Sains Malaysiana 47(6)(2018): 1327–1335
http://dx.doi.org/10.17576/jsm-2018-4706-30

Discrete Hopfield Neural Network in Restricted Maximum *k*-Satisfiability Logic Programming

(Rangkaian Neural Hopfield Diskret dalam Pengaturcaraan Logik Maksimum *k*-Kepuasan Terhad)

MOHD SHAREDUWAN MOHD KASIHMUDDIN*, MOHD ASYRAF MANSOR & SARATHA SATHASIVAM

ABSTRACT

Maximum k-*Satisfiability (MAX-*k*SAT) consists of the most consistent interpretation that generate the maximum number of satisfied clauses. MAX-*k*SAT is an important logic representation in logic programming since not all combinatorial problem is satisfiable in nature. This paper presents Hopfield Neural Network based on MAX-*k*SAT logical rule. Learning of Hopfield Neural Network will be integrated with Wan Abdullah method and Sathasivam relaxation method to obtain the correct final state of the neurons. The computer simulation shows that MAX-*k*SAT can be embedded optimally in Hopfield Neural Network.*

Keywords: Hopfield Neural Network; Maximum k-*Satisfiability; Wan Abdullah method*

ABSTRAK

Maksimum k-*Kepuasan (MAX-*k*SAT) terdiri daripada penyelesaian yang paling tekal untuk menghasilkan bilangan klausa yang betul secara maksimum. MAX-*k*SAT merupakan perwakilan logik yang penting dalam pengaturcaraan logik kerana tidak semua masalah kombinatori boleh dipuaskan. Kertas ini membentangkan Rangkaian Neural Hopfield berdasarkan peraturan logik MAX-*k*SAT. Pembelajaran Rangkaian Neural Hopfield akan diintegrasikan dengan kaedah Wan Abdullah dan kaedah rehat Sathasivam untuk mendapatkan tahap akhir neuron yang betul. Simulasi komputer menunjukkan bahawa MAX-*k*SAT boleh diintegrasi secara optimum dalam Rangkaian Neural Hopfield.*

Kata kunci: Kaedah Wan Abdullah; maksimum k-*Kepuasan; Rangkaian Neural Hopfield*

INTRODUCTION

Recently, Maximum *k*-Satisfiability (MAX-*k*SAT) problem have drawn attention by scholars and produced a prolific amount of research in artificial intelligence (Pipatsrisawat et al. 2008; Raman et al. 1998). Basically, MAX-*k*SAT is the notable counterpart of the Boolean satisfiability (SAT) optimization problem, represented in Conjunctive Normal Form (CNF) form (Layeb et al. 2010). In theory, MAX-*k*SAT problem can be defined as the maximum number of satisfied clauses achieved by any optimum interpretation (Borchers & Furman 1998; Madsen & Rossmanith 2004). Berg and Järvisalo (2015) proposed the implementation of MAX-*k*SAT incorporated with the data mining and constrained clustering. The recent work on MAX-*k*SAT by Bouhmala (2016) emphasized on the neighbourhood search as solver for MAX-*k*SAT problem. Nevertheless, in this research, MAX-*k*SAT will be embedded as a logical rule in Hopfield Neural Network.

The actual concept of contemporary artificial neural network inspired by the biological nervous system to abstract the computations employed by the human brain (Rojas 1996). Among a vast neural network, one of the well-known network implemented for optimization is the Hopfield Neural Network (Hopfield & Tank 1985). Hopfield Neural Networks have been attracted many momentous contributions to various applications, such as combinatorial optimization, pattern recognition, scheduling and data mining (Kumar & Singh 1996; Sulehria & Zhang 2007). The momentous breakthrough is the integration of neuro symbolic which combine the logic programming and Hopfield Neural Network. Kowalski (1979) introduced the main concept of logic as a programming language to represent and interpret a problem. Thus, the logic programming can be interpreted as a problem in combinatorial optimization standpoint. Pinkas (1991) expanded the idea of logic program by integrating the competent propositional knowledge or logical mapping system via symmetric connectionist network. Hence, the proposed symmetric connectionist network (SCN) has attracted researchers to revive many domains of artificial neural network such as Hopfield Neural Network, Boltzmann Machine, Harmony Theory and Mean Field Theory. Pursuing that, Wan Abdullah (1992) proposed a method to compute the synaptic weight of the network correspond to the propositional logic entrenched to the system. Hence, the synaptic weight computing method proposed by Wan Abdullah is still relevant especially when dealing with recurrent neural network. Following that, Hölldobler et al. (1999) proved that the logic program can work effectively with recurrent neural network. Sathasivam (2008) deployed the Wan Abdullah method in order to compute the synaptic weights for the Horn logic programming in Hopfield Neural Network. In this work, the retrieved neuron states are computed by using Lyapunov energy function. Furthermore, Sathasivam (2010) upgraded the Horn logic programming in Hopfield Neural Network by incorporating the effective relaxation method in generating the optimum final neuron states. Sathasivam (2011) further developed the stochastic method for logic programming in Hopfield Neural Network. This stochastic approach has reduced the neuron oscillations during retrieval phase deployed by Hopfield Neural Network. Hamadneh et al. (2012) presented the logic programming in Radial basis function neural network in single operator logic. Inevitably, the Radial basis function worked well with logic programming. Consequently, Velavan et al. (2016) portrays the flexibility of logic programming in HNN with Mean Field Theory. In addition, Mansor and Sathasivam (2016) introduced the application of activation function in Horn logical rules with Hopfield Neural Network. Kasihmuddin et al. (2017) has fruitfully implemented the *k*-SAT logical rule in Hopfield neural network. The proposed non-horn logical rules bind very well with the HNN. In another development, Alzaeemi et al. (2017) has successfully applied logic programming in Kernel Hopfield Neural Network (KHNN). Thus, KHNN is an existing method in MAX-*k*SAT logic programming. In term of MAX-*k*SAT, there is no recent effort to combine the advantages of insatisfiable logic programming with Hopfield Neural Network.

In this paper, we will combine the benefits of the Hopfield Neural Network, logic programming, MAX-*k*SAT, Wan Abdullah method and Relaxation method. The proposed hybrid model will be developed based on MAX-2SAT and MAX-3SAT clauses. Hence, the main focus of this paper was to analyze the effectiveness of Hopfield Neural Network based on the proposed MAX-*k*SAT logical rule. Thus, the effectiveness of the proposed model in doing MAX-*k*SAT logic programming, HNN-MAX*k*SAT will be compared with Kernel Hopfield network, KHNN-MAX*k*SAT.

RESTRICTED MAXIMUM *K*-SATISFIABILITY

Restricted Maximum *k*-Satisfiability problem (MAX-*k*SAT) is a vital generalization of Satisfiability problem. Given a Boolean formula P in conjunctive normal form (CNF) with n clauses containing variable each and positive integer g where $g \leq n$. MAX-*k*SAT can be defined implicitly as a pair (λ, θ) where λ is the set of all possible solution $\{1, -1\}^n$ and θ is a mapping of $\lambda \rightarrow T$ which denotes the score of the assignments (Layeb et al. 2010). T is scored based on true clauses (Satisfied clause). Therefore, MAX-*k*SAT problem consists of defining the best bipolar/binary assignments to the k variables per clause in P that simultaneously satisfies at least g of the n clauses (Madsen & Rossmanith 2004).

In other words, the task is to determine the 'optimized' assignment that can satisfy the maximum number of clauses containing k variables. Basically, there are 2^n potential solutions to this problem. There are many variances of the MAX-*k*SAT such as weighted MAXSAT (Borchers & Furman 1998) and Partial MAX-*k*SAT (Menaï & Batouche 2005). Hence, MAX-*k*SAT is one of the constrained optimization problem that can be include in maximization problem. In our case, we considered $k = 2$ and $k = 3$. Consider the following MAX-2SAT formula:

$$P = (x \vee y) \wedge (x \vee \neg y) \wedge (\neg x \vee y) \wedge (\neg x \vee \neg y) \quad (1)$$

Equation (1) comprise of variables x and y. P is unsatisfiable since there is no specific inteerpretation that make constraint P become true. By assigning $x = 1$ and $y = 1$, P will obtain 3 out of 4 satisfied clauses. According to several studies (Liu & de Melo 2017; Santra et al. 2014; Yannakakis 1994), any of MAX-*k*SAT formula that has $0.7 \leq RSC < 1$ is considered similar to one another. Note that RSC is given as the ratio of satisfied clauses in MAX-*k*SAT logic. This studies can only be applied to the respective counterpart of MAX-2SAT and MAX-3SAT.

LOGIC PROGRAMMING IN DISCRETE HOPFIELD NEURAL NETWORK

The discrete Hopfield Neural Network (HNN) is a simple and powerful method to find high quality solution to hard optimization problem. HNN is an auto associative model and systematically store patterns as a content addressable memory (CAM) (Muezzinoglu et al. 2003). Theoretically, HNN comprises of interconnected units called neurons, forming a recurrent network (Sathasivam & Fen 2013). The network comprises of N recognized neurons, each is described by an Ising spin variable (Hopfield 1982). General updating rule in HNN is given as follows:

$$S_i = \begin{cases} 1 \text{ if} & \sum_j W_{ij} S_j > \psi_i \\ -1 & \text{Otherwise} \end{cases} \quad (2)$$

where W_{ij} is the synaptic weight from unit j to I; S_j is the state of unit j ; and ψ_i is the pre defined threshold of unit i. The connection in the HNN has no connection with itself, $W_{jj} = W_{ii} = 0$ and connections are symmetrical or bidirectional (Sathasivam 2011). Neurons in HNN are bipolar. $S_i \in \{1, -1\}$ in nature. In terms of MAX-*k*SAT representation, each variable in MAX-*k*SAT formula will be represented in terms of N neurons. The synaptic weight will represent the connection between the variable and the clauses in MAX-*k*SAT formula. The connection model can be generalized to embrace higher order connection. This modifies the field to:

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

The updating rule maintains as follows:

$$S_i(t+1) = \text{sgn}[h_i, (t)] \quad (4)$$

where h_i is the local field of the network. The final state of the neurons will be examined by using Lyapunov energy function which is given as follows:

$$H_P = \;\ldots - \frac{1}{2}\sum_i\sum_j W_{ij}^{(2)} S_i S_j - \sum_i W_i^{(1)} S_j, \quad k=2 \quad (5)$$

$$H_P = \;\ldots - \frac{1}{3}\sum_i\sum_j\sum_k W_{ijk}^{(3)} S_i S_j S_k - \frac{1}{2}\sum_i\sum_j W_{ij}^{(2)} S_i S_j - \sum_i W_i^{(1)} S_j, \quad k=3 \quad (6)$$

Worth mentioning that, the updating rule in (4) guarantees the energy will decrease monotonically with the network. In this paper, the learning MAX-kSAT in HNN can be abbreviated as HNN-MAXkSAT. Hence k = 2 and k = 3 will be abbreviated as HNN-MAX2SAT and HNN-MAX3SAT, respectively. Equations (5) – (6) are vital to establish the degree of convergence of the neurons in HNN (Ioneschu et al. 2010; Mathias & Rech 2012). Thus, the energy value is vital to separate local minimum and global minimum solution. Global minimum energy supposed to be $H_P^{\min}$ can be pre-calculated because the total magnitude of the energy that corresponds to MAX-kSAT clauses is always constant (Pinkas 1991; Wan Abdullah 1993). The retrieval power of HNN always depends on how the synaptic weights are computed. In this paper, we implemented the Wan Abdullah method to obtain the synaptic weights for our network.

WAN ABDULLAH METHOD IN LEARNING MAX-*k*SAT CLAUSES

Wan Abdullah method is one of the earliest learning method that extracted synaptic weight based on logical inconsistencies (Wan Abdullah 1992). This can be done by storing truth values of the atoms and creating a minimized cost function when maximum clauses are satisfied. Consider the following MAX-2SAT and MAX-3SAT formula with α and ϕ, respectively.

$$\alpha = (A \vee B) \wedge (A \vee \neg B) \wedge (\neg A \vee B) \wedge (\neg A \vee \neg B) \wedge (C \vee D) \quad (7)$$

$$\begin{aligned}\phi = {} & (P \vee Q \vee R) \wedge (\neg P \vee Q \vee R) \wedge (P \vee \neg Q \vee R) \wedge \\ & (P \vee Q \vee \neg R) \wedge (\neg P \vee \neg Q \vee R) \wedge (\neg P \vee Q \vee \neg R) \wedge \\ & (P \vee \neg Q \vee \neg R) \wedge (\neg P \vee \neg Q \vee \neg R) \wedge (S \vee T \vee U)\end{aligned} \quad (8)$$

Finding inconsistencies by taking into account the negation of (7) and (8)

$$\neg\alpha = (\neg A \wedge \neg B) \vee (\neg A \wedge B) \vee (A \wedge \neg B) \vee (A \wedge B) \vee (\neg C \wedge \neg D) \quad (9)$$

$$\begin{aligned}\neg\phi = {} & (\neg P \vee \neg Q \vee \neg R) \wedge (P \vee \neg Q \vee \neg R) \wedge (\neg P \vee Q \vee \neg R) \wedge \\ & (\neg P \vee \neg Q \vee R) \wedge (P \vee Q \vee \neg R) \wedge (P \vee \neg Q \vee R) \wedge \\ & (\neg P \vee Q \vee R) \wedge (P \vee Q \vee R) \wedge (\neg S \vee \neg T \vee \neg U)\end{aligned} \quad (10)$$

Hence, the cost function for (9) and (10) are as followed:

$$\begin{aligned}E_\alpha = {} & \tfrac{1}{2}(1-S_A)\tfrac{1}{2}(1-S_B) + \tfrac{1}{2}(1-S_A)\tfrac{1}{2}(1+S_B) + \\ & \tfrac{1}{2}(1+S_A)\tfrac{1}{2}(1-S_B) + \tfrac{1}{2}(1+S_A)\tfrac{1}{2}(1+S_B) + \\ & \tfrac{1}{2}(1-S_C)\tfrac{1}{2}(1-S_D)\end{aligned} \quad (11)$$

$$\begin{aligned}E_\phi = {} & \tfrac{1}{2}(1-S_P)\tfrac{1}{2}(1-S_Q)\tfrac{1}{2}(1-S_R) + \tfrac{1}{2}(1+S_P) + \\ & \tfrac{1}{2}(1-S_Q)\tfrac{1}{2}(1-S_R) + \tfrac{1}{2}(1-S_P)\tfrac{1}{2}(1+S_Q)\tfrac{1}{2}(1-S_R) + \\ & \tfrac{1}{2}(1-S_P)\tfrac{1}{2}(1-S_Q)\tfrac{1}{2}(1+S_R) + \tfrac{1}{2}(1+S_P) \\ & \tfrac{1}{2}(1+S_Q)\tfrac{1}{2}(1-S_R) + \tfrac{1}{2}(1+S_P)\tfrac{1}{2}(1-S_Q)\tfrac{1}{2}(1+S_R) \\ & + \tfrac{1}{2}(1-S_P)\tfrac{1}{2}(1+S_Q)\tfrac{1}{2}(1+S_R) + \tfrac{1}{2}(1+S_P) \\ & \tfrac{1}{2}(1+S_Q)\tfrac{1}{2}(1+S_R) + \tfrac{1}{2}(1-S_T)\tfrac{1}{2}(1-S_U)\tfrac{1}{2}(1-S_V)\end{aligned} \quad (12)$$

Since it is impossible to find consistent interpretation that leads to $E_\alpha = 0$ and $E_\phi = 0$, the focus of the network will be shifted by finding the least value E_α and E_ϕ. The corresponding synaptic weight of HNN-MAXkSAT can be computed by comparing the cost function (11) and (12) with (5) and (6), respectively. Synaptic weights from HNN-MAXkSAT are summarized in Tables 1 and 2.

NETWORK RELAXATION

The quality of the final state obtained by HNN-MAXkSAT can be affected by various factors such as parameter setting in the energy function. According to Zeng and Martinez (1999), one of the important factors that influence the quality of the final state is the frequency of the information transferred and received by a particular neuron. In this view, neuron relaxation is one of the essences of getting the correct final state. In detail, the neuron is updated according to Sathasivam Relaxation method (Sathasivam 2010):

$$\frac{dh_i^{new}}{dt} = R\frac{dh_i}{dt} \quad (13)$$

where R denotes the relaxation rate; and h_i is the local field computed by HNN-MAXkSAT. The relaxation rate R will reflect how fast the network relaxed. R is an adjustable parameter and can be determined empirically. Sathasivam

TABLE 1. Synaptic weight for α based on Wan Abdullah method

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

TABLE 2. Synaptic weight for ϕ based on Wan Abdullah method

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

(2010) suggested that the best value to get the optimal solution is when $R = 2$.

COMPUTER SIMULATION

The simulations for HNN-MAX*k*SAT and KHNN-MAX*k*SAT were executed by using Dev C++ Version 5.11 in Windows 10, Intel Core i3, 1.7 GHz processor with 8GB RAM. In order to comply with conventional studies (Alzaeemi et al. 2017; Sathasivam & Abdullah 2007) the data set employed in this computer simulation is simulated data set. Initially, the state of the neurons will be randomized by the program. The main objective of the program was to find the optimum 'model' has that MAX-*k*SAT logical representation. The following algorithms depicts the implementation of the models in the program.

Translate the MAX-*k*SAT clauses into Boolean algebra form (if any); Assign neuron to each variable in MAX-*k*SAT formula; Initialize all the synaptic weight; Check inconsistency of the MAX-*k*SAT logic; Derive cost function for MAX-*k*SAT by assigning $X = \frac{1}{2}(1+S_X)$ and $\overline{X} = \frac{1}{2}(1-S_X)$. The state of the neuron learned true when $S_X = 1$ and false when $S_X = -1$; Check clauses satisfaction of the neurons correspond to the MAX-*k*SAT logic. Maximum consistent interpretation will be stored as content addressable memory (CAM); Obtain synaptic weights correspond to the MAX-*k*SAT logic; Compute the expected global minimum energy supposed to be $H_P^{\min}$ by using (5) or (6); Apply Sathasivam Relaxation method by using (13); Compute the corresponding local field for all neurons by using (3). The local field for KHNN has some modification as shown in Alzaeemi et al. (2017); Find the corresponding final energy by using (5) or (6). Verify whether the final energy is a local minimum energy of global minimum energy.

Each simulation runs 100 trials with 100 neuron combinations in order to reduce statistical error. According to Sathasivam (2010), $Tol = 0.001$ is chosen as an ideal tolerance value for Lyapunov energy since it gives us a better filtering mechanism to distinguish the global minima or local minima solution effectively.

PERFORMANCE EVALUATION METRIC

Under this section, total of four performance evaluation metrics will be investigated in order to analyze the efficiency of the HNN-MAX*k*SAT model. Each of the HNN-MAX*k*SAT model will be evaluated based on global minima ratio, ratio of satisfied clauses, fitness evaluation and CPU time.

GLOBAL MINIMA RATIO

Global minima ratio, *Zm* is defined as the ratio between the total global minimum energy and total number of runs (Sathasivam 2010). Since HNN and KHNN model will produce 10000 solutions per execution, finding *Zm* will be relevant in this study. Each of the computed H_P of the neurons in HNN will be filtered by specific value *Tol* . If the H_P of the model is within the *Tol*, H_P will be considered as global minimum energy. The equation of *Zm* is found to be as follows:

$$Zm = \frac{1}{tc}\sum_{i=1}^{n} N_{H_P} \tag{14}$$

where *t* is the number of trial; *c* is the neuron combination; and N_{H_P} is the number of global minimum energy of the proposed model. According to Sathasivam and Velavan (2014), a particular model is considered robust if the value of *Zm* tends to 1.

RATIO OF SATISFIED CLAUSE

Since it is impossible for MAX-*k*SAT logical rule to be satisfied, ratio of satisfied clause (RSC) of the program will be evaluated. The equation of RSC is as follows (Feige & Goemans 1995):

$$\text{RSC} = \frac{f_{MAXkSAT}}{NC} \tag{15}$$

where *NC* and $f_{MAXkSAT}$ are the total number of clauses and the fitness of MAX*k*SAT solution, respectively, in HNN-MAX*k*SAT model. The capability of the model will be evaluated based on increasing number of variables (number of neurons).

FITNESS LANDSCAPE EVALUATION

When one of the stored pattern ξ^v is given to the HNN-MAX*k*SAT as an initial state, the state of neurons may oscillate. In order for the network to function as an associative memory, the final states false of the neuron *N* must be similar to the initial state. The similarity function of time is defined as (Imada & Araki 1998),

$$m^v(t) = \frac{1}{N}\sum_{i=1}^{N} \xi_i^v s_i^v \tag{16}$$

The fitness function is obtained by averaging the values over all the given MAX-*k*SAT pattern. The fitness

$$f = \frac{1}{t_0 \cdot p}\sum_{t=1}^{t_0}\sum_{v=1}^{p} m^v(t) \tag{17}$$

where, t_0 is fixed to twice the number of neurons (*2N*); and *p* is based on the state of neurons. In this study, different number of neurons are used to test the performance of the network.

CPU TIME

CPU time is defined as a time acquired by a particular model to complete one execution. CPU time signifies the efficiency and the stability of the HNN models. Good HNN-MAX*k*SAT model can complete the whole execution in a shorter period of time. In that sense, each HNN model will be executed with equivalent processor to cancel off the effect of bad sector and memory buildup. Equation of the CPU time is given by Sathasivam (2010),

$$CPU\_Time = Learning\_Time + Retrieval\_Time \tag{18}$$

RESULTS AND DISCUSSION

The performance of simulated program for HNN-MAX*k*SAT models and existing KHNN-MAX*k*SAT models will be compared in terms of global minima ratio, ratio of satisfied clauses, fitness energy landscape and CPU time.

Table 3 delineates the obvious variation in the *Zm* obtained by HNN-MAX2SAT, HNN-MAX3SAT, KHNN-MAX2SAT and KHNN-MAX3SAT. Performance of HNN-MAX*k*SAT can be determined by checking the quality of the energy obtained from the network. *Zm* = 0.9720 for global minima ratio is defined as 9720 global minimum energy and 280 local minimum energy. If the *Zm* of proposed network closer to one, almost all neurons reached the correct final state during retrieval phase. Effective relaxation method by Sathasivam relaxation method stabilize the neuron state during retrieval phase. Stable neuron's state produced by HNN-MAX*k*SAT cause the computed energy to converge to global minimum energy. HNN-MAX2SAT is shown to give a better *Zm* ratio compared to HNN-MAX3SAT due to higher number of variable in MAX-3SAT compared to MAX-2SAT. As the number of neuron increased (number MAX-*k*SAT constraint increased), some of the neuron states retrieved might trapped at local minimum solution (suboptimal solution). HNN-MAX*k*SAT requires more computation time to avoid 'inconsistencies' of MAX-*k*SAT before the network can enter the relaxation phase. Without proper relaxation, neuron is shown to oscillate with their assigned states. The implementation of HNN-MAX*k*SAT has enhanced the quality of the solutions compared to conventional KHNN-MAX*k*SAT. HNN-MAX*k*SAT can stabilize the state of neurons by squashing the collective output from the neurons. This cause the states of the neurons converge to global minimum. Conventional KHNN-MAX*k*SAT is not able to withstand the complexity of 50 neurons. Generally, HNN-MAX*k*SAT is still able to produce more than 80% global minimum solution (correct solution).

Figure 1 shows the RSC value for all HNN-MAX*k*SAT and KHNN-MAX*k*SAT models. The higher the RSC value, the more clauses will be satisfied in any randomized MAX-*k*SAT. Effective neuron's state during retrieval phase by HNN-MAX*k*SAT helps the network to retrieve more correct states. In this case, the amount of satisfied MAX-*k*SAT clauses will be maximized during states retrieval. HNN-MAX*k*SAT integrated is proven to reduce the unforeseen changes in neuron states (spurious states) that may lead to false clauses compared to existing model, KHNN-MAX*k*SAT. The quality of the RSC dropped as the number of clauses increased since the solution of MAX*k*SAT might trapped at suboptimal solution. Suboptimal solution will produce neuron state that has less number of satisfied clauses. Despite that, HNN-MAX*k*SAT managed to produce most of the correct clauses in MAX-*k*SAT. However, KHNN-MAX*k*SAT produced lower RSC value due to the complexity in Kernel Hopfield network.

Table 4 examines the CPU time recorded for all models. HNN-MAX*k*SAT is expected to consume more time compared to other studies such as Kasihmuddin and Sathasivam (2016) and Mansor and Sathasivam (2016). This is due to the fact that MAX*k*SAT representation will

TABLE 3. Global Minima Ratio (*Zm*) of the models

NN	KHNN-MAX2SAT	HNN-MAX2SAT	KHNN-MAX3SAT	HNN-MAX3SAT
10	0.9385	0.9720	0.9208	0.9650
20	0.9070	0.9452	0.9004	0.9399
30	0.8911	0.9285	0.8829	0.9103
40	0.8742	0.9007	0.8640	0.8835
50	0.8446	0.8994	0.8375	0.8760
60	-	0.8744	-	0.8559

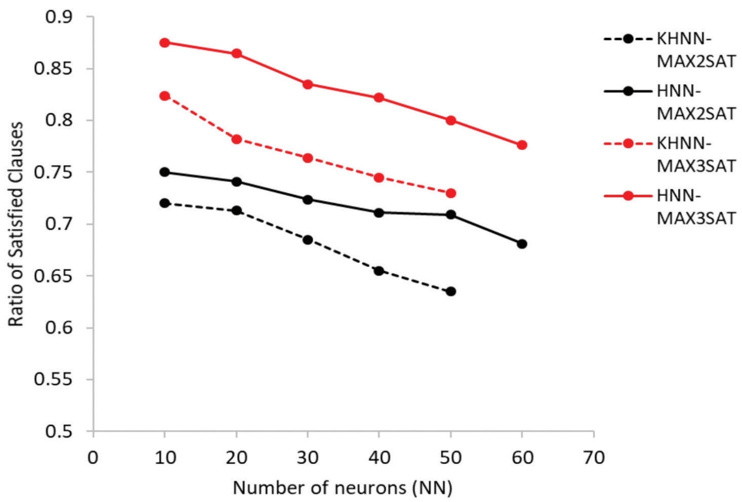

FIGURE 1. Comparison of ratio of satisfied clauses (RSC) for the HNN-MAX*k*SAT and KHNN-MAX*k*SAT

TABLE 4. CPU time for the models

NN	KHNN-MAX2SAT	HNN-MAX2SAT	KHNN-MAX3SAT	HNN-MAX3SAT
10	78	24	92	32
20	332	108	384	159
30	1909	357	2661	482
40	8400	2880	9235	3461
50	25336	11452	29544	13708
60	-	88562	-	99350

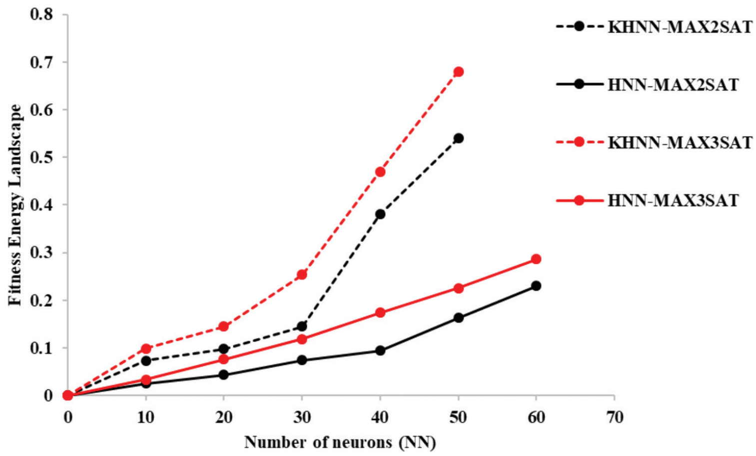

FIGURE 2. Comparison of the fitness energy evaluation for HNN-MAX*k*SAT and KHNN-MAX*k*SAT

never be fully satisfied. During the learning phase, HNN-MAX*k*SAT requires time to search the most consistent interpretation for MAX-*k*SAT representation. Higher number of neurons will increase the search space of HNN-MAX*k*SAT. Hence, the probability of HNN-MAX*k*SAT to avoid logical inconsistencies will reduce dramatically (Abdul Rahman 2017). During retrieval phase, CPU time in HNN-MAX*k*SAT has a slight edge compared to KHNN-MAX*k*SAT. KHNN-MAX*k*SAT only able to compute up to 50 neurons due to the complexity of the network. HNN-MAX*k*SAT is able to reduce neuron oscillation and stabilize the neuron to reach the final state faster. Reduction in neuron oscillation reduce the computation time during retrieval phase. Overall, the HNN remains competent in minimizing the MAX*k*SAT inconsistencies and compute the global solution within the acceptable CPU time.

Figure 2 depicts the fitness energy evaluation obtained for KHNN-MAX2SAT, HNN-MAX2SAT, KHNN-MAX3SAT and HNN-MAX3SAT with different NN. In this study, the fitness energy evaluation is computed based on (16) and (17). According to Figure 2, the fitness energy for the HNN-MAX2SAT and HNN-MAX3SAT are closer to zero for both models. Hence, it indicates the effectiveness of HNN-MAX*k*SAT in synaptic weight computation via the Wan Abdullah method. Thus, the well-defined synaptic weight of HNN-MAX*k*SAT will drive the final state of neuron to the non-oscillatory neuron state. The flatness of the energy landscape in HNN-MAX*k*SAT will reduce the probability of the network to stuck at suboptimal energy (local minima solution). Another point to ponder is as the number of neurons increased, the fitness energy landscape will be higher due to the complexity of the network. This will lead to some solutions to be trapped at spurious states. However, the HNN-MAX*k*SAT is still maintain lower fitness of the neuron states during retrieval phase compared to KHNN-MAX*k*SAT.

CONCLUSION

The work, reported in this paper, showed solid performances of Hopfield Neural Network (HNN) in doing MAX-*k*SAT programming compared to the existing method Kernel Hopfield neural network (KHNN). According to the experimental results, the HNN-MAX*k*SAT outperformed the KHNN-MAX*k*SAT. The proposed HNN-MAX*k*SAT model gives us an acceptable *Zm*, lower CPU time, higher RSC value and lower fitness energy evaluation. For future work, the HNN provides protractile platform for evaluating different variant of satisfiability logic such as majority satisfiability, quantified maximum satisfiability and weighted maximum satisfiability. Additionally, robust metaheuristic techniques such as swarm intelligence and evolutionary algorithms can be added to reduce the complexity of HNN-MAX*k*SAT model during learning phase.

REFERENCES

Abdul Rahman, N. 2017. Binomial distribution of 2sat interpretation. Private Communication.

Alzaeemi, S.A., Sathasivam, S., Adebayo, S.A., Kasihmuddin, M.S.M. & Mansor, M.A. 2017. Kernel machine to doing logic programming in hopfield network for solve non horn problem-3sat. *MOJ Applied Bionics and Biomechanics* 1(1): 1-6.

Berg, J. & Järvisalo, M. 2015. Cost-optimal constrained correlation clustering via weighted partial maximum satisfiability. *Artificial Intelligence* 244: 110-142.

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

Bouhmala, N. 2016. A variable neighbourhood search structure based-genetic algorithm for combinatorial optimisation problems. *International Journal of Intelligent Systems Technologies and Applications* 15(2): 127-146.

Feige, U. & Goemans, M. 1995. Approximating the value of two power proof systems, with applications to max 2sat and max dicut. In *Theory of Computing and Systems Proceedings, Third Israel Symposium*. pp. 182-189.

Hamadneh, N., Sathasivam, S., Tilahun, S.L. & Choon, O.H. 2012. Learning logic programming in radial basis function network via genetic algorithm. *Journal of Applied Sciences* 12(9): 840-847.

Hölldobler, S., Kalinke, Y. & Störr, H.P. 1999. Approximating the semantics of logic programs by recurrent neural networks. *Applied Intelligence* 11(1): 45-58.

Hopfield, J.J. & Tank, D.W. 1985. Neural computation of decisions in optimization problems. *Biological Cybernatics* 52: 141-152.

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

Imada, A. & Araki, K. 1998. What does the landscape of a Hopfield associative memory look like? *International Conference on Evolutionary Programming*. pp. 647-656.

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

Kasihmuddin, M.S.M. & Sathasivam, S. 2016. Accelerating activation function in higher order logic programming. *Advance in Industrial and Applied Mathematics: Proceedings of 23rd Malaysian National Symposium of Mathematical Sciences (SKSM23)* 1750: 030008-1–030008-6 https://doi.org/10.1063/1.4954544.

Kasihmuddin, M.S.M., Mansor, M.A. & Sathasivam, S. 2017. Hybrid genetic algorithm in the Hopfield network for logic satisfiability problem. *Pertanika Journal of Science & Technology* 25(1): 139-152.

Kowalski, R.A. 1979. *Logic for Problem Solving*. New York: Elsevier Science Publishing,

Kumar, S. & Singh, M.P. 2010. Pattern recall analysis of the Hopfield network with a genetic algorithm. *Computer and Mathematic with Applications* 60: 1049-1057.

Layeb, A., Deneche, A.H. & Meshoul, S. 2010. A new artificial immune system for solving the maximum satisfiability problem. In *Trends in Applied Intelligent Systems*. Berlin, Heidelberg: Springer. pp. 136-142.

Liu, S. & de Melo, G. 2017. Should algorithms for random sat and max-sat be different? *AAAI*. pp. 3915-3921.

Madsen, B.A. & Rossmanith, P. 2004. Maximum exact satisfiability: NP-completeness proofs and exact algorithms. *BRICS Report Series* 04(19): 1-20.

Mansor, M.A. & Sathasivam, S. 2016. Performance analysis of activation function in higher order logic programming. *Advance in Industrial and Applied Mathematics: Proceedings of 23rd Malaysian National Symposium of Mathematical Sciences (SKSM23)* 1750: 030007-1–030007-7 https://doi.org/10.1063/1.4954543.

Mathias, A.C. & Rech, P.C. 2012. Hopfield neural network: The Hyperbolic tangent and the piecewise-linear activation functions. *Neural Networks* 34: 42-45.

Menaï, M.E.B. & Batouche, M. 2005. A backbone-based co-evolutionary heuristic for partial MAX-SAT. In *Artificial Evolution*. Berlin Heidelberg: Springer. pp. 155-166.

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

Pinkas, G. & Dechter, R. 1995. Improving connectionist energy minimization, *J. Artif. Intell. Res. (JAIR)* 3: 223-248.

Pinkas, G. 1991. Symmetric neural networks and propositional logic satisfiability. *Neural Computation* 3(2): 282-291.

Pipatsrisawat, K., Palyan, A., Chavira, M., Choi, A. & Darwiche, A. 2008. Solving weighted Max-SAT problems in a reduced search space: A performance analysis. *Journal on Satisfiability, Boolean Modeling and Computation* 4: 191-217.

Raman, V., Ravikumar, B. & Rao, S.S. 1998. A simplified NP-complete MAXSAT problem. *Information Processing Letters* 65(1): 1-6.

Rojas, R. 1996. *Neural Networks: A Systematic Introduction*. Berlin: Springer.

Santra, S., Quiroz, G., Ver Steeg, G. & Lida, D.A. 2014. Max 2-sat with up to 108 qubits. *New Journal of Physics* 16(4): 045006.

Sathasivam, S. & Fen, N.P. 2013. Developing agent based modelling for doing logic programming in Hopfield network. *Applied Mathematical Sciences* 7(1): 23-35.

Sathasivam, S. & Velavan, M. 2014. Boltzmann Machine and Hyperbolic Activation Function in Higher Order Network. *Modern Applied Science* 8(3): 140.

Sathasivam, S. & Wan Abdullah, A.T.W. 2007. Flatness of the energy landscape for horn clauses. *Matematika* 23: 147-156.

Sathasivam, S. 2008. Learning in the recurrent Hopfield network. *Computer Graphics, Imaging and Visualisation, CGIV'08. Fifth International Conference on IEEE*. pp. 323-328.

Sathasivam, S. 2010. Upgrading logic programming in Hopfield network. *Sains Malaysiana* 39(1): 115-118.

Sathasivam, S. 2011. Boltzmann machine and new activation function comparison. *Applied Mathematical Sciences* 5(78): 3853-3860.

Sulehria, H.K. & Zhang, Y. 2007. Hopfield neural networks: A survey. *Proceedings of the 6th Conference on 6th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases*. pp. 125-130.

Velavan, M., Yahya, Z.R., Abdul Halif, M.N. & Sathasivam, S. 2016. Mean field theory in doing logic programming using hopfield network. *Modern Applied Science* 10(1): 154-160.

Wan Abdullah, A.T.W. 1992. Logic programming on a neural network. *International Journal of Intelligent Systems* 7(6): 513-519.

Wan Abdullah, A.T.W. 1993. The logic of neural networks. *Physics Letters A* 176(3-4): 202-206.

Yannakakis, M. 1994. On the approximation of maximum satisfiability. *Journal of Algorithms* 17(3): 475-502.

Zeng, X. & Martinez, T. 1999. A new relaxation procedure in the hopfield network for solving optimization problems. *Neural Processing Letters* 10(3): 211-222.

Zhang, H., Hou, Y., Zhao, J., Wang, L., Xi, T. & Li, Y. 2017. 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: 1035-1043.

Mohd Shareduwan Mohd Kasihmuddin* & Saratha Sathasivam
Pusat Pengajian Sains Matematik
Universiti Sains Malaysia
11800 USM, Pulau Pinang
Malaysia

Mohd Asyraf Mansor
Pusat Pengajian Pendidikan Jarak Jauh
Universiti Sains Malaysia
11800 USM, Pulau Pinang
Malaysia

*Corresponding author; email: shareduwan@usm.my

Received: 19 August 2017
Accepted: 19 January 2018