Estimation of Distribution Algorithm with Discrete Hopfield Neural Network for GRAN3SAT Analysis

Yuan, Y.G, Gao
School of Medical Information Engineering, Chengdu University of Traditional Chinese Medicine, Chengdu 611137, China; School of Mathematical Sciences, Universiti Sains Malaysia, Penang 11800, Malaysia
gaoyuan@student.usm.my

Chengfeng, C.Z, Zheng
School of Mathematical Sciences, Universiti Sains Malaysia, Penang 11800, Malaysia
1002953832@qq.com

Ju, J.C, Chen*
School of Medical Information Engineering, Chengdu University of Traditional Chinese Medicine, Chengdu 611137, China; School of Mathematical Sciences, Universiti Sains Malaysia, Penang 11800, Malaysia
chenju@cdutcm.edu.cn

Yueling, Y.G, Guo
School of Mathematical Sciences, Universiti Sains Malaysia, Penang 11800, Malaysia
guoyueling1982@163.com

ABSTRACT

The Discrete Hopfield Neural Network introduces a G-Type Random 3 Satisfiability logic structure, which can improve the flexibility of the logic structure and meet the requirements of all combinatorial problems. Usually, Exhaustive Search (ES) is regarded as the basic learning algorithm to search the fitness of neurons. To improve the efficiency of the learning algorithm. In this paper, we introduce the Estimation of Distribution Algorithm (EDA) as a learning algorithm for the model. To study the learning mechanism of EDA to improve search efficiency, this study focuses on the impact of EDA on the model under different proportions of literals and evaluates the performance of the model at different phases through evaluation indicators. Analyze the effect of EDA on the synaptic weights and the global solution. From the discussion, it can be found that compared with ES, EDA has a larger search space at the same efficiency, which makes the probability of obtaining satisfactory weights higher, and the proportion of global solutions obtained is higher. Higher proportions of positive literals help to improve the model performance.

CCS CONCEPTS

• **CCS CONCEPT Computing methodologies** → Artificial intelligence.

KEYWORDS

Hopfield Neural Network, Exhaustive Search, Estimation of Distribution Algorithm, Meta-heuristic

*Corresponding author.

CACML 2023, March 17–19, 2023, Shanghai, China

ACM ISBN 978-1-4503-9944-9/23/03…$15.00
https://doi.org/10.1145/3590003.3590021

ACM Reference Format:
Yuan, Y.G, Gao, Chengfeng, C.Z, Zheng, Ju, J.C, Chen, and Yueling, Y.G, Guo. 2023. Estimation of Distribution Algorithm with Discrete Hopfield Neural Network for GRAN3SAT Analysis. In *2023 2nd Asia Conference on Algorithms, Computing and Machine Learning (CACML 2023), March 17–19, 2023, Shanghai, China.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3590003.3590021

1 INTRODUCTION

The principle of Hopfield neural network (HNN) simulating human memory, proposed by J. J. Hopfield [1] in 1982, can solve pattern recognition problems and combinatorial optimization problems. HNN guarantees convergence to a local minimum, but it is possible to converge to a non-global minimum. Discrete Hopfield neural network (DHNN) is the earliest proposed binary neural network. The input and output of the neuron only take {0, 1} or {-1, 1}. In 1990, W. McCulloch and W. Pitts [2] proposed that the relationship between neurons can be handled by propositional logic. In 1992, Wan Abdullah [3] proposed the model of introducing the satisfiability problem into DHNN. Since then, the DHNN based on Satisfiability (SAT) logic programming has undergone a series of developments, including 3SAT [4], MAX3SAT [5], RAN3SAT [6], YRAN2SAT [7], r2SAT [8] combination with DHNN. In 2022, Gao Yuan proposed G-Type Random 3 Satisfiability (GRAN3SAT) [6] so that all combination problems can be satisfied. As the number of clauses increases, the model's task of finding consistent explanations becomes more complex. The exhaustive search (ES) is used as the learning algorithm in the learning phase [9]. ES is less efficient when the search space is large. Estimation of Distribution Algorithm (EDA), also known as the genetic algorithm based on the probability distribution model, was first proposed in 1996 [10]. In 2010, Peralta [11] evaluated methods for evolving neural network architectures using EDA. In 2013, Donate [12] used EDA to introduce the automatic design of artificial neural networks for forecasting time series to improve the final forecast accuracy. According to these related works, we can deduce that the EDA can be combined with the neural network to obtain faster convergence. Based on this, EDA can be considered a promising algorithm to facilitate the learning phase. However, no attempt has been made to exploit the EDA algorithm as an optimal learning method in Discrete Hopfield Neural Networks (DHNN), especially in optimizing GRAN3SAT logic representation and analysis.

In this paper, the GRAN3SAT logic structure is applied in DHNN. To efficiently find a consistent solution to the satisfiability problem, EDA is used as a learning algorithm to search the fitness of neurons. EDA achieves the search and convergence of satisfactory solutions through continuous updating and testing. This study focuses on different proportions of positive and negative literals, analyzes the performance improvement of EDA as a learning algorithm for GRAN3SAT compared with ES, and evaluates the behavior changes of the model at different phases through evaluation indicators.

2 EDA IN THE DHNN BASED ON GRAN3SAT

2.1 Logic Rules of GRAN3SAT.

GRAN3SAT is a novel non-systematic SAT logic structure represented in conjunctive normal form. Logic consists of a set of different literals and clauses. GRAN3SAT mainly consists of third-order, second-order, and first-order logic clauses randomly. Each literal value is of the form {-1, 1}. The general formula of GRAN3SAT is P_G, detailed as follows.

a) A set of NN literals: $A_1, A_2, A_3, \ldots\ldots, A_{NN}$. Randomly generated for each literal state.
b) The number of clauses $\{x_i, y_i, z_i\}$is randomly generated.$x_i$is the number of third-order logic clauses, $y_i$is the number of second-order logic clauses, and z_i is the number of first-order logic clauses.
c) The representation of a clause:

Third-order logic clause: $C_1^{(3)}, C_2^{(3)}, \ldots\ldots\ldots, C_{x_j}^{(3)}$, whereby $C_{m_j}^{(3)} = (A_m \vee A_n \vee A_k), m, n, k \in N*$.

Second-order logic clause: $C_1^{(2)}, C_2^{(2)}, \ldots\ldots\ldots, C_{y_j}^{(2)}$, whereby $C_{n_j}^{(2)} = (A_m \vee A_n), m, n \in N\ *$.

First-order logic clause: $C_1^{(1)}, C_2^{(1)}, \ldots\ldots\ldots, C_{z_j}^{(1)}$, whereby $C_{k_j}^{(1)} = A_m, m \in N\ *$.

$$P_G = \wedge_{i=1}^{x_j} C_i^{(3)} \wedge_{i=1}^{y_j} C_i^{(2)} \wedge_{i=1}^{z_j} C_i^{(1)} \tag{1}$$

2.2 DHNN

The bipolar neurons of DHNN used in this study are represented by {-1, 1}, and the update equation of neuron state of DHNN is as follows:

$$S_i = \begin{cases} 1, & \sum_{jk} W_{ijk} S_j S_k \geq \theta_i \\ -1, & \sum_{jk} W_{ijk} S_j S_k < \theta_i \end{cases} \tag{2}$$

whereby S_i is the neuron state,W_{ij} is the synaptic weight in the two neurons. θ_i is the threshold. The cost function of GRAN3SAT-DHNN is Cost_{P_G}, and the formula is as follows:

$$\mathrm{Cost}_{P_G} = \frac{1}{2^3} \sum_{j=1}^{x_i} \delta_{j_1}^{(3)} \delta_{j_2}^{(3)} \delta_{j_3}^{(3)} + \frac{1}{2^2} \sum_{j=1}^{y_i} \delta_{j_1}^{(2)} \delta_{j_2}^{(2)} + \frac{1}{2} \sum_{j=1}^{z_i} \delta_{j_1}^{(1)} \tag{3}$$

$$\delta_{j_x}^{(k)} = \begin{cases} 1 + S_{A_{j_x}}, & when \quad A_{j_x} \\ 1 - S_{A_{j_x}}, & when \quad \neg A_{j_x} \end{cases} \tag{4}$$

Whereby $k = 1, 2, 3$. The value of η_{P_G} describes the logical consistency of P_G. When the logical inconsistency reaches the minimum value, the weight can be obtained through the Wan Abdullah method [5]. The probability formula for consistent interpretation is as follows:

$$\lambda(\eta_{P_G} = 0) = \left(1 - \frac{1}{2^3}\right)^{x_i} \left(1 - \frac{1}{2^2}\right)^{y_i} \left(1 - \frac{1}{2}\right)^{z_i} \tag{5}$$

In the testing phase, the Formula (6)(7) represent the local field formula and the update neuron state. The activation function is the Hyperbolic Tangent Activation Function (HTAF) [7].

$$h_i = \sum\nolimits_{k \neq i, j} \sum\nolimits_{j \neq i,} W_{ijk} S_j S_k + \sum\nolimits_{j \neq i} W_{ij} S_j + W_i \tag{6}$$

$$S_{u_i} = \begin{cases} 1, & \sum_{k \neq i, j} \sum_{j \neq i,} W_{ijk} S_j S_k + \sum_{j \neq i} W_{ij} S_j + W_i \geq 0 \\ -1, & \sum_{k \neq i, j} \sum_{j \neq i,} W_{ijk} S_j S_k + \sum_{j \neq i} W_{ij} S_j + W_i < 0 \end{cases} \tag{7}$$

S_i and S_{u_i} represent the initial state and the update state. W_{ijk},W_{ij}, and W_i represent the weights of the third, second and first order of DHNN. The Lyapunov energy function En_{P_G} is obtained by formula(8), and the minimum energy $En_{P_G}^{min}$ is obtained by formula(9).

$$En_{P_G} = -\frac{1}{3} \sum_i \sum_{j \neq i} \sum_{k \neq i, j} W_{ijk} S_i S_j S_k - \frac{1}{2} \sum\nolimits_i \sum\nolimits_{j \neq i} W_{ij} S_i S_j - \sum\nolimits_i W_i S_i \tag{8}$$

$$En_{P_G}^{min} = -\left(\frac{x_i}{2^3} + \frac{y_i}{2^2} + \frac{z_i}{2}\right) \tag{9}$$

The current DHNN convergence formula (10) is as follows,tv means tolerance value.

$$\left| En_{P_G} - En_{P_G}^{min} \right| \leq tv \tag{10}$$

Introduce GRAN3SAT into DHNN to become $GRAN3SAT_{DHNN}$.

2.3 EDA

The EDA is a population evolution algorithm based on statistical theory. By establishing a probability formula to describe the distribution information of satisfiable solutions in the search range, a new population is generated by random sampling. The evolution of the population is achieved through repeated iterations.The EDA flow is as follows:

Step 1: Initialization.

Initialize the number of populationsN_p. The dimension of each individual is N_n. The initial population is $X_i = \{x_{ij} | i = 1, 2\ldots, N_p; j = 1, 2\ldots, N_n\}$, where $x_{ij} \in \{1, -1\}$.

Step 2: Calculate the fitness function.

The neuron fitness of the above X_i is calculated using the following formula:

$$f(X_i) = NC - \left(\sum_m C_i^{(3)} + \sum_n C_i^{(2)} + \sum_k C_i^{(1)}\right) \tag{11}$$

$$C_i^{(x)} = \begin{cases} 1, & if\,\text{clauseissatisfied} \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

where NC is the number of clauses. The larger the $f(X_i)$, the greater the number of unsatisfied clauses.

Step 3: probability model.

Select dominant populations based on $f(X_i)$. Construct a probability model [13] based on N dominant population, where $N <$

Table 1: Main parameters of $GRAN3SAT_{DHNN}$.

Parameter	Value
Different proportions of negative literals (P_N)	0.1,0. 3,0.5,0.7,0.9
Number of neurons(NN)	6 ≤ NN ≤ 100
Activation function	HTAF [7]
Number of neuron combination	100
Relaxation rate	2 [7]
Number of learn and test trial	100
Tolerance value	0.001 [7]

Table 2: List of main parameters in the indicator.

Parameter	Explanation.
f_{NC}	Maximum fitness achieved
f_{i}	Current fitness achieved
W_{WAN}	Satisfactory synaptic weights
W_i	Number of local minimum solution
v_w	Current number of weights
v_{wc}	$v_{wc} = v_w \cdot v_{combmax}$
ε_{min}	Minimum energy value
ε_f	Final energy function value
v_G	Number of global solutions
v_t	Number of testing trials
v_{tc}	$v_{tc} = v_t \cdot v_{combmax}$

N_P,the formulais as follows.

$$P(\mathrm{x}_j) = \frac{1}{N}\sum_{i=1}^{N}\chi_{ij} \tag{13}$$

$$\chi_{ij} = \begin{cases} 1, & x_{ij} = 1 \\ 0, & x_{ij} = -1 \end{cases} \tag{14}$$

Step 4: Update data.

Randomly generate a new population with a size of N_p according to the probability model.

Step 5: Judgment.

Judging whether the conditions for ending the loop are met, if not, jump to Step 2.

Introduce EDA into $GRAN3SAT_{DHNN}$ to become $GRAN3SAT_{DHNN}^{EDA}$.

3 EXPERIMENTAL SETTINGS

This section introduces the experimental parameters and evaluation metrics of $GRAN3SAT_{DHNN}$. In the system, the main experimental parameters involved are defined in Table 1. This experiment mainly uses MATLAB 2021a for the simulation.

This paper uses four performance indicators to evaluate the effectiveness of the $GRAN3SAT_{DHNN}$. These metrics are evaluated by Mean Absolute Error (MAE) for learning error analysis, weight analysis, energy analysis, and global solution analysis. Table 2 describes the parameters used for evaluation in the testing and learning phases. The formula for the learning phase and testing phase is as follows.:

$$MAE_{\mathrm{learn}} = \sum_{i=1}^{v_l}\frac{|f_{NC} - f_{\mathrm{i}}|}{v_l} \tag{15}$$

$$MAE_{\mathrm{weight}} = \frac{\sum_{i=1}^{v_{wc}}|W_{WAN} - W_i|}{v_{wc}} \tag{16}$$

$$MAE_{\mathrm{energy}} = \frac{\sum_{i=1}^{v_{tc}}\left|\varepsilon_{min} - \varepsilon_f\right|}{v_{tc}} \tag{17}$$

$$ZM_{\mathrm{test}} = \frac{v_G}{v_{tc}} \tag{18}$$

4 RESULT AND DISCUSSION

The purpose of this work is to analyze the impact of EDA as a learning algorithm on the overall behavior of $GRAN3SAT_{DHNN}$ in the learning phase, testing phase. In $GRAN3SAT_{DHNN}$, the MAE index is used for evaluation. Compared with ES, EDA improves neuron fitness through update iterations and narrows down the search space. This paper discusses the advantages of EDA in the learning phase and testing phase.

Figure 1 and 2 show the performance changes of EDA and ES when the neuron states have different proportions under indicators $RMSE_{\mathrm{learn}}$ and $RMSE_{\mathrm{weight}}$. $RMSE_{\mathrm{learn}}$ and $RMSE_{\mathrm{weight}}$ respectively quantify the fitness and weight error of neurons. It can be seen that the error of each index of EDA is better than that of ES. There is no obvious difference between $GRAN3SAT_{DHNN}^{EDA}$ and $GRAN3SAT_{DHNN}^{ES}$ under different P_N, different proportions of literals do not affect the fitness of neuron states of different

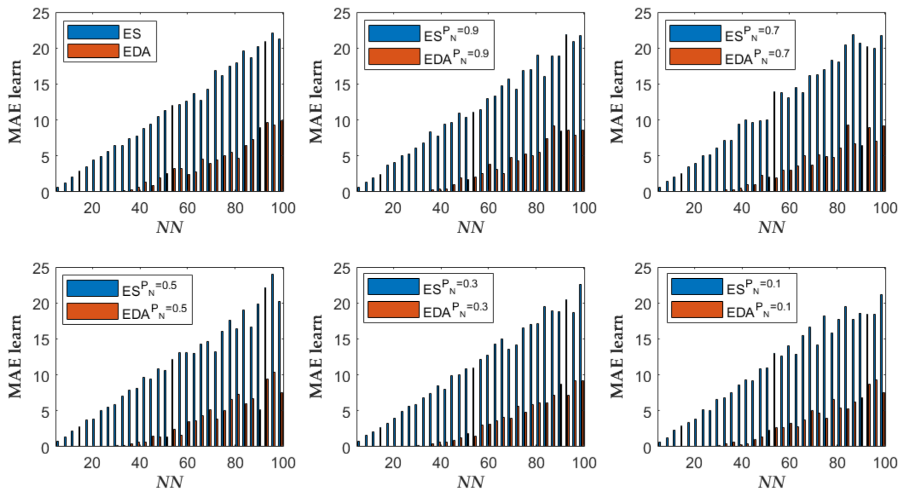

Figure 1: Changes in MAE_{learn} of EDA and ES under different P_N.

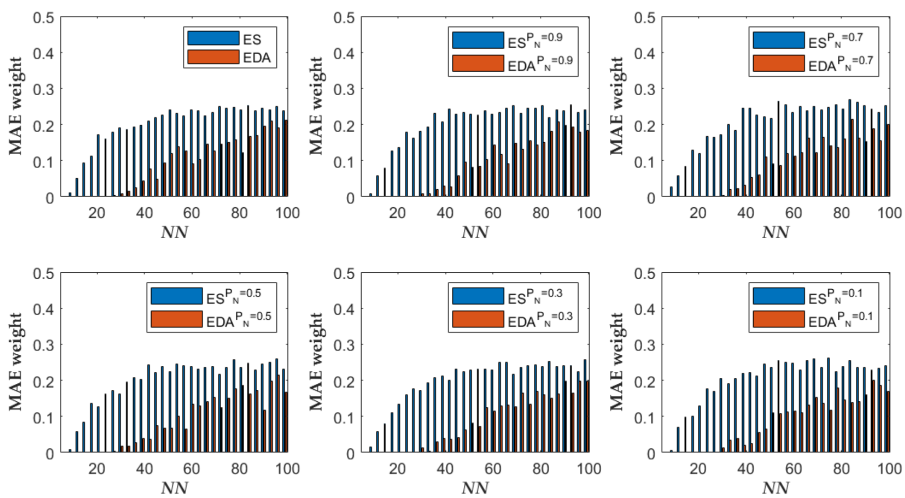

Figure 2: Changes in MAE_{weight} of EDA and ES under different P_N.

models, $RMSE^{ES}_{\text{weight}}$ generally shows a steady state after rising, and $RMSE^{EDA}_{\text{weight}}$ shows a linear rise. This is because as the NN increases, the solution search space of EDA expands, the fitness of neurons decreases, and satisfactory weights can no longer be obtained. To sum up, it is easier to find the optimal weight value for $GRAN3SAT^{EDA}_{DHNN}$ than ES in the learning phase. When adjusting the state ratio of neurons, different proportions of neuron states have no significant impact on $GRAN3SAT^{EDA}_{DHNN}$.

Figures 3 and 4 show the $GRAN3SAT^{EDA}_{DHNN}$ energy error distributionMAE_{energy} and the global solution proportion ZM_{test} under different P_N during the test phase. It can be obtained that the average energy distribution of $GRAN3SAT^{ES}_{DHNN}$ with different P_N is between 4.6 and 5.0. With the decrease of P_N under EDA, the energy distribution decreases gradually decreases from 4.5 to 2.8. The energy distribution of ES is generally higher than that of EDA. It can be seen that as the positive state of neurons increases, a is easier to obtain a smaller energy error and $GRAN3SAT^{EDA}_{DHNN}$ larger global solution ratio than $GRAN3SAT^{ES}_{DHNN}$. It is mainly due to two factors, the first is that it is easier to obtain satisfactory synaptic weights in the learning phase, and the second is that as P_N becomes larger, the generated clauses are easier to increase the number of global solutions.

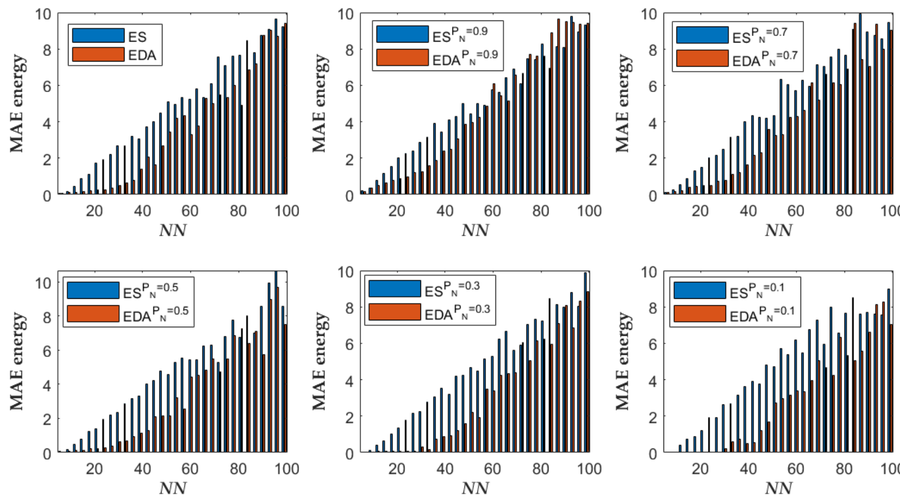

Figure 3: Changes in MAE_{energy} of EDA and ES under different P_N.

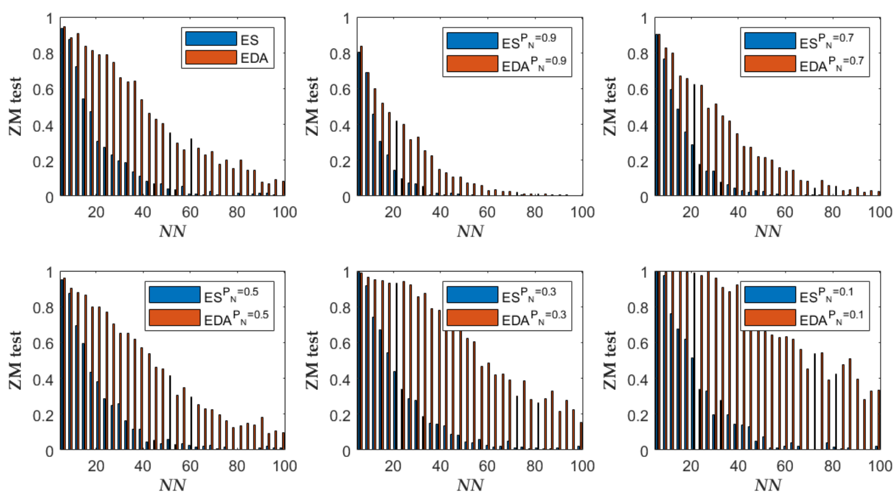

Figure 4: Changes in ZM_{test} of EDA and ES under different P_N.

5 CONCLUSIONS

The learning mechanism of $GRAN3SAT_{DHNN}$ based on the EDA algorithm has the following conclusions: Compared with ES, it has a larger search space at the same efficiency, so the probability of obtaining satisfactory weights in the learning phase is higher, and the proportion of global solutions obtained in the testing phase is higher. As the NN increases, the advantages of EDA become more obvious. Different proportions of negative literals P_N has no significant impact on neuron fitness and weight error in the learning phase, and a smaller P_N in the testing phase helps to reduce the energy distribution and increase the proportion of the global solution, thereby improving $GRAN3SAT_{DHNN}^{EDA}$ performance.

ACKNOWLEDGMENTS

Chengdu University of Traditional Chinese Medicine Xinglin Scholar Project.ZRQN2018013,QNXZ2018042.

REFERENCES

[1] Atul Adya,Hopfield, J. J. 1982. Neural networks and physical systems with emergent collective computational abilities. Proceedings of the national academy of sciences, 79(8), 2554-2558. https://doi.org/10.1073/pnas.79.8.2554

[2] McCulloch W S, Pitts W. 1990. A logical calculus of the ideas immanent in nervous activity. Bulletin of mathematical biology, 52(1), 99-115.W. A. T. W. Abdullah, International journal of intelligent systems. 7, 513-519 (1992). https://doi.org/10.1007/BF02459570

[3] Abdullah, W. A. T. W. 1992. Logic programming on a neural network. International journal of intelligent systems, 7(6), 513-519. https://doi.org/10.1002/int.4550070604

[4] Mansor M A, Sathasivam S. 2016. Accelerating activation function for 3-satisfiability logic programming. International Journal of Intelligent Systems and Applications, 8(10), 44. https://doi.org/10.5815/ijisa.2016.10.05
[5] Kasihmuddin M S M, Mansor M A, Sathasivam S. 2018. Discrete Hopfield neural network in restricted maximum k-satisfiability logic programming[J]. Sains Malaysiana, 47(6): 1327-1335. htpe://dx. doi. org/10.17576/jsm-2018-4706-30
[6] Karim, S. A., Zamri, N. E., Alway, A., Kasihmuddin, M. S. M., Ismail, A. I. M., Mansor, M. A., Hassan, N. F. A. 2021. Random satisfiability: A higher-order logical approach in discrete Hopfield Neural Network. IEEE Access, 9, 50831-50845. https://doi.org/10.1109/ACCESS.2021.3068998
[7] Guo, Y., Kasihmuddin, M. S. M., Gao, Y., Mansor, M. A., Wahab, H. A., Zamri, N. E., & Chen, J. 2022. YRAN2SAT: A novel flexible random satisfiability logical rule in discrete hopfield neural network. Advances in Engineering Software, 171, 103169. https://doi.org/10.1016/j.advengsoft.2022.103169
[8] Zamri, N. E., Azhar, S. A., Mansor, M. A., Alway, A., & Kasihmuddin, M. S. M. (2022). Weighted Random k Satisfiability for k= 1, 2 (r2SAT) in Discrete Hopfield Neural Network. Applied Soft Computing, 109312. https://doi.org/10.1016/j.asoc.2022.109312
[9] Gao, Y., Guo, Y., Romli, N. A., Kasihmuddin, M. S. M., Chen, W., Mansor, M. A., Chen, J. 2022. GRAN3SAT: Creating Flexible Higher-Order Logic Satisfiability in the Discrete Hopfield Neural Network. Mathematics, 10(11), 1899. https://doi.org/10.3390/math10111899
[10] Mühlenbein, H., Paass, G. 1996. From recombination of genes to the estimation of distributions I. Binary parameters. In International conference on parallel problem solving from nature. 178-187. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-61723-X_982
[11] Peralta, J., Gutierrez, G., Sanchis, A. 2010. Time series forecasting by evolving artificial neural networks using genetic algorithms and estimation of distribution algorithms. In The 2010 international joint conference on neural networks (IJCNN) (pp. 1-8). IEEE. https://doi.org/10.1109/IJCNN.2010.5596892
[12] Donate, J. P., Li, X., Sánchez, G. G., de Miguel, A. S. 2013. Time series forecasting by evolving artificial neural networks with genetic algorithms, differential evolution and estimation of distribution algorithm. Neural Computing and Applications, 22(1), 11-20. https://doi.org/10.1007/s00521-011-0741-0
[13] Mühlenbein, H. 1997. The equation for response to selection and its use for prediction. Evolutionary computation, 5(3), 303-346. https://doi.org/10.1162/evco.1997.5.3.303