International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249 – 8958 (Online), Volume-9 Issue-4, April, 2020

Hybrid Genetic Algorithm Model in Neuro Symbolic Integration

Shehab Abdulhabib Alzaeemi, Saratha Sathasivam, Muraly Velavan

Abstract: ***The development of artificial neural network and logic programming plays an important part in neural network studies. Genetic Algorithm (GA) is one of the escorted randomly searching technicality that uses evolutional concepts of the natural election as a stimulus to solve the computational problems. The essential purposes behind the studies of the evolutional system is for developing adaptive search techniques which are robust. In this paper, GA is merged with agent based modeling (ABM) by using specified proceedings to optimise the states of neurons and energy function in the Hopfield neural network (HNN). Hence, it is observed that the GA provides the best solutions in affirming optimal states of neurons and thus, enhancing the performance of Horn Satisfiability logical program (HornSAT) in Hopfield neural network. This is due to the fact that the GA lesser susceptive to be restricted in the local optimal or in any suboptimal solutions. NETLOGO version 6.0 will be used as a dynamic platform to test our proposed model. Hence, the computer simulations will be carried out to substantiate and authenticate the efficiency of the proposed model. The results are then tabulated by evaluating the global minimum ratio, computational time and hamming distance.***

Keywords : Logic Program, Genetic Algorithm, Hopfield Neural Network, Horn Satisfiability, NETLOGO, Agent Based Modelling.

I. INTRODUCTION

The biological and performance structure of the neural network has inspired new models to compute performing tasks for the recognition of patterns type problems. Nevertheless, those computing models aren't expecting for reaching the best levels of biological networks because of numerous reasons. One of them, the operation of biological neurons and the neural interconnection aren't completely understood until this extremely days [1].One of the customary neural networks is the Hopfield Neural Network (HNN), that was launched by John Hopfield. HNN is the recurrent neural network prominent to outstand powerful in storage, memory, and learning [2]. Its construction consists of a two-dimensional as input and output connecter the neural network in which connecting the strengths of the synaptic weights between neurons are judged based on restrictions and solution base of the optimization problems to be solved [3].

Revised Manuscript Received on April 25, 2019.
* Corresponding Author
Shehab Abdulhabib Alzaeemi1*, School of Mathematical Science, Universiti Sains Malaysia, 11800 USM, Penang Malaysia. Email: saratha@usm.my
Saratha Sathasivam, School of Mathematical Science, Universiti Sains Malaysia, 11800 USM, Penang Malaysia. Email: saratha@usm.my
Muraly Velavan, School of General and Foundation Studies, AIMST University, 08100 Semeling, Kedah.

II. RESEARCH METHOD

II.1 Hopfield Neural Network (HNN)

HNN founded by John Hopfield in 1982 [4], the HNN model is broadly used to illustrated numerous optimisation problems. Strictly speaking, the associated units in HNN are identified as the bipolar threshold unit [5], that consideration bipolar values as [-1,1]. Thus, a_i is the i^{th} neuron activation with the follow threshold fragmented function:

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

whereas W_{ij} is synaptic weights from unit j to unit i and S_j is the neuron state of j. Next, ξ_i is the unit threshold i. The network constitutes of N neurons, each is characterized by an Ising spin variable. The link of the neuron in Hopfield network is not self-connection $W_{ii} = W_{jj} = 0$. Thus it makes the Hopfield net connections became bidirectional or symmetric [6]. The state of the neurons in Hopfield net is bipolar as $S_i \in \{1,-1\}$ and follow the dynamic $S_i \to \text{sgn}(h_i)$ whereas h_i is local field in the Hopfield network for the connections of the neurons.

The local field of HNN adjusted according to this equation when deals with higher-levels connection:

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

where W_{ij} is the synaptic weight from neuron i to neuron j.

The connections is symmetric and zero-diagonal $W_{ij} = W_{ji}, W_{ii} = 0$.

The discrete Hopfield networks energy function for Horn Satisfiability (*HornSAT*) clauses are given by the following equation:

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

This energy function of the neural network is vital due to it establishes the convergence degree of the network [7]. Since the synaptic weights of units in HNN is continually symmetric the updating rule of the state is declaring in the follow equation [8]

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

This method is known as Wan Abdullah method [8].

II.2 Neural-Symbolic Integration

In the past, it was quite uncertain whether neural or symbolic procedures alone will be adequate to provide a comprehensive framework for intelligent dealings [9]. Neural symbolic integral take in both the connection system and systems of symbolic artificial intelligence (AI). Web mining is a common example of application that integrates both paradigms. Litman [10] gives the neural symbolic integration cycle as shown in Fig. 1 as the follows:

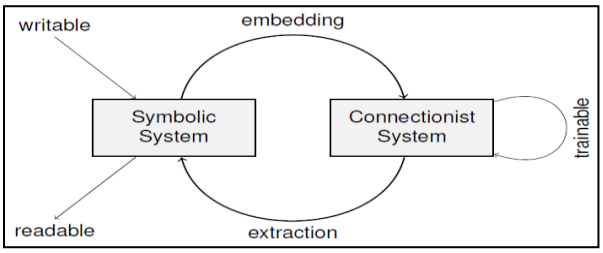

Fig. 1 Neural-Symbolic cycles [10]

II.3 Horn Satisfiability Logic Programming (HornSAT)

In this study, HornSAT logic is a logic to decide the satisfiability of sets of clauses. Horn clauses are clauses that have at most one positive literals. HornSAT Boolean variable can take the value of either -1 (false) or 1 (true). The followed components of HornSAT logic:

(a) It consists of a group n variables as $x_1, x_2 ..., x_n$

(b) Consists a collection of literals in the clauses. Variable x or its negation $\overline{x}$.

(c) A group of m feature clauses: $L_1, L_2, ..., L_m$ . . Each clauses combined by just $\wedge$ logic AND. Every clause composed of literals joint by just $\vee$ logic OR.

The aim of HornSAT logical is to decide if there subsists an assignment of the true value of the variables in the clauses that make the P become satisfiability. HornSAT is a main impulsion in this paper as show in the following example of HornSAT logical programming,

$$P = A \leftarrow B, C \wedge D \leftarrow B \wedge C \quad (5)$$

The goal will be given

$$\leftarrow G$$

$A \vee \neg B \vee \neg C$, $D \vee \neg B$, and C are sequentially transmuted from (4) and G is a conjunction or the goal of neurons[10].

II.4 HornSAT Logic Programming in Hopfield Neural Network (HNN-HornSAT)

HNN is been utilized in an assignment for doing the HornSAT logical model because of its ability for solving constrained optimisation problems. The procedure below shows steps involved in integrating Genetic Algorithm (GA) in the Hopfield neural network (HNN) for doing Horn Satisfiability logic programming (HornSAT):

Step 1

Interpret each HornSAT clauses in the logical programming that given in equation (5) in elementary Boolean algebraic form.

Step 2

Initialization of the neurons to individually ground neurons.

Step 3

Initialization every synaptic weights values to zero.

Step 4

Determine the cost function which is be linked with the negation clauses, multiplication performed a conjunction logical connective and addition performed a disjunction connective.

Step 5

Compute the synaptic weights of the neurons and stored as content addressable memory (CAM) as a building block HNN during the retrieval phase as the “correct” synaptic weight configuration that agrees to logical rules.

Step 6

Checking the clauses satisfactions by utilizing Genetic Algorithm and stored each satisfied clauses in the network.

Step 7

Randomise the state of neuron to provide chance to Hopfield neural network to be satisfiable with learned interpretation and this interpretation will be stored as Content Address Memory (CAM).

Step 8

Applying Sathasivam’s relaxation method [11] to the network to aid in controlling the energy relaxation process by adjusting the relaxation speed so that solutions with better fitnesses are attained.

Step 9

If the state of the neuron in HNN remainders unchanged for 5 runs, it is classified as a stable state [12] and compute the corresponding local field by utilizing the equation (2).

Step 10

Computing the final energy for the stable state by finding the final energy by utilizing the equation (3) [13, 14].

Step 11

The model of HNN-HornSATGA will be evaluated by using appropriate performance evaluation metrics of global minimum ratio, CPU time (computational time) and hamming distance.

II.5 Genetic Algorithm (GA)

Genetic algorithm (GA) is one of a successful evolutionary algorithm for optimisation and learning based-on several physiological evolution features [15]. GA requires principally five ingredients:

a) A way to encode chromosomes problem solutions

b) The evaluation function which reclassifies each specific chromosome

c) Method to develop the chromosome population

d) Factors that can be applied to parent when they reproduce for modifying their genetic make up.

e) Setting the parameters, operators, for algorithm and others.

The implementation of GA in HNN-HornSAT framework is presented as follow:

Stage 1: Initialization

Randomized 100 chromosomes populations as the interpretation (bit strings). Consequently, the interpretation probable of HNN-HornSAT will be illustrated by the chromosomes.

Stage 2: Fitness Evaluation

Computed chromosomes fitness based on the number of satisfaction clauses in every of the expositions. The effectiveness of the process of training will be settled by the maxima fitness.

Stage 3: Stage of Selection

Noted 10 candidate chromosomes from the 100 chromosomes have maxima fitness will proceed to follow generation and stage of GA. Following this, the elected chromosomes will be implemented for crossover procedure to magnify the variability and fitness.

Stage 4: Crossover Procedure

Crossover operative involves the principal transmutation step in the GA. During this phase, the substitution of information of the genets between the two substructure of the chromosomes (bit strings) as shown in the follow example will be carried out:

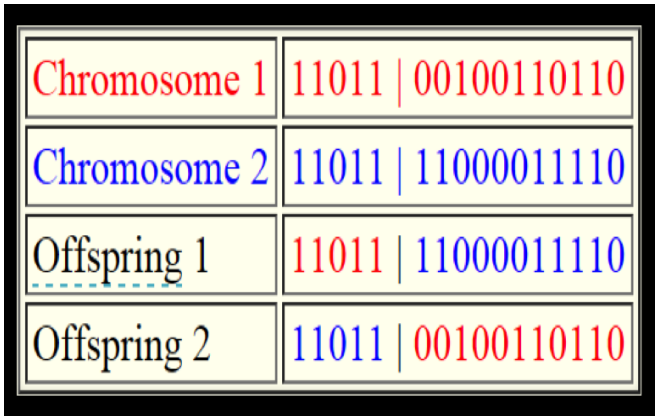

Fig. 2: Examples of crossover [15, 26]

The crossover period chromosomes are represented mutably to suffer the chromosome's genetic variety. Crossover usually raises the satisfying number of clauses for the new chromosomes pairs. This trait helps the best generation chromosome to remain and faces the enhancements during the next operation (mutation operant).

Stage 5: Mutation

In this step of GA will be enhanced the non-improving interpretations. The mutation operator of GA involves the random flip of the bit string state, either from 1 to -1 or vice versa. For example:

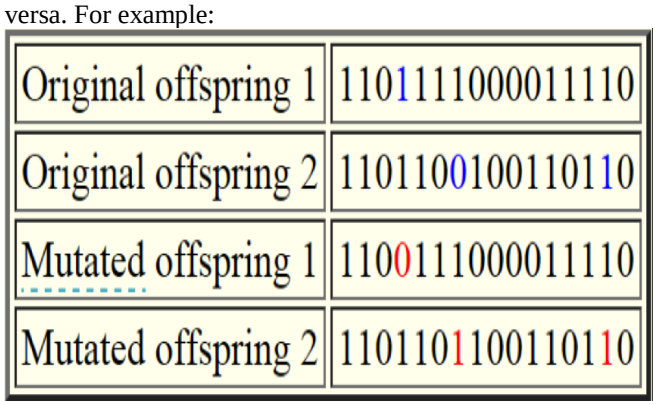

Fig. 3: Examples of mutation [15, 26]

If the maxima value of fitness is not achieved, the steps shall be repeating from stage 1. In the following Figure 4, show the implement of GA in HNN-HornSAT (HNN-HornSATGA).

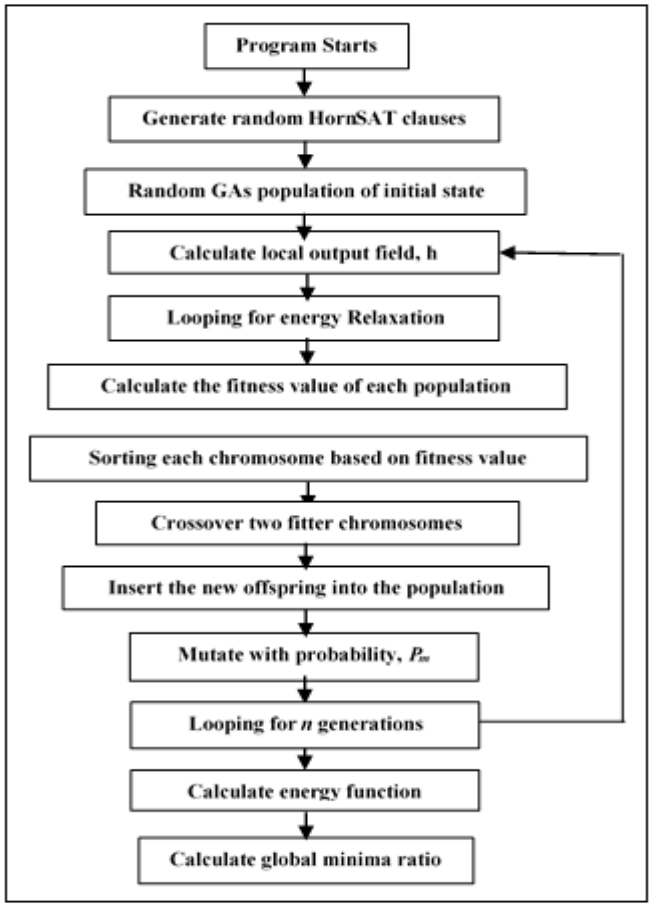

Fig. 4 Flow diagram for HNN-HornSATGA

II.6AGENT BASED MODELLING (ABM)

Agent-based modelling (ABM) is a methodology widely utilized in a wide area of social sciences [16]. This requires the development of a computer model consisting of "agents" representing both a social entity and a "world," NETLOGO being an agent-based language for programming and an essential framework for modeling. NETLOGO was planned to be a "low threshold and no ceiling" in the spirit of logical programming. NETLOGO teaches programming concepts in tortoises, patches, connections and observers using agents [17]. NETLOGO was designed for a variety of publics, particularly: educational children and field experts without programming history to model-related phenomena. NETLOGO has been used in many academic publications [18]. When connecting agents to mobile devices, they build networks, charts and aggregates to allow users to gain a greater understanding of system performance. In fact, the runs are exactly cross-platform reproducible. An agent-based simulation is used to model the interaction of HNN to do logic programming by utilized GA. Agent-Based Modeling uses NETLOGO as a platform. A flow chart below shows how this procedure is created and the following figure 5 shows a screenshot of the diagram model.

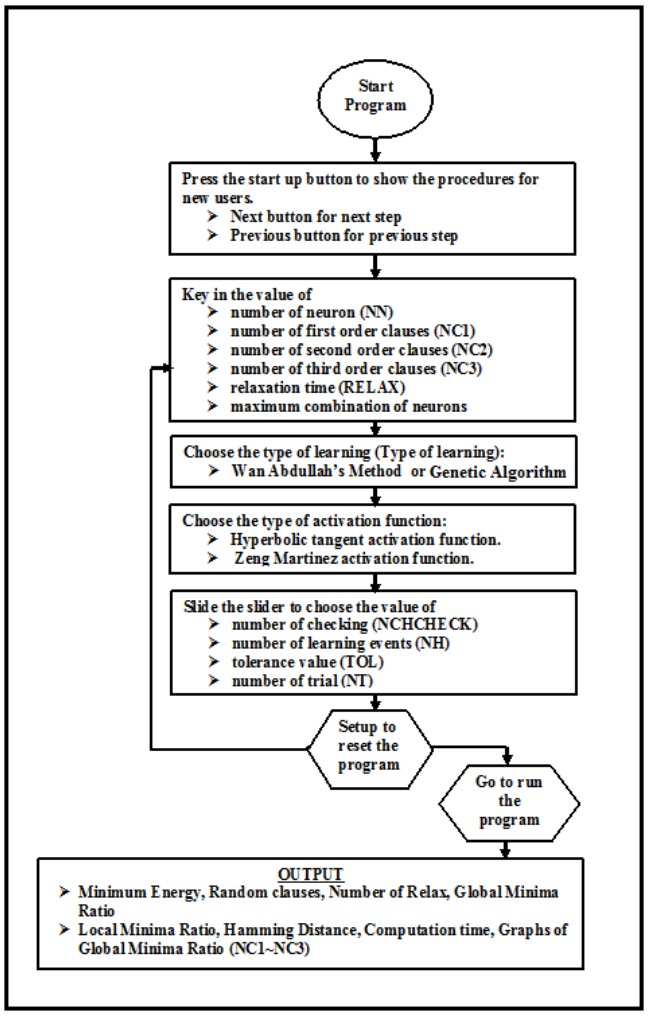

Fig. 5: Flowchart of Agent-Based Modeling for HNN-HornSATGA [19]

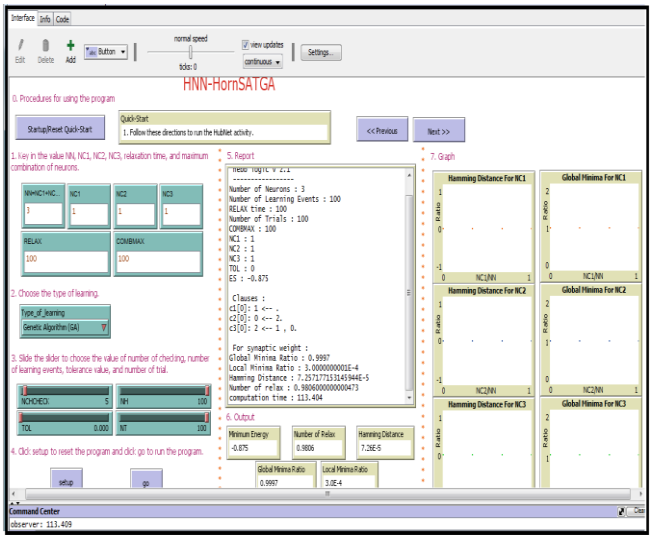

Fig. 6: The screenshot of the NETLOGO software for doing agent-based modelling

Fig. 6 above is explained as follows:

STAGE 1: Entering the Value of Each Parameters

1. Compress the startup / Reset (Quick-Start) button for the new user. Users will click the next button to take the next step and the last step.

2. In this step enter the value of number of neurons involved *NN*, and the clauses numbers *NC1*, *NC2*, *NC3*, and setups the value of Relax μ, value of COMBMAX ρ.

3. In this step we should choose type of activation function and learning.

4. Next, slides the slider to fix the number of trial τ, number of learning events δ while the maximum of tolerance ξ is 0.001.

5. After setting the values, press the setup button in the program and press set.

6. Finally, press the button "go" to run the program.

STAGE 2: Training phase

7. Initialise initial states of neuron in each clauses. Based on the design of HNN that derived from the behaviors of neurons movement in the magnetic domain, all neuron will communicate with each other in a complete bonded form as all of them tries to reach active appropriate state and this means minimal function energy. In this state, all the neurons will be rotating asychoronously. Thus, let the network evolve to the minima capacity when the minima energy equals the energy prerequisite in order to achieve the global minima.

8. The neurons consider relaxed if the state of the neuron remains unaltered after five runs. Therefore, the network system will classify the neurons have reached the stable states.

9. Computing final energy for the corresponding state of stable are carried out.

10. Accept the final solution as a global solution if the different between the minima energy and the final energy is within the tolerable value, or else go back to step 1.

11. Finally, the global minimum ratio, computational time and hamming distance (HD) are computed.

III. RESULTS AND DISCUSSION

The simulation of the models are carried out by using Netlogo version 6.0 software, on a computer (3.40GHz processor, 500GB hard disk, and 4096MB RAM) to comparatively analyze the performance of HNN-HornSATGA. The parameters in Table 1 are chosen by try and error technique.

TABLE 1 LIST OF PARAMETERS IN MFTHNN-2SAT MODEL

Parameter	Value
Relax	100
COMBMAX	100
Number of Trial	100
Number of learning	100
Number of checking	5

By using the Genetic Algorithm (GA) to accelerate the network performance, the neurons which are stucked in local minimum values enable to jump/relax into global minimum values. The following figures show the result obtained in the aspect of: global minimum ratio (zM), computational time (CPU time) and hamming distance (HD) as the performance evaluation metrics.

(i) Global Minima Ratio (*zM*)

zM is the ratio between global total minima energy and total simultaneous numbers is defined by Kasihmuddin *et al* and Alzaeemi *et a l* [20, 21].

When the final energy is within the limit, it is known as global minima energy. The global minimum ratio equation is defined as:

$$zM = \frac{1}{tc}\sum_{i=1}^{n} N_P \quad (6)$$

where c is the combination of the neuron, t is the trial, and N_P is the global minima energy number of the propose model. The higher accuracy HNN model has higher value of zM [21].

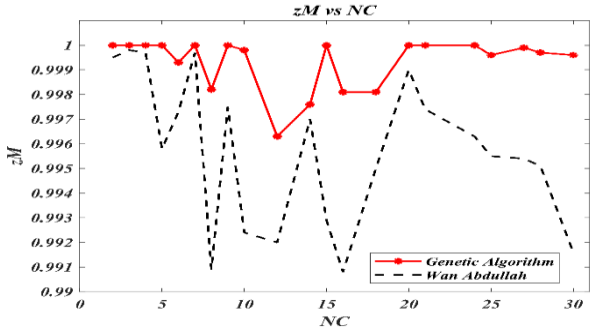

Fig. 7: Global Minima Ratio (zM) of HNN-HornSAT based GA and Wan Abdullah method

From Fig. 7, we can observe that more than 90% of the neuron's final state obtained by utilizing the GA method in doing HornSAT logic programming in HNN is the global minimum. The performance of this method is far more better and feasible than Wan Abdullah method. Although the complexity of network is increased by increasing the clauses number (NC), it doesn't affect the network stability and capacity by using HNN-HornSATGA. Based on the results from Fig. 7, global minima ratio portray that the HNN-HornSATGA is consistent and more stable as the network's complexity increases. This is possible because the genetic algorithm is less prone to been trapped in sub-optimal solutions or local minimal solutions. The performance of the network by utilizing Wan Abdullah method shows a slight inconsistency in values of the global minima which are oscillating up and down.

(ii) Computational Time (CPU)

In this study, we define the *CPU time* as the total time that is taken for the network to generate maxima satisfied clauses in the logic programming by using different activation functions [22, 23]. *CPU time* is given by the follow equation

$$CPU\ time = Retrieval\ time + Learning\ time \quad (7)$$

As defined by Sathasivam & Abdullah [24], the best model HNN has the least *CPU time* during retrieval phase and the learning phase. Hence, the best ANN models will have the shorter computational time.

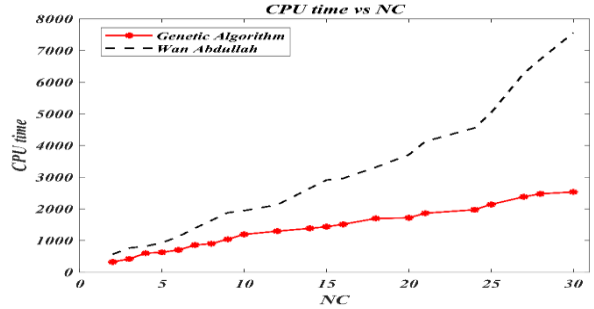

Fig. 8: CPU time of HNN-HornSAT based GA and Wan Abdullah method

Based on Fig. 8, lower CPU Time is required to complete one execution of learning and testing the HNN-HornSATGA. It is seen that the proposed Genetic Algorithm in HNN is more effective when dealing with a larger network compared with Wan Abdullah's method. Apparently, the different between the CPU time between the two techniques are significantly large as shown in Fig. 8. Moreover, as the network becomes more complex (number of clauses increased) the GAs show faster convergence compare to Wan Abdullah's method.

(iii) Hamming Distance (HD)

Hamming Distance has defined as the dimension between the global solution and the stable state of the neurons [25].

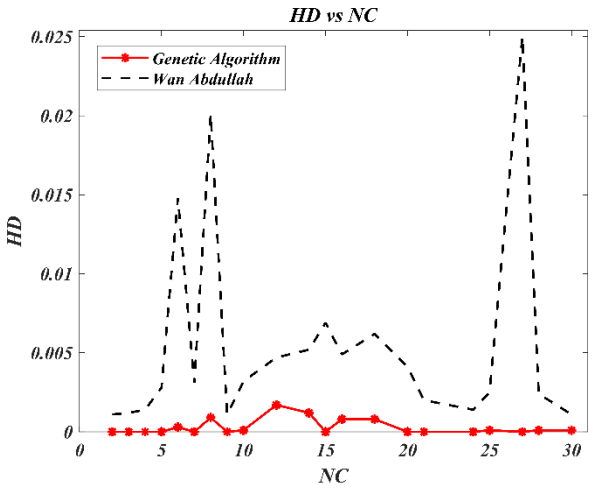

Fig. 9: Hamming distance of HNN-HornSAT based GA and Wan Abdullah method

Fig. 9 shows the hamming distance for the HNN-HornSATGA, which is near or equals to zero. We can conclude that this occured because of the fact that GA barely takes any restriction in the local minimum or any sub-optimal solution and always searching for optimal solution which epistolize to the global solution. Furthermore, by using GA, the mechanisms of crossover operation and mutation operation can be viewed as processes of moving a population round the echelon of fitness described by the fitness function. This means the population improves onto the maximum fitness is reached.

IV. CONCLUSION

In this paper, we improved the performance of doing HornSAT logic programming in HNN for optimal HornSAT by incorporating the Genetic Algorithm (GA) into HNN as a single computational model for optimal HornSAT in Boolean satisfiability representation (HNN-HornSATGA). It can be concluded that by using the GA method the network performance increased immensely. This has been supported from the computation time, hamming distance, and global minimum ratio. The outcomes also authenticated that the HNN-HornSATGA system always converge to the optimal solutions or nearly to the optimal solution. Furthermore it also maintains the candidate solutions population for the problem.

ACKNOWLEDGMENT

This research is supported by Research University Grant (RUI) (1001/PMATHS/8011131)obtained from Universiti Sains Malaysia.

REFERENCES

1. P. N. Tekwani, A. Chandwani, S. Sankar, N. Gandhi, & S. K. Chauhan, (2020). Artificial neural network-based power quality compensator. International Journal of Power Electronics, 11(2), 256-282.
2. S. A. S. Alzaeemi, et al., (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.
3. W.C. Carpenter, & J.F. Barthelemy, (1994). Common misconceptions about neural Networks as approximators. Journal of computing in civil engineering.
4. J. J. Hopfield, (1988). Neurons with graded response have collective computational properties like those of two-state neurons. In Artificial neural networks: theoretical concepts (pp. 82-86).
5. S. A. Alzaeemi. et al., (2017). "Analysis of Performance of Various Activation Functions for doing the logic programming in Hopfield Network.". Journal of Computational Bioinformatics and in Silico Modeling, 6, 2, pp. 911-921.
6. J.J. Hopfield, & D.W. Tank, (1985). Neural computation of decisions in optimization problems. Biological Cybernatics, 52, 141-152
7. S.S. Haykin, (2009). Neural Networks and learning machines. New Jersey: Prentice Hall.
8. W. A. T. W. Abdullah, (1992). "Logic programming on a neural network." International journal of intelligent systems 7.6, pp. 513-519.
9. S. Sathasivam, (2012). Applying Different Learning Rules in Neuro-Symbolic Integration. In Advanced Materials Research (Vol. 433, pp. 716-720). Trans Tech Publications Ltd.
10. D. Litman, (2016). Natural language processing for enhancing teaching and learning. In Thirtieth AAAI Conference on Artificial Intelligence.
11. S. Sathasivam, et al, (2020). "Hybrid Discrete Hopfield Neural Network based Modified Clonal Selection Algorithm for VLSI Circuit Verification." Pertanika Journal of Science & Technology 28.1.
12. S. A. Alzaeemi. et al.,(2017). "Analysis of Performance of Various Activation Functions for doing the logic programming in Hopfield Network.". Journal of Computational Bioinformatics and in Silico Modeling, 6, 2, pp. 911-921
13. S. Sathasivam, (2015). Acceleration Technique for Neuro Symbolic Integration. Applied Mathematical Sciences, 9(9), 409-417.
14. M. S. M. Kasihmuddin, M. A. Mansor, & S. Sathasivam, (2018). Discrete Hopfield Neural Network in Restricted Maximum k-Satisfiability Logic Programming. Sains Malaysiana, 47(6), 1327-1335
15. K.A. De Jong, (1975). An Analysis of the Behavior of a Class of Genetic Adaptive Systems. Ph.D Thesis, University of Michigan.
16. A. Alzaeemi, & S. Sathasivam, (2018). "Hopfield neural network in agent based modeling". MOJ App Bio Biomech, 2(6), pp. 334-341.
17. Y. Dijkxhoorn, et al. (2019). "Trusted sorghum: simulating interactions in the sorghum value chain in Kenya using games and agent-based modelling." Journal of Development Effectiveness 11.2, pp. 146-164.
18. C. Delcea, et al. (2020). "An agent-based modeling approach to collaborative classrooms evacuation process." Safety science 121, pp. 414-429.
19. S. Sathasivam, & M. Velavan, (2017). Agent Based Modelling for New Technique in Neuro Symbolic Integration. MATTER: International Journal of Science and Technology, 3(2).
20. M. S. M. Kasihmuddin, et al. (2019). "Quality Solution of Logic Programming in Hopfield Neural Network." In Journal of Physics: Conference Series, vol. 1366, no. 1, p. 012094. IOP Publishing.
21. M. Mansor, et al. (2020). Systematic Boolean Satisfiability Programming in Radial Basis Function Neural Network. Processes, 8(2), 214.
22. W. A. T. W. Abdullah, (2010). "The logic of neural networks." Physics Letters A 176.3-4, 1993, pp. 202-206.
23. S. Sathasivam, "Upgrading logic programming in Hopfield network." Sains Malaysiana 39.1, pp.115-118
24. S. Sathasivam. & Wan Abdullah, W.A.T., (2008). Logic Learning in the Hopfield Networks, Modern Applied Science, 2(3), 57-62.
25. M. Islam, et al. (2019). "Performance comparison of various probability gate assisted binary lightning search algorithm." IAES International Journal of Artificial Intelligence, pp. 228-236.
26. R.D. Munroe, Genetic programming: The ratio of crossover to mutation as a function of time. Res. Lett. Inf. Math. Sci. 2004, 6, 83–96.

AUTHORS PROFILE

Shehab Abdulhabib Alzaeemi received his Bachelor's Degree in Mathematics Science from Taiz University in 2003-2004 and his Master of Applied Mathematics Science from Universiti Sains Malaysia in 2016-2017 and now I continue in Ph.D. research in Universiti Sains Malaysia. He is fellow under the Academic Staff System in Sana'a Community College from 2005-2014. His research interests mainly focus on data mining, logic programming and neural network.

Saratha Sathasivam working as an Associate Professor in the School of Mathematical Sciences, Universiti Sains Malaysia. She completed her PhD in Universiti Malaya. MSc (Mathematics) and BSc(Ed) received from Universiti Sains Malaysia. Her current research interests are artificial intelligence, agent based modeling and numerical methods.

Muraly Velavan is a senior lecturer and also the Deputy Director in the School of General & Foundation Studies, AIMST University. He received his MSc at Universiti Malaysia Perlis. His current research interests are neural networks and agent based modeling.