no. 7, pp. 1857–1871, 2010.

[19] N.-C. Wei and I.-M. Chao, ''A solution procedure for type e simple assembly line balancing problem,'' *Comput. Ind. Eng.*, vol. 61, no. 3, pp. 824–830, Oct. 2011.

[20] S. Topaloglu, L. Salum, and A. A. Supciller, ''Rule-based modeling and constraint programming based solution of the assembly line balancing problem,'' *Expert Syst. Appl.*, vol. 39, no. 3, pp. 3484–3493, Feb. 2012.

[21] P. T. Zacharia and A. C. Nearchou, ''A meta-heuristic algorithm for the fuzzy assembly line balancing type-E problem,'' *Comput. Oper. Res.*, vol. 40, no. 12, pp. 3033–3044, Dec. 2013.

[22] R. Esmaeilbeigi, B. Naderi, and P. Charkhgard, ''The type E simple assembly line balancing problem: A mixed integer linear programming formulation,'' *Comput. Oper. Res.*, vol. 64, pp. 168–177, Dec. 2015.

[23] M. E. Salveson, ''The assembly line balancing problem,'' *J. Ind. Eng.*, vol. 6, no. 3, pp. 18–25, 1955.

[24] A. Scholl, M. Fliedner, and N. Boysen, ''Absalom: Balancing assembly lines with assignment restrictions,'' *Eur. J. Oper. Res.*, vol. 200, no. 3, pp. 688–701, Feb. 2010.

[25] D. Battini, X. Delorme, A. Dolgui, and F. Sgarbossa, ''Assembly line balancing with ergonomics paradigms: Two alternative methods,'' *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 586–591, 2015.

[26] R. K. Hwang, H. Katayama, and M. Gen, ''U-shaped assembly line balancing problem with genetic algorithm,'' *Int. J. Prod. Res.*, vol. 46, no. 16, pp. 4637–4649, 2008.

[27] R. Pitakaso and K. Sethanan, ''Modified differential evolution algorithm for simple assembly line balancing with a limit on the number of machine types,'' *Eng. Optim.*, vol. 48, no. 2, pp. 1–19, 2015.

[28] A. Mozdgir, I. Mahdavi, I. S. Badeleh, and M. Solimanpur, ''Using the Taguchi method to optimize the differential evolution algorithm parameters for minimizing the workload smoothness index in simple assembly line balancing,'' *Math. Comput. Model.*, vol. 57, nos. 1–2, pp. 137–151, Jan. 2013.

[29] J. Sternatz, "Enhanced multi-Hoffmann heuristic for efficiently solving real-world assembly line balancing problems in automotive industry," *Eur. J. Oper. Res.*, vol. 235, no. 3, pp. 740–754, Jun. 2014.

[30] M. Subramaniyan, A. Skoogh, M. Gopalakrishnan, H. Salomonsson, A. Hanna, and D. Lämkull, "An algorithm for data-driven shifting bottleneck detection," *Cogent Eng.*, vol. 3, no. 1, Dec. 2016, Art. no. 1239516.

[31] M. Lemessi, S. Rehbein, G. Rehn, and T. Schulze, "Semi-automatic simulation-based bottleneck detection approach," in *Proc. Winter Simulation Conf. (WSC)*, Dec. 2012, p. 272.

[32] H. Tang, "A new method of bottleneck analysis for manufacturing systems," *Manuf. Lett.*, vol. 19, pp. 21–24, Jan. 2019.

[33] L. Li, Q. Chang, G. Xiao, and S. Ambani, "Throughput bottleneck prediction of manufacturing systems using time series analysis," *J. Manuf. Sci. Eng.*, vol. 133, no. 2, Apr. 2011, Art. no. 021015.

[34] M. A. Biazen and S. G. Gebeyehu, "Comparative analysis of analytical and discrete-event simulation models of assembly line systems," *J. Eng., Project, Prod. Manage.*, vol. 9, no. 2, pp. 132–141, Jul. 2019.

[35] C. Hofmann, T. Staehr, S. Cohen, N. Stricker, B. Haefner, and G. Lanza, "Augmented Go & See: An approach for improved bottleneck identification in production lines," *Procedia Manuf.*, vol. 31, pp. 148–154, Jan. 2019.

[36] W. Zhou, S. Li, Y. Huang, and J. Wang, "Simulation-based planning of a kind of complex product general assembly line," *Procedia CIRP*, vol. 76, pp. 25–30, Jan. 2018.

[37] A. H. Ng, J. Bernedixen, and L. Pehrsson, "What does multi-objective optimization have to do with bottleneck improvement of production systems?" in *Proc. 6th Int. Swedish Prod. Symp.*, Gothenburg, Sweden, Sep. 2014.

[38] B. W. Pearce, "A study on general assembly line balancing modeling methods and techniques," Ph.D. dissertation, Graduate School Clemson Univ., Clemson Univ., Clemson, SC, USA, 2015.

[39] H.-Y. Zhang, "An improved immune algorithm for simple assembly line balancing problem of type 1," *J. Algorithms Comput. Technol.*, vol. 11, no. 4, pp. 1–7, 2017.

[40] P. Sivasankaran and P. Shahabudeen, "Literature review of assembly line balancing problems," *Int. J. Adv. Manuf. Technol.*, vol. 73, nos. 9–12, pp. 1665–1694, Aug. 2014.

[41] N. Manavizadeh, M. Rabbani, D. Moshtaghi, and F. Jolai, "Mixed-model assembly line balancing in the make-to-order and stochastic environment using multi-objective evolutionary algorithms," *Expert Syst. Appl.*, vol. 39, no. 15, pp. 12026–12031, Nov. 2012.

[42] X. Ji, H. Ye, J. Zhou, Y. Yin, and X. Shen, "An improved teaching-learning-based optimization algorithm and its application to a combinatorial optimization problem in foundry industry," *Appl. Soft Comput.*, vol. 57, pp. 504–516, Aug. 2017.

[43] A. A. Juan, J. Faulin, S. E. Grasman, M. Rabe, and G. Figueira, "A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems," *Oper. Res. Perspect.*, vol. 2, pp. 62–72, Dec. 2015.

[44] A. Scholl and C. Becker, "State-of-the-art exact and heuristic solution procedures for simple assembly line balancing," *Eur. J. Oper. Res.*, vol. 168, no. 3, pp. 666–693, Feb. 2006.

[45] Y. Li, D. Wang, Y. Yu, and L. Jiao, "An improved artificial immune network algorithm for data clustering based on secondary competition selection," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2016, pp. 2744–2751.

[46] N. K. Jerne, "Towards a network theory of the immune system," *Annu. Immunol.*, vol. 125, no. 3, pp. 373–389, 1974.

[47] R. J. Kuo, S. S. Chen, W. C. Cheng, and C. Y. Tsai, "Integration of artificial immune network and K-means for cluster analysis," *Knowl. Inf. Syst.*, vol. 40, no. 3, pp. 541–557, Sep. 2014.

[48] D. Lizondo, S. Rodriguez, A. Will, V. Jimenez, and J. Gotay, "An artificial immune network for distributed demand-side management in smart grids," *Inf. Sci.*, vol. 438, pp. 32–45, Apr. 2018.

[49] M. Wang, S. Feng, J. Li, Z. Li, Y. Xue, and D. Guo, "Cloud model-based artificial immune network for complex optimization problem," *Comput. Intell. Neurosci.*, vol. 2017, pp. 1–17, May 2017.

[50] S. Tonegawa, "Somatic generation of antibody diversity," *Nature*, vol. 302, no. 5909, pp. 575–581, Apr. 1983.

[51] M. V. A. R. Bahubalendruni, B. B. V. L. Deepak, and B. B. Biswal, "An advanced immune based strategy to obtain an optimal feasible assembly sequence," *Assem. Autom.*, vol. 36, no. 2, pp. 127–137, Apr. 2016.

[52] J. Kelsey and J. Timmis, "Immune inspired somatic contiguous hypermutation for function optimisation," in *Proc. Genetic Evol. Comput. Conf. (GECCO)*. Berlin, Germany: Springer, 2003, p. 202.

[53] K. Faasse, B. Yeom, B. Parkes, J. Kearney, and K. J. Petrie, "The influence of social modeling, gender, and empathy on treatment side effects," *Ann. Behav. Med.*, vol. 52, no. 7, pp. 560–570, May 2018.

[54] K. Leung, F. Cheong, and C. Cheong, "Generating compact classifier systems using a simple artificial immune system," *IEEE Trans. Syst., Man, B, Cybern.*, vol. 37, no. 5, pp. 1344–1356, Oct. 2007.

[55] B. Christian and M. Daniel, *Swarm Intelligence: Introduction and Applications* (Natural Computing Series). Berlin, Germany: Springer-Verlag, 2008.

[56] A. Scholl, "Balancing and sequencing of assembly lines," Dept. Bus. Admin., Econ. Law, Inst. Bus. Stud., Darmstadt Tech. Univ., Darmstadt, Germany, Tech. Rep. 10881, 1999.

[57] A. Corominas, A. García-Villoria, and R. Pastor, "Improving the resolution of the simple assembly line balancing problem type e," *Statist. Oper. Res. Trans.*, vol. 1, no. 2, pp. 227–242, 2016.

[58] A. S. Muhamad and S. Deris, "An artificial immune system for solving production scheduling problems: A review," *Artif. Intell. Rev.*, vol. 39, no. 2, pp. 97–108, Feb. 2013.

[59] M. Cavazzuti, *Optimization Methods: From Theory to Design Scientific and Technological Aspects in Mechanics*. Heidelberg, Germany: Springer, 2012.

[60] D. C. Montgomery, *Design and Analysis of Experiments*. Hoboken, NJ, USA: Wiley, 2017.

[61] G. P. Coelho and F. J. Von Zuben, "A concentration-based artificial immune network for continuous optimization," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2010, pp. 1–8.

[62] G. P. Coelho, F. O. de França, and F. J. Von Zuben, "A concentration-based artificial immune network for combinatorial optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2011, pp. 1242–1249.

[63] S. Guo, S. Zhong, and A. Zhang, "Privacy-preserving Kruskal–Wallis test," *Comput. Methods Programs Biomed.*, vol. 112, no. 1, pp. 135–145, 2013.

[64] E.-G. Talbi, *Metaheuristics: From Design to Implementation*, vol. 74. Hoboken, NJ, USA: Wiley, 2009.

[65] S.-C. Neoh, N. Morad, C.-P. Lim, and Z. A. Aziz, "A layered-encoding cascade optimization approach to product-mix planning in high-mix–low-volume manufacturing," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 40, no. 1, pp. 133–146, Jan. 2010.

MOHD NOR AKMAL KHALID (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees from the University of Science, Malaysia, in 2013, 2015, and 2018, respectively. He is currently an Assistant Professor with the School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), and a part-time Research Fellow with the School of Computer Sciences, University of Science. He specializes in artificial intelligence techniques, such as evolutionary algorithms, decision support systems, and game studies, such as entertainment science and games informatics. His work focuses specifically on methods and developments in the fields of entertainment technology and operational research. His research interests include but are not limited to manufacturing systems, advanced scheduling and planning, artificial intelligence techniques, game analytic and informatics, search algorithms, bio-inspired optimization techniques, and machine learning methods.

UMI KALSOM YUSOF (Member, IEEE) received the B.Sc. degree from Western Illinois, Macomb, IL, USA, in 1986, the M.Sc. degree from Universiti Sains Malaysia (USM), Penang, Malaysia, in 2004, and the Ph.D. degree in computer science from Universiti Teknologi Malaysia (UTM), Johor, Malaysia, in 2013. She is currently an Associate Professor and a Senior Lecturer with the School of Computer Sciences, USM. She has previously worked at Petronas, Toyota, ASE Electronics, and Motorola before joining the academia, in 2008. She has published research articles in national and international journals, conference proceedings, as well as book chapters. Her research interests include artificial intelligence, machine learning, computational intelligence, multi-objective optimization, evolutionary computing, and web engineering, manufacturing optimization, crowd behavior in an emergency evacuation, and health-related and global warming effect studies.