

***k* Satisfiability Programming by using Estimation of Distribution Algorithm in Hopfield Neural Network**

Norul Fazira Ahmad Rasli^{1, a)}, Mohd Shareduwan Mohd Kasihmuddin^{2, b)}, Mohd. Asyraf Mansor^{3, c)}, Md Faisal Md Basir^{4, d)} and Saratha Sathasivam^{5, e)}

^{1,2,5} School of Mathematical Sciences, Universiti Sains Malaysia,
11800 Minden, Pulau Pinang, Malaysia

³ School of Distance Education, Universiti Sains Malaysia,
11800 Minden, Pulau Pinang, Malaysia

⁴ Department of Mathematical Sciences, Faculty of Science, Universiti Teknologi Malaysia,
81310 Johor Bahru, Johor, Malaysia

^{b)}Corresponding author: shareduwan@usm.my

^{a)}norulfazira95@gmail.com

^{c)}asyrafman@usm.my

^{d)}mfaisalmbasir@utm.my

^{e)}saratha@usm.my

Abstract. Hopfield Neural Network (HNN) is a sort of neural network that is strongly dependent to energy minimization of solution. Although HNN managed to solve various optimization problem, the output of HNN suffered from a lack of interpretability and variation. This has severely limited the practical usability of HNN in doing logic programming. Inspired by random neuron perturbation, Estimation of Distribution Algorithm (EDA) has been proposed to explore various optimal neuron state. EDAs employs a probabilistic model to sample the neuron state in order to move toward the various optimal location of global minimum energy. In this paper, a new Mutation Hopfield Neural Network (MHNN) will be proposed to do *k* Satisfiability programming. Based on the experimental result, the proposed MHNN has outperformed conventional HNN in various performance metric.

Keywords: Estimation of Distribution Algorithm, Hopfield Neural Network, Satisfiability Logic Programming, Probabilistic Model, Mutation

INTRODUCTION

Due to the tremendous increase of optimization demand in various disciplines, researchers are continuously seeking comprehensive paradigm to address wide range of NP problem. Hopfield Neural Network (HNN) is the simplest network that can be used to solve numerous optimization problem [1]. Configurative speaking, HNN is a recurrent neural network invented by John Hopfield [2] which made a compelling impact in the field of Artificial Neural Network (ANN). There were two school of thoughts concerning the approach of ANN. There are researchers who consider ANN as a black box model or symbolic system. In this case, symbolic system increases the interpretability of HNN in doing various real-life problem [3]. In that regard, logic learning in HNN has been a primary work of [4] and the benefit of the two field of knowledge has been integrated as a single intelligent unit. This work proposed an optimized logic learning through synaptic weight called Wan Abdullah method. On a broader perspective, logic programming represents symbolic knowledge that will be “learned” by HNN model. The perceived knowledge will be stored and retrieved according to the given constraint optimization problem. The pursuit of creating an

optimized HNN model were critically harnessed by several researchers [5, 6]. Several interesting logical rules such as k Satisfiability (k SAT) [7], Maximum Satisfiability [8] have successfully embedded to HNN. The proposed HNN model managed to achieve more than 80% of global minimum energy. Hence, the next perspective has revealed an interesting question about the structure of the hybrid HNN that is, what is the global search capability of HNN model? Unfortunately, output from HNN experienced lack of variety and HNN only explores few solution spaces. In this case, high percentage of solution produced by HNN model will be shifted towards a single solution.

EDA has been widely introduced in various optimization problem to overcome the local minima problem of HNN. In the proposed model, probability model act as mutation operator that perturbed the current solution and move the current solution to a point beyond the neighborhood searched by the HNN. The perturbation of EDA can generate a new starting point to possibly find other global solution. Hu *et al.* [9] has proposed the combination of EDA and HNN in solving aircraft landing scheduling (ALS). The proposed hybrid HNN yielded a better performance than other conventional method such as standalone HNN and genetic algorithm. In this paper, we incorporate the global search ability of EDA into the HNN; which typically has a power local search capability. Therefore, a new mutation Hopfield Neural Network (MHNN) has been proposed to increase the number of global minimum solution as well as to increase the variety of final neuron state. This paper is organized as follows. In Section 2, a brief framework of HNN is discussed and in Section 3, 2 Satisfiability programming as a logical rule is implemented in HNN. Meanwhile, Section 4 contains the explanation regarding the structure of Estimation of Distribution Algorithm (EDA). Section 5 and 6 simulate the capability of MHNN model and concluding remarks regarding this work.

k SATISFIABILITY PROGRAMMING

k SAT representation is considered as a NP problem or non-deterministic problem. The three components of k SAT are summarized by [10] as follows:

1. Consist of a set of variables, x_1, x_2, \dots, x_m
2. A set of literals. A literal is a variable or a negation of a variable, connected by OR (\vee) operator.
3. A set of n distinct clauses: C_1, C_2, \dots, C_n . Each clause consists of only literals combined by just logical AND (\wedge). Each clause must consist of k variables.

Each of the variable can only take bipolar value which is 1 or -1 that exemplified the idea of True and False. The goal of k SAT logical rule is to determine whether there exists an assignment of truth values to variables that makes the following formula satisfiable. The flexibility of k SAT logical rule will make it compatible to the structure of HNN.

HOPFIELD NEURAL NETWORK

Hopfield Neural Network (HNN) is a simple flexible structure which contain associative content-addressable memory. Design of this artificial neural network can memorize huge amount of information and reviewing the same from available data [11]. Structurally, HNN consist of a set of N interconnected neuron with symmetrical synaptic weight and non-feedback connections. The general updating rule of HNN is given as follows:

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

where G_{ij} is the synaptic weight from unit i to j . S_j is the state of neuron j and $\theta = 0$ [12] to ensure the energy of the network decrease monotonically. The neurons have bipolar representation $\{1, -1\}$ where 1 is considered as True and -1 is considered as False. k SAT can be implemented in HNN by assigning variable to neuron. In this paper, HNN model will consider $k = 2$ which allows the network to retrieve 2 dimensional neuron configuration.

The local field of HNN is denoted by

$$h_i(t) = \sum_{j=1, i \neq j}^N G_{ij}^{(2)} S_j + G_i^{(1)} \quad (2)$$

The final neuron state can be classified based on the following

$$S_i(t+1) = \begin{cases} 1, & \sum_{j=1, i \neq j}^N G_{ij}^{(2)} S_j + G_i^{(1)} \geq 0 \\ -1, & \sum_{j=1, i \neq j}^N G_{ij}^{(2)} S_j + G_i^{(1)} < 0 \end{cases} \quad (3)$$

The Lyapunov energy function for the HNN-2SAT is given as

$$H_p = -\frac{1}{2} \sum_{i=1, i \neq j}^N \sum_{j=1, i \neq j}^N G_{ij}^{(2)} S_i S_j - \sum_{i=1}^N G_i^{(1)} S_i \quad (4)$$

where the Lyapunov energy function is decrease monotonically [13].

ESTIMATION OF DISTRIBUTION ALGORITHM

EDA is an evolutionary algorithm that use information obtained during the optimization process to build probabilistic models of the distribution of good regions in the search space and use these models to generate new solutions [14]. In this paper, the EDA will be implemented during the retrieval phase of HNN-2SAT. Conventional HNN has a high possibility to be trapped in local minima as the number of neurons involved increased. The local minima problem is caused by neuron oscillation of HNN. In this case, EDA will perturb the current neuron state and move the current neuron state to a point beyond the neighborhood searched by the HNN. Structurally, EDA will probabilistically explore more solution space in order to allow to escape from the current local optimum [9]. Univariate Marginal Gaussian distribution (UMG) is a primary probabilistic model that alter the final neuron state based on the probability value. Supplemental details of UMG are available in [15]. The combination of EDA and HNN is represented as Mutation Hopfield Neural Network or MHNN. The steps involved in MHNN in given as follows

Step 1: Given a logic program, translate all the clauses in the logic program into Boolean algebra.

Step 2: The input of HNN $S_i = S_{i,1}, S_{i,2}, \dots, S_{i,N}$ is randomly initialized.

Step 3: The output $h_i = (h_1, h_2, h_3, \dots, h_N)$ of the HNN for every $S_i = S_{i,1}, S_{i,2}, \dots, S_{i,N}$ is computed

Step 4 Calculate mean, variance and joint probability distribution function (JPDF) are defined as [16]:

Step 5: Normalize the joint probability density function (JPDF).

Step 6: Obtain the new state $S_i^{M_i}$ by using $h_i^{M_i}$. The best solution of $S_i^{M_i}$ will be updated using Roulette wheels selection [17].

Step 7: New output $S_i^{M_i}$ based on $h_i^{M_i}$ will be retained.

Step 8: Step 2,3,4,5,6,7 and 8 are repeated until $f_{2SATES} = NC$ where f_{2SATES} is the solution of MHNN-2SAT and NC is the total number of 2SAT clause.

Generally, the implementation of EDA in HNN during retrieval phase can be shown in Figure 1.

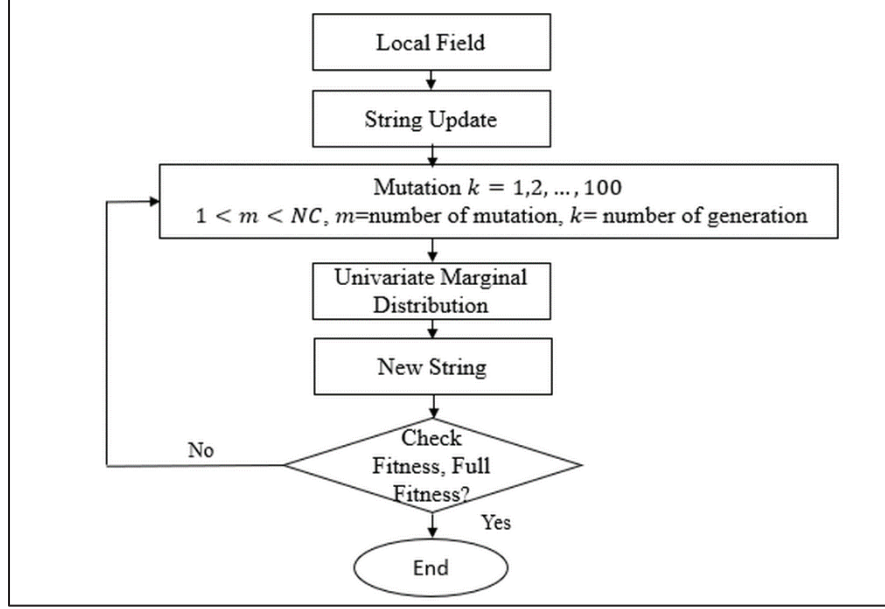


FIGURE 1. Flowchart of MHNN-2SATES

PERFORMANCE EVALUATION METRIC

To test the effectiveness of the proposed method, the performance of all HNN will be evaluated based on error analysis, energy analysis and similarity analysis of the retrieval neurons. The equation for Global Minima Ratio [18], analysis is given follows:

$$Z_m = \frac{1}{tC} \sum_{i=1}^n N_{H_p} \quad (5)$$

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 [19]. The equation for root mean squared error (RMSE) is given follows:

$$RMSE = \sum_{i=1}^n \sqrt{\frac{1}{n} (H_{\min}^P - H_i^P)^2} \quad (6)$$

where the global minimum energy H_i^P is given in equation (4).

Benchmark State

The key component of analyzing the final state of neuron is by comparing the retrieved state with an “ideal” neuron state. In this section, analysis of final state of neuron in HNN-2SAT model will be studied. Benchmark state is defined as the ideal neuron state retrieved from the HNN model. The benchmark neuron state is given as follows:

$$S_i = \begin{cases} 1 & , M \\ -1 & , -M \end{cases} \quad (7)$$

where M and $\neg M$ are positive and negative literal in 2SAT formula respectively. Consider the logical rule reads $P_{2SAT} = (K \vee L) \wedge (\neg M \vee N) \wedge (O \vee \neg P)$, the benchmark state of the neuron is given as $S_K = 1, S_L = 1, S_M = -1, S_N = 1, S_O = 1, S_P = -1$ or $S_i^{\max} = (1, 1, -1, 1, 1, -1)$. Worth mentioning that the final energy

of $S_i^{M_i}$ is always global minimum solution or $\left| H_{P_{S_i^{\max}}} - H_{P_{S_i^{\min}}} \right| \leq \xi$ [10]. ξ is a tolerance value for energy difference in HNN. Since most of the neuron state retrieved in HNN achieve global minimum energy [7], S_i^{\max} is a perfect benchmark state in comparing the final state of different HNN model.

Similarity Metrics

Analyzing the behavior of the final neuron state is a challenging task. In this section, the final neuron state retrieved that corresponds to 2SAT logical rule will be analyzed by using similarity metrics. Several similarity metrics were identified to explore the lack of variation of HNN models. In this case, instead of comparing logic with logic, the comparison will be made based on the individual neuron state. Hence, the general comparison between benchmark state and the final neuron state is as follows:

$$C_{S_i^{\max} S_i} = \{(S_i^{\max}, S_i) | i = 1, 2, \dots, n\} \quad (8)$$

The further specification of the variable is defined as follows:

p is the number of (S_i^{\max}, S_i) where both elements have the value 1 in $C_{S_i^{\max} S_i}$.

q is the number of (S_i^{\max}, S_i) where S_i^{\max} is 1 and S_i is -1 in $C_{S_i^{\max} S_i}$.

r is the number of (S_i^{\max}, S_i) where S_i^{\max} is -1 and S_i is 1 in $C_{S_i^{\max} S_i}$.

s is the number of (S_i^{\max}, S_i) where both elements have the value -1 in $C_{S_i^{\max} S_i}$.

The size of the neuron string is given as $n = p + q + r + s$. By using the above information, similarity coefficient for all HNN model is given as follows
Jaccard's Index [20]

$$J(S_i^{\max}, S_i) = \frac{p}{p + q + r} \quad (9)$$

Sokal and Sneath-2 [21]

$$SS(S_i^{\max}, S_i) = \frac{p}{p + 2(q + r)} \quad (10)$$

Worth mentioning that, high similarity index signifies low variation of final neuron state compared to benchmark neuron state. The aim of this paper is to find HNN model that has the highest variation of final neuron state. Based on the experimental result, the proposed MHNN has outperformed conventional HNN in various performance metrics such as Global Minima Ratio, Root Mean Square Error, Jaccard and Sokal and Sneath-2 Index.

RESULTS AND DISCUSSION

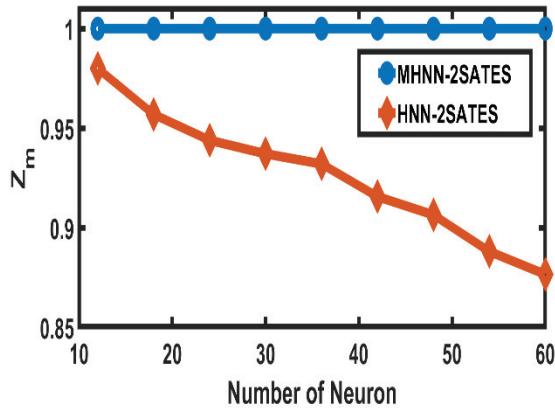


FIGURE 2. Global Minima Ratio for HNN-2SATES Models

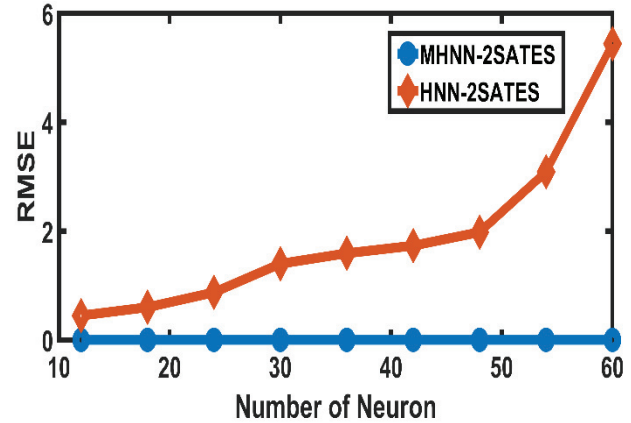


FIGURE 3. RMSE for HNN-2SATES Models

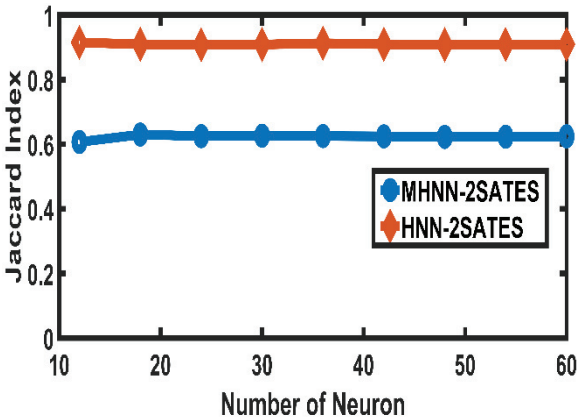


FIGURE 4. Jaccard Index for HNN-2SATES Models

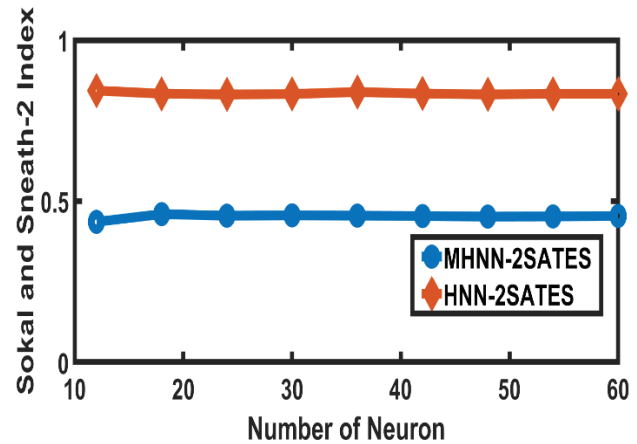


FIGURE 5. Sokal and Sneath-2 Index for HNN-2SATES Models

We used Dev C++ as a platform to simulate and train the programs. In our analysis, we presented HNN-2SATES [10] and MHNN-2SATES in doing 2SAT logic programming. We use Global Minima Ratio and RMSE to evaluate the performance of both HNN-2SATES model. Meanwhile, we use similarity analysis such as Jaccard Index and Sokal and Sneath-2 Index to find the model that can produce high variation of final neuron state. The result in Figure 2 until Figure 4 allow the following observations:

1. Figure 2 illustrate the graph for the global minima ratio, Z_m from the computer simulation that we have carried out. Z_m for HNN-2SATES decrease rapidly as number of neuron increases. This is due to the complexity of the HNN-2SATES intensified and more solution trapped in local minima. On the other hand, Z_m for MHNN-2SATES approaching 1 because EDA utilizes probabilistic model to achieved global solution. The mutation helps the neuron to escape the stable state of HNN an explore other global solution.
2. From the data obtained in Figure. 3, it is proven that MHNN-2SATES give more accurate solutions since the testing error are much closer to zero compared to HNN-2SATES. This is due to the higher stability of the neurons during the retrieval phase. Meanwhile, error for HNN-2SATES keep increasing as the number of neurons increase because it traps in trial and search.

3. From the graphs in Figure. 4 and Figure. 5, MHNN-2SATES will have less similar index value compared to HNN-2SATES. Therefore, it can be concluded that MHNN-2SATES has achieved our objective in order to increase the variation of the solution and reduce overfitting of final neuron states.
4. In this case, the neuron retrieved from MHNN-2SATES has a lowest similarity with the benchmark state. HNN-2SATES has the highest similarity with the benchmark state. In this case, the HNN-2SATES fails to explore more state that lead to global minimum energy.
5. MHNN-2SATES demonstrates higher value of variation compared to HNN-2SATES model. The variation value shows that MHNN-2SATES can locate other state that leads to global minimum energy. The updating rule of MHNN-2SATES may increase the accuracy of HNN models.

CONCLUSION

We have presented EDA in HNN to reduce overfitting and to increase the variation of final neuron states. We can validate that, finding optimal solution for HNN-2SAT problem can be efficient and accurate by using EDA mutation operator (MHNN-2SATES) compared to conventional HNN (HNN-2SATES). This finding was supported by good performance of MHNN-2SATES in term of Z_m , RMSE and similarity analysis. In this paper, the similarity index does not consider pair of $(S_i^{\max}, S_i) = (-1, -1)$. In that regard, other similarity index such as Kulczynski index [22] will be able to evaluate the mentioned pair. It will be interesting to note that, the capability of MHNN will be more transparent if Kulczynski index has been used.

ACKNOWLEDGMENTS

This research was supported by Fundamental Research Grant Scheme (FRGS), Ministry of Education Malaysia, grant number 205/PMATH/6711801 and Universiti Sains Malaysia (USM).

REFERENCES

1. H. K. Sulehria, and Y. Zhang, "Hopfield neural networks: a survey," Proceedings of the 6th Conference on 6th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases, (2007), pp. 125-130.
2. J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," Proceedings of the national academy of sciences, (1982), pp. 2554-2558.
3. M. S. M. Kasihmuddin, M. A. Mansor, and S. Sathasivam, "Satisfiability based reverse analysis method in diabetes detection," *AIP Conference Proceedings*, (Vol. 1974), (AIP Publishing, 2018), pp. 020020.
4. W. A. T. Wan Abdullah, *International journal of intelligent systems* 7, 513–519 (1992).
5. M. A. Mansor, M. S. M. Kasihmuddin, and S. Sathasivam, *International Journal of Intelligent Systems and Applications (IJISA)* 8, 27 (2016).
6. M. A. Mansor, M. S. M. Kasihmuddin, and S. Sathasivam, *International Journal of Intelligent Systems and Applications (IJISA)* 8, 22-29 (2016).
7. M. S. M. Kasihmuddin, M. A. Mansor, and S. Sathasivam, *Pertanika Journal of Science & Technology* 25, (2017).
8. M. S. M. Kasihmuddin, M. A. Mansor, and S. Sathasivam, *International Journal of Interactive Multimedia & Artificial Intelligence* 4, (2016).
9. L. Hu., F. Sun, H. Xu, H. Liu, and X. Zhang, *Information Sciences* 181, 92-105 (2011).
10. M. S. M. Kasihmuddin, "Satisfiability Logic Programming Incorporating Metaheuristics in Hopfield Neural Networks", PhD Thesis, Universiti Sains Malaysia, Malaysia, (2017).
11. T. Deb, A. K. Ghosh, and A. Mukherjee, "Singular value Decomposition applied to Associative Memory of Hopfield Neural Network," *Materials Today, Proceedings*, (2018), pp. 2222-2228.
12. A. Barra, M. Beccaria, and A. Fachechi, *Neural Networks* 106, 205-222 (2018).
13. H. Yan, L. Zhao, L. Hu, X. Wang, E. Wang, and J. Wang, "Nonequilibrium landscape theory of neural networks," *Proceedings of the National Academy of Sciences*, (2013), pp. 4185-4194.
14. M. Pelikan, D. E. Goldberg, and F. G. Lobo, *Computational optimization and applications* 21, 5-20 (2002).

15. Y. Wang, and B. Li, "A restart univariate estimation of distribution algorithm: sampling under mixed Gaussian and Lévy probability distribution," IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), (IEEE, 2008), pp. 3917-3924.
16. S. Gao, and C.W. de Silva, [Applied Mathematics and Computation](#) **339**, 323-345 (2018).
17. D. E. Goldberg and K. Deb, "A comparative analysis of selection schemes used in genetic algorithms," Foundations of genetic algorithms, (Vol. 1), (Elsevier, 1991), pp. 69-93.
18. S. Sathasivam, *Sains Malaysiana* **3**, 115-118 (2010).
19. M. S. M. Kasihmuddin, M. A. Mansor, and S. Sathasivam, *Sains Malaysiana* **47**, 1327-1335 (2018).
20. S. Bag, S. K. Kumar, and M. K. Tiwari, *Information Sciences*, (2019).
21. M. Pachayappan, and R. Panneerselvam, "A Comparative Investigation of Similarity Coefficients Applied to the Cell Formation Problem using Hybrid Clustering Algorithms," *Materials Today: Proceedings*, (2018), pp. 12285-12302.
22. S. Kulczynski, "Classe des sciences mathématiques et naturelles," *Bulletin International de l'Academie Polonaise des Sciences et des Lettres*, (1927), pp. 57-203.