Genetic Algorithm for Restricted Maximum *k*-Satisfiability in the Hopfield Network

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

School of Mathematical Science, Universiti Sains Malaysia

***Abstract* — The restricted Maximum k-Satisfiability MAX-kSAT is an enhanced Boolean satisfiability counterpart that has attracted numerous amount of research. Genetic algorithm has been the prominent optimization heuristic algorithm to solve constraint optimization problem. The core motivation of this paper is to introduce Hopfield network incorporated with genetic algorithm in solving MAX-kSAT problem. Genetic algorithm will be integrated with Hopfield network as a single network. The proposed method will be compared with the conventional Hopfield network. The results demonstrate that Hopfield network with genetic algorithm outperforms conventional Hopfield networks. Furthermore, the outcome had provided a solid evidence of the robustness of our proposed algorithms to be used in other satisfiability problem.**

***Keywords* — Exhaustive Search, Genetic Algorithm, Hopfield Neural Network, Restricted Maximum k-satisfiability.**

I. Introduction

Since decades ago, optimization has provided intensified algorithmic research for the development of constraint satisfaction and Boolean satisfiability. Hybridization between both field are motivated by application in scheduling, VLSI circuit, pattern reconstruction and many other applications. However, the main problem for both area is to assign interpretation or value to variables such that it makes the whole system become feasible [1]. Before it can be considered as optimization problem, both field must merge their character and able to produce their hybridized cost function [2]. Motivated by notable counterpart of Boolean satisfiability, restricted maximum k-satifiability (MAX-kSAT) has been a wide subject in constraint optimization problem. Restricted MAX-kSAT can be defined as a problem to assign value to Boolean variable with k literal per clause that maximize the number of satisfied clauses.

The idea of implementing artificial neural network to provide solution to optimization problem has been utilized by various researcher in artificial intelligence field. It is a fascinating field of study because it provides an alternatives way of doing computation and it is a paradigm towards understanding of the intelligence. It is a pursuit to visualize and represent information processing capabilities of an actual nervous system [3]. Hopfield neural network [4] is a simple recurrent network that has an efficient associative memory and resembled the biological brain [5]. The important property of the Hopfield neural network is the minimization of energy whenever there is any change in inputs. Due to effectiveness of energy changes in Hopfield neural network, several researchers have merged the idea of logic programming with Hopfield neural network. Several celebrated models were developed by Sathasivam [6] and Wan Abdullah [3]. Most of the model employed the cost function based on inconsistencies of the Horn clauses [5]. The cost function of the logic will be exploited in order to find the connection strength that act as a building block of the energy minimization.

Due to the complexity of network when the number of neuron increased, method of searching satisfied interpretation in a given MAX-*k*SAT clause should comply with traditional Hopfield network. The easiest method that compliment with Hopfield network is exhaustive search method (ES). The combination of exhaustive search method (ES) and Hopfield network namely HNN-MAX*k*SATES will be utilized to represent the conventional Hopfield network [32].

Genetic algorithm (GA) is increasingly viewed as optimization technique to a wide range of problem. Strictly speaking, genetic algorithm combines the idea of evolutionary improvement, recombination and mutation among the candidate solution. Since Hopfield neural network often providing a local minimum to solution [7], genetic algorithm will be incorporated with Hopfield network to do MAX-*k*SAT problem. The combination of genetic algorithm and Hopfield neural network were proven effective by many researchers in solving various optimization problem [8, 9, 10]. Thus, genetic algorithm is introduced in this study to supplement the Hopfield neural network to facilitate the search process of MAX-*k*SAT solution. HNN-*k*SATGA indicates the combination of Hopfield network and Genetic algorithm in solving any given MAX-*k*SAT problem. Although the solution obtained may stuck at local minima, the performance of the MAX-*k*SAT solution based on this hybrid algorithm was indeed promising.

This paper has been organized as follows. Section II introduces the *k*-satisfiability (*k*-SAT) and maximum *k*-satisfiability (MAX-*k*SAT). In section III, neuro searching methods including exhaustive search (ES) and genetic algorithm method (GA) in doing MAX-*k*SAT will be discussed. In section IV, neuro-logic paradigm comprises of the Hopfield model, Wan Abdullah's method and Sathasivam's relaxation method will be discussed. Furthermore, the implementation of our proposed method will be discussed in section V. Finally, section VI and VII enclose the experimental results and conclusion.

II. Maximum k-Satisfiability Problem

A. *k*-Satisfiability Problem

The *k*-SAT problem can be delineated as a conundrum of determining satisfiability of sets of clauses comprise of at most *k* literals per clause (*k*-CNF formulas). It is a general form of satisfiability problem that can be divided into the randomized satisfiability and maximum satisfiability [41]. Additionally, *k*-SAT problem can be expressed as *k*-CNF (*k*-Conjunctive Normal Form) or Krom formula [38]. Besides, *k*-SAT problem is considered as a NP problem or non-deterministic problem. Hence, the *k*-SAT problem involves logic formula that can be translated into an optimization problem. Therefore, the three core components of *k*-SAT are simplified as follows:

1. Consists of a set of *m* variables, $x_1, x_2, \ldots\ldots, x_m$

DOI: 10.9781/ijimai.2016.429

2. A set of literals. A literal is a variable or a negation of a variable.
3. A set of n distinct clauses: $C_1, C_2 .... C_n$. Each clause consists of only literals combined by just logical OR "V". Each clause must contain of variables.

In addition, the Boolean values are bipolar, consisting of 1 and -1 that could have exemplified the idea of true or false [45]. Hence, the goal of the k-SAT problem is to decide whether there exits an assignment of truth values to variables that makes the following formula satisfiable.

$$P = \bigwedge_{i=1}^{n} C_i \tag{1}$$

Where $\wedge$ is a logical AND connector, P denotes the entire Boolean formula for k-SAT. C_i is a clausal form of DNF with k variables. In our case, we investigated k=2 and k=3 for our satisfiability problem where the clause in randomized 2-SAT and 3-SAT has the following form:

$$C_i = \bigvee_{j=1}^{k} \left(x_{ij}, y_{ij}\right),\ k = 2 \tag{2}$$

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

B. Restricted Maximum k-Satisfiability

Restricted maximum k-Satisfiability problem (MAX-kSAT) can be defined as generalized form of Boolean satisfiability problem [37]. Given a Boolean formula P in conjunctive normal form (CNF) with n clauses containing variable each and positive integer g where $g \le m$. MAX-kSAT can be defined specifying implicitly a pair (λ, θ) [44] where λ is the set if all possible solution $\{1, -1\}^n$ and θ is a mapping $\lambda \rightarrow \xi$ which is denotes the score of the assignments. ξ is scored based on true clauses. Therefore, MAX-kSAT problem contains of defining the best bipolar/binary assignments to the variables in P that simultaneously satisfies at least g of the m clauses. Moreover, the mission is to decide the "optimized" assignment that can satisfy the maximum number of clauses containing k variables. Fundamentally, there are 2^n possible solutions to this problems. It was proven that MAX-kSAT is NP-complete problem for any $k \ge 3$.

There are numerous classifications of the MAX-kSAT namely, weighted MAXSAT [42] and Partial MAXSAT [43]. However, restricted MAX-kSAT constrained optimization problem that can be included in maximization problem [39]. Additionally, restricted MAX-kSAT can be ventured in logic programming [40]. In this exploration, we limit our analysis to k=2 and k=3. For instance, we can form the following 2-SAT formula:

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

Equation (4) is not possible to satisfy because no particular assignment will drive to all the clauses true. The following Table 1 portrays the truth table for P.

Table 1 depicts that entire assignments will not be able to make formula P true. Hence, every clause will be checked in order to compute the maximum number of satisfied clause. All in all, the maximum number of clauses satisfied by the assignment is 3 out of 4.

TABLE I
TRUTH TABLE FOR P

x	y	$(x \vee y)$	$(x \vee \neg y)$	$(\neg x \vee y)$	$(\neg x \vee y)$	P
1	-1	1	1	-1	1	-1
-1	1	1	-1	1	1	-1
-1	-1	-1	1	1	1	-1
1	1	1	1	1	-1	-1

III. Neuro- Searching Paradigm

Neuro-searching paradigm consists the algorithmic method in finding the solutions. Previously, Hopfield neural network alone has been utilized in doing logic programming. The usage of Hopfield neural network (HNN) in doing logic programming is proven effective when the number of neurons were small. In order to make a fair comparison between standalone Hopfield neural network and Hopfield neural network incorporated with genetic algorithm, we embedded exhaustive search technique to enhance the traditional HNN in doing MAXkSAT. In this paper, neuro-searching paradigms were used in hunting the maximum number of clauses for restricted maximum 2-satisfiability and restricted maximum 3-satisfiability problem.

A. Exhaustive Search (ES)

Exhaustive search (ES) algorithm can be demarcated as a local search technique for an element with a particular property among combinatorial forte such as permutations, combinations, logics, satisfiability or subsets of a set [32].

Roughly speaking, the ES algorithm will brutally hunt for the total potential clause, even if the search dimension was getting bigger and more complex [5]. Technically, the exhaustive search is the most primitive algorithm for checking the logic satisfaction. ES is theoretically simple to implement. In our exploration, we embedded ES to traditional Hopfield neural network in order to enhance the primitive solution checking by Hopfield neural network. The exhaustive search will facilitate traditional Hopfield neural network to check the satisfaction clause by clause in order to generate the maximum number of satisfied clause.

However, the main drawback of exhaustive search (ES) is the speed of the algorithm [35]. Subsequently, exhaustive search devours more computation time in searching for the maximum number of satisfied clauses completely [28]. In this paper, we will generate random bit strings and compute the number of satisfied clauses directly, clause by clause. It will be a huge possibility that the bit strings are not converging to global maxima during the first iteration of ES. Thus, the iterations will be repeated 100 times. The ES might look decent for the simpler case, but what would happen if we increase the number of clauses?

We will encounter with the complexity of the hybrid network when we attempted with more complex bit strings. Thus, the computation time will become very high if we increase the complexity of the hybrid network. Therefore, the computation complexity is represented as $O(2^n)$ [6]. For the ES algorithm, the satisfied assignment is gained after performing a 'trial and error' procedure exhaustively. Henceforth, the correct assignment will be stored into the Hopfield's artificial brain in the form of content addressable memory (CAM). Some related work on exhaustive search has been done by a few neural network practicioners such as Aiman & Asrar [27], Kaushik [28], and Zinovik *et al*. [32]. In this paper, we hybridized ES algorithm with the Hopfield neural network as a network based on logic programming to solve MAX-kSAT problems (HNN-MAX2SATES and HNN-MAX3SATES).

B. .Genetic Algorithm (GA)

Genetic algorithms are robust evolutionary paradigms that have attracted a prolific amount of research in optimization and maximization problem [46]. According to mathematicians' standpoint, the genetic algorithm is a staple computational paradigm inspired from the Darwin's theory of evolution, namely survival for the fittest model [29, 30]. For instance, every generation is represented by an array of bit strings similar to the chromosomes of DNA. In our case, we have a set of bit string that represent the interpretation of the MAX*k*SAT. This is motivated by the previous work by Aiman and Asrar [27] that highlighted on the genetic algorithm to solve randomized 3-SAT problem. On the contrary, the fundamental impetus of genetic algorithm is to find the bit string that maximize the number of satisfied clauses before we incorporate with Hopfield network. Specifically, the genetic algorithm in doing MAX*k*SAT consists of distinctive stages.

Stage 1: Initialization

In this stage, 100 random populations in the form of bit strings were initialized [27]. Each assignment consists of possible solution to randomized MAX*k*SAT problem.

Stage 2: Fitness evaluation

Next. all the bit string will undergo fitness evaluation. Each of the correct bit which result in satisfied MAX*k*SAT clause will be "awarded". During fitness evaluation, the number of satisfied clauses will represent the fitness of the chromosome (bit string). The fitness function is widely used as an objective function by a few notable works [46, 47, 48]. The objective function of the genetic algorithm is as follows:

$$f_{MAXkSAT} = \max[c_1(x) + c_2(x) + c_3(x)... + c_N(x)] \tag{5}$$

Where $c_1, c_2, c_3 ......c_N$ are the number of clause checked by genetic algorithm 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

During this stage, 10 candidate bit strings with the highest fitness will go to the next generation. The selected candidate bit strings will have the privilege to perform the crossover process.

Stage 4: Crossover

During crossover, bit strings will be chosen randomly and the exchange of information between two sub-structure of the bit strings occurred. Bit string. For example,

Before crossover

Bit string A = -1 1 1 1

Bit string B = 1 -1 1 -1

After crossover =

Bit string A = -1 -1 1 1

Bit string B = 1 1 1 -1

The location of crossover in a particular bit string is randomly defined since we want to maintain genetic diversity of the bit strings. Crossover usually increase the number of satisfied clause of the newly bit strings children. This feature helps the best bit string of the generation to survive and improve further. On top of that, the crossover operator imitates the biological amalgamation between two single-chromosomes (haploid) in organisms. After crossover, all the bit strings children undergo fitness evaluation in order to check their corresponding fitness.

Stage 5: Mutation

Mutation operator is the real game changer for genetic algorithm. Local maxima could occur during simulation. Local maxima occurred when the fitness of the bit string varies significantly to the expected maximum fitness. This will create non-improving solution of MAX*k*SAT clauses. In order to create "out of the blue" bit string, we utilized mutation. Mutation involve flipping the state of the bit string from 1 to -1 or -1 to 1 [27]. Position of the mutation in a particular bit string is random. For example,

Before mutation

Bit string A = -1 -1 1 1

After mutation

Bit string A = -1 1 1 1

The second position of the bit string was flipped from -1 to 1. In this case, different bit string was created after mutation.

Thus, we can calculate the fitness value for the newly formed bit strings [27, 36].

Genetic algorithm in doing MAX*k*SAT has been proven effective by previous researchers to avoid global maxima. If the bit string does not achieve the desired fitness (local maxima), the current bit string will improve further during the next generation via crossover and mutation. Most of the researchers set up to 100 to 1000 generations in order to improve the solution. Since we are dealing bipolar search which only involve 1 or -1, it will be easy for bit strings to converge to global maxima (Maximum fitness). In this paper, we hybridized GA algorithm with the Hopfield neural network as a network based on logic programming to solve MAX-*k*SAT problems (HNN-MAX2SATGA and HNN-MAX3SATGA).

Figure 1 shows the algorithm for this paradigm.

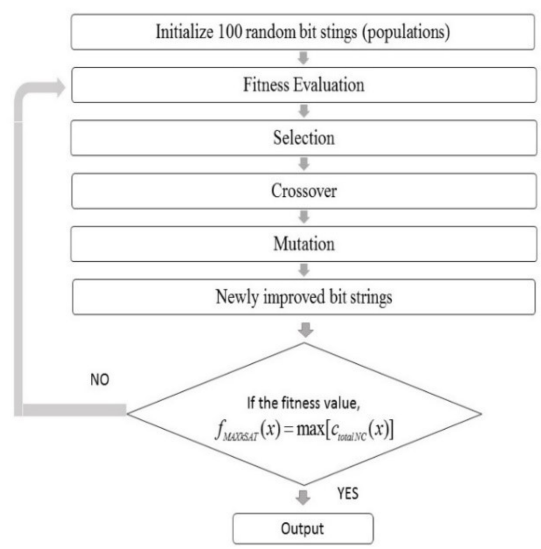


Fig. 1. Algorithm/flowchart for genetic algorithm.

IV. Neuro- Logic In Hopfield Neural Network

A. The Hopfield Neural Network

For many years, Hopfield model has been recognized as an effective optimization method [4]. Since the first application of Hopfield network to optimization problem, this approach has well drawn many attention towards various field such as computer network, pattern recognition and scheduling problem.

Theoretically, the model comprises of interconnected unit called neurons, forming a network. Computation in Hopfield network is executed by collections of interconnected neurons [4, 11]. Most of the literature suggest Hopfield network contains good properties including parallel execution for fast solutions to computationally intensive optimization problems with exceptionally good accuracy [9]. In this connection, we choose Hopfield network to do logic programming because it is well distributed, consist of CAM [12], smooth implementation and easy to blend with other algorithm.

The units in Hopfield nets are called binary threshold unit [13] which can only take bipolar values such as 1 and -1. The paramount definition for unit I's activation, a_i are given:

$$a_i = \begin{cases} 1 & if \sum_j W_{ij} S_j > \xi_i \\ -1 & Otherwise \end{cases} \tag{6}$$

Where W_{ij} is the connection strength from unit j to i . S_j is the state of unit j and ξ_i is the threshold of unit i . The network comprises of N recognized neurons, each is described by an Ising spin variable. The connection in Hopfield net contain no connection with itself $W_{ii} = W_{jj} = 0$. Thus it makes the Hopfield connections became symmetric or bidirectional [4, 14]. Neuron is basically bipolar $S_i \in \{1, -1\}$ thus it follows the dynamics $S_i \rightarrow \mathrm{sgn}(h_i)$ where h_i is the local field of the connection. When dealing with higher order connection, the local field modifies to

$$h_i = ...+\sum_j W_{ijk}^{(3)} S_j S_k + \sum_j W_{ij}^{(2)} S_j + W_i^{(1)} \tag{7}$$

Since the weight (connection strength) in Hopfield network is constantly symmetrical, the updating rule maintains as follows [15]:

$$S_i\left(t+1\right) = \mathrm{sgn}\left[h_i\left(t\right)\right] \tag{8}$$

The dynamic is to ensure the energy decrease monotonically which following the activation system. The generalized lypunov energy equation is as followed:

$$= -\frac{1}{3}\sum_i\sum_j\sum_k W_{ijk}^{(3)} S_i S_j S_k - \frac{1}{2}\sum_i\sum_j W_{ij}^{(2)} S_i S_j - \sum_i W_i^{(1)}{}_j \tag{9}$$

This energy function is significant because it establishes the degree of convergence of the network [16, 4]. The energy value obtained from the equation will be checked through and will be classified as global or local minimum energy. As it stands, the network is hunting for global minimum energy (correct solution) compared to local minimum energy (wrong solution). The process of obtaining global minimum energy always associated with how we define the weight of the network. In this work, we implemented Wan Abdullah's updating technique to obtain the weights for our network [3, 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 being carried out on Hopfield neural network. Wan Abdullah's method became the pioneer in weight extraction based on logical inconsistencies [17]. Truth values were assigned to each atoms. The minimized cost function can be created by maximizing the number of satisfied clauses.

For example, Consider the following MAX-2SAT and MAX-3SAT problem with α and ϕ

$$\alpha = \left(A \vee B\right) \wedge \left(A \vee \neg B\right) \wedge \left(\neg A \vee B\right) \wedge \left(\neg A \vee \neg B\right) \tag{10}$$

$$\begin{aligned} \phi = & (P \vee Q \vee R) \wedge (\neg P \vee Q \vee R) \wedge (P \vee \neg Q \vee R) \\ & \wedge (P \vee Q \vee \neg R) \wedge (\neg P \vee \neg Q \vee R) \wedge (\neg P \vee Q \vee \neg R) \\ & \wedge (P \vee \neg Q \vee \neg R) \wedge (\neg P \vee \neg Q \vee \neg R) \end{aligned} \tag{11}$$

Cost function f_{cost} for both equation (8) and (9) are as followed

$$\begin{aligned} f_{\mathrm{cost}\,\alpha} = & \frac{1}{2}\left(1-S_A\right)\frac{1}{2}\left(1-S_B\right) + \frac{1}{2}\left(1-S_A\right) + \frac{1}{2}\left(1+S_B\right) \\ & + \frac{1}{2}\left(1+S_A\right)\frac{1}{2}\left(1-S_B\right) + \frac{1}{2}\left(1+S_A\right)\frac{1}{2}\left(1+S_B\right) \end{aligned} \tag{12}$$

$$\begin{aligned} f_{\mathrm{cost}\phi} = & \frac{1}{2}(1-S_P)\frac{1}{2}(1-S_Q)\frac{1}{2}(1-S_R) + \frac{1}{2}(1+S_P)\frac{1}{2}(1-S_Q)\frac{1}{2}(1-S_R) \\ & + \frac{1}{2}(1-S_P)\frac{1}{2}(1+S_Q)\frac{1}{2}(1-S_R) + \frac{1}{2}(1-S_P)\frac{1}{2}(1-S_Q)\frac{1}{2}(1+S_R) \\ & + \frac{1}{2}(1+S_P)\frac{1}{2}(1+S_Q)\frac{1}{2}(1-S_R) + \frac{1}{2}(1+S_P)\frac{1}{2}(1-S_Q)\frac{1}{2}(1+S_R) \\ & + \frac{1}{2}(1-S_P)\frac{1}{2}(1+S_Q)\frac{1}{2}(1+S_R) + \frac{1}{2}(1+S_P)\frac{1}{2}(1+S_Q)\frac{1}{2}(1+S_R) \end{aligned} \tag{13}$$

By comparing equation (12), (13) with equation (9), we obtained synaptic weight for α and ϕ . The synaptic weights are shown in Table 2 and Table 3.

TABLE II

Synaptic Weight for α based on Wan Abdullah's Method

W	C_1	C_2	C_3	C_4
$W_A^{(1)}$	1/4	1/4	-1/4	-1/4
$W_B^{(1)}$	1/4	-1/4	1/4	-1/4
$W_{AB}^{(2)}$	-1/2	1/2	1/2	-1/2

TABLE III

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
$W_P^{(1)}$	1/8	-1/8	1/8	1/8	-1/8	-1/8	1/8	-1/8
$W_Q^{(1)}$	1/8	1/8	-1/8	1/8	-1/8	1/8	-1/8	-1/8
$W_R^{(1)}$	1/8	1/8	1/8	-1/8	1/8	-1/8	-1/8	-1/8
$W_{PQ}^{(2)}$	-1/8	1/8	1/8	-1/8	-1/8	1/8	1/8	-1/8
$W_{QR}^{(2)}$	-1/8	-1/8	1/8	1/8	1/8	1/8	-1/8	-1/8
$W_{PR}^{(2)}$	-1/8	1/8	-1/8	1/8	1/8	-1/8	1/8	-1/8
$W_{PQR}^{(3)}$	1/16	-1/16	-1/16	-1/16	1/16	1/16	1/16	-1/16

Normally, weight can be determined by using traditional Hebbian learning concept [18]. Sathasivam has shown that the weight obtained by using Wan Abdullah's method are similar due to clausal MAX-kSAT similarity. Although both method is expected to produce the similar weight, Wan Abdullah's method is proven to minimize the spurious minima produced by logic compared to Hebbian learning [17, 18].

C. Network Relaxation

The nature of the solution obtained by Hopfield network can affected by multiple factors. According to Zeng & Martinez [19], the firing and receiving information among neurons can influence the output of the network. In this case, network relaxation helps the network to exchange information efficiently. As the number of neuron increased, more interconnected neurons involved in firing and receiving information. Without proper relaxation mechanism network tend to produce more local minima solution. Since MAX-kSAT contain more clausal constrained, we applied Sathasivam's relaxation technique [14] to ensure the network relaxed to equilibrium states. Information exchange between neurons will be updated based on the following equation

$$\frac{dh_i^{new}}{dt} = R\frac{dh_i}{dt} \tag{14}$$

Where R denotes the relaxation rate and h_i refers to the local field of the network as listed in equation (7). In this case, we consider a constant relaxation R since it will improve the network relaxation compared to dynamic relaxation.

D. Hyperbolic Activation Function

Other than relaxation rate, the choice of activation function can affect the performance of the network. Traditional McCulloch-Pitts activation function is prone to few weaknesses such as computational burdening and lack of efficiency on producing desired result [20]. In order to get network's full potential, we utilized Hyperbolic tangent activation function. The Hyperbolic tangent activation function is written as follows:

$$g(h_i) = \frac{e^{h_i} - e^{-h_i}}{e^{h_i} + e^{-h_i}} \tag{15}$$

Where h_i refers to the local field of the network. The Hyperbolic tangent can act as an efficient squashing function for local field and produce a well-defined output (between 1 and -1). In addition, the usage of activation function is to avoid the network from collapse into a simple linear function [21].

V. Implementation

The simulations for HNN-MAXkSATGA and HNN-MAXkSATES were executed on Microsoft Visual C++ 2013 for Windows 10. Firstly, the restricted MAX-kSAT clauses were generated randomly. After that, the initial states for the neurons were initialized in the MAX-kSAT clauses. The network evolved until it reached the final state. Once the program had reached the final state, the neuron state was updated via equation (7). As soon as the network relaxed via equation (14), the final state obtained. Furthermore, if the state had remained unaffected for five runs, neurons achieved stable states. Hence, by permitting an ANN to evolve, sooner or later, shall lead to a stable state where the energy function obtained would not change further. Subsequently, the corresponding final energy for the stable state was calculated. If the difference between the final energy and the global minimum energy is within the tolerance value, the solution would be considered as a global solution. Both algorithms were repeated 100 times with 100 neuron combinations. The termination criteria for the final energy was 0.001. Sathasivam et al. [11] highlighted the fact that 0.001 was selected as the termination criteria because it could minimize the statistical errors. The analysis will involve the global minima ratio, ratio of satisfied clause, fitness landscape value, Hamming distance and computation time as the performance measure and indicator.

VI. Result and Discussion

A. Global Minima Ratio

Global minima ratio is defined as the ratio between the global solutions over total number of runs [18]. Each simulation will produce 10000 bit strings solutions. 0.9524 global minima ratio value shows 9524 bit strings are global minimum and 476 bit strings are local minimum.

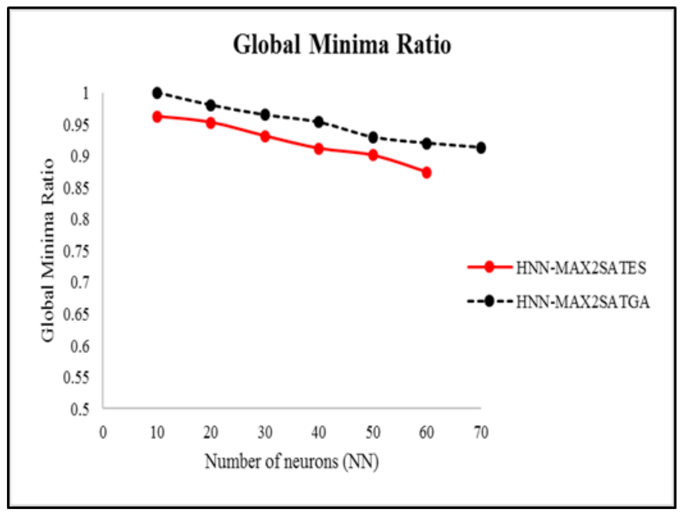


Fig. 2. Global minima ratio for HNN-MAX2SATES and HNN-MAX2SATGA.

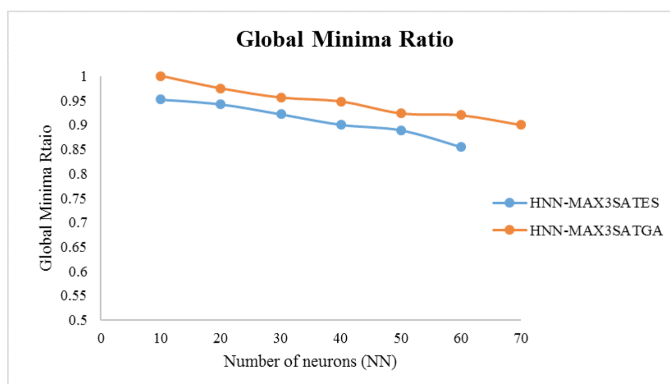


Fig. 3. Global minima ratio for HNN-MAX3SATES and HNN-MAX3SATGA.

Figure 2 and 3 represent the global minima ratio obtained by HNN-MAX2SATGA, HNN-MAX2SATES and HNN-MAX3SATGA, HNN-MAX3SATES. The efficiency of both networks can be determined by calculating their global minima ratio. According to Sathasivam [18], if the global minima ratio of the proposed network close to one, almost all solutions in the network reached global minimum energy (global solution). HNN-kMAXGA is able to recall more correct states compared to HNN-MAX-*k*SATES. The limit for HNN-MAX-*k*SATES is 60 neurons. After 60 neurons, the network in HNN-MAX-*k*SATES trapped in trial and error state and consume more time to find the solution. On contrary, HNN-*k*MAXGA is able to withstand number of neurons up to 70 neurons. Genetic algorithm is proven to reduce the complexity of the searching technique. Unsatisfied bit string can be improved through the crossover among the best offspring (highest fitness). The bit strings are expected to improve in term of fitness (satisfied clauses) as the number of generation increased. As a result, the bit string produced by genetic algorithm achieved global minima compared to traditional exhaustive search method. Mutation in genetic algorithm was added to reduce the chances for the bit string to reach local maxima (non-improving solution) [22, 23]. Besides, less complexity during searching can gives more time for the network to relax. Effective relaxation will reduce the number of suboptimal solution during the computation [14].

B. *Ratio of Satisfied Clauses*

Ratio of satisfied clauses can be defined as the total number of satisfied clauses over the total number of clauses [24]

TABLE IV

Ratio of Satisfied Clause

NN	HNN-MAX2SATES	HNN-MAX2SATGA	HNN-MAX3SATES	HNN-MAX3SATGA
10	0.745	0.750	0.800	0.875
20	0.723	0.750	0.787	0.850
30	0.716	0.743	0.764	0.834
40	0.700	0.739	0.732	0.827
50	0.688	0.732	0.700	0.815
60	0.632	0.728	0.688	0.799
70	-	0.722	-	0.774

NN=Number of neurons.

Table 4 depicts the ratio of the satisfied clauses over total clause obtained HNN-MAX2SATGA, HNN-MAX2SATES and HNN-MAX3SATGA, HNN-MAX3SATES. In maximum satisfiability problem, MAX-2SAT and MAX-3SAT clauses will never be fully satisfied. 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*SATGA is proven to obtain more satisfied clauses in MAX-*k*SAT compared to traditional exhaustive search method. As the number of neurons increased, the HNN-MAX*k*SATGA is still able to maintain the quality of the ratio. On the other hand, HNN-MAX*k*SATES will produce a lower ratio of satisfied clauses since most of the solution obtained trapped at suboptimal solution (local minima).

C. *Fitness Energy Landscape Value*

Fitness energy landscape value is associated with each point according to the pattern storing capability. Since Hopfield network concern about the ruggedness of the energy model, the fitness energy landscape must be taken into account. The fitness energy landscape value is based on Kauffman's model [25].

Figure 4 and 5 depicts the fitness energy landscape value obtained for HNN-MAX*k*SATGA and HNN-MAX*k*SATES. As observed, the difference in energy for HNN-MAX*k*SATGA is almost flat (zero) compared to HNN-MAX*k*SATES. MAX-*k*SAT clauses is always related to the ruggedness of the energy landscape. The more rugged the energy landscape, the harder it will to obtain good solution [26]. Since the complexity of the solution searching has been reduce drastically by genetic algorithm in HNN-MAX*k*SATGA, more relaxation time was added before the network retrieve the final states. As a result, HNN-MAX*k*SATGA has a greater capability to store MAX*k*SAT pattern compared to HNN-MAX*k*SATES. Hence, more global minimum energy produced.

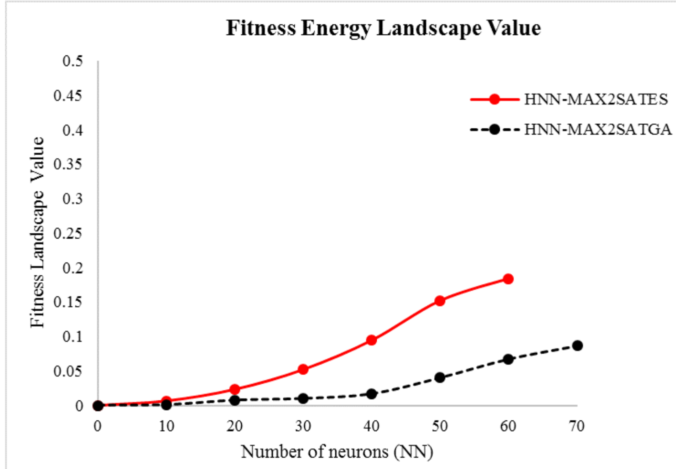


Fig. 4. Fitness energy landscape value for HNN-MAX2SATES and HNN-MAX2SATGA.

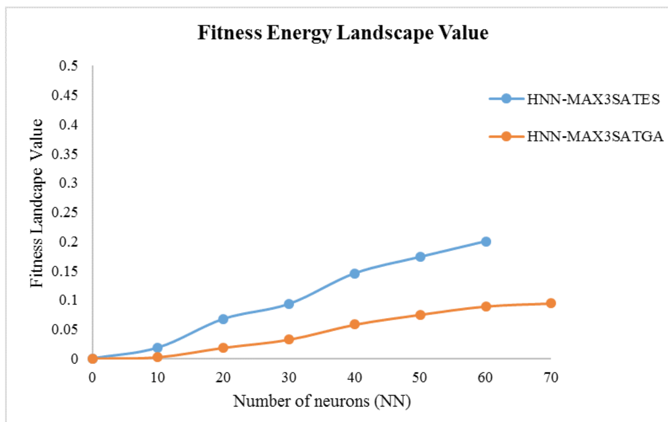


Fig. 5. Fitness energy landscape value for HNN-MAX3SATES and HNN-MAX3SATGA.

D. *Hamming Distance*

Hamming distance is demarcated as the number of positions at which the corresponding binary values between two strings are different. In our context, Hamming distance measures the closeness of bits between the stable state and the global state of the neurons upon relaxation process [14].

TABLE V

Global Hamming Distance

NN	HNN-MAX2SATES	HNN-MAX2SATGA	HNN-MAX3SATES	HNN-MAX3SATGA
10	0.00804	0.00215	0.00986	0.00382
20	0.01558	0.00624	0.01980	0.00845
30	0.02340	0.01002	0.02976	0.01367
40	0.03478	0.01555	0.03867	0.01822
50	0.04682	0.01930	0.04922	0.02138
60	0.06099	0.02889	0.07133	0.02990
70	-	0.03302	-	0.03673

NN=Number of neurons.

Table 5 portrays the obvious success of genetic algorithm compared to exhaustive search in generating the maximum satisfied clauses. According to Sathasivam [17], if the Hamming distance of the network close to zero, almost all outputs produced by the network are considered as global solutions. Based on table 4, Hamming distance for HNN-MAX*k*SATGA and HNN-MAX*k*SATES are close to zero.

This is due to the power of GA in ascertaining the satisfied clause, especially during the crossover stage where the clause was being improved by certain rate to achieve the highest fitness value. Additionally, HNN-MAX*k*SATGA would be able to recall the correct states that contributed to the lower hamming distance. Conversely, the exhaustive search algorithm emphasized the trial and error process during clause satisfaction process. When the complexity increased, HNN-MAX*k*SATES were able to sustain up to 60 neurons and HNN-MAX*k*SATGA with the limitation until 70 neurons. The main reason is due to the nature of exhaustive search that increased the computation burden to get correct neuron states. Hence, the ability to sustain huge number of neurons is due to the special ability of GA that reduces the computation burden in hunting the correct states.

E. Computation Time

The computation time is an important measure or indicator to analyze the performance of our proposed algorithm. According to our exploration context, the computation time can be delineated as the expanse of time for which our network was used to complete the whole computation process [6, 13]. The computation process involves the training and generating the maximum satisfied clauses via our proposed paradigm [14].

TABLE VI

Computation Time

NN	HNN-MAX2SATES	HNN-MAX2SATGA	HNN-MAX3SATES	HNN-MAX3SATGA
10	24	1	32	2
20	108	17	159	23
30	357	76	482	98
40	2880	280	3461	340
50	11452	633	13708	784
60	88562	1550	102855	1699
70	-	3124	-	3322

NN=Number of neurons.

Table 6 depicts the computation time for our proposed algorithms, HNN-MAX2SATGA and HNN-MAX3SATGA together with the conventional algorithm, HNN-MAX2SATES and HNN-MAX3SATES. A nearer look at the running time indicates as the network become more complex, more computation time are needed to generate the global solutions. 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 neuron increased, the computation time taken 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 6, HNN-MAX2SATGA and HNN-MAX3SATGA require less computation time compared to the other counterparts, HNN-MAX2SATES and HNN-MAX3SATES. 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 [14]. Additionally, the training process by using exhaustive search requires more computational time due to the trial and error process in hunting the maximum number of satisfied assignments. One of the important fact is for the maximum satisfiability problem, MAX-2SAT and MAX-3SAT clauses are never be satisfied 100%. On the contrary, when we implemented genetic algorithm, the computation time was faster due to the crossover and mutation process that speed up the training process. This is due to the fact that the unsatisfied bit string can be enhanced through the crossover among the finest offspring. The mutation process can avoid the bit string to achieve local minima. Hence, the bit string created by genetic algorithm achieved global minima swiftly compared to traditional exhaustive search method.

VII. Conclusion

Inspired by the Darwin's survival of the fittest theory together with biological genetic operators and engaging concept in artificial intelligence, a hybrid paradigm had been proposed. We had successfully develop a network by using genetic algorithm incorporated with Hopfield neural network in performing restricted maximum k-satisfiability logic programming (HNN-MAX*k*SATGA). The proposed model, later, was compared with a conventional technique; ES with Hopfield neural network (HNN-MAX*k*SATES). The work, reported in this paper, revealed decent performances of HNN-MAX*k*SATGA in terms of the global minima ratio, ratio of satisfied clause, Hamming distance, fitness landscape value and the compuatation time. According to the experimental results, the HNN-MAXkSATGA outperformed HNN-MAX*k*SATES in all of those measures. In addition, the proposed framework provides solid platform for evaluating various type of satisfiability problem. Our future work revolves on the robustness of other metaheuristic technique to solve restricted maximum k-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] D. Givry, Simon, J. Larrosa, P. Meseguer, and T. Schiex. "Solving Max-SAT as weighted CSP." In *Principles and Practice of Constraint Programming–CP 2003*, Springer Berlin Heidelberg, 2003, pp. 363-376.

[2] Papadimitriou, Christos, and M. Yannakakis. "Optimization, approximation, and complexity classes." In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, ACM, 1988, pp. 229-234.

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

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

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

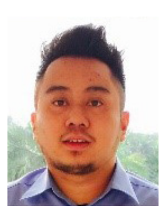
[6] S. Sathasivam, Learning in the Recurrent Hopfield Network, Proceedings of the Fifth International Conference on Computer Graphics, Imaging and Visualisation, 323-328, 2008.

[7] M. Peng, N. K. Gupta, & A. F. Armitage, An investigation into the improvement of local minima of the Hopfield network. *Neural networks*, 9(7), pp. 1241-1253, 1996.

[8] S. Salcedo-Sanz, C. Bousoño-Calzón, & A. R Figueiras-Vidal, A mixed

neural-genetic algorithm for the broadcast scheduling problem.*Wireless Communications, IEEE Transactions on*, vol. 2, no. 2, pp. 277-283, 2003.
[9] 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.
[10] C. Y. Ngo, & V. O. Li, Fixed channel assignment in cellular radio networks using a modified genetic algorithm. *Vehicular Technology, IEEE Transactions on*, vol. 47, no. 1, pp. 163-172, 1998.
[11] S. Sathasivam, P.F. Ng, N. Hamadneh, Developing agent based modelling for reverse analysis method, 6 (22), pp. 4281-4288, 2013.
[12] 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.
[13] S. Haykin, *Neural Networks: A Comprehensive Foundation*, New York: Macmillan College Publishing, 1999.
[14] S. Sathasivam, Upgrading Logic Programming in Hopfield Network, Sains Malaysiana, 39, 115-118, 2010.
[15] M. Velavan, Z. R. Yahya, M. N. A. Halif, & S. Sathasivam, (2015). Mean Field Theory in Doing Logic Programming Using Hopfield Network. *Modern Applied Science*, vol. 10, no. 1, p. 154, 2015.
[16] 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.
[17] W. A. T. W. Abdullah, Logic Programming on a Neural Network. Malaysian Journal of computer Science, 9 (1), 1-5, 1993.
[18] S. Sathasivam, "Learning Rules Comparison in Neuro-Symbolic Integration." *International Journal of Applied Physics and Mathematics*, vol.1, no.2, pp. 129, 2011.
[19] X. Zeng, and T. R. Martinez, "Improving the performance of the Hopfield network by using a relaxation rate." In *Artificial Neural Nets and Genetic Algorithms* Springer Vienna.s, 1999, pp. 67-72.
[20] M. S. Kasihmuddin, "Modifying Activation Function in Neuro Symbolic Integration," M.S. thesis, Dept. Mathematical Science., Universiti Sains Malaysia, Penang, Malaysia, 2014.
[21] M. Velavan, Boltzman Machine and Hyperbolic Activation Function in Higher Order Network, 9 (2), 140-146, 2014.
[22] 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
[23] C. K. Chong, M. S. Mohamad, S. Deris, M. S. Shamsir, Y. W. Choon, and L. E. Chai, Improved differential evolution algorithm for parameter estimation to improve the production of biochemical pathway. *International Journal of Interactive Multimedia and Artificial Intelligence*, no.5, 2012.
[24] E. Nudelman, K. Leyton-Brown, H. H. Hoos, A. Devkar, and Y. Shoham, Understanding random SAT: Beyond the clauses-to-variables ratio. In *Principles and Practice of Constraint Programming–CP 2004*, Springer Berlin Heidelberg, 2004, pp. 438-452.
[25] A. Imada and K. Araki, What does the landscape of a Hopfield associative memory look like? In *Evolutionary Programming VII*, 1998, Springer Berlin Heidelberg, pp. 647-65.
[26] S. Sathasivam, W. A. T. W. Abdullah, Flatness of the Energy Landscape for Horn Clauses. *SSJ*, vol 1, no. 2, pp. 2, 2008.
[27] Aiman, U. and Asrar, N., Genetic Algorithm Based Solution to SAT-3 Problem. *Journal of Computer Sciences and Applications*, *3*(2), pp. 33-39, 2015.
[28] Zinovik, I., Kroening, D. and Chebiryak, Y., Computing binary combinatorial gray codes via exhaustive search with SAT solvers. *Information Theory, IEEE Transactions on*, *54*(4), pp.1819-1823, 2008.
[29] Xie, J. and Dong, J., Heuristic genetic algorithms for general capacitated lot-sizing problems. *Computers & Mathematics with applications*, *44*(1), pp. 263-276, 2002.
[30] Salcedo-Sanz, S. and Yao, X., A hybrid Hopfield network-genetic algorithm approach for the terminal assignment problem. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, *34*(6), pp. 2343-2353, 2004.
[31] Pooranian, Z., Shojafar, M., Tavoli, R., Singhal, M. and Abraham, A., A hybrid metaheuristic algorithm for Job scheduling on computational grids. *Informatica*, *37*(2), pp.157, 2013.
[32] Kaushik, M., Comparative analysis of exhaustive search algorithm with ARPS algorithm for motion estimation. *International Journal of Applied Information Systems*, 1(6), pp. 16-19, 2012.
[33] Luke, S., Essentials of metaheuristics, 2nd edition. United States: Lulu, 2013.
[34] Mark, W.G., & Lee C.G., Routing in random multistage interconnections networks: Comparing exhaustive search, greedy, and neural network approaches. International Journal of Neural System, 2 (3), pp. 125-142, 1992.
[35] Hooker, J.N., Unifying local and exhaustive search. In L. Villasenor and A.I Martinez, editors. *Proceeding of ENC 2005- Sixth Mexican International Conference on Computer Science*, p. 237-243, 2005.
[36] Kumar, S., and Singh, M. P., Pattern recall analysis of the Hopfield neural network with a genetic algorithm. Computers & Mathematics with Applications, 60(4), pp. 1049-1057, 2010.
[37] Raman, V., Ravikumar, B., and Rao, S. S., A simplified NP-complete MAXSAT problem. Information Processing Letters. 65(1), pp. 1-6, 1998.
[38] Fernandez, W., Random 2-SAT: Result and Problems. Theoretical Computer Science. 265 (1): pp. 131-146, 2001.
[39] Pipatsrisawat, K., Palyan, A., Chavira, M., Choi, A., and Darwiche, A., Solving weighted Max-SAT problems in a reduced search space: A performance analysis. Journal on Satisfiability, Boolean Modeling and Computation. 4: pp. 191-217, 2008.
[40] Hansen, P. and Jaumard, B., Algorithms for the maximum satisfiability problem. *Computing*, *44*(4), pp.279-303, 1990.
[41] Broder, A.Z., Frieze, A.M. and Upfal, E., On the satisfiability and maximum satisfiability of random 3-CNF formulas. In *SODA* (Vol. 93, pp. 322-330), 1993.
[42] Madsen, B. A., and Rossmanith, P. Maximum exact satisfiability: NP-completeness proofs and exact algorithms. BRICS Report Series. 11(19), 2004.
[43] Borchers, B., & Furman, J., A two-phase exact algorithm for MAX-SAT and weighted MAX-SAT problems. Journal of Combinatorial Optimization, 2(4), pp. 299-306, 1998.
[44] Layeb, A., Deneche, A. H., and Meshoul, S., A new artificial immune system for solving the maximum satisfiability problem. Trends in Applied Intelligent Systems, pp. 136-142, 2010.
[45] Kowalski, R. A., The early years of logic programming. Communications of the ACM. 31(1): pp. 38-43, 1988.
[46] Gottlieb J., Marchiori E., & Rossi C. Evolutionary algorithms for the satisfiability problem. Evolutionary Computation. 2002;10(1):35-50.
[47] Rossi, C., Marchiori, E., & Kok, J. N. (2000, March). An adaptive evolutionary algorithm for the satisfiability problem. In Proceedings of the 2000 ACM symposium on Applied computing-Volume 1 (pp. 463-469). ACM.
[48] Marchiori, E, & Rossi, C. (1999, July). A flipping genetic algorithm for hard 3-SAT problems. In Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 1 (pp. 393-400). Morgan Kaufmann Publishers Inc.
[49] J. Singha, & R. H. Laskar, (2016). Hand gesture recognition using two-level speed normalization, feature selection and classifier fusion. *Multimedia Systems*, pp. 1-16.

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, constrained optimization, neural network and logic programming.



Mohd Asyraf Mansor was born in Sarawak, Malaysia in 1990. He 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.

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.