Robust Artificial Immune System in the Hopfield network for Maximum *k*-Satisfiability

Mohd Asyraf Bin Mansor, Mohd Shareduwan Bin Mohd Kasihmuddin, and Saratha Sathasivam

School of Mathematical Sciences, Universiti Sains Malaysia

Abstract — **Artificial Immune System (AIS) algorithm is a novel and vibrant computational paradigm, enthused by the biological immune system. Over the last few years, the artificial immune system has been sprouting to solve numerous computational and combinatorial optimization problems. In this paper, we introduce the restricted MAX-*k*SAT as a constraint optimization problem that can be solved by a robust computational technique. Hence, we will implement the artificial immune system algorithm incorporated with the Hopfield neural network to solve the restricted MAX-*k*SAT problem. The proposed paradigm will be compared with the traditional method, Brute force search algorithm integrated with Hopfield neural network. The results demonstrate that the artificial immune system integrated with Hopfield network outperforms the conventional Hopfield network in solving restricted MAX-*k*SAT. All in all, the result has provided a concrete evidence of the effectiveness of our proposed paradigm to be applied in other constraint optimization problem. The work presented here has many profound implications for future studies to counter the variety of satisfiability problem.**

Keywords —**Artificial Immune System, Brute Force Algorithm, Hopfield Neural Network, Maximum *k*-satisfiability.**

I. Introduction

The astonishing power of the biological systems has been a core impetus for the researcher to enhance and create a computational paradigm [1]. One of the brand new bio-inspired metaheuristic techniques is the artificial immune system (AIS) that enthused from the robust vertebrate immune system [2]. The artificial immune system (AIS) has been transformed into a rapid growing heuristics method in parallel computation and optimization problem. In point of fact, most of AIS practitioners have dedicated on the learning and memory domains of the immune system in order to resemble it with the human's immune system [3]. The breakthrough work by Farmer *et al*. (1986) [4] was the eye-opener for the other researchers to venture and improve the artificial immune system algorithm. After a few years, we can deduce that the main advances within AIS have revolved on three important immunological forte namely, the immune clonal selection, immune networks, and negative selection [5]. Thus, it was proven as a notable searching technique by various researchers. The AIS algorithm has been applied to wide range problems such as global optimization [6], pattern recognition [7], multiple sequence alignment [8] and shop scheduling conundrum [9]. The related work by Layeb *et al*. [10] has demonstrated the robustness of the artificial immune system in tackling the general maximum satisfiability problem. Hence, we will improve the work by taking the advantage of clonal selection power in artificial intelligence together with the Hopfield network to solve maximum *k*-satisfiability problem.

The main motivation of this paper is to compare the effectiveness of the searching techniques incorporated with the Hopfield neural network in solving MAX-*k*SAT problem. Since a few decades ago, Boolean satisfiability has emerged from a classical problem into a bunch of various hard problems [11]. Hence, MAX-*k*SAT is a counterpart of classical Boolean satisfiability that has captured the attention of many researchers in the optimization field [12]. Specifically, the MAX-*k*SAT can be delineated as the maximum number of satisfied clauses achieved by any complete assignment [13]. The integration of artificial immune system and Hopfield neural network has been proven in solving some real life or industrial computational problems [43, 44]. The Hopfield neural network is vital due to its resemblance to the human biological brain [14]. Thus, the output we obtained after performing the searching technique will be stored as a content addressable memory (CAM). Thus, it is vital in identification or pattern recognition problem [45]. Hence, the Hopfield neural network will train the output systematically in order to drive towards the final output (global minima).

The idea of applying Hopfield neural network to hunt solution has been supported by numerous notable works [15, 16, 17]. Hopfield neural network comprises of a simple recurrent network that has an efficient associative memory and resembles the biological brain [14]. The important property of the Hopfield neural network is the minimization of energy whenever there is any change in inputs [15]. Moreover, the Hopfield neural network minimizes Lyapunov energy by utilizing the physical Ising spin of the neuron states. On top of that, the network produces global output by minimizing the network energy. Gadi Pinkas [16] and Wan Abdullah [17] described a bi-directional mapping between logic and energy function of symmetric neural network. Besides, both methods are the building blocks for a corresponding logic program. Due to effectiveness of energy changes in Hopfield neural network, several researchers have combined the idea of logic programming with Hopfield neural network. Several renowned models were developed by Sathasivam [18] and Wan Abdullah [17].

The main work of this paper is to propose a hybrid computational model by incorporating artificial immune system (AIS) and Hopfield neural network (HNN-MAX*k*SATAIS) in solving maximum *k*-satisfiability (MAX-*k*SAT) problem. Specifically, the work will focus on the capability of our proposed hybrid network in obtaining the maximum satisfied clauses in any MAX-*k*SAT formula. Therefore, the novelty can be found in the proposed hybrid technique since most of the researchers are only focusing on the standalone Hopfield neural network or metaheuristic to solve any maximum *k*-satisfiability problem. To frame the novelty, we will compare the HNN-MAX*k*SATAIS with conventional method, via integrating brute force algorithm with Hopfield neural network (HNN-MAX*k*SATBF).

The remaining frameworks of the paper are organized as follows. In Section II, the noteworthy concept of Boolean satisfiability (SAT) and maximum *k*-satisfiability (MAX-*k*SAT) problem are introduced. After that, we discuss in brief about the neuro-searching techniques namely Brute force (BF) search and artificial immune system (AIS) in Section III. Section IV discusses the Hopfield model, Wan Abdullah's

DOI: 10.9781/ijimai.2017.448

method and Sathasivam's relaxation method. Furthermore, we present the theory implementation of our proposed network in Section V. Moving on, Section VI encloses the experimental results for HNN-MAX*k*SATAIS and HNN-MAX*k*SATBF in terms of global minima ratio, ratio of satisfied clause, fitness energy landscape value and the computation time. As a final point, we conclude with some remarks in Section VII.

II. Maximum *k*-Satisfiability Problem

A. k-Satisfiability Problem

The *k*-SAT problem can be defined as a problem to determine the satisfiability of a particular Boolean formula containing *k* literals per clause [36]. *k*-SAT is generally expressed in terms of *k* – CNF (Conjunctive Normal Form) or Krom formula [37]. *k*-SAT has been a special NP problem that represents various optimization problems such as circuit and pattern recognition. Solutions of the *k*-SAT are corresponding to the solution of the various optimization problems. *k*-SAT consists of a set of *m* variables where $x_1, x_2,......,x_m$ with a set of literal or negation. These variables accommodate the *n* clauses $C_1, C_2........C_n$. The finest definition of *k*-SAT Boolean formula is as followed:

$$P = \overset{n}{\underset{i=1}{\wedge}} C_i \tag{1}$$

where $\wedge$ is a logical AND connector and *P* denotes the entire Boolean formula for *k*-SAT. C_i is a clausal form of DNF with *k* variables. Boolean value or truth value of the variable only consists of 1 or -1. These two values impersonate the information of True and False respectively. The goal of *k*-SAT problem is to decide whether there exists an assignment that makes *P* become satisfiable. In this paper, randomized 2-SAT and 3-SAT will be the benchmark to our satisfiability problem. Randomized 2-SAT and 3-SAT clauses are as followed:

$$C_i = \overset{k}{\underset{j=1}{\vee}} \left(x_{ij}, y_{ij}\right),\ k=2 \tag{2}$$

$$C_i = \overset{k}{\underset{j=1}{\vee}} \left(x_{ij}, y_{ij}, z_{ij}\right),\ k=3 \tag{3}$$

B. Restricted Maximum k-Satisfiability

Restricted maximum *k*-Satisfiability problem (MAX-*k*SAT) is a variant of Boolean SAT problem. Given a Boolean formula *P* in conjunctive normal form (CNF) with *n* clauses containing *k* number of variables per clause and positive integer g where $g \le n$. MAX-*k*SAT can be defined implicitly as a pair of (λ, θ) [10] where λ is the set of all possible solution $\{1,-1\}^n$ bit string and θ is a mapping $\lambda \rightarrow \xi$ which denotes the score of the assignments. ξ is scored based on correct clauses. Therefore, MAX-*k*SAT problem consist of defining the best bipolar/binary assignments to the variables in *P* that simultaneously satisfy at least *g* of the *n* clauses. The aim of the system is to decide the "optimized" assignment that can satisfy the maximum number of clauses containing *k* variables. It was proven that MAX-*k*SAT is NP-complete problem. There are numerous classifications of the MAX-*k*SAT namely, weighted MAXSAT [33] and Partial MAXSAT [34]. In our case, restricted value of *k* in MAX-*k*SAT only allowed $k=2$ and $k=3$. Restricted MAX-*k*SAT can be included as combinatorial problem that works in parallel with logic programming. Equation (4) is an example of MAX-*k*SAT formula:

$$P = (x \vee y) \wedge (x \vee \neg y) \wedge (\neg x \vee y) \wedge (\neg x \vee \neg y) \wedge (z \vee w) \tag{4}$$

Equation (4) consists of variables $x, y, z\ and\ w$ and formula *P* is impossible to satisfy since there are no assignments that make the above constraint become true. As the number of clauses increases, finding satisfying assignment will be terribly complex.

III. Neuro- Searching Paradigm

Neuro-searching techniques are staple in the computational method in order to hunt for the optimal solutions. The implementation of Hopfield neural network in logic programming has been successfully done by a few researchers. However, we require a solid neuro-searching paradigm to facilitate and foster the training process especially when the problem is getting complex. The standalone Hopfield network has limitations especially in terms of complexity [18]. The complexity will determine whether our models able to withstand more complex problems or not. Thus, we embedded the conventional brute force search as a heuristic method to the Hopfield neural network in solving MAX-*k*SAT problem (HNN-MAX*k*SAT). Conversely, we proposed the artificial immune system (AIS) algorithm as searching technique for the Hopfield neural network (HNN-MAX*k*SATAIS). In this paper, neuro-searching techniques were implemented in computing the maximum number of clauses for maximum 2-satisfiability (MAX-2SAT) and maximum 3-satisfiability problem (MAX-3SAT).

A. Brute Force (BF) algorithm

Brute-force (BF) algorithm can be delineated as a straight forward local search method for an element with a specific property between combinatorial domains including probabilities, permutations, combinations, logics, satisfiability or arrays of a set [23]. It is a conventional technique and the simplest algorithm, extensively used in pattern searching problem [24, 37]. The brute force searching paradigm is based on "generate and test" principle to enumerate all possible solutions in a search space. In our context, we will implement the brute force algorithm as a searching technique in the Hopfield network for solving maximum *k*-satisfiability problem (HNN-MAX*k*SATBF).

The brute force algorithm is a generalized problem-solving paradigm that usually enumerates all possible solutions and check whether the solutions satisfies the given formula [24]. It will return the satisfying assignment for the formula, if such exists. In essence, MAX*k*SAT formula with a specific number and combinations of clauses is addressed as a combinatorial optimization problem to be solved by our proposed technique. Furthermore, the brute force algorithm will explore any combinations of assignments and directly compute the total number of satisfied clauses. Technically, the brute force algorithm will allow our paradigm to hunt for the total satisfied clauses brutally, even in tremendous search dimension [25]. Specifically, the brute force search will evaluate the candidate solutions clause by clause in order to obtain the feasible solution. The feasible assignment will be stored in the Hopfield memory as content addressable memory (CAM) [30].

The brute force algorithm (exhaustive search) has been widely used by the researchers in solving numerous satisfiability problems, including the maximum satisfiability problem [23, 24]. Hence, it can be considered as the primitive searching tool for the standalone Hopfield neural network. Therefore, the brute force searching method is practically easy to implement by the researchers [25].

The underlying reason we venture this conventional method is to ascertain the degree of effectiveness of HNN-MAX*k*SATBF. Theoretically, the brute force algorithm devours more computation time in searching for the maximum satisfied assignment completely. Hence, the computation complexity is represented as $O(2^n)$. This hybrid network will encounter higher complexity as the program attempts large and more constrained clauses. Thus the computation time will be intensified and might end up in infinite loop. In this paper, we will generate random bit strings (represented the assignment of the MAX-*k*SAT logic) and straight away record the number of satisfied clauses. Hence, it is not guaranteed that the bit strings are not converging to global maxima during the first iteration of BF.

The brute force algorithm is given as follows:

Step 1

Generate the candidate bit strings.

Step 2

Test the candidate bit strings and compute the number of satisfied clauses.

Step 3

If

Return the assignment with the maximum number of satisfied clauses as an output.

Else

Repeat Step 1 and 2.

B. *Artificial Immune System (AIS)*

The artificial immune system is a population based paradigm where every individual represents a potential solution to the problem [6]. The artificial immune system is popularized by Framer *et al*. (1986) [4] by modelling the Jerne's Immune network theory. On top of that, the artificial immune system (AIS) algorithm can be illustrated as a distributed network and able to do parallel processing. We proposed the binary artificial immune system according to the immune clonal selection perspective. Technically, the binary artificial immune system has been implemented by several researchers for binary optimization and pattern recognition [6, 7].

Furthermore, the complex interactions between entities within each level will ensure the immune system to shield the body after any harmful entity and exogenous agent, known as antigen. A particular form of cell, identified as B-cells, is in charge for the destruction of the antigen. Hence, the B-cell produces antibodies that bind with the antigens and mark them for damage [3]. In addition, the power of the antibody or antigen binding is called antigenic affinity [8]. Robust features of that immune system have fortified their adaptation to information technology in solving numerous problems. In this paper, we only focused on clonal selection that will be implemented in our binary AIS.

The remarkable feature in our biological immune system is the capability to build antibodies to combat the new antigens or pathogens [1, 3]. Hence, the immune clonal selection process depicts the fundamental structures of an immune response towards an antigenic stimulus. By all means, it inaugurates the idea that only those cells can identify the antigen proliferate; thus, being nominated against those that do not [5]. Specifically, the B-cells will produce the antibodies if any incoming antigen is discovered. Then, the particular B-cells distinguish the antigen proliferate via cloning process. Significantly, the main event during clonal mutation is called somatic hypermutation, whereby the genetic maturation and variation can be improved [7, 9]. The B-cells with higher affinity will be differentiated into plasma and memory cell, whereby the worst one will be destroyed [35].

In our paper, we developed a hybrid paradigm by implementing the Hopfield network and binary AIS to do a MAX-*k*SAT logic programming HNN-MAX*k*SATAIS. In our exploration on binary AIS, the binary strings were illustrated as the B-cells. Thus, the bit string of 1 and -1 will be represented as "true" and "false" respectively.

The artificial immune system algorithm is given as follows:

Stage 1: Initialization

Generate and initialize the 100 B-cells (bit strings). Generally, any massive and diverse population of B-cells may represent a massive space search of solutions that can lead the program to global solutions. On contrary, a smaller population size can contribute to local minima solutions.

Stage 2: Affinity evaluation

Compute the affinity of every B-cells (each of candidate solutions). The affinity measure can be delineated as the total number of satisfied clauses in any particular MAX-*k*SAT formula. The affinity evaluation of the artificial immune system is given as follows:

$$aff = c_1(x) + c_2(x) + c_3(x) ..... + c_N(x) \tag{5}$$

where $c_1, c_2, c_3 ..... c_N$ are the number of clause checked by artificial immune system and *N* is the number of clauses present in the formula. Specifically, the role of the fitness function is to evaluate the candidate bit strings.

Stage 3: Selection

Select the best 5 B-cells with the highest affinity. The selected B-cells will stand the chance to perform the cloning process.

Stage 4: Cloning

Clone and duplicate the selected B-cells by implementing the roulette wheel mechanism [42]. Therefore, the newly produced B-cells population will comprise of 200 cloned B-cells. We need to consider the initial affinity (aff_i) and the total affinity of the population to check the number of possible clone. The β is the number of population clone that the program want to introduce to the search space. In our study, we set $\beta = 200$ to be punched into formula (6).

$$\begin{pmatrix} \text{The number of} \\ \text{clone allowed} \end{pmatrix} = \frac{aff_i}{\sum aff} \times \beta \tag{6}$$

Stage 5: Normalization

Normalize the B-cells $(aff\ N_i)$ via equation (7). Thus, the antibodies that exist in a memory response will achieve a higher average affinity than those of the initial primary response. It is called the maturation of the immune response process.

$$aff\ N_i = \frac{aff_i - \min aff}{\max aff - \min aff} \tag{7}$$

Stage 6: Somatic Hypermutation

The mutation process in AIS is vital in order to improve the quality of B-cells. The process is enriched by the "somatic" principle whereby the nearer the match, the more disruptive the mutation [6]. In order to avoid possible local maxima in terms of affinity (non-improving B-cell), the somatic hypermutation might be very useful. Mutation for each B-cell works by flipping the variable from 1 to -1 or vice versa. The flipping process will improve the B-cells (bit strings) to achieve the best affinity value.

$$\begin{pmatrix} \text{Number of} \\ \text{Mutation (Nb)} \end{pmatrix} = \left(\frac{1}{NN}\right)\left(aff\ N_i\right) + \left(1 - aff\ N_i\right)\left(0.01\right) \tag{8}$$

Thus, we can compute the affinity value for the newly formed B-cells (matured population). NN denotes the number of variables in one particular randomized MAX-*k*SAT. If the B-cells are able to reach the maximum affinity, the solution will exit the program. On contrary, if any of the B-cell did not manage to achieve maximum affinity, the program will reset the affinity of all B-cells and repeat stage 1 until 6.

In this paper, we hybridized AIS algorithm with the Hopfield neural network as a network based on logic programming to solve MAX-*k*SAT problems (HNN-MAX2SATAIS and HNN-MAX3SATAIS).

IV. Neuro- Logic In Hopfield Neural Network

A. The Hopfield Neural Network

One of the milestone for neural network was the associative model proposed by Hopfield at the beginning of 1980s. This model has been celebrated by many optimization problems such as computer network, pattern recognition and scheduling problem since this network focused the ensembles of computing unit.

Hopfield design consist of interconnected unit called neurons, forming a network. Computation in Hopfield network is executed by collections of activated neurons [15, 38]. Most of the literature suggest Hopfield network contains good properties including parallel execution for fast computation and exceptional stability [39]. In particular, we choose Hopfield network to do MAX-*k*SAT problem because the truth value of the problem can be well presented in Ising model, well distributed, integration of CAM [41], simple implementation and easy to synthesis with other algorithm.

The units in Hopfield nets are called binary threshold unit [30] which can only consider bipolar values of 1 and -1. The paramount definition for unit I's activation, a_i are given:

$$a_i = \begin{cases} 1 \quad if \sum_j W_{ij} S_j > \xi_i \\ -1 \; Otherwise \end{cases} \tag{9}$$

where W_{ij} is the synaptic weight from unit j to i. S_j is the state of unit j and ξ_i is the threshold of unit i. The network's architecture comprises of N recognized neurons, each was described by an Ising spin variable model. The connections in Hopfield net contain no connection with itself $W_{ii} = W_{jj} = 0$. This gave Hopfield the symmetrical features in terms of architecture [18]. Neuron state is basically bipolar $S_i \in \{1, -1\}$ and the spin points follow in the direction of magnetic field. This causes each neuron to flip until the equilibrium is reached. Thus, it follows the dynamics $S_i \rightarrow \mathrm{sgn}(h_i)$ where h_i is the local field of the neurons connection. When dealing with higher order neurons connection, the sum of the field induced by each neuron are as followed:

$$h_i = ... + \sum_j W_{ijk}^{(3)} S_j S_k + \sum_j W_{ij}^{(2)} S_j + W_i^{(1)} \tag{10}$$

Since the synaptic weight in Hopfield network is constantly symmetrical and does not contain zero diagonal, the updating rule maintains as follows [31]:

$$S_i(t+1) = \mathrm{sgn}\left[h_i(t)\right] \tag{11}$$

Equation (10) is necessary in order to ensure the energy decrease monotonically with the network. The generalized Lypunov energy equation that accommodates more neurons connections is as followed:

$$E = -\frac{1}{3}\sum_i \sum_j \sum_k W_{ijk}^{(3)} S_i S_j S_k - \frac{1}{2}\sum_i \sum_j W_{ij}^{(2)} S_i S_j - \sum_i W_i^{(1)} S_j \tag{12}$$

This energy function is significant because it establishes the degree of convergence of the network [41]. The energy value obtained from the equation will be verified as global or local minimum energy. For our case, the network will produce the correct solution when the induced neurons state reached global minimum energy. The process of obtaining global minimum energy is always associated with the correct model of synaptic weight. For MAX-*k*SAT, we implemented Wan Abdullah's updating technique to obtain the synaptic weight for our network [17].

B. Wan Abdullah's Method in Learning MAX-kSAT Clauses

MAX-*k*SAT can be treated as one of the constrained optimization problem that is being carried out on Hopfield neural network. Wan Abdullah's method became the pioneer in synaptic weight extraction based on logical inconsistencies [17]. Cost function that corresponds to MAX-*k*SAT clauses is the minimization of logical inconsistencies.

$$\min_{i \in (0,\infty),\, C_i = 1} \neg C_i \tag{13}$$

As the number of "wrong" assignment decreases, the number of satisfied MAX-*k*SAT clauses will increase. For example, consider the following randomized MAX-2SAT and MAX-3SAT problem with α and ϕ

$$\begin{aligned} \alpha = &(A \vee B) \wedge (A \vee \neg B) \wedge (\neg A \vee B) \wedge (\neg A \vee \neg B) \\ &\wedge (C \vee D) \end{aligned} \tag{14}$$

$$\begin{aligned} \phi = &(P \vee Q \vee R) \wedge (\neg P \vee Q \vee R) \wedge (P \vee \neg Q \vee R) \\ &\wedge (P \vee Q \vee \neg R) \wedge (\neg P \vee \neg Q \vee R) \wedge (\neg P \vee Q \vee \neg R) \\ &\wedge (P \vee \neg Q \vee \neg R) \wedge (\neg P \vee \neg Q \vee \neg R) \wedge (S \vee T \vee U) \end{aligned} \tag{15}$$

Cost function $f_{\mathbf{cost}}$ for both equations (14) and (15) are as followed:

$$\begin{aligned} f_{\text{cost }\alpha} = &\frac{1}{2}(1-S_A)\frac{1}{2}(1-S_B) + \frac{1}{2}(1-S_A) + \frac{1}{2}(1+S_B) \\ &+ \frac{1}{2}(1+S_A)\frac{1}{2}(1-S_B) + \frac{1}{2}(1+S_A)\frac{1}{2}(1+S_B) \\ &+ \frac{1}{2}(1-S_C)\frac{1}{2}(1-S_D) \end{aligned} \tag{16}$$

$$\begin{aligned} f_{\text{cost}\phi} = &\frac{1}{2}(1-S_P)\frac{1}{2}(1-S_Q)\frac{1}{2}(1-S_R) + \frac{1}{2}(1+S_P)\frac{1}{2}(1-S_Q)\frac{1}{2}(1-S_R) \\ &+ \frac{1}{2}(1-S_P)\frac{1}{2}(1+S_Q)\frac{1}{2}(1-S_R) + \frac{1}{2}(1-S_P)\frac{1}{2}(1-S_Q)\frac{1}{2}(1+S_R) \\ &+ \frac{1}{2}(1+S_P)\frac{1}{2}(1+S_Q)\frac{1}{2}(1-S_R) + \frac{1}{2}(1+S_P)\frac{1}{2}(1-S_Q)\frac{1}{2}(1+S_R) \\ &+ \frac{1}{2}(1-S_P)\frac{1}{2}(1+S_Q)\frac{1}{2}(1+S_R) + \frac{1}{2}(1+S_P)\frac{1}{2}(1+S_Q)\frac{1}{2}(1+S_R) \\ &+ \frac{1}{2}(1-S_T)\frac{1}{2}(1-S_U)\frac{1}{2}(1-S_V) \end{aligned} \tag{17}$$

By comparing equation (16), (17) with equation (12), we obtained synaptic weight for α and ϕ. The synaptic weights are shown in Table 1 and Table 2.

TABLE I

Synaptic Weight for α based on Wan Abdullah's Method

W	C_1	C_2	C_3	C_4	C_5
$W_A^{(1)}$	$\frac{1}{4}$	$\frac{1}{4}$	$-\frac{1}{4}$	$-\frac{1}{4}$	0
$W_B^{(1)}$	$\frac{1}{4}$	$-\frac{1}{4}$	$\frac{1}{4}$	$-\frac{1}{4}$	0
$W_{AB}^{(2)}$	$-\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{2}$	0
$W_C^{(1)}$	0	0	0	0	$\frac{1}{4}$
$W_D^{(1)}$	0	0	0	0	$\frac{1}{4}$
$W_{CD}^{(1)}$	0	0	0	0	$-\frac{1}{2}$

TABLE II

Synaptic Weight for ϕ Based on Wan Abdullah's Method

W	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
$W_P^{(1)}$	$\frac{1}{8}$	$-\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$-\frac{1}{8}$	$-\frac{1}{8}$	$\frac{1}{8}$	$-\frac{1}{8}$	0
$W_Q^{(1)}$	$\frac{1}{8}$	$\frac{1}{8}$	$-\frac{1}{8}$	$\frac{1}{8}$	$-\frac{1}{8}$	$\frac{1}{8}$	$-\frac{1}{8}$	$-\frac{1}{8}$	0
$W_R^{(1)}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$-\frac{1}{8}$	$\frac{1}{8}$	$-\frac{1}{8}$	$-\frac{1}{8}$	$-\frac{1}{8}$	0
$W_{PQ}^{(2)}$	$-\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$-\frac{1}{8}$	$-\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$-\frac{1}{8}$	0
$W_{QR}^{(2)}$	$-\frac{1}{8}$	$-\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$-\frac{1}{8}$	$-\frac{1}{8}$	0
$W_{PR}^{(2)}$	$-\frac{1}{8}$	$\frac{1}{8}$	$-\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$-\frac{1}{8}$	$\frac{1}{8}$	$-\frac{1}{8}$	0
$W_{PQR}^{(3)}$	$\frac{1}{16}$	$-\frac{1}{16}$	$-\frac{1}{16}$	$-\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$-\frac{1}{16}$	0
$W_S^{(1)}$	0	0	0	0	0	0	0	0	$\frac{1}{8}$
$W_T^{(1)}$	0	0	0	0	0	0	0	0	$\frac{1}{8}$
$W_U^{(1)}$	0	0	0	0	0	0	0	0	$\frac{1}{8}$
$W_{ST}^{(2)}$	0	0	0	0	0	0	0	0	$-\frac{1}{8}$
$W_{TU}^{(2)}$	0	0	0	0	0	0	0	0	$-\frac{1}{8}$
$W_{SU}^{(2)}$	0	0	0	0	0	0	0	0	$-\frac{1}{8}$
$W_{STU}^{(3)}$	0	0	0	0	0	0	0	0	$\frac{1}{16}$

Hebbian learning is another possible alternative to imprint the adequate synaptic weight for neurons connections. Sathasivam has shown that the synaptic weight obtained by using Wan Abdullah's method are similar due to clausal MAX-*k*SAT similarity. Although both method is expected to produce the similar synaptic weight information, Wan Abdullah's method is proven to minimize the spurious minima produced by clauses compared to Hebbian learning [16, 17]. Spurious minima can lead to local minima solutions.

C. Network Relaxation

The quality of the solutions obtained by Hopfield network can be affected by multiple factors. Prolonged firing and receiving information among neurons can reduce the quality of the solutions. As the number of neurons increased, more interconnected neurons involved in firing and receiving information. In this case, network relaxation helps the neurons to "pause" before continuing the information exchange. "Network relaxation" is a series of relaxation loop in the program after local field has been obtained. Without proper relaxation mechanism, network tends to produce more local minima solution. Since MAX-*k*SAT contains more clausal constrained, we applied Sathasivam's relaxation technique [18] to ensure the network relaxed to its final state. Information exchange between neurons will be updated based on the following equation:

$$\frac{dh_i^{new}}{dt} = R\frac{dh_i}{dt} \tag{18}$$

Where R denotes the relaxation rate and h_i refers to the local field of the network as listed in equation (10). For our case, constant relaxation $R = 2$ will be used in our program.

V. Implementation of Neuro-Heuristic Method

The simulations for HNN-MAX*k*SATAIS and HNN-MAX*k*SATBF were executed on Microsoft Visual C++ 2013 for Windows 10. The main task of the program is to find "models" that can solve MAX-*k*SAT problem. The following algorithms shows how we can implement HNN-MAX*k*SATAIS and HNN-MAX*k*SATBF.

a) Translate MAX-*k*SAT clauses into Boolean algebra (if any).
b) Assign neuron to each variable in MAX-*k*SAT clauses
c) Initialize all synaptic weight to zero.
d) Derive MAX-*k*SAT cost function by assigning $X = \frac{1}{2}(1+S_X)$ and $\overline{X} = \frac{1}{2}(1-S_X)$. The state of the neuron reads true when $S_X = 1$ and reads false when $S_X = -1$
e) By using Wan Abdullah's method, compare the derived cost function with equation (16) or (17).
f) Check clauses satisfaction by using both BF and AIS. Each satisfied clause will be stored to the network.
g) Randomize the state of the neurons.
h) Compute the corresponding local field by using equation (10). Find the final state and run it for 5 times. If the state remains unchanged, we considered it as stable state.
i) Apply Sathasivam's relaxation method to the network via equation (14).
j) Find the corresponding final energy by using equation (12). Verify whether final energy is a local minimum energy of global minimum energy.
k) Find the corresponding global minima ratio, ratio of satisfied clauses, fitness energy landscape value and computation time.

Each simulation runs 100 trials with 100 combinations of neurons in order to reduce statistical error. According to Sathasivam [18], 0.001 is chosen as tolerance value since it gives us a better filtering mechanism to separate global or local minima solution.

VI. Result and Discussion

The performance of simulated program model for both HNN-MAX*k*SATBF and HNN-MAX*k*SATAIS will be tested by a few tests. Both models will be compared in terms of global minima ratio, ratio of satisfied clauses, fitness energy landscape value and the computation time. In order to make a fair comparison for all tests, all simulations will be performed in the same device with the similar processor.

A. Global Minima Ratio

Global minima ratio is elucidated as the ratio between the total global minimum energy over total number of runs [18]. Since the program pursues the global minimum energy for every neuron state in MAX-*k*SAT, it will be wise enough to indicate the simulation by checking it's the ratio of global minima. In our case, each simulation will produce 10000 bit strings each consist of final state of the neurons. 0.9500 global minima ratio value indicates 9500 bit strings solutions are global minimum and 500 bit strings are local minimum.

TABLE III

Global Minima Ratio

NN	HNN-MAX2SATBF	HNN-MAX2SATAIS	HNN-MAX3SATBF	HNN-MAX3SATAIS
10	0.9724	0.9999	0.9702	0.9993
20	0.9645	0.9904	0.9549	0.9822
30	0.9487	0.9767	0.9300	0.9606
40	0.9133	0.9632	0.8801	0.9399
50	0.8740	0.9408	0.8554	0.9230
60	0.8346	0.9225	0.8246	0.9117
70	-	0.9123	-	0.9045

Table III depicts the global minima ratio obtained by HNN-MAX*k*SATAIS and HNN-MAX*k*SATBF respectively. Hence, the capability of both hybrid networks can be measured by taking into consideration their global minima ratio for different variants of complexity. Sathasivam [18] explained the correlation between the global minima ratio with the nature of energy achieved at the end of the computation process. Theoretically, if the global minima ratio of the proposed hybrid network is close to one, almost all solutions in the network reached global minimum energy (global solution). According to Table 3, HNN-*k*MAXAIS has the capability to retrieve more accurate state compared to MAX-*k*SATBF. This is due to the effectiveness in the neuro-searching technique employed by the HNN-*k*MAXAIS. The ability of the B-cells in fighting the pathogens and improving affinity of the bit string in artificial immune system algorithm helps the program to search the solution optimally compared to the traditional brute force search algorithm. Hence, more solutions had successfully achieved the global minimum energy compared to the local minima (non-improving solution). Somatic hypermutation helps the program to avoid local maxima in terms of affinity (non improving affinity). Variation of mutation rate for each group of B-cells helps the program to spread the potential search space. This property causes AIS algorithm to find the correct solution and converge to global minima compared to traditional BF.

On the other hand, the limit for HNN-MAX-*k*SATBF is 60 neurons. After 60 neurons, the network in HNN-MAX-*k*SATBF trapped in trial and error state and consume more time to find the solution. On contrary, HNN-*k*MAXAIS is able to withstand number of neurons up to 70 neurons. The artificial immune system has been proven in reducing the complexity of the network. As the number of neurons increased, the complexity of the network increased, since the size of the constraint will enlarge indefinitely. Besides, AIS algorithm was able to sort the possible candidate solution (B-cells) effectively and can cope with more constraints compared to BF algorithm. The problem with HNN-MAX*k*SATBF was the nature of the brute-force search that deployed an intensive training process in hunting the correct neuron states. Therefore, the updating rule for HNN-MAX*k*SAT-BF will generate additional abrupt energy surfaces and more solutions are not improving (local minima). Based on the obtained global minima ratio, HNN-MAX*k*SATAIS has outperformed the HNN-MAX*k*SATBF in generating the global solutions.

B. Ratio of Satisfied Clauses

As we deal with the MAX-*k*SAT problem, we are required to compute the possible satisfied clauses for each of the problems. Ratio of satisfied clauses can be defined as the total number of satisfied clauses over the total number of clauses [13].

TABLE IV

Ratio of Satisfied Clause

NN	HNN-MAX2SATBF	HNN-MAX2SATAIS	HNN-MAX3SATBF	HNN-MAX3SATAIS
10	0.725	0.750	0.805	0.875
20	0.708	0.750	0.796	0.870
30	0.700	0.748	0.784	0.852
40	0.694	0.742	0.753	0.838
50	0.680	0.735	0.718	0.825
60	0.643	0.731	0.696	0.800
70	-	0.726	-	0.788

NN=Number of neurons.

Table IV portrays the ratio of the satisfied clauses over total clause obtained HNN-MAX2SATAIS, HNN-MAX2SATBF and HNN-MAX3SATAIS, HNN-MAX3SATBF. In maximum satisfiability problem, MAX-2SAT and MAX-3SAT clauses will never be fully satisfied. Hence, the searching method will be able to return the maximum number of satisfied clauses as the output. We can further deduce that, the higher the ratio obtained, the more clauses will be satisfied in any MAX-*k*SAT problem. According to Table 4, HNN-MAX*k*SATAIS is proven to obtain more satisfied clauses in MAX-*k*SAT compared to conventional Brute force method. As the number of neurons increased, the HNN-MAX*k*SATAIS is still able to maintain the quality of the ratio. The ability of B-cells in artificial immune system algorithm to perform hyper mutation will drive into optimal solutions. Thus, it will be able to cope the higher number of neurons. Conversely, HNN-MAX*k*SATBF will produce a slightly lower ratio of satisfied clauses since most of the solutions obtained trapped at local minima. The generate and test procedure in the brute force search will cause the bit string to reach local maxima easily. This is the higher energy barrier needed during searching the optimum assignment. Thus, high stability in HNN-MAX*k*SATAIS will reduce the spurious minima which will cause the retrieved solutions become local minima solutions.

C. Fitness Energy Landscape Value

Basically, the effectiveness of a paradigm can be measured by taking into consideration the fitness energy landscape value. Hence, the fitness energy landscape value is linked with the pattern or input storing competency. The ruggedness of the energy model depicts the performance of each of our algorithms. Thus, the fitness energy landscape value is computed by using Kauffman's model [29]. Fig. 1 and 2 represent the obtained fitness energy landscape values for HNN-MAX*k*SATAIS and HNN-MAX*k*SATBF respectively.

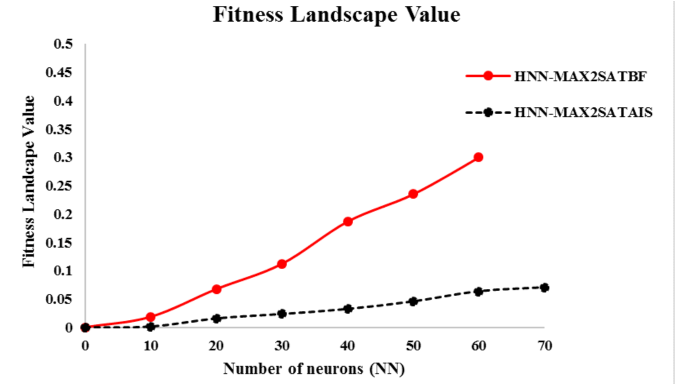

Fig. 1. Fitness energy landscape value for HNN-MAX2SATBF and HNN-MAX2SATAIS.

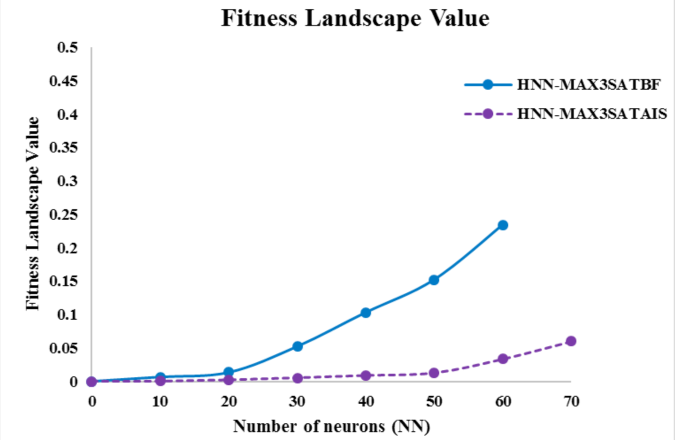

Fig. 2. Fitness energy landscape value for HNN-MAX3SATBF and HNN-MAX3SATAIS.

As shown in the Fig. 1 and Fig. 2, the difference in energy for HNN-MAX*k*SATAIS is very nearly flat (zero) compared to HNN-MAX*k*SATBF. Yet another observation that can be made from the graph is the capability of the artificial immune system algorithm to optimize the energy required throughout the exploration compared to the brute force search algorithm. Also, it was observed that the fitness energy landscape for HNN-MAX*k*SATBF can be observed as more rugged than the HNN-MAX*k*SATAIS. One thing to ponder is the MAX-*k*SAT clauses always allied to the ruggedness of the energy landscape. The more rugged the energy landscape, the harder it will be to achieve feasible solution [21]. Since the complexity of the solution searching has been reduce drastically by AIS in HNN-MAX*k*SATAIS, additional relaxation time was added before the network retrieve the final states. Consequently, HNN-MAX*k*SATAIS has a remarkable ability to store MAX*k*SAT pattern compared to HNN-MAX*k*SATBF. As a result, more global minimum energy produced that will drive to more feasible solutions. Thus, we are able to generate and retrieve the maximum satisfied clauses accurately. Other than that, HNN-MAX*k*SATBF was able to withstand up to 60 neurons due to the complexity.

D. Computation Time

The robustness of our algorithms can be approximately proved by looking at the effectiveness of the entire computation process. In our case, the computation time refers to the expanse of time for which our network was used to complete the whole computation process [31]. Thus, the computation process involves the training and generating the maximum satisfied clauses by our proposed paradigms. In this study, we measure the computation time by using the standard unit of second for the simplicity.

TABLE V
Computation Time

NN	HNN-MAX2SATBF	HNN-MAX2SATAIS	HNN-MAX3SATBF	HNN-MAX3SATAIS
10	20	1	29	2
20	95	11	126	19
30	316	53	440	64
40	2645	242	3003	303
50	10835	498	12657	685
60	86102	1325	98368	1502
70	-	3049	-	3166

NN=Number of neurons.

Table V depicts the overall computation time recorded for our proposed algorithms, HNN-MAX*k*SATAIS and the conventional algorithm HNN-MAX*k*SATBF. Since we deal with MAX-2SAT and MAX-3SAT clauses, the training process consumes more time to minimize the logical inconsistencies than the randomized *k*-SAT problem. For instance, as the number of neurons increased, the computation time to generate the maximum number of clauses also increased. This is due to the fact that maximum *k*-satisfiability problem will never be fully satisfied, but we can possibly calculate the maximum number of clauses that will be satisfied. Hence, the states retrieved from the network can improve the global solutions that maximize the number of satisfied clauses. Thus, the whole process incurs more computation time. Generally, MAX-3SAT requires more time than MAX-2SAT due to complexity as the number of literals entrenched in the formula also higher.

According to Table V, HNN-MAX2SATAIS and HNN-MAX3SATAIS require less computation time compared to the other counterparts, HNN-MAX2SATBF and HNN-MAX3SATBF. The undoubted evidence beyond that results are due to more neurons being forced to jump the energy barrier to relax into global solutions during the training process [38]. Additionally, the training process by using brute force requires more computational time due to "generate and test" process in hunting the maximum number of satisfied assignments. On the contrary, when we applied AIS algorithms, the CPU time was reported faster due to the efficiency of the B-cells to improve towards the desired solution. Before the network was fed by the solution, B-cells with high and low affinity will be considered in finding the best B-cells [7]. Hence, HNN-MAX*k*SATAIS experienced less computation burden during the training processes as compared to HNN-MAX*k*SATBF.

VII. Conclusion

A robust paradigm is developed by taking the advantages of a brand new immune inspired heuristic method, known as artificial immune system algorithm and the power of Hopfield network to solve MAX-*k*SAT problem. Hence, the performance comparison was made between our proposed paradigm with the conventional brute force method integrated with Hopfield neural network. In comparison of the algorithms, the HNN-MAX*k*SATAIS has shown more robustness to produce the global output and feasible solution compared to HNN-MAX*k*SATBF. The exploration showed solid performances of HNN-MAX*k*SATAIS in terms of the global minima ratio, ratio of satisfied clause, fitness landscape value and the computation time. In our future work, we would like to introduce our proposed algorithm to solve the other variants of satisfiability problems such as Weighted maximum satisfiability, quantified maximum satisfiability and unique satisfiability problem.

Acknowledgment

The authors fully acknowledged Ministry of Higher Education (MOHE) and School of Mathematical Sciences, Universiti Sains Malaysia for the support which makes this important research viable and effective.

References

[1] R. B. Xiao and L. Wang, "Artificial immune system: principle, models, analysis and perspectives", *CHINESE JOURNAL OF COMPUTERS-CHINESE EDITION*, vol. 25, no. 12, pp. 1281-1293, 2002.

[2] E. Hart and J. Timmis, "Application areas of AIS: The past, the present and the future", *Applied soft computing*, vol. 8, no. 1, pp. 191-201, 2008.

[3] D. Dasgupta, Z. Ji and F.A González, "Artificial immune system (AIS) research in the last five years" in *IEEE Congress on Evolutionary Computation,* vol. 1, pp. 123-130, 2003.

[4] J. D. Farmer, N.H. Packard and A. S. Perelson, "The immune system, adaptation, and machine learning", *Physica D: Nonlinear Phenomena*, vol. *22*, no. 1, pp. 187-204, 1986.

[5] J. Timmis and M. Neal, "A resource limited artificial immune system for data analysis", *Knowledge-Based Systems*, vol. 14, no. 3, pp. 121-130, 2001.

[6] N. Cruz-Cortés, D. Trejo-Pérez and C. A. C. Coello, "Handling constraints in global optimization using an artificial immune system", in *International Conference on Artificial Immune Systems*, pp. 234-247, 2005.

[7] L. N. de Castro, and J. Timmis, "Artificial immune systems: a novel paradigm to pattern recognition", *Artificial Neural networks in pattern Recognition*, no. 1, pp.67-84, 2002.

[8] A. Layeb and A. H. Deneche, "Multiple sequence alignment by immune artificial system", in 2007 IEEE/ACS International Conference on Computer Systems and Applications, pp. 336-342, 2007, May.

[9] O. Engin and A. Döyen, "A new approach to solve hybrid flow shop scheduling problems by artificial immune system", *Future generation computer systems*, vol. 20, no. 6, pp.1083-1095, 2004.

[10] A. Layeb, A. H. Deneche and S. Meshoul, "A new artificial immune system for solving the maximum satisfiability problem" in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pp. 136-142, 2010, June.

[11] P. Hansen and B. Jaumard, 1990. "Algorithms for the maximum satisfiability problem". *Computing*, *44*(4), pp.279-303.

[12] P. Crescenzi, and V. Kann, 1997, July. "Approximation on the web: A compendium of NP optimization problems". In *International Workshop on Randomization and Approximation Techniques in Computer Science* (pp. 111-118). Springer Berlin Heidelberg.

[13] A. Z. Broder, A. M. Frieze and E. Upfal, "On the satisfiability and maximum satisfiability of random 3-CNF formulas", in *SODA*, vol. 93, pp. 322-330, 1993, January.

[14] R. Rojas, Neural Networks: A Systematic Introduction. Berlin: Springer, 1996.

[15] J.J. Hopfield, D. W. Tank, "Neural computation of decisions in optimization problem", Biological Cybernatics, vol. 52, 141-152, 1985.

[16] G. Pinkas, "Symmetric neural networks and propositional logic satisfiability", *Neural Computation*, vol. 3, no. 2, pp.282-291, 1991.

[17] W. A. T. W. Abdullah, "Logic Programming on a Neural Network", *Malaysian Journal of Computer Science*, vol. 9, no. 1, 1-5, 1993.

[18] S. Sathasivam, "Upgrading Logic Programming in Hopfield Network, Sains Malaysiana", vol. 39, 115-118, 2010.

[19] W. A. T. W. Abdullah, "The logic of neural networks". *Physics Letters A*, vol. 176, no. 3, pp. 202-206, 1993.

[20] J. R. M. Fernández, J. D. L. C. B Vidal, "Improved Shape Parameter Estimation in K Clutter with Neural Networks and Deep Learning", *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 3, 2016.

[21] S. Sathasivam, W. A. T. W. Abdullah, "Flatness of the Energy Landscape for Horn Clauses", *SSJ*, vol 1, no. 2, pp. 2, 2008.

[22] U. Aiman, and N. Asrar, "Genetic Algorithm Based Solution to SAT-3 Problem", *Journal of Computer Sciences and Applications*, vol. 3, no. 2, pp. 33-39, 2015.

[23] D. Verma, N. Kakkar & N. Mehan, "Comparison of Brute-Force and KD Tree Algorithm", *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 3, no. 1, pp. 5291-5294, 2014.

[24] R. A. Abdeen, "An Algorithm for String Searching Based on Brute-Force Algorithm", *International Journal of Computer Science and Network Security*, vol. 11, no. 7, pp. 24-27, 2011.

[25] I. Zinovik, D. Kroening, and Y. Chebiryak, "Computing binary combinatorial gray codes via exhaustive search with SAT solvers", *Information Theory, IEEE Transactions,* vol. 54, no. 4, pp. 1819-1823, 2008.

[26] K. Haddouch, K. Elmoutaoukil, and M. Ettaouil, "Solving the Weighted Constraint Satisfaction Problems Via the Neural Network Approach", *International Journal of Interactive Multimedia and Artificial Inteligence*, *4* (Special Issue on Artificial Intelligence Underpinning), 2016.

[27] B. Borchers and J. Furman, "A two-phase exact algorithm for MAX-SAT and weighted MAX-SAT problems", *Journal of Combinatorial Optimization*, vol. 2, no. 4, pp. 299-306, 1998.

[28] R. A. Kowalski, "The early years of logic programming", Communications of the ACM, vol. 31, no. 1, pp. 38-43, 1988.

[29] A. Imada and K. Araki, "What does the landscape of a Hopfield associative memory look like?" in *Evolutionary Programming VII*, Springer Berlin Heidelberg, pp. 647-65, 1998.

[30] S. Haykin, *Neural Networks: A Comprehensive Foundation*, New York: Macmillan College Publishing, 1999.

[31] M. Velavan, Z. R. Yahya, M. N. A. Halif, & S. Sathasivam, "Mean Field Theory in Doing Logic Programming Using Hopfield Network", *Modern Applied Science*, vol. 10, no. 1, p. 154, 2015.

[32] E. Elbeltagi, T. Hegazy and D. Grierson, Comparison among five evolutionary-based optimization algorithms. *Advanced engineering informatics*, vol. 19, no. 1, pp. 43-53, 2005

[33] B. A. Madsen and P. Rossmanith, "Maximum exact satisfiability: NP-completeness proofs and exact algorithms", BRICS Report Series, vol. 11, no.19, 2004.

[34] B. Borchers and J. Furman, "A two-phase exact algorithm for MAX-SAT and weighted MAX-SAT problems", *Journal of Combinatorial Optimizatio*n, vol. 2, no. 4, pp. 299-306, 1998.

[35] A. Baldominos Gómez, N. Luis Mingueza, and M. C. García del Pozo, "OpinAIS: An Artificial Immune System-based Framework for Opinion Mining", *International Journal of Interactive Multimedia and Artificial Inteligence, 2015.*

[36] V. Raman, B. Ravikumar and S. S. Rao, "A simplified NP-complete MAXSAT problem", *Information Processing Letters*, vol. 65, no. 1, pp. 1-6, 1998.

[37] L. Bordeaux, Y. Hamadi, and L. Zhang, "Propositional satisfiability and constraint programming: A comparative survey", *ACM Computing Surveys (CSUR)*, vol. 38, no. 4, p.12, 2006.

[38] S. Sathasivam, P.F. Ng, N. Hamadneh, "Developing agent based modelling for reverse analysis method", *Research Journal of Applied Sciences, Engineering and Technology*, vol. 6, no. 22, pp. 4281-4288, 2013.

[39] X. G. Ming and K. L. Mak. "A hybrid Hopfield network-genetic algorithm approach to optimal process plan selection." *International Journal of Production Research* 38, no. 8, pp. 1823-1839, 2000.

[40] U. P. Wen, K. M. Lan & H. S. Shih, "A review of Hopfield neural networks for solving mathematical programming problems", *European Journal of Operational Research*, vol. 198, no. 3, pp. 675-687, 2009.

[41] L. M. Ionescu, A. G. Mazare, & G. Serban, "VLSI Implementation of an associative addressable memory based on Hopfield network model", *IEEE Semiconductor Conference*, vol. 2, pp. 499-502, 2010.

[42] D. E. Goldberg, and K. Deb, "A comparative analysis of selection schemes used in genetic algorithms.", *Foundations of genetic algorithms*, vol. 1, pp. 69-93, 1991.

[43] V. B. Semwal, K. Mondal, and G. C. Nandi, "Robust and accurate feature selection for humanoid push recovery and classification: deep learning approach", *Neural Computing and Applications*, pp. 1-10, 2015.

[44] J. Singha, and R. H. Laskar, "Hand gesture recognition using two-level speed normalization, feature selection and classifier fusion", *Multimedia Systems*, pp.1-16, 2016.

[45] P. Kumari, and A. Vaish, "Feature-level fusion of mental task's brain signal for an efficient identification system", *Neural Computing and Applications*, vol. 27, no. 3, pp.659-669, 2016.

Mohd Asyraf Mansor obtained his MSc (2014) and BSc(Ed) (2013) from Universiti Sains Malaysia. He is currently pursuing Ph.D degree at School of Mathematical Science, Universiti Sains Malaysia. His current research interests include evolutionary algorithm, satisfiability problem, neural networks, logic programming and heuristic method especially Artificial Immune System.

Mohd Shareduwan bin M. Kasihmuddin received his MSc (2014) and BSc(Ed) (2013) from Universiti Sains Malaysia. He is currently pursuing Ph.D degree with School of Mathematical Science, Universiti Sains Malaysia Penang Malaysia. His current research interests include neuro-heuristic method especially Artificial Bee Colony algorithm, constrained optimization problem, neural network and logic programming.

Saratha Sathasivam is a lecturer in the School of Mathematical Sciences, Universiti Sains Malaysia. She received her MSc and BSc(Ed) from Universiti Sains Malaysia. She received her Ph.D at Universiti Malaya, Malaysia. Her current research interest are neural networks, agent based modeling and constrained optimization problem.