Journal of Informatics and Mathematical Sciences
Vol. 11, No. 1, pp. 81–98, 2019
ISSN 0975-5748 (online); 0974-875X (print)
Published by RGN Publications
DOI: 10.26713/jims.v11i1.1062

http://www.rgnpublications.com

Research Article

Modified Artificial Immune System Algorithm with Elliot Hopfield Neural Network For 3-Satisfiability Programming

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

School of Distance Education,UniversitiSains Malaysia, 11800, Pulau Pinang, Malaysia
***Corresponding author:** asyrafman@usm.my

Abstract. The modified artificial immune system algorithm, hybridized with the Elliot Hopfield Neural Network, is proposed in doing the 3-Satisfiability programming (EHNN-3SATAIS). The main impetus of this paper is to investigate the effectiveness and capability of the proposed approach in the 3-Satisfiability programming with different complexities based on the number of neurons. The performance analysis of the proposed technique is assessed by integrating the comprehensive simulation via Dev C++ Version 5.11 and compared with the state-of-the-art Elliot Hopfield networks. The Elliot Hopfield network is to be incorporated into the exhaustive search (EHNN-3SATES) and genetic algorithm (EHNN-3SATGA). The simulated results depicted the performance of the paradigms in terms of mean absolute error (MAE), Schwarz Bayesian Criterion (SBC) and CPU Time. The proposed approach, EHNN-3SATAIS outperformed its other two conventional counterparts in term of accuracy, sensitivity, and robustness during the simulation. Hence, the simulation has proven that the modified artificial immune system algorithm complied effectively in tandem with the Elliot Hopfield neural network in doing the 3-Satisfiability logic programming.

Keywords. Artificial immune system algorithm; Elliot Hopfield neural network; Elliot symmetric activation function; Satisfiability logic; Genetic algorithm

MSC. 62M45; 78M32

Received: August 13, 2018 **Accepted:** October 10, 2018

1. Introduction

The research on 3-Satisfiability (3-SAT) problem is flourishing even though it was theoretically considered as a classical NP-hard problem as explained by Johnson [15]. Fundamentally, the 3-SAT problem is demarcated as a propositional logic formula represented in a conjunctive normal form (CNF) with an array of clauses comprising strictly 3 literals per clause [18]. Consequently, a 3-SAT problem permits dual choices of the representation of each variable, given as 1 or -1. Furthermore, the 3-SAT problem is a class of non-deterministic problem and widely addressed as a constraint satisfaction problem (CSP). In 3-SAT logic programming, we are required to hunt for the best 3-SAT interpretation that will converge to global minimum solution. The major problem arises in computing the 3-SAT problem is the complexity. To address that problem, there are a plethora of works on the approach in solving 3-SAT problem ranging from the exact to the approximation techniques. For instance, Bünz and Lamm [2] proposed a classifier for randomized 3-SAT data sets by incorporating Graph neural network (GNN). The result recorded from the work was acceptable with the accuracy between 65% to 71%. The core motivation of 3-SAT logic programming is enthused by the development of Horn Satisfiability (HORN-SAT) inaugurated by Sathasivam [4]. Since the complexity of 3-SAT evolves exponentially, a robust training method based on the metaheuristic paradigm will be proposed.

The emergence of HNN has started in 1982, where a neuroscientist proposed a model of artificial neural network (ANN), namely Hopfield neural network (HNN) [13,28]. Thus, Hopfield [13] introduced an associative network that propelled up the field of ANN in solving various problems. Theoretically, the HNN was utilized to solve many class of optimization and constraint satisfaction problems [7,12]. The architecture of HNN includes of a simple recurrent network that has awell-organized associative memory and impersonates the biological brain [20,21]. Wen *et al*. [37] proposed that the HNN is one major neural networks specialized and crafted for solving the constraint optimization or mathematical programming problems. The power of HNN is that the architecture can be applied in verification of an electronic circuit, probably on a very large-scale integration circuit [16,28]. Pinkas [24] elaborated that the structure of HNN minimizes Lyapunov energy by utilizing the physical Ising spin of the neuron states. Pinkas [24] and Wan Abdullah [36] described a bi-directional mapping between logic and energy function of symmetric neural network. Thus, the HNN can be utilized as a black-box model in solving the any variant of satisfiability logic program.

The competencies of energy optimization in HNN has motivated a prolific number of researchers to hybridize and incorporate the idea of logic programming in HNN. For instance, the state-of-the-art hybrid HNN models were developed by Sathasivam and Wan Abdullah [27] and Kasihmuddin and Sathasivam [16]. Henceforth, Velavan *et al*. [35] describes the flexibility of HNN to integrate with the searching algorithm such as Mean Field theory. Furthermore, the work by Zhang *et al*. [41] has proposed the classification by utilizing the Hopfield associative memories for Chernoff face image recognition. As a result, the work reported the welding quality of Chernoff face image had been successfully classified even though it was under abnormal welding conditions. In this paper, Elliot function proposed by Sibi *et al*. [31] will be embedded to the Hopfield neural network as an Elliot Hopfield neural network (EHNN).

The works on the artificial immune system (AIS) algorithm has been motivated by the capability and robustness in performing some computation. To begin with, AIS is a class of evolutionary algorithm and metaheuristic method, inspired by the mechanism in the immune system based on the work of Dasgupta *et al*. [5]. The capability of AIS as a searching and accelerating paradigm has benefited the researcher to solve numerous constraint satisfaction problems and real-life data [19]. The studies on AIS can be divided into 3 variants such as immune clonal selection, immune networks, and negative selection of the B-cells [32]. The focus on AIS revolves on the learning and memory. In fact, AIS will be deployed in the learning phase of EHNN in doing 3-SAT logic programming. A prolific amount of works on artificial immune systems from the revolution research by Farmer *et al*. [6] has changed the AIS into a vigorous metaheuristic model to solve numerous problems ranging from simulated to the real-life problem. Besides, the AIS inaugurates the idea that only those B-cells can identify the antigen proliferate; thus, it is being nominated against those that do not [33]. Moreover, the AIS paradigm proposed in this work is basically impersonates the clonal selection process as in biological immune system. Hence, the binary AIS is proposed by modifying the standard AIS by Layeb [19] in the way of affinity computation and hybridization with Elliot Hopfield neural network (EHNN). To evaluate the performance of modified AIS, we compared with the standard model of genetic algorithm (GA) and exhautive search (ES).

This paper has been organized as follows. In Section 2, the background and fundamental formulation of 3-Satisfiability (3-SAT) is highlighted. Section 3 emphasizes the formulation of Hopfield neural network with Elliot activation function. Then, Section 4 discusses the learning method via the modified artificial immune system. Meanwhile, in Section 5 the genetic algorithm and exhaustive search are vividly introduced. Furthermore, Section 6 and 7 discuss the performance evaluation metrics; the implementation of the EHNN networks are also discussed. Section 8 and 9 encompass the full results, comprehensive discussions, and conclusion.

2. 3-Satisfiability (3-SAT) Logic Programming

3-Satisfiability (3-SAT) is a class of constraint satisfaction problem, widely used in the field of mathematics, computer sciences and physics. The main task is to enumerate a set of interpretations consisting of binary or bipolar values that make the 3-SAT formula to be satisfied. In fact, there are numerous constraint satisfaction problems dealing with non-Boolean quantities can also be converted into the 3-SAT problem [14]. Therefore, 3-SAT logical rule will be entrenched to a network to perform the computation or data processing. Ideally, 3-SAT is a variant of Boolean logic consists of three literals per clause with the bipolar values of either 1 or -1.

In this paper, 3-SAT logic program will be embedded to the EHNN with different learning algorithm. An example of a 3-SAT logical rule is given:

$$P = (\overline{A} \vee B \vee \bar{C}) \wedge (D \vee \overline{E} \vee F) \wedge (G \vee H \vee \overline{I}). \tag{1}$$

Equation (1) shows a 3-SAT formula, P with 3 clauses connected by AND operators. Each clause containing 3 different literals connected by OR operators. Henceforth, the number of neurons corresponds to the total number of literals involved in a particular 3-SAT logic programming.

Therefore, the generalized form of 3-SAT formula is shown in equation (2).

$$P = \wedge_{i=1}^{n} T_i \tag{2}$$

whereby T symbolizes the clause and i denotes the number of clauses involved in 3-SAT formula P.

The four important connotations of 3-SAT logic formula are simplified:

(i) The variables denote as $x_1, x_2, \ldots, x_n$ for a particular clause, where the variables is strictly equal to $n = 3$ for 3-SAT logic.

(ii) An array of m clauses in a 3-SAT formula, Z. $\exists\ m : Z = c_1 \wedge c_2 \wedge \ldots \wedge c_m$.

(iii) An array of literals in respective clauses. As for 3-SAT formula, we are strictly set 3 literals per individual clause. Next, each clause $c_k$contained of literals connected by the logical operator OR. $\forall\ 1 \le k \le m : c_k = (l_{k,1} \vee l_{k,2} \vee l_{k,3})$.

(iv) An array of literals which are the negation of the variable or the variable itself. $\forall\ 1 \le k \le m,\ 1 \le i \le 3 : l_{k,i} = x_p$ or $l_{k,i} = \neg x_p$ for $1 \le p \le n$.

3. Elliot Hopfield Neural Network (EHNN)

Hopfield neural network (HNN) is a variant of recurrent neural network, widely applied with logic programming and data mining. It comprises the interconnected neurons that forming a network that impersonates the human biological brain system. It is often known as a black-box model but comprises astounding properties such as parallel execution, faster computation, exceptional stability, and good associative memory system in the form of content addressable memory (CAM) (Sathasivam and Wan Abdullah [27]).

In this research, we consider discrete HNN to hybridized with Elliot activation function to form Elliot Hopfield neural network (EHNN). As we consider the discrete EHNN, the interconnected neurons will be assigned to their respective states, $S_i \in \{-1, 1\}$, where their bipolar activation units are either 1 or -1 [28].

Equation (3) shows the neuron state activation units in a more precise form.

$$S_i = \begin{cases} 1 & \text{if } \sum_j W_{ij} S_j > \xi_i \\ -1 & \text{otherwise}, \end{cases} \tag{3}$$

where $W_{ij}$is the neuron synaptic weight from unit j to i. Thus, the state S_j denotes the state of a neuron corresponds to unit neuron j. On the other hand, ξ_i depicts the threshold unit of the neuron i. Therefore, the connection in the Hopfield network naturally has no connection with itself $W_{ii} = 0$ or $W_{jj} = 0$. Moreover, the EHNN comprises of an array of symmetrical weights. Notionally, the Hopfield neural network works asynchronously with each neuron by deterministically updating their state. The system consists of N formal neurons, each is described by an Ising variable. The neurons are bipolar $S_i \in \{1, -1\}$ following the dynamics $S_i \rightarrow sgn(h_i)$ whereby the local field denotes as h_i. The local field function of HNN is given as follows:

$$h_i = \sum_k \sum_j W_{ijk} S_j S_k + \sum_j W_{ij} S_j + W_i . \tag{4}$$

Since the synaptic weight in HNN is always symmetrical and without zero diagonal, the updating rule is maintained as in equation (5):

$$S_i(t+1) = sgn[h_i(t)], \tag{5}$$

whereby *sgn* is a signum function. According to equation (4) the updating rule decreases monotonically with the dynamics. In standard HNN, the state of the network changes from a initial state to a final state where it is a global minimum of the Lyapunov function [24, 35].

$$E_{Lyapunov} = \ldots - \frac{1}{2} \sum_i \sum_j W_{ij} S_i S_j - \sum_i \xi_i S_j . \tag{6}$$

In HNN, the energy function is a dynamic filtering paradigm of global minima solution or local minima solution. Therefore, equation (6) is recrafted into 3-SAT Lyapunov energy that accommodates more complex order of neurons as shown in equation (7).

$$E = -\frac{1}{3} \sum_i \sum_j \sum_k W_{ijk} S_i S_j S_k - \frac{1}{2} \sum_i \sum_j W_{ij} S_i S_j - \sum_i W_i S_i . \tag{7}$$

Additionally, the energy function verifies the degree of convergence of the HNN in doing 3-SAT logic programming. Sibi *et al*. [21] defines the Elliot symmetric activation function as a variant of transfer function with sophisticated and swiftness approximation. Hence, this Elliot symmetric activation function is a simple function without connection of exponential or trigonometric functions. Consequently, the following equation manifests the output classification via the Elliot activation function proposed by Mansor *et al*. [22].

$$g(h_i) = \begin{cases} 1 & f(h_i) \geq 0 \\ -1 & \text{otherwise} \end{cases} \tag{8}$$

whereby

$$f(h_i) = \frac{h_i}{1 + |h_i|}, \tag{9}$$

where h_i represents the local field as in equation (4) of the EHNN network. The hybridization of Elliot symmetric activation and HNN has formed Elliot Hopfield neural network (EHNN). Hence, in order to bridge the 3-SAT and EHNN, we require a systematic and robust learning algorithm such as metaheuristic algorithm. In this paper, we proposed a newly artificial immune system (AIS) algorithm and the performance the learning method with EHNN will be compared with genetic algorithm (GA) and exhaustive search (ES).

4. Modified Artificial Immune System Algorithm

The building block of modified artificial immune system algorithm is based on the clonal selection theory inspired by the maneuver of biological immune system in fighting the pathogen. In this paper, a newly modified artificial immune system with Elliot Hopfield neural network is proposed in 3-Satisfiability logic programming. A prolific amount of works on artificial immune system algorithms ranging from the constraint optimization, real-life applications and numerical optimization have been the motivation of applying that algorithm in doing 3-SAT logic programming.

The work by Timmis and Neal [32] discussed the comprehensive implementation of restricted binary AIS in processing numerous real-life data mining. The modified AIS in this work is based on Layeb [19] proposed affinity-based function for artificial immune system (AIS) algorithm with the tabu search approach. Hence, we formulated the cost function as the affinity function of the B-cells (solutions) to comply with discrete 3-SAT logic programming.

The AIS algorithm involves a few stages. Firstly, the 3-SAT interpretations were illustrated as the B-cells in an immune system. For instance, if an antigen or pathogen attacks the organism, the antibodies (B-cells) that identify these antigens will survive. Thus, we will be able to calculate the fitness of the B-cells via affinity function. Moreover, the affinity of every B-cells was computed based on the number of satisfied clauses in the 3-SAT formula. During the selection process, a particular number of B-cells with better affinity value will be selected. Pursuing that, the selected B-cells could be cloned and duplicated by utilizing the classical roulette wheel paradigm by Kumar [17]. The remarkable operator in AIS is the somatic hypermutation will improve the B-cells. Finally, the newly improved B-cells with maximum affinity will be stored into the memory cells. In this research, the B-cells refer to the interpretation and the memory cells denote content addressable memory (CAM) in EHNN.

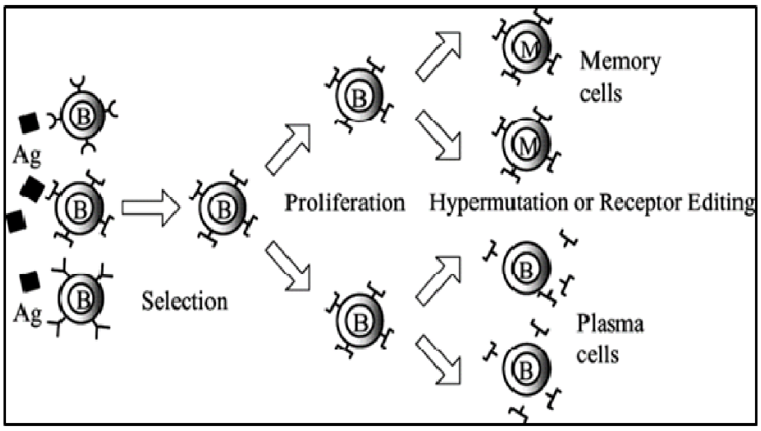

Figure 1. Illustration of the biological artificial immune system process

The implementation of the modified artificial immune system algorithm is summarized as follows:

Stage 1: B-cells Initialization

Initialize the population of 100 B-cells (interpretations) in the system. The ideal initial B-cells complies with the work of Layeb [19].

Stage 2: Affinity Evaluation

Compute the affinity of every respective B-cells from the populations in Stage 1. Affinity measure refers to the entire number of satisfied clauses per interpretation for a particular 3-SAT formula.

Next, the modification is made in terms of the affinity evaluation to comply with and fulfill our 3-SAT programming problem. Hence, the affinity can be calculated as follows:

$$f_{affinity} = c_1(x) + c_2(x) + c_3(x) \ldots + c_{total\,NC}(x), \tag{10}$$

where c represents the number of clauses during learning phase and NC symbolizes the number of clauses embedded into the 3-SAT formula. Similarly, equation (10) will assess the affinity of the individual B-cells (interpretation) before proceeding to the next operator.

Stage 3: Selection

Select the best 5 B-cells with the top maximum affinity. Next, the selected B-cells will be chosen to undergo the cloning operator to further diversify the population to attain better affinity.

Stage 4: Cloning

The cloning starts with the replicating the selected B-cells by using the classical roulette wheel selection to the system [17]. After that, the B-cells will evolve until the production of 200 B-cells. The number of cloned B-cells complies with the work of [19, 32]. In addition, the number of cloned B-cells can be computed as follows:

$$\begin{pmatrix} \text{The number of} \\ \text{clone allowed} \end{pmatrix} = \frac{affinity_i}{\sum affinity} \times \beta, \tag{11}$$

where $affinity_i$ refers to the initial affinity observed by the system and β is the number of population clone that is introduced to the system. Due to good agreement with the work of Layeb [19], $\beta = 200$ is selected as a fixed parameter.

Stage 5: Normalization

Normalization is the affinity standardization process before further improvement is made during the Hypermutation operator. The normalization of the B-cells is also called maturation of the immune response. The standard formulation for B-cells normalization is given in equation (12).

$$affinity\,N_i = \frac{affinity_i - \min affinity}{\max affinity - \min affinity}, \tag{12}$$

where $affinityN_i$ refers to the normalized affinity.

Stage 6: Somatic Hypermutation

Somatic hypermutation operator will enhance B-cells to achieve the maximum affinity that will converge towards a feasible solution. Theoretically, the somatic hypermutation is more powerful than the normal mutation due to the concept of the nearer the match, the more disruptive the mutation. Specifically, the B-cells (interpretation) flipping will enhance the solution to obtain the maximum affinity values [33]. That mechanism works by randomly flipping the one or more string of the B-cells from -1 to 1 or vice versa. The number of mutation can be determined by equation (13).

$$\begin{pmatrix} \text{Number of} \\ \text{Mutation } (Nb) \end{pmatrix} = \left(\frac{1}{\text{Number of variable}}\right)(affinity\,N_i) + (1 - affinity\,N_i)(0.01). \tag{13}$$

Next, the affinity of the new generation of B-cells will be calculated. If the value of affinity is equivalent to the number of clauses, the solution will be selected. Consequently, if a particular of B-cells fails to attain the required affinity, Stage 1 until Stage 6 will be repeated until the maximum affinity is achieved.

In this research, any satisfied interpretation (B-cells with maximum affinity) will be kept in CAM to be retrieved by the Elliot Hopfield neural network.

5. Genetic Algorithm and Exhaustive Search

In this section, the fundamental concept and procedure of both standard searching techniques are discussed.

A. Genetic Algorithm

In this work, the standard genetic algorithm procedure will be applied to our Elliot Hopfield neural network. Genetic algorithm is a population-based evolutionary searching paradigm enthused by the notable Darwin's theory of evolution [1, 8]. Thus, the interpretation (bit strings) is represented by an array of complicated biological units, namely chromosomes in our DNA. The inaugural work by Holland demonstrates the effectiveness of GA in finding the feasible solutions of computational intractable problem with high complexity.

Specifically, GA is a standard metaheuristic paradigm, commonly applied as the indicator of the capability of the proposed computational network. Hence, the standard GA proposed by Aiman and Asrar [1] will be integrated into the Hopfield neural network with Elliot activation function.

The process in GA involves five main stages:

Stage 1: Chromosomes Initialization

Randomize 100 chromosomes as the bipolar interpretations. Therefore, the chromosomes will represent the possible interpretation for EHNN-3SAT.

Stage 2: Fitness evaluation

Compute the fitness of the respective chromosomes according to the number of satisfied clauses in each of the interpretation. The maximum fitness implies the effectiveness of the learning process.

Stage 3: Selection stage

10 candidate chromosomes that have recorded the maximum fitness among the 100 chromosomes will continue to the following generation and stage of GA. After that, the selected chromosomes will be facing crossover procedure to enhance the fitness and variability.

Stage 4: Crossover Operator

The crossover process involves the main genetic alteration process in GA. Furthermore, the genetic exchange of information among two sub-structure of the chromosomes (bit strings) occurs here. As an illustration:

Before Crossover
Chromosome X = 1 -1 -1 1 1 -1
Chromosome Y = 1 1 1 1 1 1

Post Crossover
Chromosome X = 1 -1 -1 1 1 1
Chromosome Y = 1 1 1 1 1 -1

The crossover point in individual chromosomes is erratically described to retain the genetic diversity of the chromosomes [1]. Crossover usually increases the number of the satisfied clause of the newly chromosome pairs. This benefits the best chromosome of the generation to survive and undergo enhancement during the mutation operator.

Stage 5: Mutation

Henceforth, the non-improving interpretations will be improved during the mutation. Theoretically, the mutation in GA involves flipping of the state of the bit string from 1 to -1 or vice versa. Hence, the chromosome point of the mutation is set to be random. As an illustration:

Pre-mutation operator
Chromosome Z = -1 1 1 -1 -1 -1

Post mutation operator
Chromosome Z = -1 1 1 -1 1 -1

Based on the illustration, the fifth position of the chromosome was flipped instantaneously from -1 to 1. Consequently, the fittest chromosome will be produced after mutation operator. Finally, the fitness for the newly produced chromosomes will be assessed and calculated. If the fitness value still has not reach the maximum fitness, the first stage will be repeated.

B. Exhaustive Search

Exhaustive search (ES) algorithm is commonly used in training the Hopfield neural network. It involves the process of enumeration and brutal search for the satisfied interpretation in logic programming [9]. The searching process will hunt for the feasible interpretation even if the search space or dimension is evolving due to complexity [34]. In layman's term, ES is basically the trial and error searching paradigm. In this work, the correct interpretation will be deposited in Content Addressable Memory (CAM) before undergoing a training process. The ES algorithm has been explored in a plethora of works, generally in constraint satisfaction and constraint optimization problem. Hence, the objective function of ES is denoted as follows:

$$\max\{f_{ES}\}. \tag{14}$$

In this research, the ES algorithm is simplified into three main stages:

Stage 1: String Initialization

Initialize and enumerate the interpretations according to the number of literals and clauses.

Stage 2: Fitness Evaluation

Evaluate and calculate the fitness of the interpretation by utilizing equation (15).

$$f_{Exhaustive\_Search} = c_1(x) + c_2(x) + c_3(x) \ldots + c_{total\,NC}(x), \tag{15}$$

where NC indicates the clausal number and c signifies to the clauses in the interpretation.

Stage 3: Selection

The interpretation with the maximum fitness will be selected as an output and stored into EHNN. Then, the non-improving interpretation is enumerated by reimplementing Stage 1.

6. Performance Evaluation Metrics

To assess the capability of EHNN-3SATAIS, EHNN-3SATGA and EHNN-3SATES, 3 performance evaluation metrics were considered. The goodness of fit measure, known as the mean absolute error (MAE) was selected as accuracy indicator. Then, the model selection and robustness indicators namely, Schwarz Bayesian Criterion (SBC) and CPU Time were also applied.

A. Mean Absolute Error (MAE)

The MAE is widely applied because the capability in error estimation [39]. Willmott and Matsuura [38] recognized the MAE as one of reliable performance measures to detect the accumulation of uniformly distributed error. According to Pwasong and Sathasivam [26], the MAE is calculated by taking the absolute value of the difference between the estimated forecast and the actual value over a number of iterations. Typically, the MAE is minimally prone to outliers as asserted by Shi *et al*. [30]. The standard MAE formula by Chai and Drexler [3] is given as follows:

$$\mathrm{MAE} = \sum_{i=1}^{n} \frac{1}{n} \left|(f_{\max} - f_i)\right| . \tag{16}$$

where $f_{\max}$ represents the maximum fitness supposed to be measured from the network and f_i denotes the fitness obtained after execution.

B. Schwarz Bayesian Criterion (SBC)

Schwarz Bayesian Criterion (SBC) is used extensively in model selection and appraisal as introduced by Schwarz [29]. Specifically, it is coherent, reliable, unbiased and precise indicator for choosing the best computational model. Furthermore, it is sensible to formulate the number of parameters that are as small as possible. In term of evaluation, the best model corresponds to the minimal value of SBC after the execution. The main motivation of using SBC is based on Hamadneh [10] that utilized the SBC as the indicator to determine the ideal logic programming model. Thus, the general formula of SBC as asserted by Hamadneh [10] is as follows:

$$SBC = n\ln(MSE) + pa\ln(n), \tag{17}$$

whereby n refers to total number of iterations, pa is the magnitude of free parameters and mean square error (MSE). Since the Hopfield neural network is free from any pa, the equation has been recrafted for our case:

$$SBC = n\ln(MSE), \tag{18}$$

where the MSE is measured in calculating SBC and n depicts the number of iterations during the simulation. Hence, the formula of MSE is given as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(f_{\max} - f_i)^2 , \tag{19}$$

where $f_{\max}$ refers to the maximum fitness supposed to be and f_i denotes the fitness recorded during each of the execution.

C. CPU Time

The CPU time is commonly used as indicator of robustness of a particular computational model. Generally, CPU time refers to the total expanse time recorded by a model to execute the whole simulation. In this research, the SI unit of second will be used as the unit of CPU Time. Moreover, the CPU Time is utilized to evaluate the robustness of EHNN-3SATES, EHNN-3SATGA and EHNN-3SATAIS. The formula of CPU Time is given as follows:

$$\text{CPU Time } (s) = \text{Training Time} + \text{Retrieval Time}. \tag{20}$$

The CPU time is considered as the robustness indicator of the model [35, 40].

7. Implementation

The comprehensive computer simulations for EHNN-3SATAIS, EHNN-3SATGA, and EHNN-3SATES were performed by using Dev C++ Version 5.11 as a platform and carried out on Windows 10.1, Intel Core i7, 2.7 GHz processor with 8GB RAM.

Step 1

Translate the 3-SAT formula containing different number of clauses into a Boolean algebra form. Define inconsistencies of the formula.

Step 2

Allocate neuron to every literal of 3-SAT clauses. Next, check inconsistency of the 3-SAT logic and formulate the cost function of 3-SAT logic formula, E_p.

Step 3

Compute the cost function of the network. For instance, the building blocks of the cost function are given as $Z = \frac{1}{2}(1+S_Z)$ and $\overline{Z} = \frac{1}{2}(1-S_Z)$. The state of the neuron during training with EHNN network will become true if $S_Z = 1$ and falsified if $S_Z = -1$.

Step 4

Check clause satisfaction by using EHNN-3SATES, EHNN-3SATGA or EHNN-3SATAIS until the network converge to $E_p = 0$. Then, the satisfied interpretation will be deposited as content addressable memory (CAM) in Elliot Hopfield neural network to be used during retrieval phase.

Step 5

By equating the cost function,E_p with energy function, E to obtain the magnitudes of the connection weights of the network.

Step 6

Compute the expected global minimum energy that is supposed to be by using equation (7).

Step 7

Determine the respective local field, h_i for all neurons by implementing equation (4).

Step 8

Update local field via Elliot activation function by using equation (8) or equation (9).

Step 9

Compute the corresponding final energy based on the final states updated by using equation (7). According to Sathasivam [28], $Tol = 0.001$ is considered as a standard termination criterion for Lyapunov energy function to systematically classify the minimum energy. Henceforth, each execution considers 100 trials with 100 neuron combinations to reduce the statistical error per execution. Thus, the simulations are repeated by using a different number of neuron (NN) until $NN = 180$.

8. Results and Discussion

The comparative analysis of EHNN-3SATAIS, EHNN-3SATGA, and EHNN-3SATES based on 3 performance valuation metrics, namely MAE, SBC and CPU time. The results are portrayed in Figure 2 until Figure 4, respectively. The results were attained from the computer simulation via Dev C++ Version 5.11 from $NN = 9$ until $NN = 180$.

A. Mean Absolute Error (MAE)

The values of MAE for EHNN-3SATASI, EHNN-3SATGA and EHNN-3SATES were recorded and analyzed to assess the performance of these models.

Figure 2 manifests the mean absolute error (MAE) recorded for EHNN-3SATES, EHNN-3SATGA, and EHNN-3SATAIS correspondingly after a single execution with a different number of neuron (NN). Based on MAE recorded from $NN = 9$ until $NN = 180$, EHNN-3SATAIS outclassed the EHNN-3SATGA and EHNN-3SATES during the learning phase. Generally, the results demonstrate the magnitudes of MAE for EHNN-3SATAIS are significantly smaller than the other two counterparts even though the number of neurons (NN) have increased. This reveals that the interpretations generated from EHNN-3SATAIS are less deviated from the feasible solutions.

When modified AIS algorithm was applied during learning process, the probability of generating correct interpretations will be particularly higher. The effective global and local search techniques in EHNN-3SATAIS during cloning and somatic hypermutation operator has improved the affinity of B-cells (interpretations). In fact, the somatic hypermutation in AIS differs with mutation operator in GA in the way of the flipping process. Somatic hypermutation in EHNN-3SATAIS will enhance the probability of B-cells in attaining possible maximum affinity before being stored in HNN (Layeb, 2012). According to optimization standpoint, less iterations were needed in generating the interpretations with the highest affinity regardless the complexity will become higher. Furthermore, EHNN-3SATAIS could sort the enumerate and generate the 3-SAT interpretations effectively and can withstand more number of neurons as compared to the EHNN-3SATES and EHNN-3SATGA.

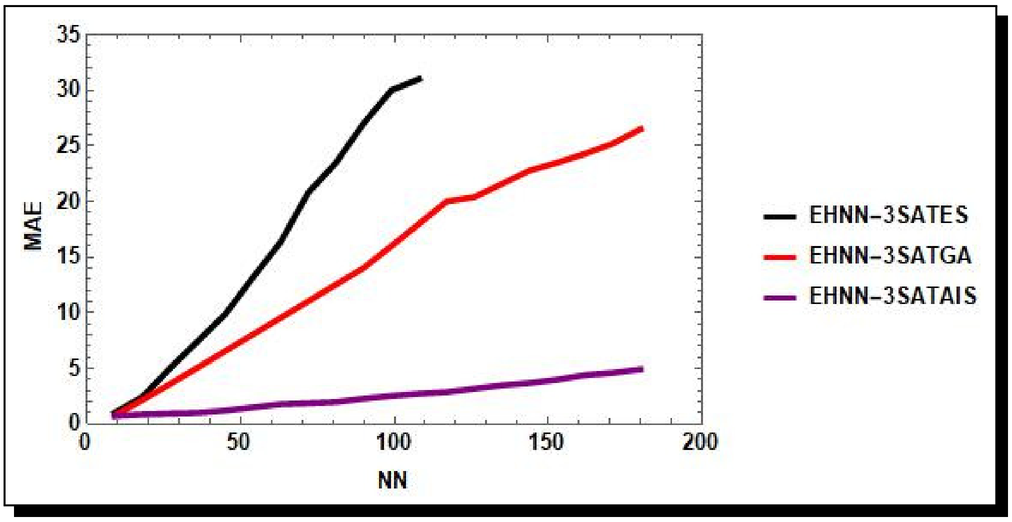

Figure 2. Mean absolute error (MAE) for EHNN models

Conversely, EHNN-3SATES records a significantly higher value MAE from $NN = 9$ until $NN = 108$. This suggests that the EHNN-3SATES experiences massive memory deterioration due to the complexity devoured during the learning phase of the network. Furthermore, the mechanism of the EHNN-3SATES that deploys the "generate and test" method to enumerate the correct solutions in a search space of the network [23]. Consequently, the MAE for EHNN-3SATES is significantly higher than EHNN-3SATES and EHNN-GA. It is shown that more iterations needed to generate the satisfied interpretation during the learning phase of EHNN-3SATES. Moreover, the MAE obtained by EHNN-3SATGA are slightly lower than EHNN-3SATES and significantly higher than EHNN-3SATAIS. The EHNN-3SATGA improves the chromosomes (interpretations) before undergoing the crossover operator and requires less iterations compared to EHNN-3SATES [1]. For instance, the solutions will be improved using the mutation process to attain maximum fitness in EHNN-3SATGA. In general, the value of MAE will be increasing as the number of neurons increase. Therefore, this is due to the number of clauses corresponds to the number of neurons during the learning phase of the network.

B. Schwarz Bayesian Criterion (SBC)

In this study, Schwarz-Bayesian information criterion (SBC) is used to measure the capability of the computational model. The mean square error (MSE) is taken into account when computing the SBC values. The relationship between these two units can be formulated. Hence, in general when the MSE is lower, the SBC will be apparently lower. According to Hamadneh (2013), the best computational model will be chosen among the lowest SBC value during the execution. Cumulatively, the MSE during learning and retrieval phase will be utilized in determining the SBC value.

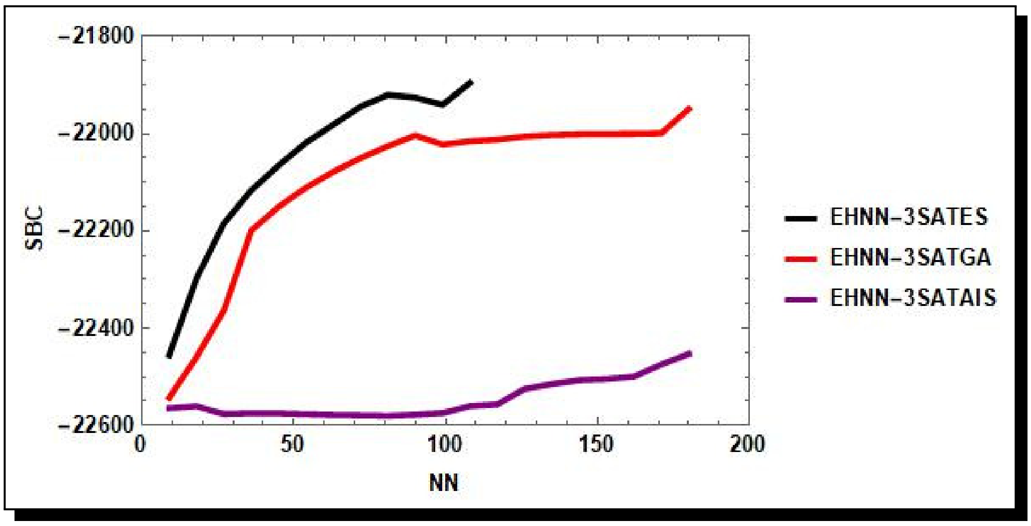

Figure 3. Schwarz-Bayesian information criterion (SBC) for EHNN models

Figure 3 illustrates the SBC recorded for EHNN-3SATAIS, EHNN-3SATGA, and EHNN-3SATES during the learning phase of the network. Based on SBC evaluation, it was clearly shown that EHNN-3SATAIS outclasses the EHNN-3SATGA and EHNN-3SATES. For this reason, the optimization operator such as somatic hypermutation in AIS will make the searching process much easier without consuming extra iterations. In fact, the non-improving interpretations will be enhanced via somatic hypermutation operator during learning phase in EHNN. On contrary, the SBC for EHNN-3SATGA is apparently higher than EHNN-3SATAIS due to early improvement in order for the crossover process to take place. Therefore, EHNN-3SATES has recorded the accumulation of MSE during learning and retrieval phase due to more iterations needed to attain the convergence. The accumulation of MSE will penalize the SBC values. Thus, the SBC for EHNN-3SATES is the highest compared to the other two counterparts. In term of SBC evaluation, EHNN-3SATAIS is an acceptable approach than EHNN-3SATGA and EHNN-3SATES.

C. CPU Time

The robustness of EHNN-3SATAIS, EHNN-3SATGA, and EHNN-3SATES can be assessed based on the elapsed time for a complete simulation with a different number of neuron (NN).

Figure 4 depicts the CPU time taken for the EHNN-3SATAIS, EHNN-3SATGA, and EHNN-3SATES from $NN = 9$ until $NN = 180$. The learning phase needs to be enhanced by using the effective global or local search techniques. Moreover, Figure 4 manifests that the EHNN-3SATAIS outperforms the other two counterparts, EHNN-3SATES and EHNN-3SATGA. The CPU Time of EHNN-3SATAIS is obviously faster due to the capability of B-cells (interpretations) to navigate and converge towards the maximum affinity. As discussed, this is due to the fact that the somatic hypermutation will improve the affinity in fewer iterations. On the contrary, EHNN-3SATGA is apparently faster than EHNN-3SATES because of the genetic diversity operators such as crossover and mutation. Both operators have enhanced the non-improving interpretation in attaining the convergence [25]. Thus, EHNN-3SATES experience the slowest convergence during learning phase of the network because of the extensive iterative process by deploying "trial and error" in enumerating the correct interpretation. Henceforth, EHNN-3SATAIS experiences minimum computational burden throughout the learning phase in generating the satisfied interpretations as compared to EHNN-3SATGA and EHNN-3SATES. As the number of neurons increases, the CPU time also increases for EHNN-3SAT models. All in all, the MAE, SBC and CPU Time have proved the effectiveness of the proposed algorithm in term of accuracy, sensitivity, and robustness.

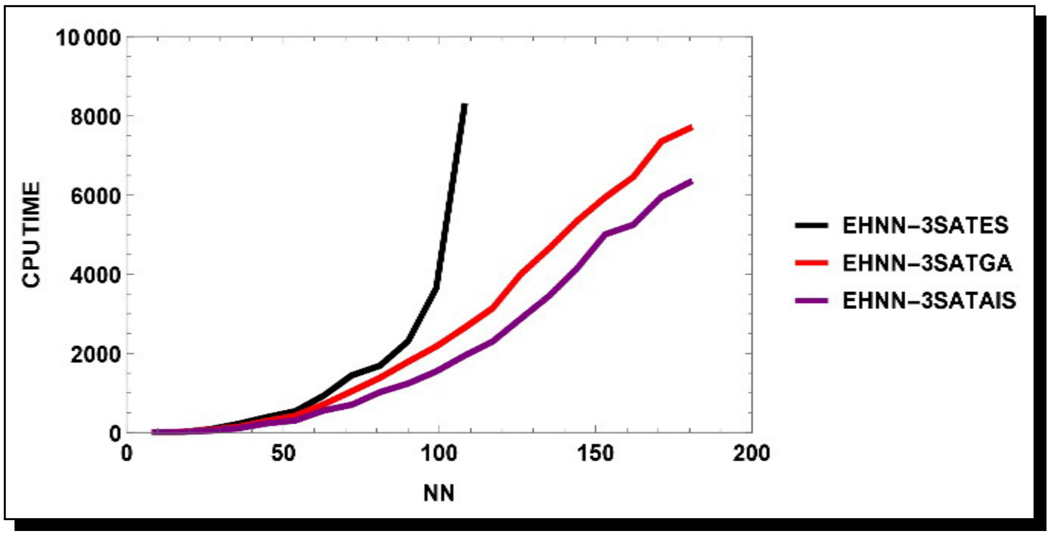

Figure 4. CPU Time for EHNN models

9. Conclusion

We have investigated the effectiveness of EHNN-3SATAIS, EHNN-3SATGA, and EHNN-3SATES in 3-SAT programming. It was shown that EHNN-3SATAIS had outperformed the other two models in terms of MAE, SBC and CPU Time recorded during the simulation by using a different number of neurons. Our work can be developed further by exploring other variants of the satisfiability problem to see whether our model will comply with other higher-order satisfiability logic.

Acknowledgement

We would like to thank School of Distance Education and School of Mathematical Sciences, Universiti Sains Malaysia for the support of the registry and approval to publish this research article.

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

[2] B. Bunz and M. Lamm, Graph neural networks and boolean satisfiability, *Cornell Journal of Computer Sciences* (2017).

[3] T. Chai and R. R. Draxler, Root mean square error (RMSE) or mean absolute error (MAE)?, *Geoscientific Model Development* **7**(3) (2014), 1247 – 1250.

[4] C. A. C. Coello and N. C. Cortés, Solving multiobjective optimization problems using an artificial immune system, *Genetic Programming and Evolvable Machines* **6**(2) (2005), 163 – 190.

[5] D. Dasgupta, S. Yu and F. Nino, Recent advances in artificial immune systems: models and applications, *Applied Soft Computing* **11**(2) (2011), 1574 – 1587.

[6] J. D. Farmer, N. H. Packard and A. S. Perelson, The immune system, adaptation, and machine learning, *Physica D: Nonlinear Phenomena* **22**(1) (1986), 187 – 204.

[7] L. Gao and X. Liu, Global optimization algorithm for Hopfield network based on simulated annealing, *Journal of Liaoning Technical University (Natural Science)* **1** (2009), 43 – 50.

[8] S. Goudarzi, W. H. Hassan, S. A. Soleymani and M. H. Anisi, Hybridisation of genetic algorithm with simulated annealing for vertical-handover in heterogeneous wireless networks, *International Journal of Ad Hoc and Ubiquitous Computing* **24**(1-2) (2017), 4 – 21.

[9] M. W. Goudreau and C. L. Giles, Routing in random multistage interconnections networks: Comparing exhaustive search, greedy and neural network approaches, *International Journal of Neural Systems* **3**(2) (1992), 125 – 142.

[10] N. Hamadneh, *Logic Programming in Radial Basis Function Neural Networks*, PhD Thesis, Universiti Sains Malaysia, Malaysia (2013).

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

[12] J. J. Hopfield and D. W. Tank, Neural computation of decisions in optimization problem, *Biological Cybernatics* **52** (1985), 141 – 152.

[13] J. J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, *Proceedings of the National Academy of Sciences* **79**(8) (1982), 2554 – 2558.

[14] K. E. Iverson, A programming language, in: *Proceedings of the Spring Joint Computer Conference*, May 1-3, 1962, 345 – 351.

[15] J. L. Johnson, A neural network approach to the 3-satisfiability problem, *Journal of Parallel and Distributed Computing* **6**(2) (1989), 435 – 449.

[16] M. S. M. Kasihmuddin, M. A. Mansor and S. Sathasivam, Bezier Curves Satisfiability Model in Enhanced Hopfield Network, *International Journal of Intelligent Systems & Applications* **8**(12) (2016), 9 – 17.

[17] R. Kumar, Blending roulette wheel selection & rank selection in genetic algorithms, *International Journal of Machine Learning and Computing* **2**(4) (2012), 365.

[18] K. Kutzkov, New upper bound for the 3-sat problem, *Information Processing Letters* **105**(1) (2007), 1 – 5.

[19] A. Layeb, A clonal selection algorithm based tabu search for satisfiability problems, *Journal of Advances in Information Technology* **3**(2) (2012), 138 – 146.

[20] Y. Liang, Combinatorial optimization by Hopfield networks using adjusting neurons, *Information Sciences* **94**(1-4) (1996), 261 – 276.

[21] J. Ma'ndziuk, Optimization with the Hopfield network based on correlated noises: experimental approach, *Neurocomputing* **30**(1) (2000), 301 – 321.

[22] M. A. Mansor and S. Sathasivam, Accelerating activation function for 3-satisfiability logic programming, *International Journal of Intelligent Systems and Applications* **8**(10) (2016), 44 – 50.

[23] J. Nievergelt, Exhaustive search, combinatorial optimization and enumeration: Exploring the potential of raw computing power, in: *International Conference on Current Trends in Theory and Practice of Computer Science*, Springer (2000), 18 – 35.

[24] G. Pinkas, Symmetric neural networks and propositional logic satisfiability, *Neural Computation* **3**(2) (1991), 282 – 291.

[25] S. Prakash and D. P. Vidyarthi, A hybrid immune genetic algorithm for scheduling in computational grid, *International Journal of Bio-Inspired Computation* **6**(6) (2014), 397 – 408.

[26] A. D. Pwasong and S. Sathasivam, Forecasting performance of random walk with drift and feed forward neural network models, *International Journal of Intelligent Systems and Applications* **7**(9) (2015), 49 – 56.

[27] S. Sathasivam and W. A. T. W. Abdullah, The satisfiabilty aspect of logic on little Hopfield network, *SSJ* **1**(2) (2010), 2 – 17.

[28] S. Sathasivam, Upgrading logic programming in Hopfield network, *SainsMalaysiana* **39** (2010), 115 – 118.

[29] G. Schwarz, Estimating the dimension of a model, *The Annals of Statistics* **6**(2) (1978), 461 – 464.

[30] W. Shi, J. Liu, Z. Du, Y. Song, C. Chen and T. Yue, Surface modelling of soil ph. *Geoderma* **150**(1) (2009), 113 – 119.

[31] P. Sibi, S. A. Jones and P. Siddarth, Analysis of different activation functions using back propagation neural networks, *Journal of Theoretical and Applied Information Technology* **47**(3) (2013), 1264 – 1268.

[32] J. Timmis and M. Neal, A resource limited artificial immune system for data analysis, *Knowledge-Based Systems* **14**(3) (2011), 121 – 130.

[33] J. Timmis, A. Hone, T. Stibor and E. Clark, Theoretical advances in artificial immune systems, *Theoretical Computer Science* **403**(1) (2008), 11 – 32.

[34] B. Tobias and K. Walter, An improved deterministic local search algorithm for 3-SAT, *Theoretical Computer Science* **329** (2004), 303 – 313.

[35] M. Velavan, Z. R. Yahya, M. N. Abdul Halif and S. Sathasivam, Mean field theory in doing logic programming using Hopfield network, *Modern Applied Science* **10**(1) (2016), 154 – 160.

[36] W. A. T. Wan Abdullah, Logic programming on a neural network, *Malaysian Journal of computer Science* **9**(1) (1993), 1 – 5.

[37] U. P. Wen, K. M. Lan and H. S. Shih, A review of Hopfield neural networks for solving mathematical programming problems, *European Journal of Operational Research* **198**(3) (2009), 675 – 687.

[38] C. J. Wilmott and K. Matsura, Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance, *Climate Research* **30**(1) (2005), 79 – 82.

[39] C. J. Wilmott, K. Matsura and S. M. Robeson, Ambiguities inherent in sums-of-squares-based error statistics, *Atmospheric Environment* **43**(3) (2009), 749 – 752.

[40] L. Xiao, H. X. Wang, B. Z. Wang, G. Zheng and P. Chen, An efficient hybrid method of iterative mom-po and equivalent dipole-moment for scattering from electrically large objects, *IEEE Antennas and Wireless Propagation Letters* (2017).

[41] H. Zhang, Y. Hou, J. Zhao, L. Wang, T. Xi and Y. Li, Automatic welding quality classification for the spot welding based on the Hopfield associative memory neural network and Chernoff face description of the electrode displacement signal features, *Mechanical Systems and Signal Processing* **85** (2017), 1035 – 1043.