Grey Wolf Optimization algorithm with Discrete Hopfield Neural Network for 3 Satisfiability analysis

Mohd. Asyraf Mansor¹, Mohd Shareduwan Mohd Kasihmuddin², Saratha Sathasivam²

¹School of Distance Education, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia.
²School of Mathematical Sciences, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia.
E-mail: shareduwan@usm.my

Abstract. An optimal learning algorithm contributes to the quality of the neuron states in the form of 3 Satisfiability logical representation during the retrieval phase of the Discrete Hopfield Neural Network. Based on that basis, we proposed a modified bipolar Grey Wolf Optimization algorithm with a Discrete Hopfield Neural Network for Boolean 3 Satisfiability analysis by manipulating the different levels of complexities. This work concerns the improvement in the learning phase which requires a robust iterative metaheuristic algorithm in minimizing the cost function of 3 Satisfiability logical representation with less iteration. Under some reasonable conditions, the proposed hybrid network will be assessed by employing several performance measures, in terms of learning errors, minimum energy evaluations, variability, and similarity analysis. To verify the compatibility of the Grey Wolf Optimization algorithm as a learning paradigm, the comparison was made with the hybrid model with an Exhaustive search. Thus, the results proved the capability of the proposed learning algorithm in optimizing the learning and generating global minimum solutions for 3 Satisfiability logic based on the analysis obtained via various performance metrics evaluation.

1. Introduction
Grey wolf optimization (GWO) algorithm is a powerful nature-inspired algorithm that impersonates the survival, maneuvers, leadership stratum and social hunting behaviour of the grey wolf in their favorable habitat [1]. Mirjalili et al. [2] proposed Grey wolf optimization (GWO) algorithm to effectively portray the social dominant hierarchy of the grey wolf from alpha, beta, delta and omega. According to the Mathematics standpoint, the highest level of grey wolf leadership stratum is an alpha wolf, which depicts the best solutions. The robustness and flexibility of the Grey wolf optimization (GWO) algorithm have been demonstrated in a plethora of works. Emary et al. [1] has proposed the binary Grey wolf optimization approach to search and enumerate the optimum feature in the classification of 18 UCI data sets. The work has successfully proved the performance and convergence of binary Grey wolf optimization algorithm in feature selection as compared with the Binary Genetic algorithm and binary particle swarm optimization algorithm. Sujatha and Punithavathani [3] has applied the Grey wolf optimizer in deciding the multi-focused image processing. Moreover, Panwar et al. [4] focused on the implementation of the GWO algorithm in solving large scale unit commitment problem. Recently, Sharma et al. [5] conducted work to diagnose the early stage of Parkinson’s disease by using the Grey Wolf optimization algorithm. The main motivation of this GWO algorithm is due
to the simplicity of the formulation without any complex fitness equations. The free parameter is very minimal and regulated to avoid any complexities \[2, 6\]. Thus, the simulation can be done without consuming more iterations during the training phase. Pursuing that, Tawhid and Ali \[7\] have hybridized the standard GWO with Genetic algorithm to minimize the energy profile of the molecule. In the recent development by Hu et al. \[8\], the binary GWO has been successfully being used in feature selection via a stochastic approach. According to these related works, we can deduce that the GWO algorithm can be hybridized with intelligent networks to attain faster convergence, better accuracy, acceptable sensitivity, and specificity. In fact, based on the premises, the performance of GWO can be seen as promising algorithm in facilitating the learning phase. However, there is no attempt to utilize the GWO algorithm as an optimal learning method in Discrete Hopfield Neural Network (DHNN), specifically in optimizing the 3 Satisfiability (3SAT) logical representation and analysis.

Discrete Hopfield Neural Network (DHNN) is a variant of the recurrent neural network consists of output and input layer, without any intervention of the hidden layer. The earliest work on Hopfield Neural Network (HNN) can be seen in the work of Hopfield \[9\], which became the breakthrough of different classes such as HNN for discrete \[10\] and continuous class of network. Garcia et al. \[11\] focused on the improvement of DHNN in solving various traveling salesman problem with minimized computational cost. Then, Hillar and Tran \[12\] discussed the remarkable exponential memory in the HNN in the form of content-addressable memories (CAM). Cabrera and Sossa \[13\] outlined the method of generating stable states for HNN. According to the work of Kasihmuddin et al. \[14, 15\], the Discrete Hopfield Neural Network and learning methods formed a robust hybrid model in Satisfiability programming. In the recent work of Rajagopal et al. \[16\] also highlighted the capability of Hopfield Neural Network in circuit design. Thus, the project will implement the Discrete Hopfield Neural Network due to the adaptability with other algorithms and tremendous memory capability. Generally, in this work, the DHNN formulation will be based on the work of Sathasivam et al. \[17\] and Zamri et al. \[18\]. The hybrid model with Exhaustive Search is inspired by the work of Aiman and Asrar \[19\]. Exhaustive Search is a standard searching method and guaranteed to converge to the global solution even though more iterations are needed in a particular search space. In this work, the structure of Boolean 3 Satisfiability logical representation as proposed by Mansor and Sathasivam \[20\] will be considered.

The contributions of this research can be divided into 3 main domains such as: (1) To implement a bipolar Grey Wolf Optimization (GWO) algorithm in optimizing the learning mechanism for 3 Satisfiability (3SAT) analysis in Discrete Hopfield Neural Network (DHNN) by adopting various layer of stochastic operators. (2) To formulate the GWO algorithm that complies in minimizing the cost function for 3SAT logical representation during the learning phase. (3) To assess the compatibility of the hybrid Grey Wolf Optimization with Discrete Hopfield Neural Network for 3 Satisfiability logic as a single model, namely DHNN-3SATGWO in terms of learning error, global energy evaluation, similarity analysis, and variability assessment, as opposed with the standard model with Exhaustive Search, called DHNN-3SATES. The comparative analysis will be conducted by deploying suitable performance evaluation measures with manipulating different complexity in terms of the number of neurons. A series of computer simulations will be used to verify the compatibility and capability of our newly proposed model in minimizing the cost function of 3SAT, which eventually will lead to effective final states retrieval phase in DHNN. This work will reveal the effectiveness of our proposed algorithm by utilizing different level of complexities, based on the number of neurons.
2. 3 Satisfiability logical representation via Discrete Hopfield Neural Network

In this work, we utilized the 3 Satisfiability Logic Analysis in Hopfield Neural Network (DHNN-3SAT) as a model to be assessed by using learning error, similarity, energy analysis and variability metrics. 3SAT problem is a discrete logic representation, with strictly 3 literals per clause [20]. The 3SAT equation in Boolean algebra form is given in Equation (1).

\[
\kappa_{3SAT} = (K \lor \neg L \lor M) \land (\neg E \lor F \lor \neg G) \land (\neg X \lor Y \lor Z),
\]

From the aforementioned 3SAT logic, the cost function is formulated as follows:

\[
E_{\kappa_{3SAT}} = \frac{1}{2^N} \sum_{i=1}^{NC} \left( \prod_{j=1}^{3} C_i \right)_j
\]

given

\[
C_i = \begin{cases} 
1 - SC, & \text{if } \neg C_i \\
1 + SC, & \text{otherwise} 
\end{cases}
\]

where \( NC \) refers to the number of clauses, \( C_i \) denotes the clause in the \( \kappa_{3SAT} \) and \( SC \) is the state corresponds to the literal in the clause. Thus, the learning process will be completed if the model obtain the satisfied states that leads to \( E_{\kappa_{3SAT}} = 0 \).

Additionally, the fitness of \( \kappa_{3SAT} \) can be computed as in Equation (4), whereby the maximum fitness, \( f_{max} \) correspond to the maximum values of \( f_i \).

\[
f_i = \sum_{i=1}^{n} C_i^{(3)},
\]

where

\[
C_i = \begin{cases} 
1, & \text{if } E_{\kappa_{3SAT}} = 0 \\
0, & \text{otherwise} 
\end{cases}
\]

The DHNN 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 [9, 10]. The retrieval phase of DHNN also play an important role in logic programming. The updating process for the final neuron states during retrieval phase in DHNN are driven by the local field function as follows:

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

where \( W_i^{(1)} \), \( W_{ij}^{(2)} \) and \( W_{ijk}^{(3)} \) are the synaptic weight corresponded to neurons \( N \) and \( S_i \) refers to the neuron state. These local field will establish the usefulness and adaptability of the final states attained by DHNN.

The final energy will be computed via:

\[
H_{\kappa_{3SAT}}^{min} = -\frac{1}{3} \sum_{i=1, i \neq j \neq k}^{N} \sum_{j=1, i \neq j \neq k}^{N} \sum_{k=1, i \neq j \neq k}^{N} W_{ijk}^{(3)} S_i S_j S_k - \frac{1}{2} \sum_{i=1, i \neq j}^{N} \sum_{j=1, i \neq j}^{N} W_{ij}^{(2)} S_i S_j - \sum_{i=1}^{N} W_i^{(1)} S_i,
\]

The final energy function will be the filtering mechanism, whether the solution is local minimum energy or global minimum energy [17]. The model will be simulated by using different complexity in terms of different number of neurons (NN).
3. Grey Wolf Optimization algorithm as learning paradigm

Grey Wolf Optimization (GWO) algorithm is a variant of swarm metaheuristic method, being modelled mathematically by the survival and hunting mechanism adopted by grey wolves in specific habitat. The GWO algorithm has been developed by Mirjalili et al. [2] based on leadership hierarchy or strata in a grey wolf community. The 4 main strata can be classified as alpha, beta, delta and omega wolves [5].

Step 1: Grey Wolf Parameter Initialization
The grey wolf parameters are initialized and formulated via:

\[
\vec{e} = \frac{2T_{MAX} - 2n}{T_{MAX}}, \quad (8) \\
\vec{E} = 2\vec{e} \cdot \vec{r}_1 - \vec{e}, \quad (9) \\
\vec{F} = 2 \cdot \vec{r}_2, \quad (10)
\]

where \( \vec{r}_1 \) and \( \vec{r}_2 \) refers to the random vector between [0, 1].

Step 2: Wolf String Generation
100 wolf strings are randomly generated by the network by taking the bipolar values of neuron state. The formula can be generalized as follows:

\[
\vec{X}_n = \{S_1, S_2, S_3, ..., S_N\}. \quad (11)
\]

Step 3: Fitness Evaluation of Wolf
The fitness of the wolves are computed by taking:

\[
f_{\vec{X}_n} = \sum_{N=1}^{m} C_N^{(3)}, \quad (12)
\]

where \( N \) denotes the number of clauses and \( m \) refers to the maximum number of clauses of the logic.

Step 4: Wolf Position Update
The best 4 wolves are elected to be in the leadership hierarchy. The alpha wolf (\( X_\alpha \)), beta wolf (\( X_\beta \)), delta wolf (\( X_\delta \)) and omega wolf (\( X_\omega \)) are selected and updated based on the position in the grey wolf in their community strata.

Step 5: Hunting (Encircling Prey)
The distance of the hunting grey wolves refers to the closeness between the wolves while encircling prey during hunting. Mathematically, the equation has been crafted based on Hu et al. [8]. The distance of the grey wolves during this phase can be modelled as follows:

\[
\vec{D}_\alpha = |\vec{F} \cdot \vec{X}_\alpha - \vec{X}_i|, \quad (13) \\
\vec{D}_\beta = |\vec{F} \cdot \vec{X}_\beta - \vec{X}_i|, \quad (14) \\
\vec{D}_\delta = |\vec{F} \cdot \vec{X}_\delta - \vec{X}_i|, \quad (15)
\]
where $\vec{D}_\alpha$, $\vec{D}_\beta$ and $\vec{D}_\delta$ refer to the distance of alpha wolf, beta wolf and gamma wolf respectively.

Step 6: Wolf Classification Via Sigmoid Function
The sigmoid function is used to normalize the distance of the wolf as in the previous step.

$$\vec{g}_\alpha = \frac{1}{1 + exp^{-10(\vec{E} \cdot \vec{D}_\alpha - 0.5)}} \quad (16)$$

$$\vec{g}_\beta = \frac{1}{1 + exp^{-10(\vec{E} \cdot \vec{D}_\beta - 0.5)}} \quad (17)$$

$$\vec{g}_\delta = \frac{1}{1 + exp^{-10(\vec{E} \cdot \vec{D}_\delta - 0.5)}} \quad (18)$$

Then the bipolar vectors can be formed by considering the following equations.

$$\vec{B}_\alpha = \begin{cases} 1, & \text{if } \vec{g}_\alpha \geq \text{random}(0,1) \\ -1, & \text{otherwise} \end{cases} \quad (19)$$

$$\vec{B}_\beta = \begin{cases} 1, & \text{if } \vec{g}_\beta \geq \text{random}(0,1) \\ -1, & \text{otherwise} \end{cases} \quad (20)$$

$$\vec{B}_\delta = \begin{cases} 1, & \text{if } \vec{g}_\delta \geq \text{random}(0,1) \\ -1, & \text{otherwise} \end{cases} \quad (21)$$

Again, the bipolar vectors for each of wolves are based on another layer of stochastic randomization to diversify the solutions. Then, the generated bipolar vectors for each of wolves are essential for the transformation in the next step.

Step 7: Wolf Transformation
The wolves are further updated by the transformation by referring to the following piecewise equations:

$$\vec{X}_\alpha^* = \begin{cases} 1, & \text{if } \vec{X}_\alpha + \vec{B}_\alpha \geq 1 \\ -1, & \text{otherwise} \end{cases} \quad (22)$$

$$\vec{X}_\beta^* = \begin{cases} 1, & \text{if } \vec{X}_\beta + \vec{B}_\beta \geq 1 \\ -1, & \text{otherwise} \end{cases} \quad (23)$$

$$\vec{X}_\delta^* = \begin{cases} 1, & \text{if } \vec{X}_\delta + \vec{B}_\delta \geq 1 \\ -1, & \text{otherwise} \end{cases} \quad (24)$$

Thus, the wolf solutions will be updated and a random number between [0, 1] is generated.
Step 8: Stochastic Crossover
The position of the wolves will be updated after improving the $\vec{X}^*_\omega$ via stochastic crossover process.

$$\vec{X}^*_\omega = \begin{cases} 
\vec{X}^*_\alpha, & \text{if } (\text{rand} < \frac{1}{3}) \\
\vec{X}^*_\beta, & \text{if } \frac{1}{3} \leq \text{rand} < \frac{2}{3} \\
\vec{X}^*_\delta, & \text{else}
\end{cases} \quad (25)$$

Previously, the omega wolf $\vec{X}^*_\omega$ is yet to be determined. Thus, by considering the equation above, we can select the partner in which the $\vec{X}^*_\omega$ will be updated via stochastic crossover operator.

Step 9: The Best Wolf String Selection
Calculate the fitness of $\vec{X}^*_\alpha$, $\vec{X}^*_\beta$, $\vec{X}^*_\delta$ and $\vec{X}^*_\omega$. Update the wolf with the highest fitness as the new alpha wolf, $\vec{X}^*_\alpha$. Thus, the newly $\vec{X}^*_\alpha \neq NC$, otherwise Step 4 need to be repeated. Therefore, for the case of $\vec{X}^*_\alpha = NC$, the learning process can be assured to complete.

4. Experimental setup and implementation
The DHNN-3SAT model is implemented in Dev C++ Version 5.11 in Windows 10 Intel Core i5 with 2.2 GHz processor. The simulations are conducted by manipulating different number of neurons (NN) ranging from $15 \leq NN \leq 120$. In this work, the simulated dataset will be obtained by generating random clause and literal for each 3SAT logic. The fundamental parameters involved in both DHNN-3SAT models are set according to Table 1.

| Parameter                   | Values          |
|-----------------------------|-----------------|
| Number of Neurons           | $15 \leq NN \leq 120$ |
| Neuron Combinations ($q$)   | 100             |
| Tolerance Value ($\psi$)    | 0.001           |
| Number of Trials ($p$)      | 100             |
| CPU Time Threshold          | 24 hours        |
| Maximum Iteration ($T_{MAX}$) | 100          |

The full implementation of DHNN-3SATGWO and DHNN-3SATES are summarized as follows:
Calculate the local field and final states of $\kappa_{3SAT}$ and derive the cost function, $E_{3SAT}$.

Compute the synaptic weight of $\kappa_{3SAT}$ and calculate $H_{3SAT}^{min}$ of DHNN-3SAT.

Check the clauses satisfaction that lead to $E_{3SAT} = 0$ via:

- Grey Wolf Optimization (GWO)
- Exhaustive Search (ES)

Retrieval phase of DHNN-3SAT

Calculate the local field and final states of $\kappa_{3SAT}$

Compute final energy of DHNN-3SAT, $H_{3SAT}$

\[ |H_{3SAT} - H_{3SAT}^{min}| < \psi \]

Yes

- Global Minimum Energy
- Local Minimum Energy

No

Learning Error, Energy Analysis, Variability and Similarity Analysis

End

Figure 1. Algorithm and flowchart of DHNN-3SAT GWO and DHNN-3SATES.
5. Performance evaluation metrics

The learning process of the DHNN-3SAT is an intensive phase that contributes to the effectiveness of the final states obtained during retrieval phase. A consistent learning can be evaluated based on different domains such as accuracy, sensitivity and forecasting ability adopted by DHNN-3SAT.

Mean Absolute Error (MAE) is the standard error based on the average difference from the computed fitness values and expected fitness of the solutions of DHNN-3SAT obtained during learning phase. According to Willmott and Matsuura [22], MAE is apparently a reliable metric in assessing the accuracy of the learning phase of the neural network model. Thus, MAE for DHNN-3SAT learning phase is recrafted as follows:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |f_{max} - f_i|$$ (26)

where $f_i$ refers to the obtained fitness values, $n$ denotes the number of iterations and $f_{max}$ is the maximum number of fitness corresponds to the number of clauses of 3SAT logic.

Sum of Squared Error (SSE) is a statistical metric adopted to measure the dispersion of the data from the expected values [24]. In this work, SSE will be utilized in computing the dispersion of the fitness values obtained by DHNN-3SAT as compared to the maximum fitness during learning phase.

$$SSE = \sum_{i=1}^{n} (f_i - f_{max})^2$$ (27)

where $f_i$ refers to the obtained fitness values and $f_{max}$ is the maximum number of fitness corresponds to the number of clauses of 3SAT logic.

Mean Absolute Percentage Error (MAPE) is an extended version of MAE, where the values are normalized into percentage [21]. It measures the deviation of the fitness values obtained as compared to the maximum fitness of the optimal solutions.

$$MAPE = \sum_{i=1}^{n} 100 \frac{|(f_{max} - f_i)|}{f_i}$$ (28)

where $f_i$ denotes the fitness value during learning phase and $f_{max}$ is the expected maximum fitness value. Theoretically, the MAPE values are ranging $MAPE = [0, 100]$.

Energy analysis can be performed by observing the quality of the solutions corresponds whether it is global minima solution (corresponds to global minimum energy) as coined by Kasihmuddin et al. [15].

The Global Minima Ratio [17] can be calculated as follows:

$$\Phi_{3SAT} = \frac{1}{pq} \sum_{i=1}^{NN} N_{H_{3SAT}}$$ (29)

where $NN$ denotes the number of neurons, $p$ refers to the number of trials and $q$ depicts the neuron combinations.

In this work, 4 similarity analysis metrics will be utilized to compare the final states obtained by DHNN-3SAT. Theoretically, most of the neuron states being retrieved by DHNN have attained global minimum energy. By taking inspiration from work of [25], the similarity metrics will be further employed to assess the 3SAT logical representation form of final states retrieved by the network. The comparison will be done by taking the benchmark states $S_i^{max}$ with the
states attained by the network $S_i$. The formula of the general comparison of the benchmark state and the final state is given as follows:

$$C_{S_{i}^{\text{max}},S_i} = \{(S_{i}^{\text{max}}, S_i) | i = 1, 2, 3, ..., n\}.$$  \hspace{1cm} (30)

The standard specification variables can be defined by considering the following domains such as:

- $a$ refers the total number of occurrences for $(S_{i}^{\text{max}} = 1, S_i = 1)$ in $C_{S_{i}^{\text{max}},S_i}$
- $b$ denotes the total number of occurrences for $(S_{i}^{\text{max}} = 1, S_i = -1)$ in $C_{S_{i}^{\text{max}},S_i}$
- $c$ is the total number of occurrences for $(S_{i}^{\text{max}} = -1, S_i = 1)$ in $C_{S_{i}^{\text{max}},S_i}$

Ochiai Index is a variant of similarity index, measured based on the exclusion of the negative co-occurrences, with emphasis given in the positive states [23]. In this work, the Ochiai Index is modified to comply in finding the similarity between final states generated at the end of the simulation, $S_i$ and the benchmark final states, $S_{i}^{\text{max}}$.

$$OC(S_{i}^{\text{max}}, S_i) = \frac{a}{\sqrt{(a+b)(a+c)}}$$  \hspace{1cm} (31)

The correlation of Ochiai similarity index with the neuron variation will be closely be investigated in this work.

6. Results and discussion

The compatibility and performance of our proposed hybrid model, DHNN-3SATGWO with the standard model, DHNN-3SATES was evaluated by utilizing various performance evaluation metrics. The comparisons are discussed according to the analysis of Learning Mean Absolute Error (MAE), Learning Sum of Squared Error (SSE), Global Minima Ratio, Neuron Variation, and Ochiai Index.

Figure 2 until Figure 7 manifests the capability of the Grey Wolf algorithm with a Discrete Hopfield Neural Network (DHNN-3SATGWO) as compared with DHNN-3SATES in generating global solutions for 3-Satisfiability logic programming. The simulations were conducted by manipulating the different Number of Neurons (NN), ranging from $NN = 15$ until $NN = 120$.

The learning mean absolute error (MAE) achieved by the proposed model, DHNN-3SATGWO, and the standard model, DHNN-3SATES are shown in Figure 2. What stands out in Figure 2 is the consistent lower MAE, indicating fewer iterations needed during the learning phase. This suggests that the accuracy of DHNN-3SATGWO in minimizing the cost function of $\kappa_{3SAT}$ logical rule within less number of iterations. The $\kappa_{3SAT}$ leads to the minimum cost function easily due to various parameters in GWO. The updates in the solution vectors can be seen in hunting (encircling the prey) [2] and stochastic crossover [8], which increases the chance of obtaining the best solution that leads to $E_{\kappa_{3SAT}} = 0$. Apart from that, the effective randomization in most of the steps in GWO ensures continuous updates of the non-fit solutions in lesser learning iterations [7]. On the contrary, DHNN-3SATES is performed poor based on the MAE due to the trial and enumerate procedure in fully minimizing the cost function of $\kappa_{3SAT}$, despite the guarantee in the convergence [14]. As for DHNN-3SATGWO and DHNN-3SATES, the learning MAE is peaked at $NN = 120$ due to the higher complexity of the neurons.

The sensitivity of the hybrid models towards the possible error during the learning phase can be explained in Figure 3. DHNN-3SATGWO exhibits the lowest value of SSE, indicating that the model is less prone to any incoming error. Henceforth, the sensitivity of DHNN-3SATGWO is lower than DHNN-3SATES. The stochastic operators have ensured the update in the solutions.
and the final selection will be iteratively computed in attaining the best solution vector with the maximum fitness. Besides, there is a possibility of the omega wolf, the non-fit vector solutions to improve during the stochastic crossover phase. The most interesting trend in Figure 3 is the sharp increase of SSE in DHNN-3SATES indicating a non-effective learning phase. The trial and error mechanism [10] requires additional learning iterations to achieve \( E_{\kappa_{3S\text{SAT}}} = 0 \).

According to Figure 4, it was observed that the MAPE for DHNN-3SAT models possesses a similar trend. When the numbers of neurons are higher, the MAPE for the both hybrid models will be increase gradually. However, DHNN-3SATGWO generates lesser MAPE due to the capability of the models to attain the global convergence in fewer iterations. The learning MAPE for DHNN-3SATGWO is substantially better than DHNN-3STAES. The optimization operator in GWO especially during the wolves encircling process and stochastic crossover have enhanced the non-fit solution to become global solutions without consuming trial and error stage. Thus, it was clear that the percentage of MAPE depicts the minimum values as compared to the standard model. This is the result of the searching and updating operator in GWO that mimics the hunting mechanism of the grey wolf in their habitat. The learning errors such as MAE, SSE, and MAPE confirm that DHNN-3SATGWO a good choice for 3SAT logic programming under different levels of complexities.

Figure 5 elucidates the energy analysis in terms of the global minima ratio for the DHNN-3SAT models provided the complexity of the network ranging from \( NN = 15 \) until \( NN = 120 \). Based on Figure 5, both models have consistently recorded the global minima ratio ranging from \([0.96, 1.0]\). It is worth discussing these interesting facts revealed by the retrieval capability of DHNN in ensuring the final states of the neurons lead to global convergence. It is worth to mention that DHNN-3SATGWO did outperform DHNN-3SATES during \( NN = 105 \) and \( NN = 120 \), proving that the slight impact of the learning algorithm towards optimal retrieval phase. This has been driven by the robust operator in GWO such as the randomization [8] during the transformation phase for the solution vectors (wolves). However, the energy analysis might not be sufficient to determine the quality of the solutions obtained by the proposed model.

The quality of the solutions can be analyzed by using variability analysis as shown in Figure 6 and similarity index as given in Figure 7. Generally, the neuron variations for DHNN-3SATGWO is apparently higher than DHNN-3SATES for most of the \( NN \). This implies the impact of the GWO learning mechanism in attaining better retrieval capability, thus generating the final states with higher variability. Therefore, less overfitting states can be found in our proposed model. According to Figure 7, the Ochiai similarity index provides the trends as DHNN-3SATGWO recorded lower similarity values as opposed to the standard model. Theoretically, the Ochiai index has revealed the retrieved final states in the form of \( \kappa_{3S\text{SAT}} \) skewed towards the positive values as compared to the benchmark states. Based on Figure 6, the variability was inversely correlated with the similarity analysis as shown in Figure 6 and Figure 7. Therefore, the higher the neuron variations, indicated the lower the similarity index recorded in terms of the Ochiai Index. However, the differences and fluctuations in the result were comparable for both models due to the existence of the small amount of overfitting final states attained during the retrieval phase.
Figure 2. Mean Absolute Error (MAE) during learning for DHNN-3SATGWO and DHNN-3SATES.

Figure 3. Sum of Squared Error (SSE) during learning for DHNN-3SATGWO and DHNN-3SATES.
Figure 4. Mean Absolute Percentage Error (MAPE) during learning for DHNN-3SATGWO and DHNN-3SATES.

Figure 5. Global minima ratio analysis of DHNN-3SATGWO and DHNN-3SATES.
Figure 6. Neuron variation for DHNN-3SATGWO and DHNN-3SATES.

Figure 7. Ochiai similarity index for DHNN-3SATGWO and DHNN-3SATES.
7. Conclusion

Overall, this work has successfully demonstrated the capability and compatibility of robust swarm metaheuristics namely, the Grey Wolf Optimization (GWO) algorithm as a learning algorithm for 3 Satisfiability logic programming analysis in Discrete Hopfield Neural Network (DHNN). The findings confirmed the effectiveness of DHNN-3SATGWO as compared with the standard model DHNN-3SATES in terms of error evaluations, energy analysis, the variability of the neuron, and similarity analysis with the different number of neuron and number of learning. Also, we have concluded that the neuron variation is inversely correlated to the number of neuron variations recorded by DHNN-3SATGWO and DHNN-3SATES. Although DHNN-3SATGWO outperformed DHNN-3SATES for the learning error and energy analysis, the similarity analysis of both models was possibly closer due to the existence of overfitting final states. In the future, the extended modification in the retrieval phase of DHNN is certainly required to disentangle the possible weaknesses such as the complexities and overfitting solutions. In addition, further research directions include the incorporation of different variants of non-systematic Boolean satisfiability logical representation ranging from the Random Satisfiability, Weighted Satisfiability, and Maximum Satisfiability logical rule.

Acknowledgments

This research is financially supported via Fundamental Research Grant Scheme (FRGS), Ministry of Education Malaysia, grant number 203/PJJAUH/6711751 and Universiti Sains Malaysia.

References

[1] Emary E, Zawbaa H M and Hassanien A E 2016 Binary grey wolf optimization approaches for feature selection Neurocomputing 8(172) 371-381
[2] Mirjalili S, Mirjalili S M and Lewis A 2014 Grey wolf optimizer. Advances in engineering software 1(69) 46-61
[3] Sujatha K and Punithavathani D S 2018 Optimized ensemble decision-based multi-focus image fusion using binary genetic Grey-Wolf optimizer in camera sensor networks Multimedia Tools and Applications 77(2) 1735-1759
[4] Panwar L K, Reddy S, Verma A, Panigrahi B K and Kumar R 2018 Binary grey wolf optimizer for large scale unit commitment problem Swarm and Evolutionary Computation 1(38) 251-266
[5] Sharma P, Sundaram S, Sharma M, Sharma A and Gupta D 2019 Diagnosis of Parkinson’s disease using modified grey wolf optimization Cognitive Systems Research 1(54) 100-115
[6] Abdel-Basset M, El-Shahat D, El-henawy I, de Albuquerque VH and Mirjalili S 2020 A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection Expert Systems with Applications 139 112824
[7] Tawhid M A and Ali A F 2017 A hybrid grey wolf optimizer and genetic algorithm for minimizing potential energy function Memetic Computing 9(4) 347-359
[8] Hu P, Pan J S and Chu C S 2020 Improved Binary Grey Wolf Optimizer and Its application for feature selection. Knowledge-Based Systems 195 105746
[9] Hopfield JJ 1982 Neural networks and physical systems with emergent collective computational abilities Proceedings of the national academy of sciences 79(8) 2554-2558
[10] Sathasivam S 2010 Upgrading logic programming in Hopfield network. Sains Malaysiana 39(1) 115-118
[11] Garcia L, Talaván P M and Yáñez J 2017 Improving the Hopfield model performance when applied to the traveling salesman problem Soft Computing 21(14) 3891-3905
[12] Hillar C J and Tran N M 2018 Robust exponential memory in Hopfield networks. The Journal of Mathematical Neuroscience 8(1) 1-20
[13] Cabrera E and Sosa H 2018 Generating exponentially stable states for a Hopfield Neural Network Neurocomputing 275 358-365
[14] Kasihmuddin M S M, Mansor M A and Sathasivam S 2017 Hybrid Genetic Algorithm in the Hopfield Network for Logic Satisfiability Problem Pertanika Journal of Science and Technology 25(1) 139-152
[15] Kasihmuddin M S M, Mansor M A and Sathasivam S 2018 Discrete Hopfield Neural Network in Restricted Maximum k-Satisfiability Logic Programming Sains Malaysiana 47(6) 1327-1335
[16] Rajagopal, K., Munoz-Pacheco J M, Pham V T, Hoang D V, Alsaudi F E and Alsaudi F E 2018 A Hopfield
neural network with multiple attractors and its FPGA design The European Physical Journal Special Topics 227 811-820

[17] Sathasivam S, Mamat M, Mansor M and Kasihmuddin M S M 2020 Hybrid Discrete Hopfield Neural Network based Modified Clonal Selection Algorithm for VLSI Circuit Verification Pertanika Journal of Science Technology 28(1) 227-243

[18] Zamri N E, Mansor M A, Kasihmuddin M S M, Alway A, Jamaludin M S Z and Alzaeemi S A 2020 Amazon Employees Resources Access Data Extraction via Clonal Selection Algorithm and Logic Mining Approach Entropy 22(6) 596

[19] Aiman U and Asrar N 2015 Genetic algorithm based solution to SAT-3 problem Journal of Computer Sciences and Applications 3(2) 33-39

[20] Mansor M A and Sathasivam S 2016 Accelerating activation function for 3-satisfiability logic programming International Journal of Intelligent Systems and Applications 8(10) 44

[21] Kasihmuddin M S M, Mansor M A, Jamaludin S.Z.M. and Sathasivam S 2020. Systematic Satisfiability Programming in Hopfield Neural Network-A Hybrid Expert System for Medical Screening. Communications in Computational and Applied Mathematics, 2(1) 1-6

[22] Willmott C J, and Matsuura K 2005 Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing averagemodel performance Climate research 30 79–82

[23] Bolton, H.C., 1991. On the mathematical significance of the similarity index of Ochiai as a measure for biogeographical habitats. Australian journal of zoology 39(2) 143-156

[24] Mansor M, Mohd Jamaludin S Z , Kasihmuddin M M S, Alzaeemi S A, Md Basir M F and Sathasivam S 2020. Systematic boolean satisfiability programming in radial basis function neural network. Processes 8(2) 214

[25] Sathasivam S, Mansor M A, Kasihmuddin M S M and Abubalar H 2020 Election Algorithm for Random k Satisfiability in the Hopfield Neural Network Processes 8(5) 568