

Modified election algorithm in hopfield neural network for optimal random k satisfiability representation

Hamza Abubakar^{1,*}, Shamsul Rijal Muhammad Sabri¹, Sagir Abdu Masanawa², and Surajo Yusuf³

¹ School of Mathematical Sciences, Universiti Sains Malaysia, Pualau Penang, 11800 USM, Penang, Malaysia

² Department of Mathematics, Federal University Dutsin-Ma, Nigeria

³ Department of Mathematics, Isa Kaita College of Education, Dutsin-Ma, Nigeria

Received: 4 April 2020 / Accepted: 20 June 2020

Abstract. Election algorithm (EA) is a novel metaheuristics optimization model motivated by phenomena of the socio-political mechanism of presidential election conducted in many countries. The capability and robustness EA in finding an optimal solution to optimization has been proven by various researchers. In this paper, modified version of EA has been utilized in accelerating the searching capacity of Hopfield neural network (HNN) learning phase for optimal random- k SAT logical representation (HNN-R \mathcal{S} ATEA). The utility of the proposed approach has been contrasted with the current standard exhaustive search algorithm (HNN-R \mathcal{S} ATES) and the newly developed algorithm HNN-R \mathcal{S} ATICA. From the analysis obtained, it has been clearly shown that the proposed hybrid computational model HNN-R \mathcal{S} ATEA outperformed other existing model in terms of global minima ratio (Zm), mean absolute error (MAE), Bayesian information criterion (BIC) and execution time (ET). The finding portrays that the MEA algorithm surpassed the other two algorithms for optimal random- k SAT logical representation.

Keywords: Election algorithm / Hopfield neural networks / exhaustive search / random satisfiability / logic programming

1 Introductions

Artificial neural networks (ANNs) belong to the family of the computational architectural-based model, viewed as equivalent to a brains' programming by imitating its design and attempts to mimic nervous system activity through which information is handled by the brains [1]. It consists of many basic processing components (called artificial neurons) that are loosely based on biological neurons. This learns about the relations between the processing elements and the system parameters through a cycle of adjustment. It contains several interconnected neurons that have a specific synaptic weight that essentially influences how much the neuron output can influence the input to the next neurons. The information is preserved in the weight and learned by the network. This process is done by several known algorithms. There has been an enormous development in computational technology and increase interest in the possible application and use of artificial intelligence in various domains of applications [2].

Hopfield neural networks (HNNs) are regarded as a specific type of neural networks that can preserve certain memories or trends in a way similar to the human brain which can be recovered if only partial, distorted or noisy information is presented to the network [3]. This brain capacity is often referred to as memory that can be associative or content-based i.e. content addressable memory (CAM). Hopfield proposed that stable patterns/memories are energy minima of HNN. The neural model initially proposed by Hopfield as associative memory either in its original version "first-order Hopfield networks" or its modified/extended version of high order [4]. The dynamics of activation function that decreases as the network evolves dynamically. Consequently, the network is expected to reach a state of equilibrium that does not correspond to an optimization problem [4]. Significant activity in this field is the quest for evolutionary metaheuristics algorithm techniques to push the network out of local minima and carry it to a global minimum. These limitations motivated researchers to propose different hybrid optimization technique to speed the accuracy and stability of HNN in solving various optimization problems. A review of HNN for solving mathematical programming problems has been presented [5]. The studies described HNN as one of the major neural network (NN) for solving various

* e-mail: zeeham4u2c@yahoo.com

optimization or mathematical programming (MP) problems. It introduced HNN structure on an electronic circuit for an on-line solver with such a parallel distribution process. A strategy for solving an economic power dispatch problem has been introduced with a piecewise quadratic cost function using the HNN in [6]. The technique suggested is much easier and the results are very similar to those of the number system. A method based on fused imagery as an additional source of information for super-resolution mapping using the HNN approach has been presented [7]. The study incorporated forward and inverse models in the HNN to support a new reflectance constraint added to the energy function. An algorithm named HNN Spatio-temporal data fusion model (HNN-SPOT) has been developed [8]. Using the HNN optimization principle for a fusion of Spatio-temporal images. The HNN-SPOT tackles the issue in particular because the fine resolution and coarse resolution images are collected over the same regional area from specific satellite overpassing periods. HNN-SPOT provides a fairly new method in a fusion of Spatio-temporal data. The Hopfield Lagrange Network (HLN) approach is used to solve the problem of optimum load loading (OLD) in the dynamic electric market [9]. The duty of the HLN in this study was conducted to determine the Optimum productive power output of thermal generation units to optimize the advantages of producing energy from all the available equipment. A Modified HNN algorithm (MHNNA) with remote sensing imaging was used to identify proportions of total suspended solids (TSS) in coastal waters of Langkawi Island, Malaysia [10]. The modification was made to allow the HNN to translate and identify colour images from satellites. For the problem of frequency assignment (FAP) in satellite communications, a hybrid Hopfield-simulated annealing algorithm (HopSA) is introduced [11]. The goal of the NP-complete problem is to reduce the cochannel conflict between satellite communications systems by rearranging the frequency assignment, so the systems can satisfy the demands. The main challenge in these studies is to implement the optimal method of learning in HNN.

Boolean Satisfiability (SAT) is a well-known decision problem, that consists in deciding whether a propositional logic formula can be satisfied given suitable value assignments to the variables of the formula. SAT is a widely used modelling framework for solving combinatorial problems. SAT is also a well-known NP-complete decision problem [11]. Many NP problems that can be transformed and represented into SAT. Besides the practical application, SAT has also influenced many related decision and optimization problems, which will be referred to as extensions of SAT. Many practical applications mathematics and engineering fall into the category of optimization that can easily be transformed into SAT. The performance improvements made to SAT solvers since the mid 90s motivated their application to a wide range of practical applications, this include, cross talk noise prediction [12], Modulo theory [13] and convex optimization [14] and formula verification [15]. Additional successful examples of practical applications of SAT include knowledge compilation [16] as well as software model checking [17].

Advances neural-symbolic computation as a fundamental approach for applied machine learning and reasoning have been pioneered in [18]. The integration explains the usefulness of the technique by outlining the main characteristics of the methodology, the main convergence of neural processing with the symbolic representation of intelligence and reasoning that allows for the development of explainable AI systems. The main idea of logic as a programming language in neural networks was introduced in [19] to serve, represent and interpret a problem. The motivating force behind it is the notion that a single formalism is adequate both for logic and computing, and that it subsumes computation. An algorithm may be known to consist of a logic component that defines the intelligence to be used in optimizing a given problem, and perhaps a control component that decides the problem-solving techniques by which that knowledge is utilized. Therefore, from the perspective of combinatorial optimization, it can be viewed as a problem. The logic programming idea was extended incorporating the competent logical mapping system or propositional knowledge through an asymmetric network of connectionists [20]. The proposed new symmetric connectionist network (SCN) has therefore attracted the interest of AI and computational scientist communities to combine the advantage of both the logic program and neural network as a single network [20]. Wan Abdullah developed a learning method which is used in calculating the synaptic weight of the HNN corresponds to the suggested logical system, and the technique is still applicable particularly when working with an RNN [21]. The Horn logic programming was upgraded by integrating the efficient relaxation method to generate the optimum final neuron states in [21]. The Boltzmann machine based on the stochastic approach for HNN model has been proposed in [22], which reduced neurons oscillations during the HNN model recovery phase. The notion of logic programming merged with radial basis function neural network as a single model for computation has been achieved in [23]. The result reveals that RBFNN in logical programming inevitably worked well. Consequently, in Mean Field Theory depict HNN logic programming flexibility has been proposed in [18,24]. A comprehensive investigation of the potential effect of systematic Satisfiability programming as a logical rule to optimize the output weights and parameters in RBFNN [25]. The study investigated the effectiveness of 2SAT logical rule in reducing the computational burden for RBFNN by obtaining the parameters in RBFNN. Successful implementation of k SAT as a logical representation in HNN has been proposed [26–27].

The benefit of metaheuristics algorithms in HNN is that a network can be inferred at different stages such as weight training and adjustment, system adaptation for determining the number of layers, node transfer functions, retrieval phase and learning rules. MEA has emerged a new evolutionary metaheuristics algorithm that has been applied in finding an optimal solution to computational optimization and engineering application [28–31]. Unlike many other metaheuristics that are mainly inspired by swam or natural evolutionary process. EA is adopting the socio-political process of presidential elections conducted in

many countries. The synthesis of metaheuristics and the HNN have been proven effective in carrying out satisfiability as a logic programming [18,28,29]. Unfortunately, to the best of our knowledge, no effort combined the global optimization ability of Election algorithm to facilitate the training phase of HNN in carrying out random satisfiability optimization (random- k SAT). Therefore, we proposed a new hybrid computational model by incorporating hybrid election algorithm (HEA) to foster the learning phase of HNN in optimizing random k -satisfiability logic representation in attaining better accuracy, sensitivity, and robustness for higher-order networks. The contributions of the present study include the following:

- To integrate the new logical representation namely random- k SAT in HNN.
- To implement newly proposed EA incorporated with random k SAT into discrete Hopfield Neural Network (HNNR k SATEA).
- To explore the feasibility of HNNR k SATEA model and measure its performance in term of accuracy with the existing models for HNN.
- To establish a comparison of the HNNR k SATEA with two existing HNN models.

The proposed hybrid computational model will be beneficial to mathematics and computational science communities by providing an alternative method in carrying combinatorial optimization problem (TSP, scheduling problem, graph colouring problem, time-tabling problems, etc), mathematical logic, mathematical approximation, the Lie group integrator, computational algebra and computational sciences. Our results explored that the new proposed hybrid computational model improves the learning phase of HNNR k SATEA by demonstrating the best performance compared to other existing work.

This paper has been organized as follows. In Section 2, random k satisfiability problems have been presented which include random satisfiability in discrete Hopfield neural network. Section 3, Neuro-searching techniques employed in this research which include, Exhaustive Search Algorithm (ES) and modified version of Election Algorithm (EA) have been discussed. Section 4, presented an experimental model setup. Finally, Sections 5 and 6 enclose graphs of the simulated results, discussion, and the conclusion of this exploration.

2 Mathematical model of discrete Hopfield neural network

Hopfield network to model biological memory, proposed the so-called, Hopfield Associative Memory (HAM). Such an HNN model is based on the Mc-Culloch-Pitts paradigm of artificial neuron [4]. In such a HANN, the state space constitutes the symmetric unit hypercube (i.e. the components of state vectors are bipolar $[1,-1]$). The associated convergence theorem ensures that in the serial mode of operation, the initial state converges to a stable state and in the fully parallel mode of operation [31]. The stable states are realized as the “memory states” of the associative memory. HNN was naturally interested in synthesizing a CAM, with certain “desired stable states”

(that are preselected). It is a novel neural computational framework through the implementation of an auto-associative memory. They belong to recurrent or fully interconnected categories of neural networks. All neurons are connected, but there is no self-recurrent connection between the neurons. HNN consists of N interconnected nodes; each of the nodes can be expressed by a simplified *Ising* variable of spin glass in statistical mechanics [32]. The neurons in HNN are considered bipolar, $L_j \in [-1, 1]$ conform to the dynamics $L_j \rightarrow \text{sign}(h_j)$ The model dynamical system including the state-evolving equation is calculated based on general asynchronous updating rule of HNN given in equation (1):

$$L_j = \begin{cases} 1, & \text{if } \sum_k^N U_{jk} L_k(t) > \tau_j \\ -1, & \text{otherwise} \end{cases} \quad (1)$$

where U_{jk} is the synaptic connection matrix that established the strong connection between j and k neurons, L_j defines the unit condition k and τ_j described the threshold function of neurons j . Several studies [16,28,33–38] defined $\tau_j = 0$ to verify that the HNN always leads to a decrease in energy monotonically. Each time neuron was connected with U_{jk} , the value of the synaptic connection will be preserved as a stored pattern in an interconnected vector where $U^{(1)} = [U_{jk}^{(1)}]_{n \times n}$ and $U^{(2)} = [U_{jk}^{(2)}]_{n \times n_T}$ for N -dimensional variable vectors $\tau = (\tau_1, \tau_2, \dots, \tau_N)^T$ as observed in [16,20] that the constraint of synaptic weight matrix $U^{(1)}$ and does not allow self-loop neuron connection $U_{jj}^{(2)} = U_{kk}^{(2)}, \dots, = U_{ii}^{(2)} = 0$ and symmetrical neuron synaptic weight matrix $U_{kj}^{(2)} = U_{jk}^{(2)}$. The HNN energy dynamics function and CAM offers a versatile system with high capacity, error tolerance, rapid memory recovery and partial inputs [20,31,32]. To make HNN ideal for incorporation with combinatorial optimization such as random k SAT. HNN can be used as a logical rule to instruct the behaviour of the system configuration based on the synaptic strength matrix. The logical formulation, in this scenario, consisting of variables vectors, that is framed in the form of N neurons. The implementation of the random- k SAT logical rule in HNN is translated in abbreviated terminology as HNN-R k SAT and the main objective is to minimize the logical inconsistencies by reducing network cost functions towards a minimum solution. Random k SAT can be integrated into HNN by assigning each variable with neurons L_j with the defined cost function. Hence, the cost function $E_{\text{QR-}k\text{SAT}}$, that governs the combinations of HNN and random k SAT is given in equation (2).

$$E_{\text{QR-}k\text{SAT}} = \sum_{j=1}^{NC} \prod_{k=1}^T C_{jk} \quad (2)$$

where NC and T described the number of clauses generated and the number of variable vector in the solution space respectively. The inconsistencies of the logical clause $E_{\text{QR-}k\text{SAT}}$ are provided in equation (3) as follows,

$$C_{jk} = \begin{cases} \frac{1}{2}(1 - L_y), & \text{if } \neg y \\ \frac{1}{2}(1 + L_y), & \text{if } y \end{cases} \quad (3)$$

The weight matrix will represent the synaptic connection matrix between the clauses and variables in a given logical formula. This is done by equating the cost function of HNN ($E_{\mathbb{Q}_{R-kSAT}}$) to the final energy dynamics of the network ($H_{\mathbb{Q}_{R-kSAT}}$) symbolizing the neural state of HNN in which each variable vector L_j at a time t is $L_j(t)$. The local field of HNN can be represented in equation (4) as follows;

$$h_j(t) = \sum_{j=1, k \neq j}^N U_{jk}^{(2)} L_k + U_j^{(1)} \quad (4)$$

$$L_j(t+1) = \begin{cases} 1, & \sum_{j=1, i \neq k}^N U_{jk}^{(2)} L_k + U_j^{(1)} \geq 0 \\ -1, & \sum_{j=1, i \neq k}^N U_{jk}^{(2)} L_k + U_j^{(1)} < 0 \end{cases} \quad (5)$$

where $U_{ij}^{(2)}$ and $U_i^{(1)}$ are second and first order synaptic weight of the embedded random $kSAT$. Equation (4) and (5) are vital to ensure the neurons L_j will always converge to $E_{\mathbb{Q}_{R-kSAT}} \rightarrow 0$. The Lyapunov energy function $H_{\mathbb{Q}_{R-kSAT}}$ has utilized to ensure the energy dynamics for the network decrease monotonically. The final energy dynamics of HNN is provided in equation (6) as follows;

$$H_{R-2SAT} = -\frac{1}{2} \sum_{j=1, j \neq k}^N \sum_{k=1, i \neq k}^N U_{jk}^{(2)} L_j L_k - \sum_{j=1}^N U_j^{(1)} L_k. \quad (6)$$

The aim is to find a feasible solution for which the cost function $H_{\mathbb{Q}_{R-kSAT}}$ is optimal. The convergence toward minimum energy will be considered as an optimal solution to the optimization search. The value of $H_{\mathbb{Q}_{R-kSAT}}$ signifies the value of the energy concerning the absolute final energy $H_{\mathbb{Q}_{R-kSAT}}^{min}$ obtained from random- $kSAT$. Hence the quality of the final neuron configuration can be properly examined based on the condition [15,20–34] defined in equation (7)

$$|H_{\mathbb{Q}_{R-kSAT}} - H_{\mathbb{Q}_{R-kSAT}}^{min}| \leq \xi. \quad (7)$$

If the embedded random- $kSAT$ in HNN does not satisfy equation (7), the final state obtained is most likely trapped in local minimum solution. The synaptic connection, $U_{ij}^{(2)}$ and $U_i^{(1)}$ can be effectively obtained by using Wan Abdullah method [14] or Hebbian learning rule [33]. The energy dynamics of an HNN has many local minima. Consequently, the network is likely to reach an equilibrium state which does not satisfy a solution to the problem. A major task in this “field” is to search for evolutionary strategies like MEA to move the network out of local minima.

2.1 Random satisfiability in discrete hopfield neural network

Random k satisfiability (random $kSAT$) is a logical representation which consists of a non-systematic number of a literal per clause. Random $kSAT$ is a variant of Boolean formula that can be represented in conjunctive normal form (CNF). According to [35], in a scale-free model, given

n , m and β , to formulate a random $kSAT$, m clauses are generated from n logic variables independently at random from the set of $2^k \binom{n}{k}$ clauses, sampling every possible clause with a probability of 0.5. In this paper, we are limiting our investigation $k \leq 2$. For the case of $k \leq 2$, random $kSAT$ has the following properties:

- A set of variables $\{x_1, x_2, x_3, \dots, x_n\}$, in a clause C_i where $i = 1, 2, 3, \dots, n$ that consists of x or $-x$.
- Randomly select 2 variables from the set of n variables with 0.5 probability of negating each variable in the clause [34,35].
- Each x_i in C_i is connected by a disjunction (\vee).
- Each clause C_i is connected by a conjunction (\wedge).

The Boolean values for each x_i are bipolar $x_i \in \{-1, 1\}$ that exemplifies the notion of FALSE and TRUE respectively. One of the examples for random $kSAT$ formulation is as follows:

$$\mathbb{Q}_{R-2SAT} = (\neg F_1 \vee F_2) \wedge (E_1 \vee \neg E_2) \wedge \neg D \quad (8)$$

where \mathbb{Q}_{R-2SAT} described the random 2SAT logical rules that consist of literals F_1, F_2, E_1, E_2 & D and logical clauses $(\neg F_1 \vee F_2), (E_1 \vee \neg E_2)$ & $\neg D$ generated at random. In general, the formulation can be generated in different combinations of neurons (atoms) as the number of neurons (atoms) fluctuated. Comparatively, a high number of neurons (atoms) per number of the clause would increase the probability of many neurons beings satisfied [16–18,34].

The costs function of the negated random 2SAT for bipolar follows.

$$E_{\mathbb{Q}_{R-2SAT}} = \frac{1}{2}(1 + S_{F_1}) \frac{1}{2}(1 - S_{F_2}) + \frac{1}{2}(1 - S_{E_1}) \frac{1}{2}(1 + S_{E_2}) + \frac{1}{2}(1 + S_D) \quad (9)$$

Random 2-SAT can be regarded as one of the constrained optimizations in the HNN model. This can be implemented in the network by storing atom truth values and generating a optimize cost function when maximum clauses are fulfilled. Equation (9) can be expanded into equation (10) as follows.

$$E_{\mathbb{Q}_{R-2SAT}} = \frac{1}{4}(S_{F_1} - S_{F_1}S_{F_2} + S_{E_2} - S_{F_2} - S_{E_1} - S_{E_1}S_{E_2}) + \frac{1}{2}S_D + 1. \quad (10)$$

Consistent interpretation found leading to $E_{\mathbb{Q}_{R-2SAT}} = 0$ as the minimum value correlating to the fact that all random 2SAT clause is satisfied. The value $E_{\mathbb{Q}_{R-2SAT}}$ described as proportional to the number of the unsatisfied clause. By applying the cost function in equation (10) to the Lyapunov energy function in equation (6), the respective synaptic weight of HNN-R2SAT has been obtained based

on equation (11).

$$\begin{aligned}
 H_{\mathbb{Q}_{R-2SAT}} = & -\frac{1}{2}(2L_{[F_1F_2]}^{(2)}S_{F_1}S_{F_2} + 2L_{[F_1E_1]}^{(2)}S_KS_{E_1} \\
 & + 2L_{[F_1E_2]}^{(2)}S_{F_1}S_{E_2} + 2L_{[F_1D]}^{(2)}S_{F_1}S_D \\
 & + 2L_{[F_2E_1]}^{(2)}S_{F_2}S_{E_1} + 2L_{[F_2E_2]}^{(2)}S_{F_2}S_{E_2} \\
 & + 2L_{[F_2D]}^{(2)}S_{F_2}S_D + 2L_{[E_1E_2]}^{(2)}S_{E_1}S_{E_2} \\
 & + 2L_{[E_1D]}^{(2)}S_{E_1}S_D) - (L_{F_1}^{(1)}S_{F_1} \\
 & + L_{F_2}^{(1)}S_{F_2} + L_{E_1}^{(1)}S_{E_1} + L_{E_2}^{(1)}S_{E_2} \\
 & + L_D^{(1)}S_D). \tag{11}
 \end{aligned}$$

According to equation (8), \mathbb{Q}_{R-kSAT} consist of 5 logical variables that are randomly chosen from the set of pre-determined n variables, 2 negative literals and 3 clauses. The outcome of equation (8) can be written as $\mathbb{Q}_{R-kSAT} = -1$ if $(F_1, F_2, E_1, E_2, D) = (1, 1, 1, 1, 1)$ with 2 clauses were satisfied. The global minimum energy is obtained $H_{\mathbb{Q}_{R-2SAT}}^{min} = -1$. The properties of HNN can be used to govern the behaviour of random $kSAT$. In term of HNN-R $kSAT$ EA, is a brand-new model as there no effort utilized the benefits of modified Election algorithm in accelerating the learning performance of HNN in finding the optimal solution to random- $kSAT$ logic representation. The proposed hybrid model developed based on random- $kSAT$ logical clauses. The main focus of this study is, therefore, to explore the feasibility of the hybrid modified election algorithm incorporated in the Hopfield network model optimal random- $kSAT$ logical representation, therefore, HNN-R $kSAT$ EA will equate the efficacy of the proposed model with other current Hopfield models (HNN-R $kSAT$ ES and HNN-R $kSAT$ ICA).

3 Neuro-searching techniques

3.1 Exhaustive search algorithm

The aim of selecting the traditional searching framework is to determine the degree of efficacy of HNN-R2SATES model in carrying out random 2SAT logic programming. Apart from that, there are feasible satisfiable assignments given for any random 2SA logical representation [34–36]. The satisfied clause for the ES heuristic is extracted after a brutal “trial and error” procedure is carried out. The exhaustive search does the survey required to produce a precise topographical map. This approach requires that an extremely large but limited solution space be checked with the number of combinations of various variable values.

The objectives function of ES is expressed as follows:

$$\max |f_{ES}| \tag{12}$$

where,

$$f = \prod_{j=1}^{NC} C_j \tag{13}$$

Stage 1: Initialization

The random population of an individual in the form of a variable vector S_j where $S_j(t) \in [-1, 1]$ is initiated.

Stage 2: present input vector

Present the input pattern $S_j = (S_{j1}, S_{j2}, \dots, S_{jN})$ to the network and store it, where $j = (1, 2, \dots, m)$.

Stage 3: Fitness Evaluation

The variable vector is tested based on the fitness $S_j(t)$ to a quantified variable position in the solution space,

$$f_j(p_j(t)) = \sum_{j=1}^{NC} C_j \tag{14}$$

where f_j designates the number of different neurons combination, NC designates the number of a logical clauses in the random 2SAT and C_j the number of different values of logical clauses passed through the eligibility stage as follows;

$$C_j = \begin{cases} 1, & \text{eligible} \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

Stage 4: clause evaluation

Preserve the clause with the highest possible fitness. Otherwise, identify a new candidate variable vector. Exhaustive searches do not get trapped in local minima with fine enough sampling and worth for both continuous and discontinuous variables function [30,34]. Nonetheless, achieving a global minimum requires an extremely long time. Another drawback of this strategy is that the global minimum may be skipped because of undersampling. When measuring the cost function, it is simple to use the sample whenever it takes a long time. For this reason, exhaustive searches are only practical for a small number of variables as observed in [16,17,20].

3.2 Election algorithm (EA)

Election algorithm is a simple computational system inspired by the socio-political phenomenon of presidential election conducted in many countries. The MEA begins by describing the optimization variables, the cost function, and the cost. It ends by checking for convergence like other optimization algorithms. It is considered a robust evolutionary methodology attracting a talented amount of research in optimization research. If an environment is considered as a function, MEA serves as a function optimizer. In this case, each individual in a population is a sample point in the function space [28,29].

In EA simulation, random weight matrix (population) is generated. In every phase, the weight matrix will be updated through the campaign and coalition mechanism, and their eligibility values will be evaluated to ascertain the best candidates or voters’ position. The cycle of reorganizing the new weight vector variable with repeat the best individuals vector and continue the searching until an appropriate solution is reached (best solution) [28,29]. The HNN energy dynamics is used as the second eligibility assessment method to pick the most appropriate weight

matrix to solve the problem of random- k SAT. On the contrary, the basic motivation of the election algorithm is to discover the variables that optimize the number of random clauses optimize before joining the HNN. The election algorithm in random- k SAT is specifically made up of distinctive stages 1–5.

In general, global optimization can be demonstrated as below (without the loss of generality minimization problem being considered as follows.

$$\max |f_{MEA}| \quad (16)$$

$$f_{MEA} = \prod_{j=1}^{NC} C_j \quad (17)$$

where f_{MEA} is the eligibility function (Objectives), NC describes as the number of a clause in random- k SAT and C_j designates the number of clauses tested by the EA given as follows,

$$C_j = \begin{cases} 1, & \text{eligible} \\ 0, & \text{otherwise} \end{cases} \quad (18)$$

Stage 1: Initialization

A random population of individuals are initialized based on certain features and the size of a population in form decision variables, p_j , where $p_j(t) \in [1, -1]$. Each interpretation is a potential solution to the random- k SAT clause randomly generated. The MEA procedure begins by splitting the entire population into P parties. Initially, the number of individuals will be denoted by N_p in each party, in which, $N_1, \dots, = N_p, \dots, = N_P$, where, $p = 1, 2, \dots, P$

Stage 2: Eligibility evaluation

All randomized variable vector will undergo eligibility function assessment f_{MEA} . Each of the correct variable vectors yields in the satisfied random- k SAT clause will be “awarded”. The number of achieved clauses constitutes the eligibility of the individuals (variable) during the eligibility assessment. The objective function $S_j(t)$ of the election algorithm is as follows.

$$f_j(p_j(t)) = \sum_{j=1}^{NC} C_j. \quad (19)$$

The role of the eligibility function is to measure the goodness of each variable.

Stage 3: Create initial parties and their supporters.

Split the sampling space connected with each variable vector x_j , $j = 1, \dots, N$ into P party’s space i.e.

$$\forall x_j N_c^p = [N_c^{l,p}, N_c^{u,p}], p = 1, \dots, P, j = 1, \dots, N \quad (20)$$

from each party $p = 1, \dots, P$ subspace, $N_c^p = [N_c^1, N_c^2, \dots, N_c^P]$ associated with each variable vector x_j , $j = 1, \dots, N$, N_p values are selected and assessed based on the associated objective functions. Then, the individuals connected with each party $p = 1, \dots, P$. $p = 1, \dots, P$ could be designated as follows:

$$\forall p f(v_{1,m_p}, \dots, v_{j,m_p}, \dots, v_{N,m_p}), m_p = 1, \dots, M_P$$

Or

$$p = \begin{bmatrix} v_{1,1}, \dots, v_{j,1}, \dots, v_{N,1} \\ \vdots \\ v_{1,m_p}, \dots, v_{j,m_p}, \dots, v_{N,m_p} \\ \vdots \\ v_{1,M_P}, \dots, v_{j,M_P}, \dots, v_{N,M_P} \end{bmatrix}, p = 1, \dots, P. \quad (21)$$

Select the best value to serve as the initial number of candidates N_C in forming the initial parties [28]. Thus, the number of individuals sampled as initial candidates represented in equation (22) as follows,

$$N_C = \tau_r N_{pop} \quad (22)$$

where N_C described the total number of candidates, τ_r describe the candidate rate, N_{pop} defined as the total population. The remaining is the total number of such candidates’ advocates (voters) which is represented in equation (23) as follows,

$$\mu_v = N_{pop} - N_C \quad (23)$$

where μ_v described the total number of voters, N_{pop} is the total population and N_C designated as the total number of candidates. All supporters are divided among the candidates based on their similarity. The Euclidian distance matrix is used as a similarity measure is used as follows.

$$d(v_i, c_j) = \text{dist}(\delta_{ir}^f, \delta_{jr}^f) \quad (24)$$

where δ_{ir}^f indicates the voter’s eligibility and δ_{jr}^f indicates the j th candidate’s eligibility. The party whose candidate is closest to it is assigned to each voter.

Stage 4: Advertising Campaign: this stage involves three steps as follows.

Step 1: Positive advertisement

The process of modelling EA positive mechanism, the vector variable of a candidate in the solution space is randomly selected. Random numbers are sampled to pick the position of vector variables (voters) to be substituted. The selection rate is donated by $\lambda_s = \text{rand} \in [0, 1]$. The number of variable vector values that are transferred from a candidate toward its supporters is represented in eq. (25) as follows.

$$\psi_S = \lambda_s S_C \quad (25)$$

where ψ_S is defined as the number of sampled vector variables to be replaced λ_s define the selection rate and S_C described the total number of vector variables of the candidate in the solution space. It is clear that, in an MEA party based on the eligibility Euclidean metric between a candidate and its relevant supporters, the effectiveness of advertisement varies. Positive advertisement happens, whereby the candidates that seem to have an excellent plan and idea if the decision was taken by the voters is to be influenced, the number of its supporters will increase and the chances of increasing the quality of the party’s plans will increase.

To model this goal, we represented eligibility distance coefficient (π_e) as follows.

$$\pi_e = \frac{1}{\|\rho_e - \rho_v\| + 1} \quad (26)$$

where ρ_e and ρ_v designated the eligibility of the candidate e and v , respectively. MEA applied the Euclidean metric to measure the distance between the vector variable ρ_e and ρ_v in the solution space. In MEA, the advertising mechanism, after selecting λ_S and measured π_e the vector values of identified vector variables from the candidate are multiplied in coefficient π_e and the replaced with the identified vector of the associated voters. In other words, given ε_{iold} be the value of i th chosen vector variables of the voters before advertising advances, then after a campaign, the updated value of the identified vector is given as follows [28,29].

$$\varepsilon_{inew} = \pi_e \varepsilon_i \quad (27)$$

based on the equation (27) in the campaign process, the near supporters are much more influenced by their associated candidate than by other followers.

Step 2: Negative advertisement

In the implementations of MEA, contrast advertisement is used among different negative campaigning strategy. Candidates, by their campaign of resistance, seek to fascinate the members of other parties towards themselves. This leads to an upsurge in support of the popular parties and a decline in popularity of the marginalized parties. The difference in eligibility between the voters and the candidates is measured at first in the defender group obeying the Euclidean metric in equation (24).

Step 3: Coalition

Candidates confederate if they shared the same ideas; In MEA, sometimes two or more parties with the same ideas and goals in solution space can come together to create a new party. So some candidates are leaving the campaign and joining another one called "leader." The candidate leaving the arena of the election is called "follower." The candidate leaving the election arena is referred to as "follower." The candidates of the followers collate with the leader and encourage their supporters to follow the leader. All the followers' supporters become the leader's supporters. In our applications, among the candidates $N_c^p = [N_c^1, N_c^2, \dots, N_c^P]$ who wish to unite, a candidate is randomly selected obeying equation (28) as the successor candidate and the remaining candidates are considered as followers [28,29].

If parties candidates unite occurs, the rest of the candidates N_c^p will be randomized according to the following equation;

$$N_c^p = \begin{cases} N_c^p, & \text{rand}[0, 1] \geq \xi \\ N_P, & \text{Otherwise} \end{cases} \quad (28)$$

where $c, p \in [1, N]$ and $\xi = \text{rand}_j.[0, 1]$. Other remaining candidates and their supporters will be allocated to a new candidate.

Stage 5: Election Day (Stopping condition)

Until a condition of termination is met, three different operators, positive advertising, negative advertising and coalition will be applied to update the population. Ultimately a candidate who gets the most votes will declare himself the winner and is equal to the best solution found for the problem of optimization and search [28]. Modified election algorithm incorporated in the Hopfield

neural network model to carry out random- k SAT is expected to be feasible. If the vector variable does not achieve the desired eligibility; the present vector variable and coalition strategies. To boost the solution space, 100 to 10000 iterations usually set in most of the metaheuristics is also considered in our case [16,20,34]. A bipolar search involving on 1 and -1 is used since it is simple for a variable to converge to global maxima. In this research, a modified election algorithm is proposed to accelerate the learning phase of the HNN as a single model in optimizing random 2SAT logic representation.

Algorithm 1 Pseudocode of the Election algorithm

BEGIN

1. Generate initial population in form decision variables,

$$p_j, p_j(t) \in [1, -1]$$

2. Compute eligibility of each individual.

$$f_j(p_j(t)) = \sum_{j=1}^{NC} C_j$$

3. Create initial parties: initial candidates and their advocators.

$$\forall x, N_c^p = [N_c^{l,p}, N_c^{u,p}], p = 1, \dots, P, j = 1, \dots, N$$

REPEAT /* advertising campaign (ad days) /*

For candidate size $N_c^p = [N_c^p, N_c^p]$ do

// positive advertisement

Candidates advertise their plans and improve their position by learning new ideas

// negative advertisement

Candidates battle with one another to increase their advocators.

// coalition

Candidates confederate if they have same ideas.

Compute eligibility of candidates.

END FOR

UNTIL population has converged /* (Election Day) /*

END

4 HNN model experimental setup

In this study, MEA has been incorporated in HNN in the search for an optimal solution for random k SAT logic representation. The proposed hybrid computational model will be compared with the existing HNN-random k SATES [16] and HNN-random 2SATICA [37] models. Both HNN models employed simulated datasets to establish random k SAT logical clauses. To achieve a meaningful comparison between the existing HNN models, all source code was formulated based on the simulation program developed in Dev C++ release version 5.11 is running on a device with Intel[®] Celeron[®] CPU B800@2GHz processor with 4 GB RAM utilizing Windows 8.1. Table 1–3 indicates the appropriate parameters during each HNN model execution.

4.1 Experimental results & discussion

In this study, the training phase of the HNN-random k SATEA model in comparison against the other existing HNN models in the literature will be explored. To prove

Table 1. List of parameters used HNN-R2SATES [23].

Parameter	Value
Neuron Combination	100
Number of Trials	100
Tolerance Value	0.001
Number of String	100
Selection_Rate	0.1

Table 2. List of parameters used in HNN-R2SATICA [37].

Parameter	Value
Neuron Combination	100
Number of Trials	100
Tolerance Value	0.001
Initial Empires	10
Parameter	0.05
Termination Value	0.05
Revolution rate	0.2
Assimilation Angle Coefficient	0.5

Table 3. List of parameters used in HNN-R2SATEA.

Parameter	Value
Neuron combination	100
Number of trials	100
Tolerance value	0.001
Number of learning	100
Positiovecampaign_Rate	0.2
Coalition_rate	0.3
Candidate_rate	0.07

the efficacy in the performance of HNN-random *k*SATEA model, we compared the proposed algorithm with HNN-random *k*SATES and HNN-random *k*SATICA in finding the global optimum based on the global minimum ratio (Z_m), mean absolute error (MAE), Bayesian information criterion (BIC) and execution time (ET).

4.1.1 Global minima ratio (Z_m)

This describes the ratio of global minimum energy combined with the number of neurons in the solution space [23]. The Z_m formula is given in equation (29) as follows.

$$Z_m = \frac{1}{(NN)(COMBMAX)} \sum_{i=1}^n H_{Q_{R-2SAT}}^{min} \quad (29)$$

where NN represents the number of neurons trials, $H_{Q_{R-2SAT}}^{min}$ described the global minimum solutions achieved and $COMBMAX$ described the maximum neurons combination generated in the solution space.

4.1.2 Mean absolute error (MAE)

This described as the average difference found between the expected and the actual values in the solution space in each data. It is used to ascertain the proximity of forecasts to potential outcomes in a given distribution. It is commonly used because it can estimate the error in the data. It is identified as one of the effective metrics used to identify the accumulation of uniformly distributed error in each sample [34,37,38]. The MAE equation is presented as follows.

$$MAE = \sum_{i=1}^n \frac{1}{n} |f_{max} - f_i| \quad (30)$$

where f_i and f_{max} describe the fitness value observed and the maximum fitness respectively.

4.1.3 Bayesian information criterion (BIC)

This described a criterion for the selection of models amongst a finite set of models in a given distribution. It is used to assess the computational efficiency of a model. The Mean square error (MSE) is taken into consideration when computing the BIC values. In general, when the MSE is lower, the BIC tends to be lower. The model which has the minimum BIC value is regarded as the model of interest.

Cumulatively, the MSE will be used to evaluate the BIC value during the training and recovery process. The BIC formula is described as follows:

$$BIC = n \ln(MSE) + pa \cdot \ln(n) \quad (31)$$

where n , pa , and MSE indicate the number of solutions obtained, its parameters and the mean square error used in the model respectively. Since the HNN is free from any pa [17], the equation is re-written as,

$$BIC = n \ln(MSE) \quad (32)$$

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

$$MSE = \frac{1}{n} \sum_{i=1}^n (q)^2 \quad (33)$$

where q is the difference between f_i and f_{max} describe the fitness value observed during each of the execution and the maximum fitness respectively.

4.1.4 Execution time (ET)

Execution time or commonly known as a computational time, represent the time taken to complete an implementation cycle [17,19]. It is represented in equation (34) as follows,

$$E_Time(s) = Training_Time(s) + Recovery_Time(s) \quad (34)$$

5 Results and discussions

Figures 1–4 show search performances between the proposed model and other existing models. It exhibits the behaviours of the errors observed in the searching process from $5 \leq NN \leq 100$. In general, the HNNRkSATES algorithm produces the highest number of errors as demonstrated in Figures 2 and 3. When the problem becomes larger, the disparity is more noticeable

(as the number of neurons increases, HNNR kSATEA manages to maintain lower error margin compares with the existing models whose errors margin increases more as the number of neurons raises). The supremacy of the MEA in searching for the optimal solution for RkSAT logical representation is shown in our study.

Figure 1 shows the apparent success of the modified election algorithm (EA) in comparison with the exhaustive search (ES) and imperialist competitive algorithm (ICA) in generating the optimally satisfied clauses. According to Figure 1, HNNRkSATEA and HNNRkSATICA can retrieve a more accurate state than HNNRkSATICA. Meanwhile, the ES stressed the ‘exhaustive’ trial and error searching techniques during clauses compliance and could only accommodate a maximum of $NN \leq 60$, as the learning model crosses the CPU time threshold. This can be due to the nature of a thorough search, which raised the pressure of computing to achieve the correct configuration. On the other hand, the ability of MEA to control a high number of neurons from $5 \leq NN \leq 100$ may be due to the sheer potential of campaign and coalition mechanism and revolutionary and competition operator in ICA, that reduce the computation burden in finding the optimal states. According to trend displays in Figure 1, HNN-random 2SATICA model some neurons states get trapped in suboptimal solution at $NN = 35, 45, 95$ but still manages to achieve more than 90% success. HNNRkSATEA model indicates a greater efficiency in neuro-searching in optimizing RkSAT logic representation. As the constraints expand forever, the network becomes more difficult as the NN rises in terms of the program’s complexity. The clarification on the connection between the global minimum ratio and the existence of the energy obtained at the end of the computation cycle has been elaborated in [16,17,20] and [34]. Potentially, if the global minimum ratio of the implemented hybrid network is close to one, almost all system solutions have achieved global minimum energy (100% satisfied clauses).

Figures 2 and 3 the MAE and BIC evaluation of various HNN model in optimizing random 2SAT logic representation. It recorded the models’ performance throughout the HNN learning phase from $5 \leq NN \leq 100$. It is explicitly shown that HNNRkSATEA outshines the HNNRkSATES

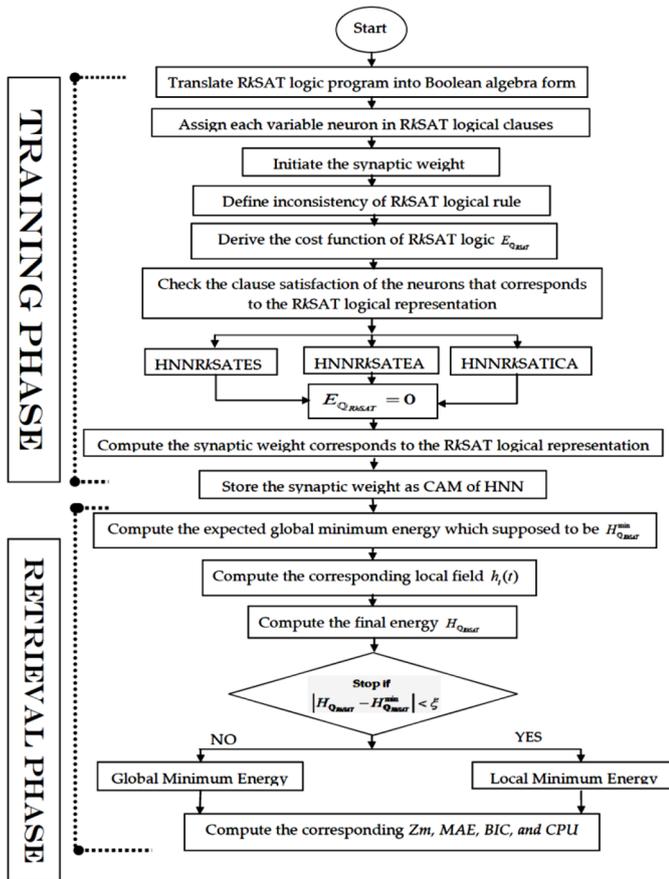


Fig. 1. Flowchart of research methodology.

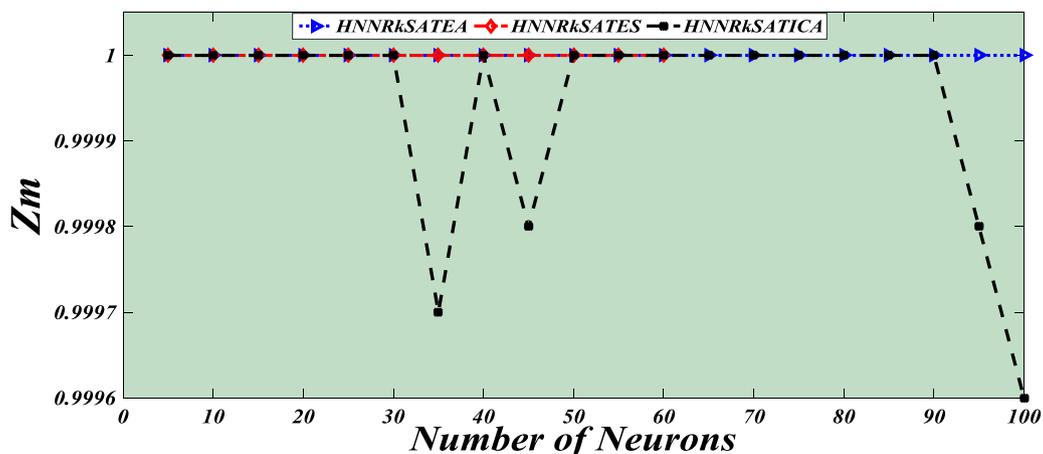


Fig. 2. Zm for HNN-R2SATEA, HNN-R2SATES & HNN-R2SATICA.

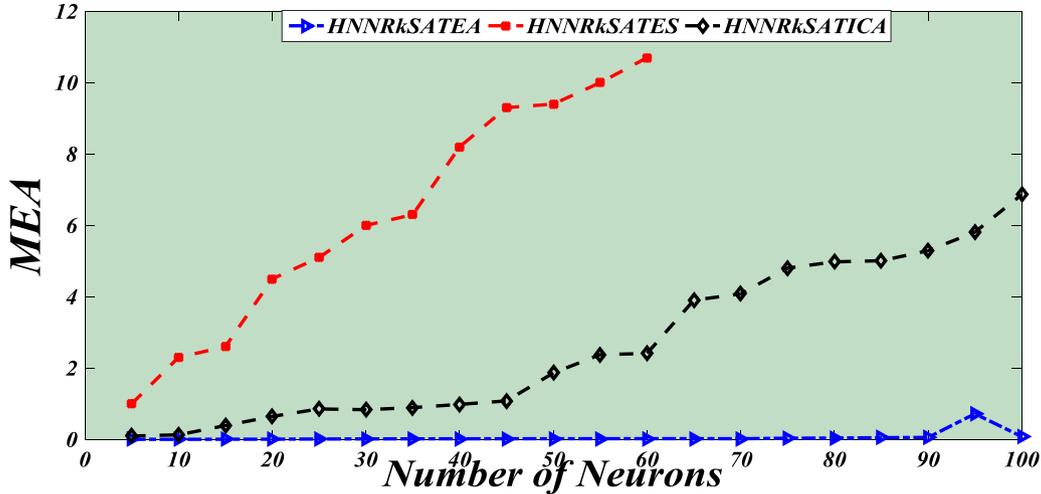


Fig. 3. MEA for HNNR2SATEA, HNNR2SATES & HNNR2SATICA.

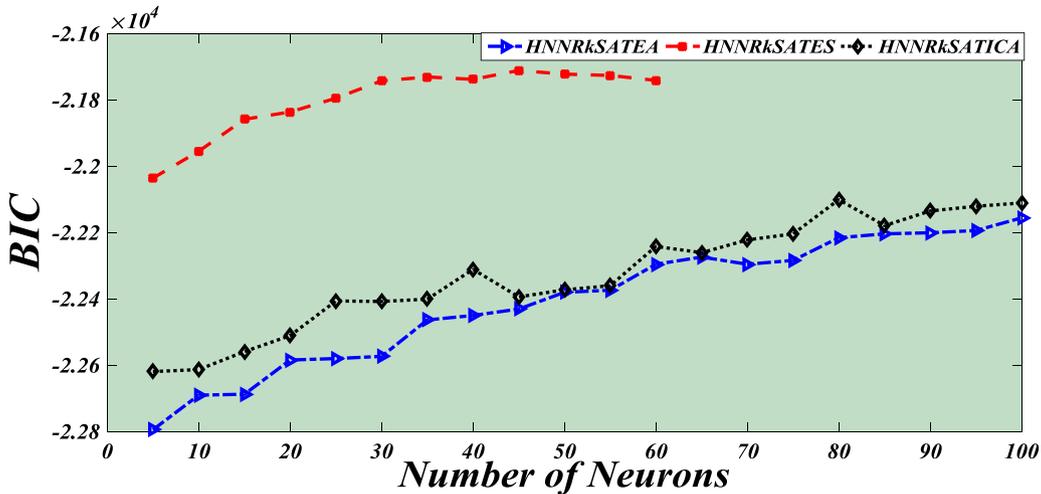


Fig. 4. BIC for HNNR2SATEA, HNNR2SATES & HNNR2SATICA.

and HNNRkSATICA models based on the MAE and BIC assessments. The HNN-R2SATES show a steady rise in errors due to the brute-force hunting for satisfiability mapping from $5 \leq NN \leq 60$. However, ICA manifests close margin with EA from $5 \leq NN \leq 10$ and very high margin as the number of neurons increases. In term of BIC, both EA and ICA display closed margin with EA at $5 \leq NN \leq 60$. The ICA displays a growing pattern of angularity, but of lower error than the ES. The colonial operator’s efficiency decreases the iterations resulted in finding global solutions better than ES. HNNRkSATEA outclasses the HNNRkSATES and HNNRkSATICA based on MAE and BIC measures. This is because the optimization mechanism, such as a coalition in the EA searching process, is much simpler without requiring additional iterations to reach a satisfying assignment. The non-improving solution will be enhanced by a coalition strategy during the HNN learning phase. The HNNR2SATES model observed to have reported an

accumulation of MSE during the learning stage, as a result, more iteration needed to achieve global convergence. The accumulation of MSE tends to penalize the values of BIC. The BIC for HNNR2SATES is, therefore, the highest compared to the other two models. In terms of MAE and BIC assessment, EA is an acceptable approach in HNN in doing random 2SAT logic programming compared to ES and ICA.

Figure 4 demonstrates the execution time for our proposed model, HNNRkSATEA in comparison with HNNRkSATICA and conventional HNNRkSATES models. A glance at the running time shows that the program is becoming more complex, that it takes more effort to find global solutions. However, all models under study display close time range from $5 \leq NN \leq 50$. HNN-R2SATES consumes more time in searching from $50 \leq NN \leq 60$. HNNRkSATES and HNNRkSATICA show closed time agreement from $5 \leq NN \leq 80$. According to Figure 5, our proposed model HNN-R2SATEA requires less execution

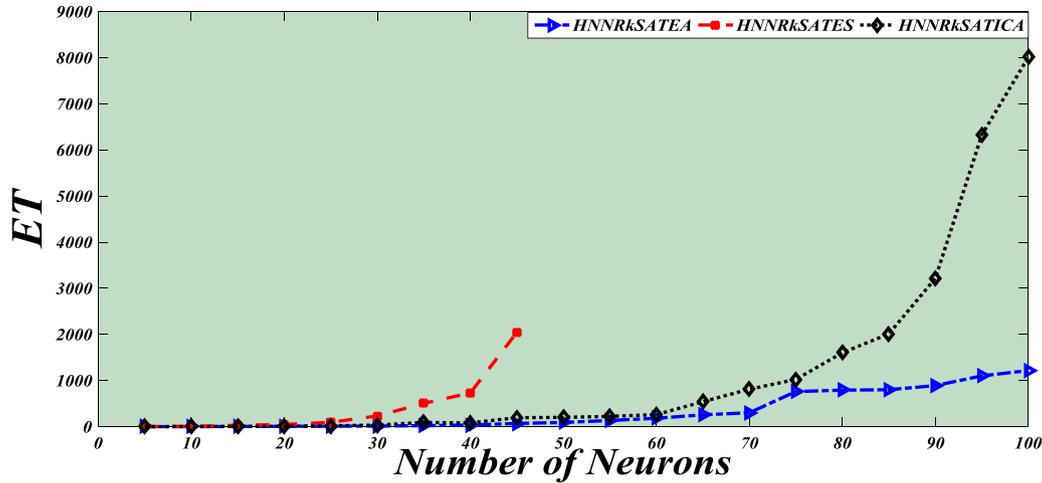


Fig. 5. ET for HNN-R2SATEA, HNN-R2SATES & HNN-R2SATICA.

time compared to HNN-R2SATES and HNN-R2SATICA. Because more neurons are required to cross the energy barrier to relax in global solutions throughout the training phases of HNN. HNNRkSATEA demonstrates the ability to be a more stable run when it produces a near-zero value of MAE, BIC and ET in the training phase of HNN, which is lower compared to other existing models. From this, it can be deduced that hybrid EA can sustain a very low value of errors in improving the training phase of HNN for optimizing random-kSAT logical representation. In other words, when the number of neurons rises, the accumulation of errors was less. Furthermore, the HNNRkSATEA required less iteration to find the desired solutions relative to other existing models. In this case, HNNRkSATES required further iterations to find a global solution, because the interpretation would collapse if the optimum fitness were not reached. This will then lead to the hunting of new interpretation with a new eligibility value. Regarding ES searching process, a suitable assignment is achieved after the ‘trial and error’ process has been carried out thoroughly [16,17,20,34].

The chances of getting more clauses satisfied (optimal) improved significantly as candidates reinforcing their positive images and qualities through campaign mechanisms. The positive campaign mechanism hopefully attracts an unsatisfied clause into a satisfied clause, this will force the model to converge to optional solution space quickly [28,29]. Sometimes RkSAT clause in the solution space may fail to give the best solution, revealing a defect that the algorithm tends to be trapped into local optima. To avoid the problem of premature convergence, balance the efficiency and global search ability of the model. EA applies the negative campaign mechanism to explore other areas of solution space to keep the clause away from an unsatisfied solution before exploring the entire solution space. This effort leads up to find the “new” search space which helps the network to avoid local maxima (non-improving solution). Therefore, because the alliance structure of the HNNRkSATEA model can combat uncertainty, the probability for global solutions is increased. As a result, these mechanisms will permit the network to search and compute the feasible solutions within a reasonable execution

time. This explores higher feasibility of neuro-searching capacity exhibits by the HNNRkSATEA model compares with the other existing models.

The searching techniques will work extensively throughout the training phase for doing RkSAT programming as the complexity of the neurons rises. Effective search techniques with optimization and robust operators are needed. The justification for this trend is that a higher number of neurons convoluted. HNNRkSAT problem is to find the consistent RkSAT assessment since the chances of finding a clear solution significantly reduced. As it stands, HNNRkSATES requires a substantial amount of time to reach the optimal eligibility [16,17,34] At $NN = 90$, HNNRkSATES failed to maintain the complexity of the training phase and the HNN model is observed to trapped in trial and error state during the learning phase.

The MEA is more successful in searching desired outcomes, even in the case of literals of high complexity. This is attributed to the campaign and coalition process to solve greater eligibility within a shorter timeframe. These features in MEA lead HNNRkSAT to complete the learning phase in a reasonable amount of time. HNN-RkSATEA is confirmed to have completed the learning process slightly quicker than other existing models in the literature. However, all HNN models remain competent in optimizing the RkSAT and sits variant and compute the global solution within a feasible CPU timeframe

6 Conclusion

From the results obtained from simulations, it can be ascertained that our proposed HNNR2SATEA model is the more improved and robust heuristic compared to HNNR2SATICA and HNN-2SATES in accelerating the learning phase of HNN in carrying out a random-2SAT logic program. Our proposed model outperformed HNN-R2SATICA and HNNR2SATES model. This has been established from the reported results in term of Zm , MAE, BIC and ET and more interestingly, a Zm of 1 is generated throughout the runs even with the complexity of the network. This also leads to the conclusion that EA is much

more robust to boost the training phase of HNN for random 2SAT logic program and it is closest towards the global minimum, irrespective of the NN release into the HNN. Finally, our future direction will focus on the implementing other metaheuristics approaches (such as dragonfly algorithm, Particle swarm optimization etc) to accelerate the computational phase of Hopfield neural network model in find optimal solution satisfiability and other combinatorial optimization problems (such as reliability problem, scheduling problem etc).

References

1. M. YahayaPudza, Z. ZainalAbidin, S. Abdul Rashid, F. MdYasin, A.S.M. Noor, M.A. Issa, Sustainable synthesis processes for carbon dots through response surface methodology and artificial neural network, *Processes* **7**, 704 (2019)
2. S. Guessasma, D. Bassir, Neural network computation for the evaluation of process rendering: application to thermally sprayed coatings, *Int. J. Simul. Multidiscipl. Des. Optim.* **8**, A10 (2017)
3. J. Hopfield, D. Tank, Neural' computation of decisions in optimization problems, *Biol. Cybernet.* **52**, 141–152 (1985)
4. J. Hertz, R. Krogh, G. Palmer, *Introduction to the Theory of Neural Computation* (Addison-Wesley, Reading, MA, 1991)
5. U.P. Wen, K.M. Lan, H.S. Shih, A review of Hopfield neural networks for solving mathematical programming problems, *Eur. J. Oper. Res.* **198**, 675–687 (2009)
6. J.H. Park, Y.S. Kim, I.K. Eom, K.Y. Lee, Economic load dispatch for piecewise quadratic cost function using Hopfield neural network, *IEEE Trans. Power Syst.* **83**, 1030–1038 (1993)
7. M.Q. Nguyen, P.M. Atkinson, H.G. Lewis, Superresolution mapping using a Hopfield neural network with fused images, *IEEE Trans. Geosci. Remote Sens.* **44**, 736–749 (2006)
8. C.H. Fung, M.S. Wong, P.W. Chan, Spatio-temporal data fusion for satellite images using Hopfield neural network, *Remote Sens.* **11**, 2077 (2019)
9. T.L. Duong, P.D. Nguyen, V.D. Phan, D.N. Vo, T.T. Nguyen, Optimal load dispatch in competitive electricity market by using different models of hopfield lagrange network, *Energies* **12**, 2932 (2019)
10. A. Kzar, M. Jafri, K. Mutter, S. Syahreza, A modified Hopfield neural network algorithm (MHNNA) using ALOS image for water quality mapping, *Int. J. Environ. Res. Public Health* **13**, 1–92 (2016)
11. S.A. Cook, The complexity of theorem-proving procedures, in *Proceedings of the third annual ACM symposium on Theory of computing*, 1971, 151–158
12. J. Marques-Silva, Practical applications of boolean satisfiability, in *2008 9th International Workshop on Discrete Event Systems*, 2008, 74–80
13. C. Barrett, C. Tinelli, *Satisfiability modulo theories*, in *Handbook of Model Checking* (Springer, Cham, 2018), pp. 305–343
14. Y. Shoukry, P. Nuzzo, A.L. Sangiovanni-Vincentelli, S.A. Seshia, G.J. Pappas, P. Tabuada, SMC: satisfiability modulo convex optimization, in *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control*, 2017, 19–28
15. X. Sun, H. Khedr, Y. Shoukry, Formal verification of neural network controlled autonomous systems, in *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control* 2019, 147–156
16. F. Capelli, Knowledge compilation languages as proof systems, in *International Conference on Theory and Applications of Satisfiability Testing* (Springer, Cham, 2019), pp. 90–99
17. J. Li, S. Zhu, G. Pu, L. Zhang, M. Vardi, YSAT-based explicit LTL reasoning and its application to satisfiability checking, *Formal Methods Syst. Des.* **54**, 164–190 (2019)
18. S. Salcedo-Sanz, R.R. Santiago-Mozos, C. Bousoño-Calzon, A hybrid Hopfield network-simulated annealing approach for frequency assignment in satellite communications systems, *IEEE Trans. Syst. Man Cybern. B* **34**, 1108–1116 (2004)
19. R.A. Kowalski, *The Logic for Problem Solving* (Elsevier Science Publishing, New York, 1979)
20. G. Pinkas, Symmetric neural networks and propositional logic SAT, *Neural Comput.* **3**, 282–291 (1991)
21. A.T.W. Wan Abdullah, Logic programming on a neural network, *Int. J. Intell. Syst.* **7**, 513–519 (1992)
22. S. Sathasivam, Upgrading logic programming in Hopfield nets, *Sains Malays.* **39**, 115–118 (2010)
23. S. Sathasivam, Boltzmann machine and new activation function comparison, *Appl. Math. Sci.* **78**, 3853–3860 (2011)
24. N. Hamadneh, S. Sathasivam, S.L. Tilahun, O.H. Choon, Learning logic programming in radial basis function network via genetic algorithm, *J. Appl. Sci.* **12**, 840–847 (2012)
25. M. Velavan, Z.R. Yahya, M.N. Abdul Halif, S. Sathasivam, Mean-field theory in doing logic programming using a Hopfield network, *Mod. Appl. Sci.* **10**, 154–160 (2016)
26. S. Alzaeemi, M.A. Mansor, M.S.M. Kasihmuddin, S. Sathasivam, M. Mamat, Radial basis function neural network for 2 satisfiability programming, *Indonesian J. Electr. Eng. Comput. Sci.* **18**, 459–469 (2020)
27. S.A.S. Alzaeemi, S. Sathasivam, Hopfield neural network in agent based modeling, *MOJ Appl. Biol. Biomech.* **2**, 334–341 (2018)
28. H. Emami, F. Derakhshan, Election algorithm: a new socio-politically inspired strategy, *AI Commun.* **28**, 591–603 (2015)
29. M. Kumar, A.J. Kulkarni, Socio-inspired optimization metaheuristics: a review, in *Socio-cultural Inspired Metaheuristics* (Springer, Singapore, 2019), pp. 241–265
30. G. Gosti, V. Folli, M. Leonetti, G. Ruocco, Beyond the maximum storage capacity limit in Hopfield recurrent neural networks, *Entropy* **21**, 726 (2019)
31. G. Gosti, V. Folli, M. Leonetti, G. Ruocco, Beyond the maximum storage capacity limit in Hopfield recurrent neural networks, *Entropy* **21**, 726 (2019)
32. A. Barra, M. Beccaria, A. Fachechi, A new mechanical approach to handle generalized Hopfield neural networks, *Neural Netw.* **106**, 205–222 (2018)
33. W. Gerstner, W.M. Kistler, Mathematical formulations of Hebbian learning, *Biol. Cybern.* **87**, 404–415 (2002)
34. S. Sathasivam, M. Mansor, M.S.M. Kasihmuddin, H. Abubakar, Election algorithm for random k satisfiability in the Hopfield neural network, *Processes* **8**, 568 (2020)
35. W. Fernandez de la Vega, Random 2-SAT: results and problems, *Theor. Comput. Sci.* **265**, 131–146 (2001)

36. D. Du, J. Gu, P.M. Pardalos, *Satisfiability Problem: Theory and Applications* (American Mathematical Society, 1997), p. 35
37. K. Vigneshwer, M.A. Mansor, M.S.M. Kasihmuddin, S. Sathasivam, Hybrid imperialistic competitive algorithm incorporated with Hopfield neural network for robust 3 satisfiability logic programming, IAES Int. J. Artif. Intell. **8**, 144–155 (2019)
38. M. Peng, N.K. Gupta, A.F. Armitage, An investigation into the improvement of local minima of the Hopfield Network, Neural Netw. **90**, 207–212 (1996)

Cite this article as: Hamza Abubakar, Shamsul Rijal Muhammad Sabri, Sagir Abdu Masanawa, Surajo Yusuf, Modified election algorithm in hopfield neural network for optimal random k satisfiability representation, Int. J. Simul. Multidisci. Des. Optim. **11**, 16 (2020)