Journal of Informatics and Mathematical Sciences
Vol. 8, No. 5, pp. 317–334, 2016
ISSN 0975-5748 (online); 0974-875X (print)
Published by RGN Publications

http://www.rgnpublications.com

Research Article

Artificial Bee Colony in the Hopfield Network for Maximum k-Satisfiability Problem

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

School of Mathematical Sciences, Universiti Sains Malaysia, Pulau Pinang, Malaysia
***Corresponding author:** iwanmaidin@gmail.com

Abstract. Artificial bee colony (ABC) is a relatively new swarm intelligence method that solves the various type of optimization problems. This algorithm utilized the behavior of the actual bees to minimize or maximize the cost function of any combinatorial problems. The aim of this study is to introduce the hybridization of artificial bee colony algorithm with Hopfield network in doing Restricted MAX-kSAT. The performance of the proposed paradigm will be compared with conventional Hopfield network. The result obtained from computer simulation indicates the beneficial features of ABC towards Hopfield network in doing MAX-kSAT. The findings have led to a significant implication on the choice of determining an alternative method to do MAX-kSAT problem.

Keywords. Artificial bee colony; Hopfield neural network; Maximum k-Satisfiability

MSC. 03B40; 62M45; 92B20

Received: August 31, 2016 **Accepted:** October 5, 2016

1. Introduction

In recent years, artificial intelligence (AI) has provided intensified algorithmic research for the development of constraint satisfaction and Boolean satisfiability. Most of the computational task in AI can be shown to be NP problem in their most general form [8]. In that sense, satisfiability problem (SAT) is a central debatable problem in artificial intelligence, logic, and computational complexity. SAT has a wide variety of applications such as consistency in expert systems knowledge bases [5, 13, 14, 45]. The essentiality of satisfiability problem is expressed as follows [16]:

Given a collection of m clauses involving n logical variable, determine whether or not there exist a truth assignment for clauses that all m clauses are simultaneously satisfied.

If the answer to the above SAT is "no", the very next question imposed to ourselves is "how close one get to the satisfiability?" Making that question precise leads to the Maximum k-satisfiability (MAX-kSAT) problem [24]. Restricted MAX-kSAT can be defined as a problem to assign value to Boolean variable with k literal per clause that maximizes the number of satisfied clauses.

The idea of implementing the artificial neural network in AI is one step forward to understand learning and memory task of the actual brain. Despite years of efforts, the precise way of learning inside the brain is still poorly understood. The similar manner of learning and memory task are believed by some [27, 46] to occur in artificial neural network system through collective representation of synaptic weight and states of the artificial neurons. Hopfield neural network (HNN) is a recurrent network that has an efficient associative memory and is simple enough to personify human brain [17]. One of the beneficial property of Hopfield neural network is the minimization of energy when there are any changes in neurons input. Due to the robustness of energy changes in Hopfield network, several researchers merged the idea of logic programming with Hopfield network. The recent celebrated models were developed by Sathasivam [38, 40] and Wan Abdullah [1, 2]. The models exploited the cost function of the logical clauses and venture the logical inconsistencies of Horn clauses [2]. The network will inherit the synaptic weight by comparing cost function with Lyapunov energy equation. In that sense, obtained synaptic weight will become the building blocks for energy minimization in Hopfield neural network [33, 35]. As the network gets larger, searching the correct assignments via Hopfield network is immensely convoluted. With an enormous amount of constraints in MAX-kSAT logic, traditional Hopfield network is expected to face complexity and most of the solution trapped at suboptimal minima. Effective searching method during training might reduce the complexity of Hopfield neural network. Since Hopfield neural network is a simple and easy to comprehend with other algorithms, established heuristic techniques [9, 28, 36, 37] were able to merge with Hopfield network as a single hybrid network.

Swarm intelligence is a meta-heuristic method in the field of artificial intelligence that utilized approximation algorithm to find optimal solutions. These swarms are able to solve complex tasks without centralized control. Exploration and exploitation are the important mechanisms in a powerful search process. While exploration process is related to the independent search for an optimal solution, exploitation utilized the existing knowledge to separate the search space [7]. Researchers have analyzed the behaviors of the swarms and designed a specific algorithm to solve specific minimization problem. Previously, artificial bee colony (ABC) algorithm introduced by D. Karaboga [22] to improve both exploration and exploitation for solving some numerical optimization problem. Recent research has shown that algorithm based on ABC algorithm have great potential [21, 23, 43]. The algorithm employs distributed agents who mimic the way real honey bees locate their nectar foods. The distinctive group of bees namely employed bees, onlooker bees, and scout bees worked in harmony to find the best food source (solution). Their foraging behavior, learning, memorizing and information sharing characteristics have been beneficial to tailor the solution as close as to reality [44]. It

was proven by some literature that ABC algorithm has successfully solved a various type of satisfiability problem [**?**,12]. Pursuing that, ABC is introduced in this study to facilitate Hopfield network in solving MAX-kSAT problem. HNN-MAXkSATABC indicates the hybridization of Hopfield network and artificial bee colony in solving any randomized MAX-kSAT problem. Although some of the solutions obtained may be trapped at local minima, the solution of the MAX-kSAT based on HNN-MAXkSATABC was indeed promising.

This paper has been organized as follows. Section 2 introduces the restricted maximum k-Satisfiability (MAX-kSAT) problem. In section 3, neuro searching methods including exhaustive search (ES) and artificial bee colony (ABC) in doing MAX-kSAT will be discussed. In section 4, neuro-logic paradigm consists of Hopfield network, Wan Abdullah method, and Sathasivam relaxation method will be discussed. Hence, the overall implementation of our proposed method will be discussed in section 5. Experimental result and conclusions were drawn in section 6 and 7.

2. Restricted Maximum k-Satisfiability Problem

In a classical satisfiability problem, we are given a conjunctive normal form (CNF) formula with an array of clauses $\{C_1, C_2, \ldots, C_n\}$. Each clause C_k) is made up from the disjunction of k literals $\{x_{k1}, x_{k2}, \ldots, x_{k1}\}$ [47]. In this research, we focus on the restricted maximum satisfiability problem. Generally speaking, the restricted maximum satisfiability problem is a counterpart of classical satisfiability problem. The maximum k-Satisfiability problem (MAX-kSAT) is a combinatorial problem and well-known to be a variant of NP-hard problem [11]. The core impetus of MAX-kSAT is to search for an optimal truth assignment with the maximum value in which each clause (constraint) has at most k literals [20, 31]. In this paper, the maximum k-satisfiability is treated as a constraint optimization problem.

Definition. Given a Boolean formula, P in conjunctive normal form (CNF), determine the maximum number of clauses of P that can be satisfied by a particular assignment. P consists of n clauses containing k number of variables per clause and positive integer g where $g \leq n$.

Given a Boolean formula MAX-kSAT can be defined implicitly as a pair of (λ, θ) where λ is the set if all possible solution $\{1, -1\}^n$ bit string and θ is a mapping $\lambda \rightarrow \xi$ which denotes the score of the assignments [26]. ξ is scored based on correct clausal assignment. Therefore, MAX-kSAT problem consists of defining the best bipolar/binary assignments to the variables in P that simultaneously satisfies at least g of the n clauses [4, 25]. The aim of the system is to decide the "optimized" assignment that can satisfy the maximum number of clauses containing k variables. In our case, restricted value of k in MAX-kSAT only allowed $k = 2$ and $k = 3$. Restricted MAX-kSAT can be included as the combinatorial problem that works in parallel with logic programming [10]. Equation (2.1) is an example of MAX-kSAT formula:

$$P = (x \vee y) \wedge (x \vee \neg y) \wedge (\neg x \vee y) \wedge (\neg x \vee \neg y) \wedge (z \vee w). \tag{2.1}$$

Equation (2.1) comprise of variables x, y, z and w formula P is unbearable to satisfy since there are no assignment that make constraint P become true. As the number of clauses increased,

finding satisfying assignments will become extremely convoluted. In order to ease the searching process, we require intricate neuro-searching paradigm.

3. Neuro-Searching Paradigm

(a) Exhaustive Search (ES) algorithm

Exhaustive search is a conventional searching technique in the Hopfield network based on the logical inconsistencies. It was successfully applied by a few researchers in modeling and representing many applications [3, 32]. Theoretically, the exhaustive (ES) algorithm is demarcated as the simplest local search algorithm, but it can be turned into computationally super expensive. The fact about ES is it exhaustively searches for the entire possible clause, even though the search space gets larger [35]. The main advantage of this algorithm is the guarantee to obtain a feasible solution (assignment with the maximum satisfied clauses) by taking into consideration bigger search space.

The exhaustive search (ES) paradigm is constructed according to "generate and test" concept to enumerate the entire possible solutions in a search space [48]. In consequence, the network requires more time to search for the maximum satisfied clauses since the search spaces include the less fit assignments. The assignment with the maximum number of satisfied clauses will be stored as a pattern in content addressable memory of neural network. In theory, the exhaustive search algorithm devours more computation time in searching for the maximum satisfied assignment completely. Therefore, the computation complexity is epitomized as $O(2^n)$ [42]. This hybrid network will encounter higher complexity as the program attempt tremendous amount of clauses. Consequently, the computation time will be intensified and potentially end up in an infinite loop. In this research, we generate random bit strings (represented the assignment of the MAX-kSAT logic) and straight away record the number of satisfied clauses. Henceforth, it is not guaranteed that the bit strings are not converging to global maxima during the first iteration of exhaustive search. In our context, we will implement the exhaustive search algorithm as a searching technique in assisting the Hopfield network for solving the maximum k-satisfiability problem (HNN-MAXkSATES). The exhaustive search algorithm is given as follows:

***Step* 1**
Enumerate the random binary bit strings. The randomized bit strings are called candidate bit strings.

***Step* 2**
Test the candidate bit strings and calculate the number of satisfied clauses.

***Step* 3**
Return the assignment (bit strings) with the maximum number of satisfied clauses as an output.
Else
Repeat Step 1 and 2.

(b) Artificial Bee Colony (ABC) algorithm

Artificial bee colony (ABC) is classified into three vital components such as employed bee, onlooker bee and scout bee with each component is equipped with unique characteristics [23]. Employed bees were assigned to locate the best food source. They are entrusted with the task of communicating the piece of solution (location of the food) with the onlooker bees. The location from the employed bees will be transferred to onlooker bees via waggle dance. If the solution does not improve, scout bees will reset and reincarnate the new search space.

In our proposed task, we are launching a new model to solve MAX-kSAT problem by means of artificial bee colony (ABC) algorithm with a view to make sure that the complexity of searching the correct assignment is brought down as little as possible. For this function, we are looking at n variables and m clauses for each solitary algorithm task. Since the complexity of the searching method grew immensely when the number of clauses increased, we have employed ABC algorithm as our innovative part technique before integrating it with neural network. Firstly, every variable in MAX-kSAT is demarcated by bee. The bees will be entrusted with the task of locating the nectar quantity (assignments) of the food source relative to fitness value (number of satisfied clauses). In our envisaged work, we have targeted our aim to increase the chances to locate the optimal assignments with a shorter period of time and avoid non-improving assignment. The ABC algorithm in solving MAX-kSAT problem is given as follows:

Stage 1. *Initialization*
Generate and initialize 50 employed bees, 50 onlooker bees, and 1 scout bee. Each bee represents the bit string of MAX-kSAT assignment. Huge amount of employed bees and onlooker bees will diversify the local search and may lead to the global solution.

Stage 2. *Fitness evaluation*
Calculate the fitness of each employed bees. This fitness evaluation is based on the number of satisfied clauses. The fitness of the employed bee is given as followed:

$$f_{bee} = \max[c_1(x) + c_2(x) + c_3(x) + c_3(x) + \ldots + c_n(x)] \tag{3.1}$$

Stage 3. *Employed bees phase*
Employed bee is entrusted with the task of hunting for the food source and conveying the bit string assignment. The adaptation of the employed bee is given as v_{ij}.

$$v_{ij} = x_{ij} \vee (\phi_{ij} \otimes (x_{ij} \wedge x_{kj})) \tag{3.2}$$

where

x_{ij} is an initial food source

x_{kj} is an observed food source

ϕ_{ij} parameter where

$$\phi_{ij} = \begin{cases} 1, & \text{rand}(0,1) < 0.5 \\ -1, & \text{rand}(0,1) \geq 0.5 \end{cases}$$

$\otimes$ is a 'XOR' operator

$\wedge$ is an 'AND' operator

$\vee$ is an 'OR' operator

After the adaptation, fitness of each employed bee will be calculated by using equation (3.1).

Stage 4. *Onlooker bees phase*

When the employed bees return to their hive, the food information (bit string assignment) will be transferred to onlooker bees via dancing. Onlooker bees will choose the information based on the following roulette wheel selection [15].

$$p_i = \frac{f_{bee}}{\sum_{i=1}^{SN} f_{bee}} \tag{3.3}$$

Where $\sum_{i=1}^{SN} f_{bee}$ is the desired fitness, while SN is denoted by the group size of the bees.

Onlooker bees were entrusted to find the food source by using equation (3.2). After the process, the fitness of the onlooker bee will be calculated by using equation (3.1). The best food source will be generated until the number of trial is equal to the predefined limit.

Stage 5. *Scout bee phase*

If the quality of the solution does not get better than the threshold fitness, the food source will be treated as excess [30]. Scout bee will abandon the search space [43]. The act of scout bee will help the algorithm to avoid local maxima or non-improving solution. Note that, if the algorithm found a solution with desired fitness, the solution will exit the algorithm easily and print the best solution. The correct solution will be improved further by using Hopfield network.

4. Neuro-Logic Paradigm

4.1 Hopfield Model

In the yesteryears, the research related to the Hopfield neural network has received a prolific amount of attention [19, 29]. The Hopfield neural network which popularized by John Hopfield [17] is well known in artificial intelligence due to the content addressable memory (CAM) and energy minimization capability [33]. The Hopfield neural network is an auto-associative model and systematically store patterns as a content addressable memory (CAM) [40]. Basically, the main architecture of the Hopfield model is comprising of a two-dimensional neural network in which the linking strengths between neurons which are binary threshold neurons are determined based on the constraints and solution criteria of the optimization problem to be solved [41]. Other than that, the Hopfield neural network is recurrent associative networks in which the output of the network will be fed back to the input units using further feedback connections [42].

The units in Hopfield nets (binary threshold unit) are represented by bipolar values of 1 and -1 [38]. Thus, the binary threshold units will bring the dynamics to the network to be applied

in other binary algorithm. Mathematically, the definition for unit I's activation a_i can be shown in equation (4.1).

$$a_i = \begin{cases} 1 & \text{if } \sum_j W_{ij} S_j > \psi_i \\ -1 & \text{otherwise} \end{cases} \tag{4.1}$$

where W_{ij} is the synaptic weight from unit j to i. S_j depicts the state of unit j and ψ_i is the threshold of unit i. The brief Hopfield model architecture involves an array of N recognized neurons, each was described by an Ising spin variable model. Point to ponder is the connection in Hopfield net enclose no connection with itself $W_{ii} = W_{jj} = 0$. This property gave the Hopfield model symmetrical features. In addition, the neuron state is basically bipolar $S_i \in \{1, -1\}$ and the spin points follow in the direction of magnetic field [2]. This cause each neuron to flip until the equilibrium is reached. Consequently, it follows the dynamics $S_i \rightarrow \text{sgn}(h_i)$ where h_i is the local field of the neurons connection [40, 42]. When dealing with higher order neurons connection, the sum of the field induced by each neurons are as followed:

$$h_i = \ldots + \sum_j W_{ijk}^{(3)} S_j S_k + \sum_j W_{ij}^{(2)} S_j + W_i^{(1)}. \tag{4.2}$$

The synaptic weight can be computed by using Wan Abdullah method [2], back-propagation [19] or Hebbian rule [38]. However, we opt to apply the Wan Abdullah technique due to the simplicity of the technique. Since the higher order constraint of synaptic weight in Hopfield network is constantly symmetrical and does not contain zero diagonal, the updating rule maintains as follows:

$$S_i(t+1) = \text{sgn}[h_i(t)]. \tag{4.3}$$

Equation (4.3) is necessary in order to ensure the energy decrease monotonically with the network. The generalized Lypunov energy equation that accommodate more neurons connections are as followed:

$$E = -\frac{1}{3} \sum_i \sum_j \sum_k W_{ijk}^{(3)} S_i S_j S_k - \frac{1}{2} \sum_i \sum_j W_{ij}^{(2)} S_i S_j - \sum_i W_i^{(1)} S_j. \tag{4.4}$$

This energy function is substantial because it forms the degree of convergence of the network. Thus, the energy value obtained from the equation will be classified as a global or local solution just by looking at the final energy. In our context, the network will produce the correct solution when the induced neurons state reached global minimum energy. Besides, the process of attaining global minimum energy always associated with the correct model of synaptic weight. Hence, we will compute the synaptic weights by using Wan Abdullah training method.

4.2 Wan Abdullah Method

The training method is absolutely vital in any neural network. In this exploration, we implement a solid training method in order to compute the synaptic weight systematically. Hence, we opt to train the HNN-MAXkSATABC and HNN-MAXkSATES by using the Wan Abdullah technique.

Strictly speaking, the Wan Abdullah's method became the inaugural technique in synaptic weight extraction based on logical inconsistencies [1, 42]. It has been proven to be a decent training method especially in obtaining the synaptic weight in the Hopfield neural network.

Basically, the MAX-kSAT will be treated as a constrained optimization problem to be further solve by using Hopfield neural network. Thus, the cost function is needed to compute the synaptic weight. Hence, the cost function that corresponds to MAX-kSAT clauses is the minimization of logical inconsistencies.

$$\min_{i\in(0,\infty),C_i=1} \neg C_i\,. \tag{4.5}$$

If the number of "wrong" assignment is minimized, the number of satisfied MAX-kSAT clauses will be maximized. For example, we denote the following MAX-2SAT and MAX-3SAT problem with α and ϕ. Thus, equation (4.6) and (4.7) were treated as the example of MAX-2SAT and MAX-3SAT problem.

$$\alpha = (A\vee B)\wedge(A\vee\neg B)\wedge(\neg A\vee B)\vee(\neg A\vee\neg B)\wedge(C\vee D), \tag{4.6}$$

$$\begin{aligned}\phi &= (P\vee Q\vee R)\wedge(\neg P\vee Q\vee R)\wedge(P\vee\neg Q\vee R)\wedge(P\vee Q\vee\neg R)\wedge(\neg P\vee\neg Q\vee R)\\ &\quad\wedge(\neg P\vee Q\vee\neg R)\wedge(P\vee\neg Q\vee\neg R)\wedge(\neg P\vee\neg Q\vee\neg R)\wedge(S\vee T\vee U).\end{aligned} \tag{4.7}$$

We require the cost function in order to compute the synaptic weights. The cost function, f_{cost} for MAX-2SAT and MAX-3SAT are shown in equation (4.8) and (4.9) respectively.

$$\begin{aligned}f_{\text{cost}\,\alpha} &= \frac{1}{2}(1-S_A)\frac{1}{2}(1-S_B)+\frac{1}{2}(1-S_A)+\frac{1}{2}(1+S_B)+\frac{1}{2}(1+S_A)\frac{1}{2}(1-S_B)\\ &\quad+\frac{1}{2}(1+S_A)\frac{1}{2}(1+S_B)+\frac{1}{2}(1-S_C)\frac{1}{2}(1-S_D),\end{aligned} \tag{4.8}$$

$$\begin{aligned}f_{\text{cost}\,\phi} &= \frac{1}{2}(1-S_p)\frac{1}{2}(1-S_Q)\frac{1}{2}(1-S_R)+\frac{1}{2}(1+S_P)\frac{1}{2}(1-S_Q)\frac{1}{2}(1-S_R)\\ &\quad+\frac{1}{2}(1-S_p)\frac{1}{2}(1-S_Q)\frac{1}{2}(1-S_R)+\frac{1}{2}(1+S_P)\frac{1}{2}(1-S_Q)\frac{1}{2}(1-S_R)\\ &\quad+\frac{1}{2}(1-S_p)\frac{1}{2}(1-S_Q)\frac{1}{2}(1-S_R)+\frac{1}{2}(1+S_P)\frac{1}{2}(1-S_Q)\frac{1}{2}(1-S_R)\\ &\quad+\frac{1}{2}(1-S_p)\frac{1}{2}(1-S_Q)\frac{1}{2}(1-S_R)+\frac{1}{2}(1+S_P)\frac{1}{2}(1-S_Q)\frac{1}{2}(1-S_R)\\ &\quad+\frac{1}{2}(1-S_T)\frac{1}{2}(1-S_U)\frac{1}{2}(1-S_V).\end{aligned} \tag{4.9}$$

By comparing equation (4.8), (4.9) with equation (4.4), we obtain the synaptic weights for α and ϕ. The synaptic weights are summarized in Table 1 and Table 2.

Table 1. Synaptic weight for α based on Wan Abdullah 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 2. Synaptic weight for ϕ based on Wan Abdullah 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	$\frac{1}{16}$

Alternatively, the Hebbian learning can be implemented to imprint the adequate synaptic weight for the neurons connection. Sathasivam [39] proved that most of the synaptic weight obtained by using Wan Abdullah method are similar due to Hebbian learning due to clausal MAX-kSAT similarity. Although both methods are expected to produce the similar synaptic weight information, Wan Abdullah method is proven to minimize the spurious minima produced by clauses compared to Hebbian learning [17]. Then, the system will undergo a series of relaxation process after taking into consideration the local field computation in equation (4.2).

4.3 Network Relaxation

Network relaxation approach is totally essential in improving the quality of the solutions. Hence, the quality of the solutions gained by the Hopfield network can be affected by various influences [40]. Persistent firing and receiving information among the neurons can reduce the quality of the solutions. Thus, the process devours more computation time. 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" can be delineated as a series of relaxation loop in the program after the local field has been obtained [38]. Without proper relaxation mechanism, network tends to produce more local minima solution. Since MAX-kSAT contains more clausal constraints, we implemented Sathasivam relaxation technique [42] to ensure the network relaxed to its final state. Hence, the information exchange between neurons will be updated based on the following equation:

$$\frac{dh_i^{\text{new}}}{dt} = R\frac{dh_i}{dt} \tag{4.10}$$

where R denotes the relaxation rate and h_i refers to the local field of the network as listed in equation (4.2). For our case, constant relaxation $R = 2$ will be used in our program.

5. Implementation

The simulations for HNN-MAXkSATABC and HNN-MAXkSATES were executed by using MICROSOFT VISUAL C++ 2013 for Windows 10. The main task of the program is to find the correct "model" that can solve randomized MAX-kSAT. Initially, all the variables and clauses are randomized. The following algorithm shows how we can implement HNN-MAXkSATABC and HNN-MAXkSATES in C++ program.

(a) Translate MAX-kSAT clauses into Boolean algebra (if any).

(b) Assign neuron to each variable in MAX-kSAT formula.

(c) Randomize the state of the neurons and initialize all synaptic weight to zero.

(d) Derive minimum cost function for MAX-kSAT by assigning $X = \frac{1}{2}(1+S_X)$ and $\bar{X} = \frac{1}{2}(1-S_X)$. The state of the neuron learned true when $S_X = 1$ and false when $S_X = -1$.

(e) By using Wan Abdullah method, compare the derived cost function with equation (4.4).

(f) Check clauses satisfaction by applying artificial bee colony and exhaustive search method. Satisfied assignment will be fed to the Hopfield network.

(g) Apply Sathasivam's relaxation method via equation (3.3) to the Hopfield network

(h) Compute the corresponding local field by using equation (4.2). Find the final state of the neurons. If the state of the neurons remains unchanged after five loops, we consider it as stable state.

(i) Find the corresponding final energy by using equation (4.4). Verify whether the final energy is a local minimum energy of global minimum energy.

(j) 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 [40], 0.001 is chosen as tolerance value for Lyapunov energy since it gives us a better filtering mechanism.

6. Result and Discussion

6.1 Global Minima Ratio

Table 3. Comparison of the global minima ratio for HNN-MAXkSATABC and HNN-MAXkSATES

Number of neurons (NN)	HNN-MAX2SATES	HNN-MAX2SATABC	HNN-MAX3SATES	HNN-MAX3SATABC
10	0.9632	0.9999	0.9480	0.9996
20	0.9563	0.9905	0.9233	0.9811
30	0.9370	0.9823	0.9108	0.9642
40	0.9052	0.9539	0.8752	0.9433
50	0.8735	0.9374	0.8438	0.9162
60	0.8440	0.9202	0.8120	0.9099
70	-	0.9023	-	0.8994

Table 3 associates the global minima ratio with different number of neurons for HNN-MAXkSATABC and HNN-MAXkSATES. According to combinatorial optimization standpoint, global minima ratio can be demarcated as the total minimum energy obtained over total number of runs processed by a particular model [42]. 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). It is evident from Table 1 that the global minima ratio for HNN-MAXkSATABC is closer to one compared to HNN-MAXkSATES. Hence, it depicts that most of the solutions converge towards global minimum energy even though the complexity of the program increased. The implementation of *Artificial Bee Colony* (ABC) algorithm in HNN-MAXkSATABC network allowed the solutions to undergo systematic improvement in order to obtain global or feasible solution. During the searching process in HNN-MAXkSATABC , the food source (solution) found by employed bee can be enhanced further via onlooker bees. Chances to obtain the correct solutions will increase when the information was transferred from the employed bee to the onlooker bees. As a result, these agents will permit the network to search and compute the feasible solutions. According to Table 1, the HNN-MAXkSATABC is able to withstand up to 70 neurons. The presence of scout bee finding the "fresh" search space will help the network to avoid local maxima (non-improving solution). Thus, the chance of

obtaining the global solutions will become higher due to the ability of HNN-MAXkSATABC to counter the complexity.

On the contrary, the global minima ratio for HNN-MAXkSATES is higher and not much closer to one. The drawback of HNN-MAXkSATES was the mechanism of the exhaustive search that deploys tedious training process in searching the correct neuron states and allow more local maxima solutions (non-improving fitness). Hence, the HNN-MAXkSATES generated more abrupt energy surface for the program to obtain more non-improving solutions. This is because the network needs to search for the correct solutions exhaustively without any improvement in the solution. However, it can be seen that HNN-MAXkSATES managed to sustain up to 50 neurons due to the complexity. The exhaustive search (ES) will be enumerating the new solutions without improving the existing one. In addition, the neurons had to jump enormous energy barrier to reach the global solutions.

6.2 Ratio of Satisfied Clauses

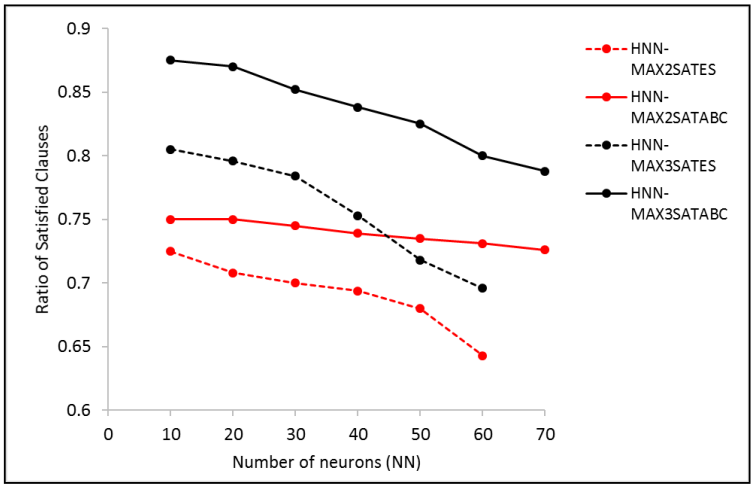

Figure 1. Comparison of the ratio of satisfied clauses for HNN-MAXkSATABC and HNN-MAXkSATES

Since we deal with highly convoluted MAX-kSAT constraint, it was wise to verify the efficiency of the hybrid system in finding the correct solution. The ratio of satisfied clauses can be defined as the total number of satisfied clauses over total number of clauses [16].

Figure 1 illustrates the ratio of satisfied clauses over total number of clauses obtained by HNN-MAX2SATES, HNN-MAX2SATABC and HNN-MAX3SATES, HNN-MAX3SATABC. In maximum satisfiability problem, the ratio of satisfied clauses will never be equal to 1. The higher the ratio of satisfied clauses, the more clauses will be satisfied in any randomized MAX-kSAT problem. As it stands, HNN-MAXkSATABC is proven to produce more satisfied clauses compared to HNN-MAXkSATES. ABC algorithm is still able to maintain the quality of ratio even though the number neuron has been escalated. The searching method via ABC helps to segregate non-solution in MAXkSAT constraint efficiently before it was fed to Hopfield network. The solutions in ABC were improved dramatically due to the bee's adaptation and transfer

of information between employed bees and onlooker bees. In contrast, conventional Hopfield network via exhaustive search method has less efficient solution filtering mechanism. The complexity of the searching method causes Hopfield network to produce more local minimum energy.

6.3 Fitness Energy Landscape

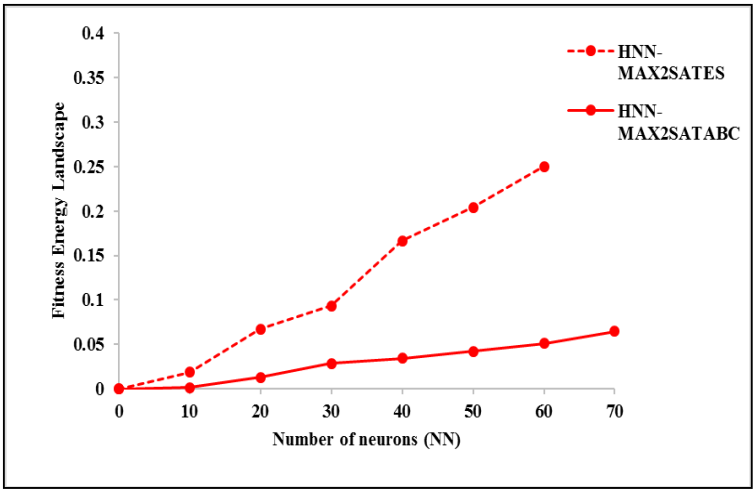

Figure 2. Comparison of the fitness energy landscape for HNN-MAX2SATABC and HNN-MAX2SATES

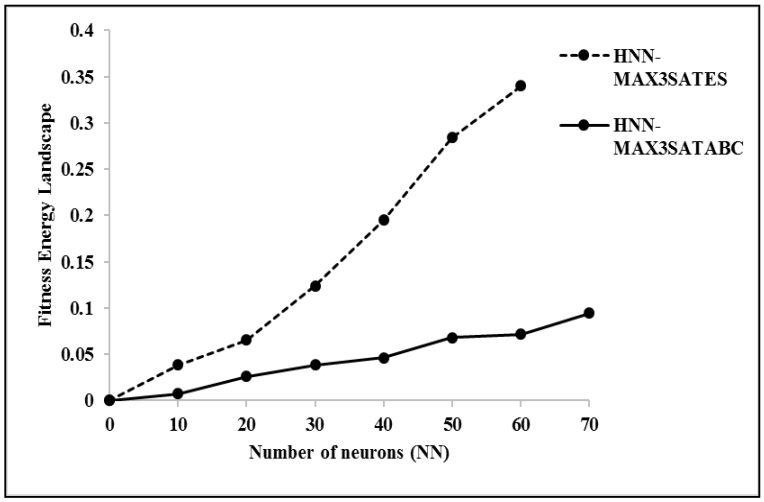

Figure 3. Comparison of the fitness energy landscape for HNN-MAX3SATABC and HNN-MAX3SATES

Fitness energy landscape is a vital indicator in appraising the performance of algorithms to solve combinatorial optimization problem. In our exploration, the competency to store the optimized patterns are systematically measured by the fitness energy landscape based on Kaufmann's model [18]. Generally speaking, the optimum energy landscape value will intensively drive the algorithm to compute the feasible solutions. Figure 2 and 3 represent the difference in

the energy landscape value against the number of neurons for HNN-MAXkSATABC and HNN-MAXkSATES. It is evident from the graphs that the fitness energy landscape for HNN-MAXkSATABC is almost flat (near to zero) even though the complexity of the network is increased. The fitness energy landscape for HNN-MAXkSATABC demonstrates the ability of the algorithm to enhance the less fit solutions. Thus, the less fit solutions are improved by the transfer of information by employed bees to onlooker bees via equation (3.2) in order to obtain feasible solutions. These two agents will provide robust mechanism to the model in minimizing the energy yet improving the solutions. In addition, the ABC has abridged the complexity in searching the feasible solutions for the higher number of neurons. On contrary, the fitness energy landscape for HNN-MAXkSAT is slightly higher and more rugged compared to the HNN-MAXkSATABC. Besides, in the case of HNN-MAXkSATES, as the number of neurons goes up, it leads to the increase of less fit solutions (local minima) and that will drive the energy landscape to be more rugged. This is due to the nature of exhaustive search that deployed the trial and error method in hunting and improving the desired solutions. As a result, the process requires more energy and it will contribute to more rugged energy landscape. On the other hand, HNN-MAXkSATES is not competent in sustaining the higher number of neurons. It is evident from Figure 2 and 3 that HNN-MAXkSATES only capable up to 60 neurons compared to our proposed technique, HNN-MAXkSATABC. In order to tackle this menace, the HNN-MAXkSATABC is vital to obtain more solid fitness landscape value.

6.4 Computation Time

Table 4. Comparison of the computation time (in seconds) for HNN-MAXkSATABC and HNN-MAXkSATES

Number of neurons (NN)	HNN-MAX2SATES	HNN-MAX2SATABC	HNN-MAX3SATES	HNN-MAX3SATABC
10	28	1	39	3
20	126	15	180	27
30	389	78	516	112
40	2934	341	3655	432
50	12678	706	14870	948
60	88743	1722	108854	2530
70	-	3345	-	4409

Computation time can be defined as the amount of time required by neurons to reach the final state after series of relaxation. Computation time is a good indicator to determine the complexity of the network. Table 4 delineates the running time obtained by HNN-MAX2SATES, HNN-MAX2SATABC and HNN-MAX3SATES, HNN-MAX3SATABC. Since we deal with MAX-kSAT problem, the training process requires more computation time compared to the ordinary

randomized kSAT problem. Under those circumstances, MAX-3SAT was expected to encounter more computation time compared to MAX-2SAT due to the number of constraints. According to Table 2, HNN-MAXkSATABC require less computation time compared to HNN-MAXkSATES. Although conventional Hopfield network might minimize the logical inconsistencies of the MAX-kSAT problem effectively, the computational burden surged indefinitely when the number of neurons increased. Exhaustive search method utilized trial and error method in finding the correct solution. Consequently, the wrong solution will be "forgotten" and the searching restart. The network is forced to bear inessential computation burden to find the next least "inconsistent" solution. At $NN = 70$, HNN-MAXkSATES was seen to trapped at trial and error state and failed to reach the relaxation phase in stipulated time. In contrast, the remarkable adaptation of the employed bees and onlooker bees via equation (3.2) and (3.3) in innovative ABC definitely improve the "location" of the required solution. Non-solution in HNN-MAXkSATABC will be developed further until the solution reaches the desired fitness. The presence of scout bee (if required) helps the network to avoid non-improving solution (local maxima). Impressively, the beneficial bees in HNN-MAXkSATABC reduce the time gap between the training and relaxation phase of the network.

7. Conclusion

A solid paradigm was successfully developed by taking the advantages of a brand new swarm-inspired heuristic method, known as artificial bee colony algorithm and the supremacy of Hopfield network as a single network to solve MAX-kSAT problem. We had fruitfully developed a network by using artificial bee colony (ABC) algorithm incorporated with Hopfield neural network in solving maximum k-satisfiability (HNN-MAXkSATABC). The proposed model was compared with a conventional technique, exhaustive search (ES) with Hopfield neural network (HNN-MAXkSATES). It is unequivocally established that HNN-MAXkSATABC is competent in term of better global minima ratio, higher number of satisfied clause, ideal fitness energy landscape value and faster computation time compared to HNN-MAXkSATES. On a more esoteric note, the results demonstrated the ability of HNN-MAXkSATABC in tackling more challenging combinatorial optimization problem. The upcoming research in this regard will be focusing on the investigation to solve the other variants of satisfiability problem such as weighted maximum satisfiability (Weighted MAX-SAT), Query-guided maximum satisfiability (Q-MAXSAT) and quantified maximum satisfiability problem by using our proposed technique.

Competing Interests

The authors declare that they have no competing interests.

Authors' Contributions

All the authors contributed significantly in writing this article. The authors read and approved the final manuscript.

References

[1] W.A.T.W. Abdullah, Logic programming on a neural network, *International Journal of Intelligent Systems* **7** (6) (1992), 513 – 519.

[2] W.A.T.W. Abdullah, The logic of neural networks, *Physics Letters A* **176** (3) (1993), 202 – 206.

[3] U. Aiman and N. Asrar, Genetic algorithm based solution to SAT-3 problem, *Journal of Computer Sciences and Applications* **3** (2) (2015), 33 – 39.

[4] T. Asano and D.P. Williamson, Improved approximation algorithms for MAX SAT, *Journal of Algorithms* **42** (1) (2002), 173 – 202.

[5] P. Asirelli, M.D. Santis and M. Martelli, Integrity constraints in logic databases, *The Journal of Logic Programming* **2** (3) (1985), 221 – 232.

[6] B. Ata and R. Coban, Artificial bee colony algorithm based linear quadratic optimal controller design for a nonlinear inverted pendulum, *International Journal of Intelligent Systems and Applications in Engineering* **3** (1) (2015), 1 – 6.

[7] A. Banharnsakun, T. Achalakul and B. Sirinaovakul, The best-so-far selection in artificial bee colony algorithm, *Applied Soft Computing* **11** (2) (2011), 2888 – 2901.

[8] S. Bart, D.G. Mitchell and H.J. Levesque, Generating hard satisfiability problems, *Artificial Intelligence* **81** (1) (1996), 17 – 29.

[9] M. Basu, Artificial bee colony optimization for multi-area economic dispatch, *International Journal of Electrical Power & Energy Systems* **49** (2013), 181 – 187.

[10] B. Borchers and J. Furman, A two-phase exact algorithm for MAX-SAT and weighted MAX-SAT problems, *Journal of Combinatorial Optimization* **2** (4) (1998), 299 – 306.

[11] A.Z. Broder, A.M. Frieze and E. Upfal, On the satisfiability and maximum satisfiability of random 3-CNF formulas, *4th ACM-SIAM Symposium on Discrete Algorithms* (*SODA*) **93** (1993), 322 – 330.

[12] H. Drias, S. Sadeg and S. Yahi, Cooperative bees swarm for solving the maximum weighted satisfiability problem, in *International Work-Conference on Artificial Neural Networks*, Springer, Berlin—Heidelberg (2005), 318–325.

[13] H. Frederick, D.A. Waterman and D.B. Lenat, *Building Expert Systems*, Reading, MA, Addisson-Wesley (1983).

[14] H. Gallaire, J. Minker and J.M. Nicolas, Logic and databases: A deductive approach, *Computing Surveys* **16** (2) (1984), 153 – 185.

[15] D.E. Goldberg and D. Kalyanmoy, A comparative analysis of selection schemes used in genetic algorithms, *Foundations of Genetic Algorithms* **1** (1991), 69 – 93.

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

[17] J.J. Hopfield and D.W. Tank, Neural computation of decisions in optimization problems, *Biological Cybernetics* **52** (3) (1985), 141 – 152.

[18] A. Imada and K. Araki, What does the landscape of a Hopfield associative memory look like, *Evolutionary Programming VII*, Springer, Berlin (1998), 647–656.

[19] L.M. Ionescu, A.G. Mazare and G. Serban, VLSI Implementation of an associative addressable memory based on Hopfield network model, *IEEE Semiconductor Conference* **2** (2010), 499 – 502.

[20] M. Jose and R. Majumdar, Cause clue clauses: error localization using maximum satisfiability, *ACM SIGPLAN Notices* **46** (6) (2011), 437 – 446.

[21] F. Kang, J. Li and Q. Xu, Structural inverse analysis by hybrid simplex artificial bee colony algorithms, *Computers & Structures* **87** (13) (2009), 861 – 870.

[22] D. Karaboga and B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *Journal of Global Optimization* **39** (3) (2007), 459 – 471.

[23] N. Karaboga, A new design method based on artificial bee colony algorithm for digital IIR filters, *Journal of the Franklin Institute* **346** (4) (2009), 328 – 348.

[24] R.M. Karp, Reducibility among combinatorial problems, *Complexity of Computer Computations*, Springer, US (1972), 85–103.

[25] A. Layeb, A new greedy randomised adaptive search procedure for solving the maximum satisfiability problem, *International Journal of Operational Research* **17** (4) (2013), 509 – 525.

[26] A. Layeb, A. H. Deneche and S. Meshoul, A new artificial immune system for solving the maximum satisfiability problem, *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems* (2010), 136 – 142.

[27] J. McCarthy, M.L. Minsky, N. Rochester and C.E. Shannon, A proposal for the dart mouth summer research project on artificial intelligence, *AI Magazine* **27** (4) (2006), p. 12.

[28] 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** (8) (2000), 1823 – 1839.

[29] M.K. Muezzinoglu, C. Guzelis and J.M. Zurada, A new design method for the complex-valued multistate Hopfield associative memory, *IEEE Transactions on Neural Networks* **14** (4) (2003), 891 – 899.

[30] A. Muthiah and R. Rajkumar, A comparison of artificial bee colony algorithm and genetic algorithm to minimize the makespan for job shop scheduling, *Procedia Engineering* **97** (2014), 1745 – 1754.

[31] R. Niedermeier and P. Rossmanith, New upper bounds for maximum satisfiability, *Journal of Algorithms* **36** (1) (2000), 63 – 88.

[32] J. Nievergelt (2000), Exhaustive search, combinatorial optimization and enumeration: Exploring the potential of raw computing power, *Sofsem 2000: theory and practice of informatics*, Springer, Berlin—Heidelberg (2000), 18 – 35.

[33] G. Pinkas, *Logical Inference in Symmetric Connectionist Networks*, Ph.D. Dissertation, Washington University, USA (1992).

[34] G. Pinkas, Reasoning, nonmonotonicity and learning in connectionist networks that capture propositional knowledge, *Artificial Intelligence* **77** (2) (1995), 203 – 247.

[35] R. Rojas, *Neural Networks: A Systematic Introduction*, Springer Science & Business Media (2013).

[36] S. Salcedo-Sanz, C. Bousoño-Calzón and A.R. Figueiras-Vidal, A mixed neural-genetic algorithm for the broadcast scheduling problem, *IEEE Transactions on Wireless Communications* **2** (2) (2003), 277 – 283.

[37] S. Sancho and X. Yao, A hybrid Hopfield network-genetic algorithm approach for the terminal assignment problem, *IEEE Transactions on Systems, Man and Cybernetics* **34** (6) (2004), 2343 – 2353.

[38] S. Sathasivam, Energy landscapes for Hopfield network programmed with program clauses, in *4th IASTED International Conference in Advances Computer Science and Technology*, Langkawi, Malaysia (2008), 179 – 182.

[39] S. Sathasivam, Energy relaxation for Hopfield Network with the new learning rule, *International Conference on Power Control and Optimization* (2009), 1 – 3.

[40] S. Sathasivam, Learning in the recurrent Hopfield network, in *IEEE Fifth International Conference on Computer Graphics, Imaging and Visualization* (CGIV'08), (2008), 323 – 328.

[41] S. Sathasivam, N. Hamadneh and O.H. Choon, Comparing neural networks: Hopfield network and RBF network, *Applied Mathematical Sciences* **5** (69) (2011), 3439 – 3452.

[42] S. Sathasivam, Upgrading logic programming in Hopfield network, *Sains Malaysiana* **39** (1) (2010), 115 – 118.

[43] A. Singh, An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem, *Applied Soft Computing* **9** (2) (2009), 625 – 631.

[44] D. Teodorović and P. Luéić, Intelligent parking systems, *European Journal of Operational Research* **175** (3) (2006), 1666 – 1681.

[45] N. Tin, W.A. Perkins, T.J. Laffey and D. Pecora, Checking an expert systems knowledge base for consistency and completeness, *Proceedings of the Ninth International Joint Conference on Artificial Intelligence* **85** (1) (1985), 375 – 378.

[46] R. Trippi and E. Turban, *Neural Networks in Finance and Investing: Using Artificial Intelligence to Improve Real World Performance*, McGraw-Hill Inc. (1992).

[47] W. Zhang, Configuration landscape analysis and backbone guided local search, Part I: Satisfiability and maximum satisfiability, *Artificial Intelligence* **158** (1) (2004), 1 – 26.

[48] I. Zinovik, D. Kroening and Y. Chebiryak, Computing binary combinatorial gray codes via exhaustive search with SAT solvers, *IEEE Transactions on Information Theory* **54** (4) (2008), 1819 – 1823.