Pattern 2 Satisfiability Analysis via Hybrid Artificial Bee Colony Algorithm as a Learning Algorithm

Mohd. Asyraf Mansor¹, Mohd Shareduwan Mohd Kasihmuddin²*, Siti Zulaikha Mohd Jamaluddin², Saratha Sathasivam²

¹School of Distance Education, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia
² School of Mathematical Sciences, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia

*Corresponding Author

Abstract: A systematic and optimal learning algorithm is an essential domain in pattern verification and constraint satisfaction logic, such as in Pattern 2 Satisfiability (P2SAT). P2SAT is a class of Boolean based satisfiability problem that focuses on generating the optimum pattern with respect to the restricted Boolean logic. The task required a dynamic searching method in order to facilitate the verification process of generating the global PSAT pattern. The existing hybrid model of Artificial Bee Colony Algorithm (ABC) and discrete Hopfield Neural Network (DHNN) only focused in 2SAT logic programming with simulated data. In this research, we propose the optimal hybrid model by employing the hybrid ABC with DHNN in P2SAT. In order to verify the capability of our proposed model, we compare by employing the Genetic Algorithm (GA) as the learning method in doing PSAT as the benchmark. The experimental results manifested the performance of the proposed model in learning and retrieving the PSAT patterns. The analysis for the performance is based on root mean square error (RMSE), mean absolute percentage error (MAPE), the Global P2SAT and the CPU time evaluations obtained from the simulation. The simulations performed on different hybrid models reveal the power of hybrid ABC with DHNN in verifying and generating global P2SAT for the higher-order patterns. This study provides new insight and approach in P2SAT, especially in enhancing the learning algorithm to be more robust and systematic.

Keywords: Pattern 2 Satisfiability, Discrete Hopfield neural network, Artificial Bee Colony Algorithm, Genetic Algorithm.

1. Introduction

Pattern satisfiability (PSAT) is the employment of Boolean satisfiability analysis that bridges the pattern recognition and higher-order logical representation. The main conundrum with higher order PSAT is the complexity problem where the process of global PSAT will trigger the attainment of local PSAT. The development of computational paradigm to generate a solid grasp of global PSAT can be done by utilizing nature inspired metaheuristic and artificial neural network as the learning and training method for PSAT. This work is inspired by the traditional approach adopted by Mansor et al. [1] that integrates the discrete Hopfield neural network in optimizing the pattern satisfiability problem. Even though the standalone DHNN might serve as a promising learning algorithm, the higher order PSAT requires a robust learning method as well as simpler logical representation. The task of PSAT has been seen in work of Alizadeh and Sharaﬁnejad [2] where the automated pattern generation in circuit is represented in terms of Boolean SAT with different level of complexities. However, the promising result suggests the opportunity to improve the existing work of PSAT to be further enhanced by taking Boolean SAT representation. Thus, the 2-Satisfiability logical representation is employed to the pattern satisfiability (P2SAT) in order to solidify the logical representation before undergoing the learning and retrieval process. Bagchi et al. [3] employed the 2SAT logical rule in fault processors diagnosis in tremendous computing systems. In another development, Alzaeeimi [4] proposed the 2SAT programming in optimizing the output weight via Radial basis function neural network (RBFNN). The work of 2SAT has been further developed by Kasihmuddin et al. [5] in Bezier curve reconstruction. Thus, the PSAT is adopted based on 2SAT logical representation, which form Pattern 2 Satisfiability (P2SAT). P2SAT will be processed by DHNN in learning and retrieval phase in order to generate the feasible patterns.

In this research, the searching capability during learning phase is prominent in order to be able to generate the Global
P2SAT by manipulating different combination of \( I_P \) per execution. The standalone learning algorithm deployed in PSAT verification with DHNN in the work of Mansor et al. [1] utilized the exhaustive search mechanism that will occasionally produce Local PSAT pattern. Thus, the idea of integrating metaheuristic approach will greatly reduce the problem in optimizing P2SAT.

Artificial bee colony (ABC) algorithm is a class of swarm based learning algorithm as developed by Karaboga and Bastürk [6] by taking the modelling foraging behaviour of honey bees into a systematic metaheuristic algorithm. In another development, Jia et al. [7] applied the ABC algorithm with bitwise operation in the improving the fast maneuvers of employed bees and onlooker bees in a particular population. The other research works of ABC can be found in Abdelhalim et al. [8], Cao et al. [9], Sonmez et al. [10] and Lin et al. [11], to name a few. In general, these works were mainly focus on the application of ABC in various fields such as image recognition, function optimization, transportation and circuit production. The main issue with standard ABC is the fitness function of the standard model does not comply with hybrid DHNN. The recent work coined by Kasihmuddin et al. [12] has adopted the modified ABC with DHNN in 2SAT logic programming by utilizing the simulated data sets. The results were exceptionally good in terms of reducing the network complexities and to reduce the model in attaining the local minimum solutions. Moreover, there is no recent effort in integrating the ABC as the learning algorithm in optimizing the P2SAT with the DHNN in a single computational network (DHNN-P2SAT).

In this research, the comparison will be made with employing the genetic algorithm (GA) as the benchmark. Genetic algorithm is a variant of evolutionary algorithm, formulated by Goldberg and Kuo [13]. The notable research on GA can be seen in Metawa et al. [14] and Mohammed et al. [15]. The effort of integrating GA into DHNN can be found in the work of Yang et al. [16] and Kasihmuddin et al. [17]. Similarly, there is minimal effort in implementing GA in P2SAT.

Discrete Hopfield neural network (DHNN) is the most celebrated class of artificial neural network (ANN), proposed by Hopfield and Tank [18] by taking into inspiration the framework of our biological brain and memory processing system. Discrete Hopfield neural network (DHNN) is a variant of recurrent neural network without the hidden layer consisting of the bipolar neuron being activated asynchronously in motion [19]. The astounding feature of DHNN is the content addressable memory (CAM) being used in the retrieval phase of the computational process [20]. Several practitioners have incorporated the DHNN into various Satisfiability logical representation and logic programming at hand of in the existing literatures namely, Kasihmuddin et al. [21], Sathasivam [22], Tatem et al. [23] and Mansor et al. [24]. Furthermore, the synaptic weight of most of DHNN models is computed by using method coined by Abdullah [25]. These works have been the motivation of hybridizing the ABC with DHNN in doing P2SAT. The integrated ABC with DHNN can be seen in the work of Kasihmuddin et al. [12], in doing 2SAT logic programming. We believe that the model can be further improved to work with semi simulated data such as in optimizing the P2SAT.

However, there is no attempt to bridge the P2SAT with DHNN as the network and ABC as the learning algorithm. The contributions of this paper are: (1) This research will investigate the impact of the 2SAT logical representation into the PSAT, to form P2SAT. (2) This research will apply ABC as the learning algorithm with DHNN in P2SAT verification (DHNN-P2SATABC). The comparison will be made with the standard GA and DHNN to process the P2SAT (DHNN-P2SATGA). The same number of patterns will be used to ensure the fair comparison.

2. Pattern 2 Satisfiability

P2SAT is a variant of SAT model, inspired by pattern recognition and circuit verification as coined in [19]. The 2SAT logical representation is chosen due to the simplicity of the logical rule. According to Bünning et al. [26], Boolean 2SAT or also known as 2 Conjunctive Normal Form (2CNF) formula is defined as the logical rule with strictly 2 literals per clause. Specifically, the Pattern-SAT, 2SAT clauses will denote the important information or \( I_P \). Illustratively, we can observe the rectangular pattern that embedded with 4 important points \( I_P \) as shown in Fig. 1.

The general formula of \( P_{P2SAT} \) based on \( I_P \) can be elucidated as follows:

\[
P_{P2SAT} = \bigwedge_{i=1}^{4} C_i \quad \text{where} \quad C_i = n \bigvee_{i=1}^{k} (x_i, y_i), k = 2
\]

whereby \( C_i = I_P \). Equation (1) can only comply with 2n cubes and must be represented in an even numbered cube. In Fig. 1, 4 cubes are selected as it satisfied with the condition of P2SAT representation. \( I_P \) is constructed based on the concept of \( C_i \) (clauses) and \( NC \) (number of clauses).

If any of the \( P_{P2SAT} = 1 \cdot I_P \) will be satisfied and activated and vice versa. The satisfied \( I_P \) will determine the characteristic of P2SAT, whether it is coloured or not. Thus, in order to maximize the generation of satisfied \( I_P \), the objective function is given:

\[
\max \left[ f_{P2SAT} \right]
\]

where \( f_{P2SAT} \) denotes the fitness values of the pattern based that can be defined and calculated as follows:

\[
f_{P2SAT} = \sum_{i=1}^{NC} I_P
\]

whereby \( NC \) is the number of 2SAT clause. Moreover, the general the classification \( I_P \) based on activation is defined as follows:

![Fig. 1 - Rectangular Pattern Embedded with \( I_P \)](http://www.fazpublishing.com/cam)
Following that, we can conclude that $C_i = IP_i$ for all clause in $P_{P2SAT}$. The important points of P2SAT will produce $IP_i = 1$ indicating P2SAT clause has consistent interpretation. Therefore, the systematic logical rule in P2SAT can be further undergo learning and retrieval phase in our network.

3. **Discrete Hopfield Neural Network**

According to Joya et al. [27], DHNN adopted a promising feature in terms of content addressable memory and capability to bind with other learning algorithms. DHNN is a type of recurrent ANN, composed of output and input neuron being activated by assigning the bipolar values of the states given by $S_i \in \{-1,1\}$. Pursuing that, the representation for $IP_i$ state activation, $S_i$ during firing in HNN is shown as follows:

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

where $W_{ij}$ is the synaptic weight from unit $j$ to $i$ whereas $\xi$ is the threshold value. In DHNN, the value of $\xi = 0$ is chosen to comply with the work in [24].

In this work, the local field formula of DHNN proposed by [22] is modified accordingly to comply with P2SAT constraint. Hence, the modified local field equation is as follows:

$$h_i(t) = \sum_{j=1, i \neq j} W_{ij}^{(2)} S_j + W_{i1}^{(1)} + \phi(t)$$

where $W_{ij}^{(2)}$ is a synaptic weight from neuron $j$ to neuron $i$ and $W_{i1}^{(1)}$ is the threshold of neuron $i$. The value of local bias, $\phi(t) = 0$ is considered in this work. The general updating rule for the state in each of $IP_i$ can be simplified as:

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

By limiting the synaptic weights to be symmetric and zero diagonal $W_{ij}^{(2)} = W_{ji}^{(2)}$, $W_{i1}^{(2)} = 0$, the deduced final energy of DHNN for the case of $P_{P2SAT}$ is given as:

$$H_{P2SAT} = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1, i \neq j} W_{ij}^{(2)} S_i S_j - \sum_{i=1}^{N} W_{i1}^{(1)} S_i$$

Additionally, in this case, we considered all possible permutations in cyclic order for any $i$ and $j$. The final energy of HNN is always decrease with the dynamics. Hence, the final energy equation is recrafted according to [17]. Apart from being used in attainment the synaptic weight, Equation (8) provides a solid filtering mechanism to discriminate between Global P2SAT and Local P2SAT effectively.

4. **Learning Pattern Satisfiability Via Artificial Bee Colony Algorithm**

In this part, the learning process via ABC will be discussed accordingly. A newly enhanced ABC enthused by Abdullah model [25] in doing P2SAT will be explained. The employment of ABC in DHNN-P2SAT is abridged as DHNN-P2SATABC. The operators and stages involve in DHNN-P2SATABC can be classified into 3 major domains namely, the employed bees, onlooker bees and scout bee [28]. The first two components are obligatory for the “bees” to forage food (consistent interpretation) in DHNN-P2SATABC. If the first two components fail to meet the food criteria, scout bee will reset the whole search space. This strategy will avert the algorithm from having local maxima (non-improving) solution. Inspired by binary ABC proposed by [7], the main stages of DHNN-P2SATABC are as follows:

**Stage 1: Initialization**

Generate and initialize 50 employed bees, 50 onlooker bees and 1 scout bee. Each bee denotes the possible consistent interpretation of $P_{P2SAT}$, where $S_i(t) \in \{1,-1\}$.

**Stage 2: Fitness evaluation**

Calculate the fitness of the employed bees.

$$f(S_i(t)) = \sum_{i=1}^{NC} C_i$$

where

$$C_i = \begin{cases} 1, & \text{true} \\ 0, & \text{false} \end{cases}$$

**Stage 3: Employed bee stage**

The adaptation for employed bee during foraging food source is given as follows:

$$v_{ij} = x_{ij} \lor (\phi_j \Box (x_{ij} \land x_{lj}))$$

where

- $x_{ij}$ is an initial food source
- $x_{lj}$ is an observed food source
- $\phi_j$ parameter where
  $$\phi_j = \begin{cases} 1, & \text{rand}(0,1) < 0.5 \\ -1, & \text{rand}(0,1) \geq 0.5 \end{cases}$$

$\Box$ is an ‘XOR’ operator
$\land$ is an ‘AND’ operator
$\lor$ is an ‘OR’ operator

After the adaption, fitness of each employed bee will be calculated by using Equation (9).

**Stage 4: Onlooker bee stage**

When the employed bees return to their hive, the food information will be transferred via dancing, Onlooker bee will
choose the information based on roulette wheel selection (RWS) \[14\]:

\[
p_i = \frac{f_{iSN}}{\sum_{j=1}^{SN} f_{j}}
\]

where \( \sum_{j=1}^{SN} f_{j} \) is the desired fitness of HNN-P2SAT and SN is denoted by the group size of the bees. The onlooker bees will find the next food source by using Equation (13). The best food generated until the number of trials is equal to the limit.

Stage 5: Scout bee stage
If the solution from the onlooker bee still does not reach the desired fitness, scout bees will abandon the search space. Note that, if the algorithm found a solution with desired fitness, the solution will exit the algorithm easily and print the best solution.

5. Learning Pattern Satisfiability Via Genetic Algorithm
GA is a standard metaheuristic algorithm, utilized as the benchmark to validate the effectiveness of the other learning algorithm. The searching mechanism in GA is inspired by the Darwin theory of natural evolution which has been explained in Metawa et al. \[14\]. The stages involve various operators starting from initialization, fitness evaluation, selection, crossover and the powerful mutation. In this method, the non-satisfied \( IP_i \) will be improved in order to attain \( P_{2SAT} = 1 \). In this research, we applied the GA as a learning algorithm in processing the \( IP_i \) in order to retrieve the complete Global P2SAT at the end of the simulation. The implementation of GA in DHNN can be found in Kashiuddin et al. \[17\], where the modification here is the logical representation and pattern reconstruction problem.

6. Methodology and Implementation
DHNN-P2SAT models will be adopted to activate all possible \( IP_i \) in P2SAT. If the proposed model is able to generate 100% Global P2SAT, the entire pattern is addressed as complete Global P2SAT as shown in Fig. 2. This depicts that all of the \( IP_i \) has successfully activated by our network. On the contrary, the existence of Local P2SAT will produce the incomplete pattern as shown in Fig. 3. Thus, the effectiveness of the learning phase of HNN-P2SATGA and HNN-P2SATABC were vital before storing the correct synaptic weight in DHNN. In other words, the main task of the HNN-P2SAT is to find hybrid “models” that activate all the clauses in P2SAT. If the HNN-P2SAT model is able to activate all the clauses, the pattern will be successfully retrieved.

The implementation of DHNN-P2SATABC and DHNN-P2SATGA can be summarized in the following algorithm:

**Step 1:**
Initialize and randomize the \( IP_i \) in terms of \( P_{2SAT} \) representation.

\[
P_{2SAT} = \begin{cases} 
1, & \text{rand}(0,1) \geq 0.5 \\
-1, & \text{Otherwise}
\end{cases}
\]

**Step 2:**
Adjust the initial corresponding synaptic weight of \( P_{2SAT} \) to be zero. Then, check the inconsistency of the \( IP_i \) respectively.

**Step 4:**
Derive the respective cost function for P2SAT as \( E_{P2SAT} \) as a measurement of \( IP_i \) satisfaction.

**Step 5:**
Check \( IP_i \) satisfaction by using HNN-P2SATGA and HNN-P2SATABC. The \( IP_i \) satisfaction is achieved when \( E_{P2SAT} = 0 \). However, if \( E_{P2SAT} \neq 0 \) the learning algorithm will improve by undergoing another iteration.

**Step 6:**
Calculate synaptic weight that corresponds to \( IP_i \) by comparing the \( E_{P2SAT} \) and global minimum energy function. Then, compute the expected global minimum energy by utilizing Equation (8).

**Step 7:**
Randomize the state entrenched in \( I_R^i \) and employ the Sathasivam relaxation paradigm [22] to DHNN-P2SAT models.

**Step 8:**
Generate the final state of \( I_R^i \) by calculating the corresponding local field via Equation (9).

**Step 9:**
Find the corresponding final energy by using Equation (10). Verify whether the final energy is a global Pattern-SAT or Local Pattern-SAT.

**Step 10:**
By using retrieved \( I_R^i \) state, convert bipolar state to binary representation. Reconstruct the \( I_R^i \) by using Equation (4). Consequently, the red colour pattern will be generated if the \( I_R^i = 1 \).

**Step 11:**
Perform the error evaluation in terms of root mean square error (RMSE), mean absolute percentage error (MAPE), ratio of Global P2SAT and CPU time.

### 7. Experimental Setup

The simulations for DHNN-P2SATGA and DHNN-P2SATABC were implemented on Microsoft Visual Studio for Windows 10. In order to retrieve the complete P2SAT pattern for each execution, the Visual Studio C# package will be utilized. The simulations are carried out by using the same device with similar processing power to avoid any biases and possible memory deteriorations. Thus, Table 1 and Table 2 manifest the entire variables or parameters involved in DHNN-P2SAT models.

The parameters in Table 1 for DHNN-P2SATABC were selected according to the good agreement in parameters applied in the work of Kasihmuddin et al. [12]. In addition, the standard variables and parameters for DHNN-P2SATGA were used in aligned with the work of Kasihmuddin et. al. [17] and Metawa et. al. [14]. The selection of the standard termination criterion for DHNN-P2SAT models was fruitfully being explained well in Sathasivam [22], due to need of filtering mechanism in distinguishing Global and Local P2SAT. The termination criterion used as a discriminating indicator whether the \( E_{P2SAT} \) is Global P2SAT or Local P2SAT.

### 8. Results and Discussion

The main purpose of integrating ABC in DHNN-P2SAT model is to improve the performance of the model in attaining Global P2SAT in different complexities. Previously, the P2SAT verification was conducted by enumerating and error method, that consistently producing Local P2SAT and errors. In addition, the main problem with the standard DHNN-P2SAT models is the capability to process the higher-order P2SAT that resulting in higher CPU time. The higher-order P2SAT might exceed the threshold time for the simulations, resulting in non-robustness in the model. The performance of DHNN-P2SAT will be compared in terms of several performance metrics such as Root mean square error (RMSE), mean absolute percentage error (MAPE), ratio of Global P2SAT and CPU Time.

As compared to the pattern satisfiability analysis coined by Mansor et. al. [1], the simulations were able to execute more complicated P2SAT with a higher number of \( I_R^i \) within 24 hours. This demonstrates the necessity of metaheuristic learning algorithm to enhance the performance.

Table 3 until Table 6 manifest the value of RMSE, MAPE, Global P2SAT ratio and CPU time obtained after the simulation employed by DHNN-P2SAT models respectively. On the similar line, the simulation is conducted by taking \( 30 \leq I_R^i \leq 300 \) in order to assess the capability of both models. Hence, the apparent differences between DHNN-P2SATABC and DHNN-P2SATGA can be observed clearly during the learning phase of the pattern reconstruction. Thus, an effective

### Table 3 – RMSE Evaluations for DHNN-P2SAT models

<table>
<thead>
<tr>
<th>Number of ( I_R^i )</th>
<th>DHNN-P2SATGA</th>
<th>DHNN-P2SATABC</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>12.4511</td>
<td>1.1008</td>
</tr>
<tr>
<td>60</td>
<td>21.2568</td>
<td>4.3560</td>
</tr>
<tr>
<td>90</td>
<td>43.8233</td>
<td>7.5000</td>
</tr>
<tr>
<td>120</td>
<td>72.4877</td>
<td>11.7890</td>
</tr>
<tr>
<td>150</td>
<td>86.0115</td>
<td>14.3434</td>
</tr>
<tr>
<td>180</td>
<td>109.9294</td>
<td>19.9867</td>
</tr>
<tr>
<td>210</td>
<td>123.3500</td>
<td>21.5277</td>
</tr>
<tr>
<td>240</td>
<td>134.7222</td>
<td>23.0000</td>
</tr>
<tr>
<td>270</td>
<td>166.0209</td>
<td>24.9325</td>
</tr>
<tr>
<td>300</td>
<td>181.2504</td>
<td>26.4530</td>
</tr>
</tbody>
</table>

### Table 4 – MAPE Evaluations for DHNN-P2SAT models

<table>
<thead>
<tr>
<th>Number of ( I_R^i )</th>
<th>DHNN-P2SATGA</th>
<th>DHNN-P2SATABC</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>40.1259</td>
<td>1.0025</td>
</tr>
<tr>
<td>60</td>
<td>44.4003</td>
<td>1.4433</td>
</tr>
<tr>
<td>90</td>
<td>47.8840</td>
<td>1.7260</td>
</tr>
<tr>
<td>120</td>
<td>49.2402</td>
<td>1.8556</td>
</tr>
<tr>
<td>150</td>
<td>49.7651</td>
<td>2.1320</td>
</tr>
<tr>
<td>180</td>
<td>50.0003</td>
<td>2.4224</td>
</tr>
<tr>
<td>210</td>
<td>50.6772</td>
<td>2.8600</td>
</tr>
<tr>
<td>240</td>
<td>50.9845</td>
<td>3.3567</td>
</tr>
<tr>
<td>270</td>
<td>51.2240</td>
<td>3.9843</td>
</tr>
<tr>
<td>300</td>
<td>51.5000</td>
<td>4.0003</td>
</tr>
</tbody>
</table>
Table 5 – Global PSAT Evaluation for DHNN-P2SAT models

<table>
<thead>
<tr>
<th>Number of $IP_i$</th>
<th>DHNN-P2SATAGA</th>
<th>DHNN-P2SATABC</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>0.9995</td>
<td>1.0000</td>
</tr>
<tr>
<td>60</td>
<td>0.9993</td>
<td>1.0000</td>
</tr>
<tr>
<td>90</td>
<td>0.9998</td>
<td>1.0000</td>
</tr>
<tr>
<td>120</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>150</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>180</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>210</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>240</td>
<td>0.9994</td>
<td>1.0000</td>
</tr>
<tr>
<td>270</td>
<td>0.9999</td>
<td>1.0000</td>
</tr>
<tr>
<td>300</td>
<td>0.9997</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

Table 6 – CPU Time Evaluation for DHNN-P2SAT models

<table>
<thead>
<tr>
<th>Number of $IP_i$</th>
<th>DHNN-P2SATAGA</th>
<th>DHNN-P2SATABC</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>60</td>
<td>36</td>
<td>24</td>
</tr>
<tr>
<td>90</td>
<td>48</td>
<td>32</td>
</tr>
<tr>
<td>120</td>
<td>54</td>
<td>43</td>
</tr>
<tr>
<td>150</td>
<td>92</td>
<td>58</td>
</tr>
<tr>
<td>180</td>
<td>108</td>
<td>100</td>
</tr>
<tr>
<td>210</td>
<td>153</td>
<td>128</td>
</tr>
<tr>
<td>240</td>
<td>226</td>
<td>187</td>
</tr>
<tr>
<td>270</td>
<td>258</td>
<td>215</td>
</tr>
<tr>
<td>300</td>
<td>440</td>
<td>260</td>
</tr>
</tbody>
</table>

The trend of CPU time might not be an effective way to quantify the quality of the global P2SAT by devouring lesser iterations. This statement demonstrates the Global P2SAT ratio recorded by DHNN-P2SATAGA. It was obvious that the trend of the Global P2SAT ratio exhibited by the proposed model was consistently equal to 1. This shows that 100% of the patterns generated by the model were Global P2SAT. This explains the superior performance of ABC to enhance the P2SAT during the learning and retrieval phase. On the contrary, the GA can be further improved in terms of the parameter selection in order to comply with the capability of ABC in DHNN-P2SAT models.

The CPU time taken by DHNN-P2SATABC was consistently lower than DHNN-P2SATAGA for the entire round of the simulations. This results highlight the role of an effective learning algorithm such as ABC algorithm in order to accelerate the execution for P2SAT verification. As shown in Table 6, the proposed model reduces the computational time by approximately 59% as compared to DHNN-P2SATAGA when $IP_i$. The main reason for the major reduction in CPU time is the capability of Onlooker bee operator in ABC to generate the global P2SAT by devouring lesser iterations. This statement outlines the capability of swarm-based metaheuristics in solving various problem, not only P2SAT but can be further utilized in pattern recognition. The CPU time might not be an effective way to quantify the quality of the global P2SAT produced at the end of the simulation.

The general conclusions and main points that can be drawn from the analysis of results in Table 2 until Table 6 are:

i. DHNN-P2SATABC records the promising results in term of RMSE, MAPE, Global P2SAT ratio and CPU time for the different number of $IP_i$. However, DHNN-P2SATAGA produces suboptimal patterns which corresponds $P_{2SAT} = 0$.

ii. The capability of DHNN-P2SATABC of generating good results in Table 2 until Table 6 due to the effectiveness of the model in checking the $IP_i$ during learning in lesser iterations. Therefore, the systematic information exchange between employed and onlooker bee will reduce the possibility of undergoing the scout bee stage.

iii. The higher number of $IP_i$ constitutes $P_{2SAT} = 0$ will generate the higher iterations and errors during learning phase. Thus, the selection of ABC has proven to generate more $P_{2SAT} = 1$ for $30 \leq IP_i \leq 300$.

iv. Worth mentioning that, ABC is expecting to outperform the method employed by Sathasivam [X]. This is due to the probability to attain $P_{2SAT} = 1$ is approximately zero when $IP_i > 300$.

This raises a few questions regarding the results obtained during the simulation, indicating the improvement for the proposed model when dealing with more complex P2SAT in the future. The limitation of the proposed model in P2SAT is possibly the lack of variation and diversification in the quality of Global P2SAT obtained after each of execution with different $IP_i$. Since the final pattern in this research only involved RBY colour (subtractive colour model) and alphabetical pattern, the researcher has opportunity to apply in a more complicated pattern or image in the future. In addition, a thorough investigation in tuning the important parameter of DHNN-P2SATABC is needed in order to determine the best parameter estimation for different type of pattern problem. Hence, the other variant of logical representation can be entrenched into pattern Satisfiability (PSAT) such as Maximum $k$-Satisfiability (MAX$k$SAT) [19], Random 2 Satisfiability (RA$2$SAT) [28] and weighted partial minimum 2 Satisfiability (WPM$2$SAT) [29] with in-depth investigation to further authenticated the effectiveness of PSAT in DHNN.

9. Conclusion

In this paper, a new approach for doing higher-order Pattern 2 Satisfiability (P2SAT) with ABC as the learning method and DHNN as the main network was proposed. Experimental results demonstrate that DHNN-P2SATABC is more competitive as compared with the standard method, HNN-P2SATAGA in attaining the global P2SAT pattern especially when dealing with the higher-order P2SAT. Thus, ABC has proven to facilitate our network in generating more global P2SAT and to minimize the existence of local P2SAT. It was found that the pattern constructed by our proposed model produced less error and optimal values of Global P2SAT. Another lead of future works would be to apply the effective learning method via ABC for more complex P2SAT with a different variant of neural networks, such as Deep Convolutional neural network, Wavelet neural network and Fuzzy neural network.

Acknowledgement
This research was funded by Short Term Grant (STG), Universiti Sains Malaysia, grant number 304/PJJAUAH/6315263.

References


