Malaysian Journal of Mathematical Sciences 14(S) : 47–61 (2020)
Special Issue: 2nd International Conference on Applied & Industrial Mathematics and Statistics 2019 (ICoAIMS 2019)

MALAYSIAN JOURNAL OF MATHEMATICAL SCIENCES

Journal homepage: http://einspem.upm.edu.my/journal

Modified Lion Optimization Algorithm with Discrete Hopfield Neural Network for Higher Order Boolean Satisfiability Programming

Mansor, M. A.[1], Kasihmuddin, M. S. M.* [2], and Sathasivam, S.[2]

[1]*School of Distance Education, Universiti Sains Malaysia, Malaysia*
[2]*School of Mathematical Sciences, Universiti Sains Malaysia, Malaysia*

E-mail: shareduwan@usm.my
* *Corresponding author*

Received: 15 January 2020
Accepted: 30 July 2020

ABSTRACT

The Lion Optimization algorithm (LOA) and discrete Hopfield neural network (DHNN) are broadly employed for solving various complex optimization problems. Specifically, the Lion Optimization algorithm (LOA) is a new iterative and robust nature-inspired swarm metaheuristic algorithm, commonly utilised as a dynamic approach to improve the learning phase and convergence of the neural network. In this paper, a Hybrid Modified Lion Optimisation algorithm (LOA) with discrete Hopfield neural network (DHNN) is proposed for Boolean Satisfiability programming with different complexities. The powerful operators in LOA can be leveraged to reduce the computational burden in DHNN. The findings manifest the performance of the hybrid DHNN model in terms of sensitivity, accuracy, convergence rate, robustness, and computational time.

Keywords: Lion Optimization algorithm, discrete Hopfield neural network, Boolean Satisfiability.

1. Introduction

The Lion Optimization algorithm (LOA) is a class of robust metaheuristic paradigm, inaugurated by Rajakumar (2012) by taking inspiration from the social behavior of lions. Lions' intriguing and fascinating social behaviors are the main reason for exposing it as the strongest mammal in the jungle (king of the jungle). This variant of the metaheuristic algorithm can be applied on any artificial neural network model according to Mohandes et al. (2019). Mathematically speaking, the Lion optimization algorithm has been modeled and crafted by Rajakumar (2012) and Ramakrishnan and Sankaragomathi (2017). The Lion's optimization algorithm hunts solutions according to survival defense and territorial takeover. According to Wang et al. (2012), the stronger pride lion will dominate compared to the territorial lion that will end up with death or migration. Hence, the stronger pride lion refers to the global minima solution whereas the territorial lion denotes as local minima solution. According to Yazdani and Jolai (2016), a group of lionesses will mutually hunt and surround the prey from different points by utilizing rapid attack. Therefore, the regulated group hunting among the lions contributes towards a massive probability of success in lion hunts and survival as reported by Ramakrishnan and Sankaragomathi (2017). The study by Rajakumar (2012) divides LOA into four powerful stages such as pride generation, mating via crossover plus mutation operator, territorial defense, and territorial takeover. After meeting the termination criteria, we will select the lion or lionesses (the best solution). The iterative process enables the algorithm in searching to converge until the global minima are reached as reported by Sirdeshpande and Udupi (2017).

Chander et al. (2018) and Lin et al. (2018) had also conducted a significant study on LOA. These works are focusing on the fundamental developments in term of mathematical equations and steps involved in Lion optimization algorithm. Hence, the flexibility of the Lion optimization algorithm is the main motivation of incorporating this algorithm with discrete Hopfield neural network. The higher order k-Satisfiability problem (k-SAT) is the backbond for a numerous challenging problem because SAT bridges the Boolean logical transformation from problems to reasoning and semantic. The most popular variant is 3-Satisfiability (3-SAT) logic where the higher order Boolean satisfiability can be converted or simplified into this form Mansor and Sathasivam (2016). The implementation of 3-SAT has been done by Aiman and Asrar (2015) by using genetic algorithm as the searching method. Pursuing that, Kasihmuddin et al. (2016) have applied the k-Satisfiability programming in discrete Hopfield neural network (DHNN). Additionally, Shazli and Tahoori (2010) has applied the higher order Boolean satisfiability problem in circuit design and verification. There are different class and variant of neural networks including the discrete Hopfield neural network as developed by Hopfield and Tank (1985).

The discrete Hopfield neural network is a bipolar recurrent network with efficient associative memory and its function as powerful storage with definite memories in a way of the biological brain process an information as demonstrated in Sathasivam (2010). Due to the effectiveness of Lyapunov energy optimization in DHNN by Rojas (2013), a prolific number of researchers have amalgamated the concept of Boolean satisfiability programming with discrete DHNN as reported in Mansor et al. (2017). The work by Velavan et al. (2016) manifests the flexibility of DHNN to work in tandem with the accelerating algorithm, for instance, Mean Field theory. Then, the work by Mansor et al. (2017) has demonstrated the effectiveness of the hybrid artificial immune system with discrete Hopfield neural network for 3-SAT problem. Pursuing that, Kasihmuddin et al. (2017) have proposed the genetic algorithm with discrete Hopfield neural network for k-Satisfiability logic programming.

A new hybrid Boolean higher order 3-SAT logic programming model has been developed by using the concept of robust nature-inspired algorithm called Lion Optimization algorithm and the recurrent neural network called discrete Hopfield neural network. The effectiveness of the proposed model will be authenticated by comparing with the metaheuristic algorithms such as genetic algorithm, exhaustive search and imperialist competitive algorithm.

2. Boolean 3-Satisfiability Logic

The Boolean Satisfiability (SAT) is the problem of determining the interpretation of assignment with a given Boolean formula that either evaluates to true or false. Thus, for any higher order Boolean logic problem with the condition where is the number of variable, the problem can be reduced to 3-SAT Shazli and Tahoori (2010). In this paper, the systematic form of SAT will be formulated. The properties of SAT that ensembles Boolean 3-SAT are as follows:

Property 1
The Boolean SAT formula comprises of n variables, $z_1, z_2, ..., z_n, z \in \{-1, 1\}$ entrenched per clause. Since $n = 3$, the Boolean SAT will consist of 3 clause.

Property 2 A set of k clauses connected by AND ($\wedge$) in a 3-SAT formula as follows: $\exists k : F = C_1 \wedge C_2 \wedge .... \wedge C_k$.

Property 3
A set of $l_{k,i}$ literals and each clause c_k, $\forall\, 1 \leq k \leq m,\ c_k = (l_{(k,1)} \vee l_{(k,2)} \vee l_{(k,3)})$ which consists of several literals connected by the classical operator OR ($\vee$).

Property 4
Then, the state of the literals can be the negation of the variables or the positive variables. $\forall\, 1 \leq k \leq m,\ 1 \leq k \leq 3 : l_{(k,i)} = z_p$ or $l_{(k,i)} = \neg z_p$ for $1 \leq p \leq n$.

The Boolean 3-SAT formula is usually specified in product of sums or conjunctive normal form or CNF. Typical example of 3-SAT formula is given:

$$P_{3-SAT} = (A \vee \neg B \vee C) \wedge (\neg E \vee F \vee \neg G) \wedge (H \vee Y \vee Z), \tag{1}$$

Equation (1) shows an example of Boolean 3-SAT logic, P_{3-SAT} with 3 literals and 3 clauses. The task of finding the satisfied bipolar combination for this logic is usually tedious. Henceforth, a powerful searching algorithm is needed to generate the truth combination that satisfy the Boolean 3-SAT formula.

3. Discrete Hopfield Neural Network

Discrete Hopfield neural network (DHNN) is a class of recurrent neural network with the complex interconnected connections. As the nature of DHNN is non-symbolic, the declarative semantic power in 3-SAT will make it more powerful with the remarkable memory. Therefore, the early work of Hopfield and Tank (1985) established the computational power of DHNN in solving the hard optimization problem, specifically the travelling salesman problem. The auto-associative model such as DHNN systematically store patterns as a content addressable memory (CAM). The excitation of the neuron in DHNN can be represented mathematically as in S_i :

$$S_i = \begin{cases} 1 & if \sum_j W_{ij} S_j > \xi \\ -1 & Otherwise \end{cases}, \tag{2}$$

where W_{ij} is the weight for unit j to i and ξ refers to the threshold. The implementation of 3-SAT in DHNN is denoted as DHNN-3SAT. DHNN-3SAT considers 3 neurons per clause. The local field is utilized to appropriately squash the retrieved output before generating the final state. Moreover, the local field formulation for $k = 3$ is shown in Equation (3) as formulated by Mansor et al. (2017).

$$h_i = \sum_{k=1, i \neq j \neq k}^{N} W_{ijk}^{(3)} S_j S_k + \sum_{j=1, i \neq j \neq k}^{N} W_{ij}^{(2)} S_j + W_i^{(1)} \quad, k = 3, \tag{3}$$

where i and j are corresponded to neurons N. These local field will establish the usefulness and adaptability of the final states attained by DHNN. Consequently, the generated final interpretation will classify whether the solution is overfit or not. Specifically, the updating rule of the states is:

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

The neuron relation is absolutely symmetric and zero diagonal for the cases; $W_{ii}^{(2)} = W_{jj}^{(2)} = W_{kk}^{(2)} = W_{iii}^{(3)} = W_{jjj}^{(3)} = W_{kkk}^{(3)} = 0$, which further derive and formulate the final energy of respective variant of DHNN-3SAT as given:

$$\begin{aligned} E_{min} = &-\tfrac{1}{3}\textstyle\sum_{i=1,i\neq j\neq k}^{N}\sum_{j=1,i\neq j\neq k}^{N}\sum_{k=1,i\neq j\neq k}^{N} W_{ijk}^{(3)} S_i S_j S_k \\ &-\tfrac{1}{2}\textstyle\sum_{i=1,i\neq j}^{N}\sum_{j=1,i\neq j}^{N} W_{ij}^{(2)} S_i S_j - \sum_{i=1}^{N} W_i^{(1)} S_i \quad, k=3, \end{aligned} \tag{5}$$

Equation (4) and equation (5) are significant to guarantee the convergence of the neurons to local or global minimum. In this paper, the value of synaptic weight will be determined by using Wan Abdullah method as coined by Abdullah (1992). Thus, it is a comprehensive learning method by computing the respective synaptic weight according to Boolean logical inconsistencies.

4. Modified Lion Optimization Algorithm

The Lion Optimization Algorithm (LOA) is a variant of bio-inspired and population-based searching method by taking socio-behavioral of lion species as reported by Rajakumar (2012). This algorithm focuses on the mathematical modelling of lions' survival, mating, and social organization in order to tackle constraint satisfaction problem. The interpretation of socio-behavioural of lion in a population will be modelled in algorithmic form to search for the global or feasible solution from a massive search space. According to Yazdani and Jolai (2016), every possible solution is represented as an individual of Lion living in a particular jungle. Thus, the lion is the bipolar interpretation to be computed by the hybrid model. Mathematically, the solutions are represented in bipolar string in order to comply with Boolean 3-SAT logic programming. The implementation of LOA is simplified as follows:

Step 1: Initialization of Pride Generation
100 Lions, L_{ij} are initialized. The lion represents the bipolar string of solutions of 3-SAT logic.

$$L_{ij} = \begin{cases} 1 & rand(0,1) \geq 0.5 \\ -1 & Otherwise \end{cases} \quad 1 \leq i \leq N, 1 \leq j \leq 100, \tag{6}$$

Step 2: Fitness Evaluation
The fitness value of the lion will be calculated. The fitness equation is modified to comply with our work. Thus, the fitness measures the total number of satisfied clauses per 3-SAT logical formula.

$$fitness_i = \sum_{i=1,j=1}^{NC} C_{ij}, \tag{7}$$

Step 3: Hunting
The lion will be divided randomly into prey lion and hunter lion. The lion with higher fitness has higher chance to be the hunter lion, see Yazdani and Jolai (2016).

$$Prey' = Prey + rand(0,1) \times PI \times (Prey - Hunter), \tag{8}$$

where $Prey$ is the prey lion, PI is the percentage of enhancement in the fitness of hunter and $Hunter$ denotes the hunter lion.

Step 4: Moving Towards Safe Place
Since the enclave of each prides consist of the best individual positions of each member, the LOA will assist to save the finest solutions obtained per iteration.

Step 5: Roaming
During this stage, the probability of the solution enhancement will be evaluated before being elected to Step 6.

$$pr_i = 0.1 + min(0.5, \frac{(Nomad_i - Best_{Nomad})}{Best_{Nomad}}) \quad i = 1, 2, ..., \tag{9}$$

where $pr_i$is the probability of the roaming, $Nomad_i$ is the current position and $Best_{Nomad}$ denotes the fitness of the best nomad lion (solutions).

Step 6: Mating
Mating is a procedure of generating the best solution via crossover, mutation

and gender grouping process. The offspring will become the nomadic lion.

$$L_{offspring} = \beta \times L_{female} + \sum \frac{(1-\beta)}{\sum_{i=1}^{NR} S_i} \times L_{males} \times S_i, \tag{10}$$

where S_i is set to be 1 , NR is number of nomadic male lions, $L_{offspring}$ is the number of the offspring produced and β is randomly generated parameter $(0,1)$.

Step 7: Territorial Defense
The territorial defense is a procedure assessing the current solution (territorial lion) and newly formed solutions (nomadic lion). Hence, the newly generated solution will be promoted to be the current solution if the fitness is higher than the existing solution.

Step 8: Territorial Takeover
In the most of nature inspired algorithm, this operator is equivalent to the selection. Therefore, the best lion or lioness will be selected by looking at Equation (11), otherwise go to Step 3.

$$fitness_i = max\{NC\}, \tag{11}$$

5. Implementation and Experimental Setup

The hybrid DHNN models that have been explored in this paper are DHNN-3SAT models such as DHNN with the modified Lion Optimization algorithm (DHNN-3SATLOA), DHNN incorporated with genetic algorithm (DHNN-3-SATGA) by Kasihmuddin et al. (2017), DHNN with exhaustive search (DHNN-3SATES) by Zhang et al. (2017) and DHNN integrated with Imperialist Competitive algorithm (DHNN-3SATICA) by Shazli and Tahoori (2010), Abdullah (1992). In this work, the DHNN models utilized the simulated datasets by generating random 3SAT clauses with different level of complexities. The implementation of the developed model, DHNN-3SATLOA and the other models is carried out via Microsoft Visual Basic C++ 2013 for Windows 10.1 and 16 GB of RAM. Similar processing system and CPU will be used in every execution to avoid possible bad sector and bias during computation. The experiments are limited until $NN = 108$ for simplicity.

Table 1: List of Important Parameters in DHNN-3SAT Models.

Parameter	Value
Tolerence Value (Tol)	0.001
Combination of Neurons	100
Number of Strings	100
Number of Neurons	$9 \leq NN \leq 108$
Relaxation Rate	5

The experimental setup and important parameters are given in Table 1. Hence, the tolerance value as in Table 1 is selected due to a good agreement with the work of Sathasivam (2010) and Kasihmuddin et al. (2018). The implementation is shown in Figure 1.

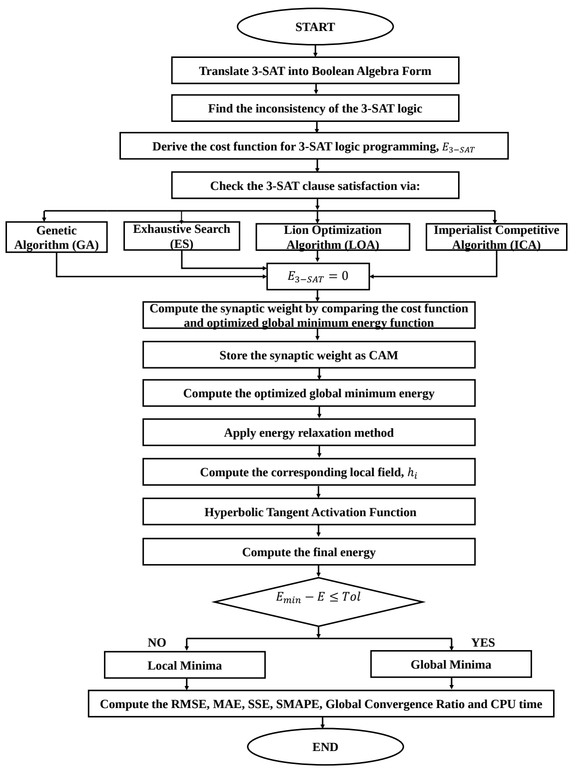

Figure 1: The Implementation of DHNN-3SAT Models in the Task of 3-SAT Programming

6. Results and Discussion

The performance and sturdiness of our proposed hybrid model, DHNN-3SATLOA with the other 3 models, DHNN-3SATES, DHNN-3SATGA and DHNN-3SATICA were evaluated by utilizing various performance evaluation metrics. The comparisons are discussed according to the analyses of Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Global Convergence Ratio, Sum of Squared Error (SSE), Symmetric Mean Absolute Percentage Error (SMAPE), and CPU Time.

Figure 2 to Figure 7 manifest the capability of the hybrid Lion Optimization algorithm with discrete Hopfield neural network (DHNN-3SATLOA) as compared with DHNN-3SATES, DHNN-3SATGA, and HNN-3SATICA in generating global solutions for 3-Satisfiability logic programming. The simulations were conducted by manipulating the different Number of Neurons (NN), ranging from $NN = 9$ until $NN = 108$.

The root mean square error (RMSE) and mean absolute error (MAE) recorded by the developed model, DHNN-3SATLOA are presented in Figure 2 and Figure 3. Figure 2 and Figure 3 demonstrate the accuracy of our proposed model in training the simulated data of 3-SAT logic without consuming the unnecessary tedious iterations in generating global solutions. This is due to the migration operator where the crossover happens and the territorial defense that improve the solutions to achieve global convergence. Therefore, fewer iterations will allow the model to attain faster convergence, resulting in minimum RSME and MAE obtained by DHNN-3SATLOA. Apparently, DHNN-3SATES performed apparently poorly due to the trial and enumerate procedure in attaining the truth combinations. Moreover, DHNN-3SATGA is still acceptable for the lower number of neurons as the non-fit strings need to be improved before undergoing the mutation operator Aiman and Asrar (2015). The capability of DHNN-3SATICA is slightly better than the two counterparts, as the RMSE and MAE approaching the one generated by our developed model. This is due to the optimization operator such as colonization and the imperialism operator in producing the best solutions Lian et al. (2012).

Figure 4 elucidates in achieving the global convergence of the DHNN-3SAT models provided the complexity of the network ranging from $NN = 9$ until $NN = 90$. Therefore, it is very obvious that DHNN-3SATLOA outperformed the other 3 hybrid models in achieving the global convergence during the task of logic programming. The developed model was able to generate more global solutions according to the ratio of global convergence obtained at the end of the simulations. From Figure 4, it is obvious that DHNN-3SATES attain less global convergence ratio due to the trial and error procedures in obtaining the solutions. As a result, more local solutions are being generated by the models, indicating the weakness of ES in training the 3-SAT logic.

The sensitivity of the hybrid models towards the accumulation of error can be explained in Figure 5. Figure 5 portrays the sum of squared error (SSE) recorded by the models in the task of logic programming for a different level of complexities. DHNN-3SATLOA exhibits the lowest value of SSE, indicating that the model is less prone towards any incoming error. Henceforth, the sensitivity of DHNN-3SATLOA is the lowest as compared to DHNN-3SATGA, DHNN-3SATES, and DHNN-3SATICA. This is due to the mating, territorial defense and territorial takeover operators that allowed the solutions to be improved vigorously without taking more iterations. Thus, the fitness of the solutions is enhanced effectively in less iterations.

According to Figure 6, it was observed that the SMAPE for DHNN-3SAT models possesses a similar trend. When the numbers of neurons are higher, the SMAPE for the four hybrid models is getting massive. However, DHNN-3SATLOA generates lesser SMAPE due to the capability of the models to attain the global convergence in fewer iterations. The optimization operator in LOA especially during territorial defense has enhanced the non-fit solution to become global solutions without consuming trial and error stage. Thus, it was clear that the percentage of SMAPE depicts the minimum values as compared to the other models. The searching operator in DHNN-3SATGA, DHNN-3SATES, and DHNN-3SATICA is also promising at NN=9 as it completes the task of 3-SAT programming effectively with SMAPE less than 5% respectively.

In addition, the robustness of the proposed algorithm with the conventional methods based on the CPU time taken to complete the execution. The CPU Time recorded by DHNN-3SATLOA, DHNN-3SATES, DHNN-3SATGA, and DHNN-3SATICA per simulation is elucidated in Figure 7. Overall, based on the CPU Time evaluation, the developed hybrid model, DHNN-3SATLOA executed the 3-SAT programming faster than the other three models. In theory, the powers of territorial defence and territorial takeover operators had enabled the non-fit solution (lion) to be further improved to achieve global convergence without undergoing additional iterations. The DHNN-3SATES exhibited the longest time to attain global solutions as the searching process was based on enumerate and generate procedures within a particular search space. Henceforth, the iterations had contributed significantly to the entire execution time for DHNN-3SATES model. On the contrary, DHNN-3SATGA model required early population adjustment before undergoing crossover and mutation operator to improve the fitness of the solutions Kasihmuddin et al. (2016). Additionally, DHNN-3SATICA is moderately reliable, even though the colonization operator in enhancing the solution might consume additional time. To sum up, DHNN-3SATLOA was apparently more robust as compared to the other three models in the task of logic programming. According to the performance metrics analysis, it can be observed that our developed model has improved the work of Aiman and Asrar (2015) and Kasihmuddin et al. (2018) in terms of hybrid model development in solving 3-SAT problems.

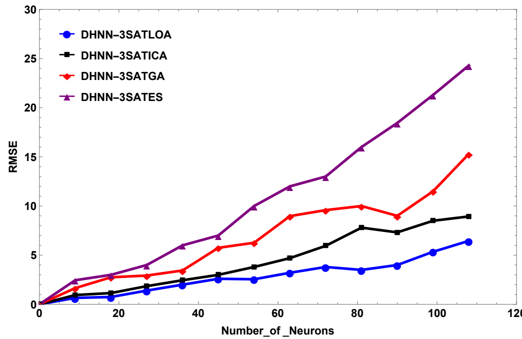

Figure 2: The RMSE for DHNN-3SAT models

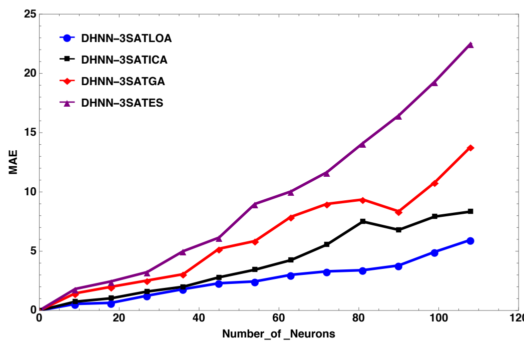

Figure 3: The MAE for DHNN-3SAT models

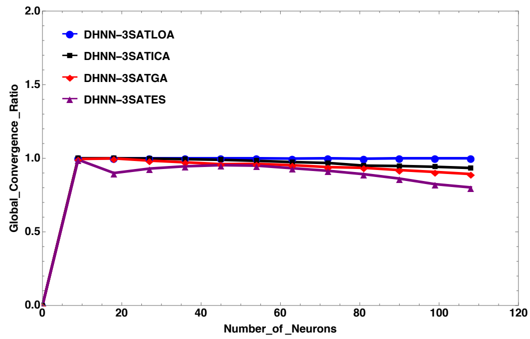

Figure 4: The Global Convergence Ratio for DHNN-3SAT models

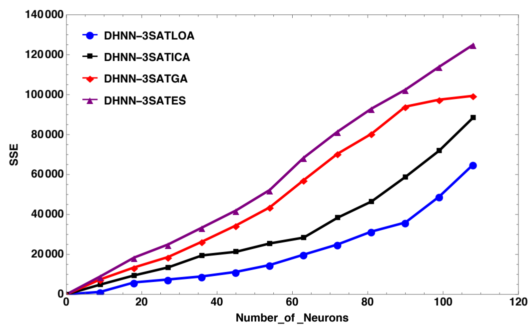

Figure 5: The SSE for DHNN-3SAT models

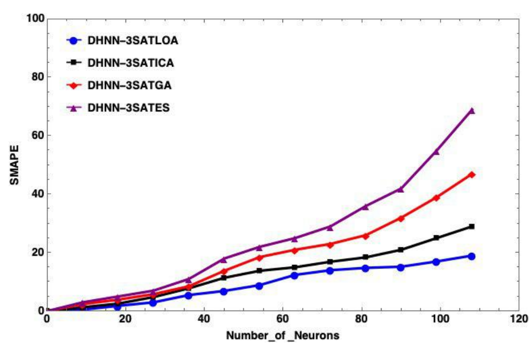

Figure 6: The SMAPE for DHNN-3SAT models

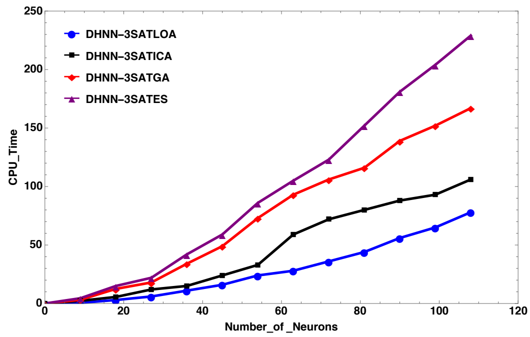

Figure 7: The CPU Time for DHNN-3SAT models

7. Concluding Remarks

The researchers have presented the effectiveness of the hybrid Lion Optimization algorithm with discrete Hopfield neural network (DHNN-3SATLOA) in executing 3-Satisfiability programming as compared to state-of-the-art models such as DHNN-3SATGA, DHNN-3SATES and DHNN-3SATICA. In order to evaluate the performance and capability of the developed model, the researchers of this study have trained and tested it by using simulated data sets, which showed that DHNN-3SATLOA outperformed the other models in terms of RMSE, MAE, SSE, Global Convergence Ratio, SMAPE and CPU Time. Despite all that, this research is still in its infancy stage, and the development of the developed hybrid model can be utilized further with Boolean-based data mining. In addition, the developed model has a profound potential in solving various constraint satisfaction and optimization problems ranging from scheduling to the shortest path problems.

Acknowledgement

This research is fully supported and funded by Short Term Research Grant (304/PJJAUH/6315263) by Universiti Sains Malaysia.

References

Rajakumar, B. 2012. The lion's algorithm: a new nature-inspired search algorithm. *Procedia Technology*, 6:126–135.

Mohandes, S. R., Zhang, X., & Mahdiyar, A. 2019. A comprehensive review on the application of artificial neural networks in building energy analysis. *Neurocomputing*, 340:55–75.

Ramakrishnan, T. & Sankaragomathi, B. 2017. Efficient implementation for classifying and segmenting of computed tomography brain tumour images using modified region growing with lion algorithm. *International Journal of Biomedical Engineering and Technology*, 23(2-4):159–179.

Wang, B., Jin, X., & Cheng, B. 2012. Lion pride optimizer: An optimization algorithm inspired by lion pride behavior. *Science China Information Sciences*, 55(10):2369–2389.

Yazdani, M. & Jolai, F. 2016. Lion optimization algorithm (loa): a nature-inspired metaheuristic algorithm. *Journal of Computational Design and Engineering*, 3(1):24–36.

Sirdeshpande, N. & Udupi, V. 2017. Fractional lion optimization for cluster head-based routing protocol in wireless sensor network. *Journal of the Franklin Institute*, 354(11):4457–4480.

Chander, S., Vijaya, P., & Dhyani, P. 2018. Multi kernel and dynamic fractional lion optimization algorithm for data clustering. *Alexandria Engineering Journal*, 57(1):267–276.

Lin, K.-C., Hung, J. C., & Wei, J.-t. 2018. Feature selection with modified lions algorithms and support vector machine for high-dimensional data. *Applied Soft Computing*, 68:669–676.

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

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

Kasihmuddin, M. S. M., Mansor, M. A., & Sathasivam, S. 2016. Genetic algorithm for restricted maximum k-satisfiability in the hopfield network. *International Journal of Interactive Multimedia & Artificial Intelligence*, 4 (2):52–60.

Shazli, S. Z. & Tahoori, M. B. 2010. Using boolean satisfiability for computing soft error rates in early design stages. *Microelectronics Reliability*, 50(1): 149–159.

Hopfield, J. J. & Tank, D. W. 1985. Neural computation of decisions in optimization problems. *Biological Cybernetics*, 52(3):141–152.

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

Rojas, R. 2013. *Neural networks: a systematic introduction.* Springer Science and Business Media, Berlin, Germany.

Mansor, M. A., Kasihmuddin, M. S. M., & Sathasivam, S. 2017. Robust artificial immune system in the hopfield network for maximum k-satisfiability. *International Journal of Interactive Multimedia & Artificial Intelligence*, 4 (4):63–71.

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

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 and Technology*, 25(1):139–152.

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

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.

Kasihmuddin, M. S. M., Mansor, M. A., & Sathasivam, S. 2018. Discrete hopfield neural network in restricted maximum k-satisfiability logic programming. *Sains Malaysiana*, 47(6):1327–1335.

Lian, K., Zhang, C., Gao, L., & Li, X. 2012. Integrated process planning and scheduling using an imperialist competitive algorithm. *International Journal of Production Research*, 50(15):4326–4343.