Pertanika J. Sci. & Technol. 25 (4): 1173 - 1188 (2017)

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

Artificial Immune System Paradigm in the Hopfield Network for 3-Satisfiability Problem

Mansor, M. A.*, Kasihmuddin, M. S. M. and Sathasivam, S.

School of Mathematical Sciences, Universiti Sains Malaysia, 11800 Minden, Pulau Pinang, Malaysia

ABSTRACT

The artificial immune system (AIS) algorithm is a heuristic technique inspired by the biological immune system. The biological immune system has been proven to be a robust system that defends our body from any pathogen attacks. This paper presents a hybrid paradigm by implementing the Hopfield neural network integrated with enhanced AIS for solving a 3-Satisfiability (3-SAT) problem. Fundamentally, a 3-Satisfiability problem is used as an ideal optimisation problem by neural network practitioners in their research. The core impetus of this study was to compare the performance of artificial immune system (AIS) algorithm and brute-force search (BFS) algorithm in doing 3-SAT logic programming. Microsoft Visual C++ 2013 was used as a dynamic platform for training, simulating and testing of the network. We restricted our analysis to 3-Satisfiability (3-SAT) clauses. The performances of both paradigms were analysed according to the following measures, namely, global minima ratio, global Hamming distance, fitness landscape value and computational time. The experimental results successfully depicted the robustness of the AIS compared to the BFS algorithm. The work presented here has profound implications for future studies of AIS to solve more complicated NP problems.

Keywords: Artificial immune system algorithm, brute-force search algorithm, Hopfield network, 3-Satisfiability, logic programming

ARTICLE INFO

Article history:
Received: 25 May 2016
Accepted: 23 February 2017

E-mail addresses:
asyrafalvez@live.com (Mansor, M. A.),
iwanmaidin@gmail.com (Kasihmuddin, M. S. M.),
saratha@usm.my (Sathasivam, S.)
*Corresponding Author

ISSN: 0128-7680

INTRODUCTION

Hybrid computational models in artificial intelligence have been mushrooming and producing a prolific amount of research. In this paper, we proposed a hybrid computational model by implementing an artificial immune system (AIS) algorithm incorporated with a Hopfield neural network to do a 3-SAT logic programming. Technically, the combination of Hopfield neural network, constrained satisfiability problem and searching techniques (metaheuristics) in logic programming as a hybrid computational network is still novel in artificial intelligence.

Firstly, the artificial immune system algorithm (AIS) is a vibrant metaheuristic paradigm, enthused by the complex biological immune system (Dasgupta et al., 2003). Furthermore, AIS can behave as an alternative machine learning network and feasibly be implemented to resolve zillions of constraint optimisation problems (Layeb, 2012). The core advances within AIS have been dedicated to three important immunological principles namely, the immune clonal selection, immune networks and negative selection (Timmis & Neal, 2001). As a matter of fact, most AIS practitioners have focussed on the learning and memory mechanisms of the immune system in order to have it resemble the human immune system. A prolific volume of works on artificial immune systems from the breakthrough research by Farmer et al. (1986) has transformed the AIS into a vital metaheuristic paradigm to solve numerous problems.

The neural network is considered as one of the most celebrated fields in artificial intelligence (AI) and mathematical computational studies (Rojas, 1999). The framework of artificial neural network is inspired by the biological nervous system to model the computations engaged in by the human brain (Zinovik et al., 2008). Strictly speaking, various types of neural networks have been sprouted such as the Hopfield network presented by Hopfield and Tank (1985). The Hopfield neural network is highlighted as a simple recurrent network equipped with an efficient associative memory. Moreover, the network can store numerous memories analogously to the human brain (Sathasivam, 2010). Additionally, it is a division of the artificial neural network that practically can be implemented to solve various mathematical problems such as the combinatorial optimisation problem, pattern recognition and hard satisfiability problem (Haykin, 1992).

Logic programming is a promising computational field that can be applied in solving numerous optimisation problem. Basically, logic programming can be delineated as an optimisation problem according to the constraint satisfiability outlook (Kowalski, 1979). Hence, logic programming requires specific clauses; the 3-SAT clause as a combinatorial optimisation problem. Generally, 3-SAT involves a massive search space, since it was depicted as a NP-hard problem (Tobias & Walter, 2004). In this paper, we used the 3-SAT clauses as the problem in logic programming. In the same way, the 3-SAT problem is applied and integrated in an artificial neural network to search for optimised global solutions (Aiman & Asrar, 2015). Therefore, the conventional model developed by Abdullah (1993) was the breakthrough in logic programming of neural networks.

Neuro-searching paradigms are getting more attention by researchers eager to solve computational problems (Luke, 2013). The conventional method, the exhaustive search (ES) and heuristics method, can be implemented as the searching technique in any constraint optimisation problem (Matsuda, 1998). Hence, we proposed the artificial immune system (AIS) as the neuro-searching technique (metaheuristic) in this study. In this research, we proposed an enhanced metaheuristic method, the artificial immune system (AIS) algorithm incorporated with the Hopfield network to hunt for the satisfied assignments within the stipulated time. The main contribution was the implementation of the artificial immune system (AIS) algorithm as a searching technique integrated with the Hopfield neural network in doing 3-SAT logic programming. The robust searching paradigm assisted the hybrid model to achieve greater convergence in solutions, better stability and faster searching time. In addition, we introduced the conventional searching method, brute-force search (BFS) algorithm, incorporated with the Hopfield neural network as the searching tool.

The overall structure of this paper had been structured systematically as follows. In Section 2, we discuss the fundamental notions of the Boolean Satisfiability Problem and the specific 3-Satisfiability concept. In Section 3, we emphasise the Hopfield neural network, content addressable memory (CAM) and logic programming in 3-SAT. Section 4 presents the neuro-searching paradigms implemented in this study, the artificial immune system (AIS) and brute-force search (BFS) algorithm. In Section 5, the implementation of the hybrid paradigms are discussed briefly. Section 6 presents the complete experimental results and expositions. Finally, Section 7 presents the conclusion and some recommendations for future research.

THE SATISFIABILITY (SAT) PROBLEM

Strictly speaking, the satisfiability problem (SAT) is one of the most celebrated topics in prepositional calculus and computer science. The SAT problem can be demarcated as the process of finding an ideal assignment based on Boolean values in order to ensure the formula is satisfied (Vilhelm et al., 2005). Hence, the core impetus of a satisfiability study is to decide the existence of an ideal assignment of truth values (assignments of 0 or 1 to each of the variables) to variables that produce any satisfied conjunctive normal form (CNF) formula (Gu, 1999). Technically, an immense amount of NP problems can be transformed in terms of SAT. Given an insight defining whether a SAT problem assignment is satisfiable or not, one can discover a satisfying assignment in time and linear based on the number of variables or literals (Ullman, 1975). Therefore, there is a possibility of transforming an NP problem in SAT in polynomial time. For instance, when we have a constraint problem of size n, there are 2^n possible assignments and also l literals to the set for each assignment where such technique requires $O(1.2^n)$ operations (Sathasivam, 2010).

3-Satisfiability (3-SAT)

In this section, we highlight 3-Satsifiability (3-SAT), which is a paradigmatic NP-complete problem. Essentially, the 3-Satisfiability (3-SAT) problem can be described as a mapping conundrum from truth values based on logic programming in 3-SAT. Technically, 3-SAT can be delineated as a conjunctive normal form formula with a collection of clauses, each comprising exactly and strictly three literals per clause (Vilhelm et al., 2005). Therefore, the 3-SAT paradigm can allow binary values of each variable, which are 1 or -1. In addition, the 3-SAT problem can be clinched as a non-deterministic problem (Tobias & Walter, 2004).

The four fundamental aspects of the 3-SAT problem in the conjunctive normal form (CNF) can be summarised as follows:

1. The SAT formula comprises an array of n variables, $z_1, z_2, ..., z_n$ inside each clause. For a 3-SAT problem, we strictly limited $n = 3$.
2. A set of m clauses in a Boolean formula. $\exists m : F = c_1 \wedge c_2 \wedge ... \wedge c_m$

3. A set of $l_{k,i}$ literals. In 3-SAT, we considered three literals in each clause. Each clause, c_k, consisted of only literals combined by the logic operator OR.

   $\forall 1 \leq k \leq m : c_k = (l_{k,1} \vee l_{k,2} \vee l_{k,3})$

4. The literals can be the variable itself or the negation of the variable.

   $\forall 1 \leq k \leq m, 1 \leq i \leq 3 : l_{\mathrm{k,i}} = z_p$ or $l_{\mathrm{k,i}} = \neg z_p$ for $1 \leq p \leq n$

In this paper, a randomised 3-SAT formula, which consisted of strictly three clauses and three literals, is emphasised. For example:

$$P = \left(D \vee E \vee \bar{F}\right) \wedge \left(\bar{D} \vee \bar{E} \vee F\right) \wedge \left(\bar{D} \vee E \vee G\right) \tag{1}$$

As an illustration, we represented the 3-SAT formula in CNF form as P in equation (1). Generally, the formula can be formed in numerous combinations (randomised) as the number of atoms can be different, excluding the literals that were rigorously equal to 3 for each clause. Correspondingly, the greater number of literals per clause will maximise the probabilities for a clause to be satisfied (Kowalski, 1979).

Neuro-Searching Paradigm

Brute-force search algorithm. Brute-force search (BFS) algorithm can be defined as a local search technique for an element with a specific property among combinatorial aspects including permutations, combinations, logics, satisfiability or subsets of a set (Mark & Lee, 1992). Additionally, the BFS algorithm will brutally search for total potential clause, even if the search dimension gets bigger and more complex (Rojas, 1999). Technically, the brute-force search algorithm is the simplest algorithm for checking the logic satisfaction problem. Even though BFS is theoretically easy to implement and frequently effective, it is occasionally considered not robust (Nievergelt, 2000). Despite the disadvantages, an exhaustive search can be guaranteed to converge towards the solution (satisfied clause) for the entire search space. Consequently, an exhaustive search consumes more computation time in searching for the satisfied interpretation completely (Zinovik et al., 2008). On the other hand, the entire bit strings (interpretation) will be collapsed when any one of the clause is not satisfied.

In our exploration, we pinpointed the complexity of the hybrid network when we ventured to work with more neurons. The CPU time was slowed down when we increased the complexity of the hybrid network. Given any 3-SAT problem, there are theoretically satisfying assignments (Gu, 1999). To sum up, the computation complexity is represented as . For the BFS algorithm, the satisfied assignment is gained after performing a brutal 'trial and error' procedure. Henceforth, the correct assignment will be stored in the Hopfield's artificial brain in the form of content addressable memory (CAM). The brute-force search algorithm performance has been explored by Aiman and Asrar (2015), Zinovik et al. (2008) and Nievergelt (2000). In this paper, we implemented the BFS algorithm with the Hopfield neural network as a hybrid network based on logic programming to solve 3-SAT problems (3SAT-BFS).

Artificial immune system algorithm. The Artificial Immune System (AIS) algorithm has emerged as a brand new metaheuristic technique based on the human immune system. The artificial immune system was popularised by Farmer et al. (1986), who modelled Jerne's Immune network theory. On top of that, the artificial immune system (AIS) algorithm can be illustrated as a distributed network able to do parallel processing (Afshinmanesh et al., 2005). The core developments of the artificial immune system (AIS) revolved around three fundamental immunological concepts, namely, the immune clonal selection, negative selection technique and immune network (Aickelin, 2008).

Researchers have investigated the learning and memory capability of the clonal selection and immune network theory. The AIS offers properties that are suitable for a computational model, such as adaptation, recognition, learning, robustness, memory and scalability (De Castro & Timmis, 2002).

The biological immune system can be classified into two key defence methods (De Castro & Von Zuben, 2002):

a) Innate immune system, which represents the non-specific defence mechanism and biological immune defences present since birth. Moreover, the innate immune system comprises the complete chemical properties contained in the antigen.
b) Adaptive immune system, which depicts the entire defences learnt over time. Additionally, adaptive immunity comprises a 'memory' that creates upcoming reactions against a particular antigen efficiently.

Furthermore, the complex interactions between entities within each level will ensure the immune system shields the body after any harmful entity and exogenous agent, known as an antigen, has attacked it. A particular form of cell, identified as the B-cell, leads in the destruction of the antigen. Hence, the B-cell produces antibodies that bind with the antigens and mark them for damage (Dasgupta et al., 2003). The strength of the antibody or antigen binding is called antigenic affinity (Layeb, 2012). Robust features of the immune system have boosted its adaptation to information technology for solving numerous problems.

Therefore, the proposed hybrid technique is a novel technology as most researchers only focus on the standalone Hopfield neural network or metaheuristic to solve 3-satisfiability problems. The brute force search (BFS) is a state-of-the-art technique, extensively applied to solve 3-satisfiability problems. Hence, the artificial immune system (AIS) algorithm needs to be compared with the brute force search (BFS) in order to highlight its computational capability. In this paper, we focussed on the clonal selection that was implemented in our binary AIS.

Clonal selection. The remarkable feature in our biological immune system is the capability to build antibodies to combat new antigens or pathogens (Dasgupta et al., 2011). Hence, the immune clonal selection process depicts the fundamental structures of an immune response towards an antigenic stimulus. It advances the idea that only the antibodies can identify the antigen proliferate, and therefore, they are the cells nominated to do the job (Timmis et al., 2008). Specifically, B-cells will produce antibodies if any incoming antigen is discovered. Then, the particular B-cells distinguish the antigen proliferate via the cloning process. Significantly, the main event during clonal mutation is somatic hypermutation, whereby genetic maturation and variation are improved (Layeb, 2012). The B-cells with higher affinity will be differentiated into plasma and memory cells, while the worst one will be destroyed.

Binary artificial immune algorithm. We proposed a binary artificial immune system based on the immune clonal selection perspective. Technically, the binary artificial immune system was implemented by several researchers for binary optimisation and pattern recognition. Previous works on binary artificial immune system include Tang et al. (1997), who proposed that the binary AIS be incorporated with the immune response network theory and Layeb (2012), who introduced the affinity-based interaction for the artificial immune system (AIS) algorithm integrated with the TABU search technique.

In our paper, we develop a hybrid paradigm by implementing the Hopfield network and binary AIS to do a 3-SAT logic programming (HNN-3SATAIS). In our exploration of binary AIS, the binary strings or Boolean interpretations were illustrated as the B-cells. Firstly, we generated and initialised 100 B-cells that represented the initial population size. Generally, any massive and diverse population represented a massive space search of solutions that can lead to global solutions. Moreover, a smaller population size can contribute to local minima solutions (Layeb, 2012).

For instance, if an antigen or pathogen attacks the organism, the antibodies (B-cells) that recognise these antigens survive. Secondly, for every iteration, the affinity of every B-cell is computed. The affinity measure was the total amount of satisfied clauses in the 3-SAT formula. After that, the best five B-cells were selected. By implementing the roulette wheel mechanism, the selected B-cells were allowed to be cloned and duplicated. Therefore, the newly produced B-cell population comprised 200 cloned B-cells. Then, we normalised the B-cells. Thus, the antibodies existing in memory response achieved a higher average affinity than those of the initial primary response (Coello & Cortes, 2005). It is called the maturation of the immune response process.

The mutation process in AIS is basically similar to the one in genetic algorithm. The process is improved by the ‘somatic’ principle, whereby the nearer the match, the more disruptive the mutation (Timmis & Neal, 2001). In order to obtain a satisfactory interpretation, somatic hypermutation might be very useful. The flipping process will improve the B-cells (interpretation) to achieve the best affinity value. The best B-cells will be selected as the candidate cells and stored in the memory cell to be retrieved to combat pathogenic attacks. The aim is to preserve the diversity between antibodies that are composed of the memory set. In our context, any satisfied interpretation will be stored in

CAM to be recalled by the network. Thus, it will converge to the global solution. The algorithm of our proposed binary AIS can be simplified as follows:

Step 1

Generate 100 random B_{cell} (random assignments)

Step 2

Check the affinity for each B_{cell}

$$affinity = c_1(x) + c_2(x) + c_3(x)..... + c_{total\,NC}(x)$$

Step 3

Take the best five B_{cell} to be cloned.

$$\begin{pmatrix}\text{The number of}\\ \text{clone allowed}\end{pmatrix} = \frac{aff_i}{\sum affinity} \times \beta$$

where β is the number of population clone that we want to produce. (Set to 200)

Step 4

Normalise the 200 B_{cell} clones: $aff\ N_i = \dfrac{aff_i - \min aff}{\max aff - \min af}$

Step 5

Calculate the number of mutations for each B_{cell} clone

$$\begin{pmatrix}\text{Number of}\\ \text{Mutation (Nb)}\end{pmatrix} = \left(\frac{1}{\text{Number of variable}}\right)(aff\ N_i) + (1 - aff\ N_i)(0.01)$$

Step 6

Based on Nb, mutate the best and the worst B_{cell} based on the flipping of the 1 and -1.

Step 7

Check the affinity of the B_{cell}.

if B_{cell} has a local maxima,

store in memory cell

otherwise, B_{cell} is the best solution

Neuro-Logic in the Hopfield Neural Network

Hopfield neural network. Inaugurated by John Hopfield in 1982 (Hopfield & Tank, 1985), the Hopfield model is widely used to elucidate various optimisation problems. Hence, the model is constructed by connecting a large number of simple processing interconnected units called neurons. Strictly speaking, the interconnected units in Hopfield neural networks are known as the binary threshold unit (Haykin, 1992), which consider the binary values, 1 and -1. After that, the state of the output is maintained until the artificial neuron is updated. In general, the network is usually designed to limit the possible value of a_i (Aiyer et al., 1990). Fundamentally, entails the following operations:

$$a_i = \begin{cases} 1 & if \sum_j w_{ij} S_j > \xi_i \\ -1 & otherwise \end{cases} \tag{2}$$

whereby w_{ij} denotes the weight from unit j to i. S_j refers to the state of unit j and ξ_i denotes the threshold of unit i. The connection in the Hopfield net normally has no connection with itself, and $w_{ij} = 0$ and connections are symmetric or bidirectional $w_{ij} = w_{ji}$ (Sathasivam et al., 2013). In this paper, the network comprises N renowned neurons, where each is defined by the well-known Ising model of a magnetism spin variable. In this model, the neuron permitted only a bipolar state $S_i \in \{1, -1\}$. Traditionally, the update of neurons is based on $S_i \rightarrow \text{sgn}(h_i)$, where h_i is the local field between the neurons. The computational model will explicitly generalise to a higher order connection. Thus, the local field can be computed using:

$$h_i = \sum_j w_{ij}^{(2)} S_j + w_i^{(1)} \tag{3}$$

Basically, the weight or connection strength in the Hopfield neural network is always symmetrical. In a higher order connection, the underlying neuron update is retained:

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

A point to ponder is the value of a neuron change asynchronously in order to minimise the energy and lastly, converge it to equilibrium form (Matsuda, 1998). Cost function is associated with energy function for minimising the inconsistencies of the 3-SAT constraint. This vital property guarantees that the energy will decrease monotonically while following the activation system. The energy function for the Hopfield neural network is denoted in equation (5).

$$E = -\frac{1}{3}\sum_i\sum_j\sum_k w_{ijk}^{(3)} S_i S_j S_k - \frac{1}{2}\sum_i\sum_j w_{ij}^{(2)} S_i S_j - \sum_i w_i^{(1)} S_j \tag{5}$$

Specifically, Hopfield's energy function is important since it will determine the degree of convergence of the network (Ionescu et al., 2010). On top of that, the energy minimisation features make the Hopfield model harmonious with other optimisation algorithms.

Content addressable memory (CAM). The Hopfield network is a contender for information processing systems due to its dynamic properties that unveil stable states that work as a basin of attraction towards which adjacent states can progress in time (Michel & Farrell, 1990). The Hopfield model has been shown to implement content addressable memory (CAM) effectively (Holland, 1975). Theoretically, content addressable memory can be demarcated as particular memories that contain information that can be retrieved from the given address of the memory location where the data is stored (Ionescu et al., 2010). Effective CAM can store an enormous library of memory in form of patterns and is able to recall particular patterns correctly when the network gets 'excited'. In this paper, we utilise the features of CAM to store the satisfied assignments that corresponded to the 3-SAT logical clause. The consistency of 3-SAT assignments (graded storage) was retrieved with the aid of CAM during the training process. With the projection storage rule, CAM was incorporated with search algorithms, namely the brute-force search (BFS) algorithm and the Artificial Immune System (AIS) algorithm to do a 3-SAT in the Hopfield network.

Logic programming in Hopfield network. Constraint satisfaction has been a subject of research in the field of artificial intelligence for many years. Logic programming is one out of many types of constraint optimisation combinatorial problems (Sathasivam et al., 2013). The recent work on logic programming in the Hopfield network was coined by Wan Abdullah, who applied the Hopfield neural network with Horn satisfiability clauses (Abdullah, 1993). Moreover, the significant work by Pinkas and Dechter (1995), which is about energy minimisation by integrating logic programming and the Hopfield neural network. Weight or synaptic strength determination based on Sathasivam's method for Horn logic programming was proven to be effective in reducing the local minima energy and achieving global convergence (Sathasivam & Wan Abdulllah, 2008). In this paradigm, 3-SAT logic was considered a constrained model and the Hopfield network was exploited to minimise constraint inconsistencies. Hence, we implemented logic programming in Hopfield for the 3-SAT clause with the neuro-search algorithms such as the brute-force search and the artificial immune system.

Implementation of 3SAT-AIS logic programming in the Hopfield neural network

i. Firstly, convert all the 3-SAT clauses into prepositional Boolean algebra with correct operators.

ii. Classify the neuron to ground neuron, respectively. Initialise the entire weights to zero.

iii. Derive and form a cost function that is associated with the negation of all 3-SAT clauses, for instance, $X = \frac{1}{2}\left(1+S_X\right)$ and $\overline{X} = \frac{1}{2}\left(1-S_X\right)$. $S_X = 1$ (True) and $S_X = -1$ (False). Multiplication is denoted as conjunction and addition symbolises the disjunction of clauses.

iv. Compare cost function for 3-SAT with energy, E, in order to obtain the values of the connection strengths or weights (Sathasivam & Wan Abdullah, 2008).

v. mplement the AIS algorithm to verify the 3-SAT clause satisfaction. The satisfied interpretations are graded in Hopfield's brain as content addressable memory storage.

vi. Next, randomise the states of the corresponding neurons. Calculate the corresponding local field $h_i\,(t)$. The hybrid network experiences a sequences of energy relaxation processes (Sathasivam, 2010) based on the following formula:

$$\frac{dh_i}{dt} = R\frac{dh_i}{dt} \tag{6}$$

where R is the rate of relaxation. Given that the final state is steady for five runs, we considered it the final state.

vii. Compute the corresponding final minimum energy, E, for the final state using the Lypunov equation. The process of authentication of the final energy will classify whether it is global minima or local minima. After that, compute the global Hamming distance for each. Record the computation time.

viii. Calculate the fitness landscape measure of the energy landscape based on the Kauffman model (Imada & Araki, 1997):

$$f = \frac{1}{t_0 p}\sum_{t=1}^{t_0}\sum_{v=1}^{p} m^v(t) \quad \text{whereby, } m^v(t) = \frac{1}{N}\sum_{i=1}^{N}\xi_i^v S_i^v(t) \tag{7}$$

THEORY IMPLEMENTATION

Firstly, we generated a randomised 3-SAT formula with three clauses. Secondly, we initialised the early states for the 3-SAT clauses in the neurons. Then, the hybrid model was evolved swiftly until the last state was reached. When the final state was achieved, Equation (4) was used to update the neuron state. The network relaxation process took place and can be computed using Equation (3). After that, the stability of the final state was verified. The stable state was considered when the state obtained was steady for five runs.

Pinkas and Dechter (1995) emphasised that allowing the ANN to evolve would contribute to a stable state, where the energy function would be obtained in optimum and equilibrium state. Consequently, the corresponding final energy for the stable state was calculated. The solution is considered a global minima solution if the difference between the final energy and the global minimum energy is within the termination criteria. Furthermore, the algorithms were repeated 100 times with 100 neuron combinations per simulation. The termination criteria for the final energy was fixed as 0.001. Sathasivam et al. (2013) emphasised that 0.001 was chosen because it offered a better performance to reduce statistical errors. We compared the following measures: global minima ratio, global Hamming distance, fitness landscape value and computation time for the brute-force search (3SAT-BFS) and the Artificial Immune System (3SAT-AIS).

RESULTS AND DISCUSSION

Global Minima Ratio and Global Hamming Distance

Global minima ratio is delineated as the ratio between the global solutions divided by the total number of iterations (Sathasivam, 2010). Since each simulation produced 10,000 alternate solutions, we computed the ratio of global minima to check the performance of each algorithm. In our context, the global Hamming distance was equivalent to the distance of the bits between the training state and global state (retrieved state) of the neurons during the energy relaxation procedure.

Table 1 describes the performance of 3SAT-BFS and 3SAT-AIS according to the global minima ratio and global Hamming distance. According to Table 1, the global minima ratio for 3SAT-AIS was close to 1, compared to the traditional 3SAT-BFS. Almost all solutions produced by 3SAT-AIS were global solutions. B-cells with high and improving affinity (fitness) in AIS were able to search the solution optimally compared to the traditional BFS. The complexity of the searching technique in the Hopfield network via AIS was reduced dramatically. Hence, more solutions had achieved the global minima compared to the local minima. The chances for AIS algorithm to converge to global minima were higher compared to BFS. As the number of neurons increased, the complexity of the network increased, since the size of the constraint enlarged indefinitely. In this case, the AIS algorithm was able to sort the possible candidate solution (B-cells) effectively (De Castro & Von Zuben, 2002) and could cope with more constraints compared to the BFS. The problem with 3SAT-BFS was the nature of the brute-force search that deployed an intensive training process in hunting the correct neuron states. Therefore, the updating rule for 3SAT-BFS generated additional abrupt energy surfaces and more solutions obtained stuck at the local minima. Based on the results, 3SAT-BFS was not able to cope with the increasing amount of constraints and did not produce a promising global minima ratio as the number of neurons increased.

Table 1
Global Minima Ratio and Global Hamming Distance for 3SAT-BFS and 3SAT-AIS

Number of Neurons (NN)	Global Minima Ratio		Global Hamming Distance	
	3SAT-BFS	3SAT-AIS	3SAT-BFS	3SAT-AIS
10	0.9941	1.0000	0.00945	0.00122
20	0.9902	0.9994	0.01920	0.00453
30	0.9816	0.9988	0.02451	0.00865
40	0.9744	0.9936	0.02876	0.01394
50	0.9628	0.9914	0.03554	0.01985
60	0.9550	0.9885	0.05002	0.02644
70	-	0.9852	-	0.02806
80	-	0.9737	-	0.03441
90	-	0.9680	-	0.04682
100	-	0.9625	-	0.05296

In comparison, 3SAT-AIS constantly achieved better results than 3SAT-BFS if we consider the global Hamming distance measure. Hence, given that the global Hamming distance was close to zero, the distance between the stable states and global states was almost zero. The global Hamming distance depicted the precision of the bit pattern compared to the expected bit output. The selection of B-cells based on affinity (fitness) helped the network to reach the correct solution effectively. In addition, the efficiency of the AIS algorithm in choosing candidate solutions (B-cells) reduced the complexity of the network. This provided an extra period for the whole network to relax via Equation (6). On the contrary, the brute-force search algorithm highlighted the trial-and-error procedure during checking of the clause satisfaction procedure. The 3SAT-BFS required time to arrive at solutions and retrieve the wrong bit pattern due to lack of relaxation time. The retrieved 3-SAT pattern in the 3SAT-AIS had better accuracy compared to the 3SAT-BFS (Sathasivam, 2010).

The proposed model, 3SAT-AIS, was able to withstand up to 100 neurons. The capability to sustain a massive number of neurons was due to the interesting feature of AIS algorithms that can avoid non-improving B-cells (local maxima assignments) during searching. As the number of neurons increased, 3-SAT constraints increased dramatically. Based on the results, the 3SAT-BFS was not able to cope with larger constraints and did not produce any results when the number of neurons exceeded 60. On the other hand, B-cells in AIS were capable of adapting to higher constrained problems due to somatic hypermutation, which always improves the affinity (fitness) of B-cells (Aickelin, 2008). Thus, high stability in 3SAT-AIS reduced the spurious minima, which caused the retrieved solutions to become local minima solutions.

Landscape Fitness Value

Viewing neural dynamics based on energy landscape can provide useful information about the efficiency of the algorithm. Figure 1 demarcates the variety in the fitness landscape value recorded for 3SAT-BFS and 3SAT-AIS. The neuron state retrieved from 3SAT-AIS was proven (from Table 1) to have a smaller global Hamming distance. As a result, the difference between the retrieved states and the training states was almost similar. Consequently, the difference in energy landscape was almost flat, since the fitness value was zero. The more rugged energy landscape in 3SAT-BFS was due to more solutions getting trapped into the local minima.

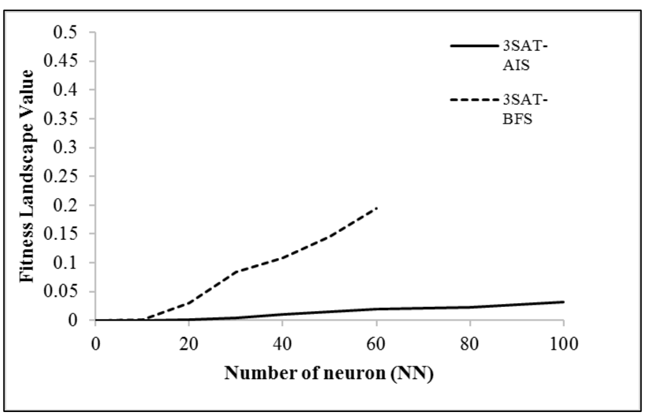

Figure 1. Fitness landscape measure for 3SAT-BFS and 3SAT-AIS

Computation Time

Computation time can be defined as the total duration for the algorithm to produce the global solutions and training process. Table 2 portrays the computation time for the proposed model, 3SAT-AIS, with the traditional method, 3SAT-BFS. According to the computational time measured, the BFS algorithm spent comparatively more computation time (CPU time) compared to the AIS paradigm. Theoretically, the training process using BFS required extra training time due to the trial-and-error process in getting the satisfied assignments. The whole built-up string of solution can collapse if one of the 3-SAT clauses is not satisfied. When this happens, BFS needs to reset the search space. On the contrary, when we applied AIS algorithms, the CPU time was faster due to the efficiency of the B-cells to improve towards the desired solution. B-cells with high and low affinity are considered in finding the best B-cells (Timmis & Neal, 2001).

Furthermore, 3SAT-AIS experienced less computation burden during the training processes as compared to 3SAT-BFS. As the number of neurons increased, 3-SAT constraint increased dramatically. The BFS algorithm required more time to arrive at the correct solution. This trend was consistent for 3SAT-AIS, even though the network complexity increased from NN=10 until NN=100. On the contrary, the 3SAT-BFS that managed to sustain up to 60 neurons. Under those circumstances, additional time was needed to relax to global solution as the number of neurons increased.

Table 2
Computation Time for 3SAT-BFS and 3SAT-AIS

Number of Neurons	Computation Time (in seconds)	
	3SAT-BFS	3SAT-AIS
10	7.23	3.28
20	92.55	13.8
30	456.0	26.54
40	1134.7	64.92
50	7440.0	88.27
60	55003.4	129.5
70	-	204.6
80	-	295.3
90	-	345.7
100	-	422.0

CONCLUSION

We presented a superior algorithm for doing 3-SAT incorporated with an artificial immune system (AIS) algorithm in the Hopfield network in this paper. An artificial immune system (AIS) was incorporated with the Hopfield neural network (3SAT-AIS) for doing 3-SAT logic programming. The hybrid paradigm was able to decrease the complexity of the network, as the number of 3-SAT clause or constraint increased. The proposed model was compared with the conventional technique, the brute-force search (BFS) hybridised with the Hopfield neural network (3SAT-BFS). The theory was supported by the tremendous differences in both performances in aspects of the global minima ratio, global Hamming distance, fitness landscape measure and the computation time. According to the experimental results, the proposed algorithm (3SAT-AIS) gave us the global minima ratio of approximately 1, faster computation time, smaller global Hamming distance and a consistent fitness landscape value, which was almost 0 compared to 3SAT-BFS. In essence, the proposed 3SAT-AIS was more robust than the 3SAT-BFS in the aspect of an exceptional global minima ratio, lower global Hamming distance, better fitness landscape value and faster computation time in doing random 3-SAT logic programming. For future work, we suggest that the AIS algorithm be used to solve other types of satisfiability problems such as maximum-satisfiability, minimum-satisfiability and quantified satisfiability problem.

REFERENCES

Abdullah, W. A. T. W. (1993). The logic of neural networks. *Physics Letters A, 176*(3), 202–206.

Afshinmanesh, F., Marandi, A., & Rahimi-Kian, A. (2005). A novel binary particle swarm optimization method using artificial immune system. In *Computer as a Tool, 2005. EUROCON 2005. The International Conference on* (Vol. 1, pp. 217–220). IEEE.

Aickelin, U. (2008). Artificial immune systems (AIS) - A new paradigm for heuristic decision making. *arXiv preprint arXiv:0801.4314*.

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

Aiyer, S. V., Niranjan, M., & Fallside, F. (1990). A theoretical investigation into the performance of the Hopfield model. *Neural Networks, IEEE Transactions on, 1*(2), 204–215.

Coello, C. A. C., & Cortés, N. C. (2005). Solving multiobjective optimization problems using an artificial immune system. *Genetic Programming and Evolvable Machines, 6*(2), 163–190.

Dasgupta, D., Ji, Z., & González, F. A. (2003). Artificial immune system (AIS) research in the last five years. In IEEE Congress on Evolutionary Computation, (1), 123–130.

Dasgupta, D., Yu, S., & Nino, F. (2011). Recent advances in artificial immune systems: Models and applications. *Applied Soft Computing, 11*(2), 1574–1587.

De Castro, L. N., & Timmis, J. (2002). Artificial immune systems: A novel paradigm to pattern recognition. *Artificial Neural Networks in Pattern Recognition, 1*, 67–84.

De Castro, L. N., & Von Zuben, F. J. (2002). Learning and optimization using the clonal selection principle. *Evolutionary Computation, IEEE Transactions on, 6*(3), 239–251.

Farmer, J. D., Packard, N. H., & Perelson, A. S. (1986). The immune system, adaptation, and machine learning. *Physica D: Nonlinear Phenomena, 22*(1), 187–204.

Gu, J. (1999). The multi-SAT algorithm. *Discrete Applied Mathematics, 96*, 111–126.

Haykin, S. (1992). Neural networks: *A Comprehensive Foundation*. New York: Macmillan College Publishing.

Holland, J. H. (1975). *Adaptation in natural and artificial systems.* Michigan: The University of Michigan Press.

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

Imada, A., & Araki, K. (1997). Application of an evolution strategy to the Hopfield model of associative memory. In *Evolutionary Computation. IEEE International Conference* (pp. 679–683). IEEE.

Ionescu, L. M., Mazare, A. G., & Serban, G. (2010). VLSI Implementation of an associative addressable memory based on Hopfield network model. *IEEE Semiconductor Conference, 2*, 499–502.

Kowalski, R. A. (1979). *Logic for problem solving*. New York: Elsevier Science Publishing.

Layeb, A. (2012). A clonal selection algorithm based tabu search for satisfiability problems. *Journal of Advances in Information Technology, 3*(2), 138–146.

Luke, S. (2013). *Essentials of metaheuristics* (2nd Ed.). United States: Lulu.

Mark, W. G., & Lee C. G. (1992). Routing in random multistage interconnections networks: Comparing exhaustive search, greedy, and neural network approaches. *International Journal of Neural System, 2*(3), 125–142.

Matsuda, S. (1998). "Optimal" Hopfield network for combinatorial optimization with linear cost function. *IEEE Transactions on Neural Networks, 9*(6), 1319–1330.

Michel, A. N., & Farrell, J. A. (1990). Associative memories via artificial neural networks. *Control Systems Magazine, IEEE, 10*(3), 6–17.

Nievergelt, J. (2000). Exhaustive search, combinatorial optimization and enumeration: Exploring the potential of raw computing power. In *Sofsem 2000: Theory and Practice of Informatics* (pp. 18-35). Berlin: Springer.

Pinkas, G., & Dechter, R. (1995). Improving energy connectionist energy minimization. *Journal of Artificial Intelligence Research, 3*, 223–237.

Rojas, R. (1999). *Neural networks: A systematic introduction.* Berlin: Springer.

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

Sathasivam, S., & Abdullah, W. A. T. W. (2008). Logic learning in Hopfield networks. arXiv preprint *arXiv:0804.4075.*

Sathasivam, S., Ng, P. F., & Hamadneh, N. (2013). Developing agent based modelling for reverse analysis method. *Journal of Applied Sciences, Engineering and Technology, 6*(22), 4281–4288.

Tang, Z., Yamaguchi, T., Tashima, K., Ishizuka, O., & Tanno, K. (1997, May). Multiple-valued immune network model and its simulations. In *Proceedings of the 1997 27th International Symposium on Multiple-Valued Logic, 1997* (pp. 233-238). IEEE.

Timmis, J., & Neal, M. (2001). A resource limited artificial immune system for data analysis. *Knowledge-Based Systems, 14*(3), 121–130.

Timmis, J., Hone, A., Stibor, T., & Clark, E. (2008). Theoretical advances in artificial immune systems. *Theoretical Computer Science, 403*(1), 11–32.

Tobias, B., & Walter, K. (2004). An improved deterministic local search algorithm for 3-SAT. *Theoretical Computer Science, 329*, 303–313.

Ullman, J. D. (1975). NP-complete scheduling problems. *Journal of Computer and System Sciences, 10*(3), 384–393.

Vilhelm, D., Peter, J., & Magnus, W. (2005). Counting models for 2SAT and 3SAT formulae. *Theoretical Computer Science, 332*(1), 265–291.

Zinovik, I., Kroening, D., & Chebiryak, Y. (2008). Computing binary combinatorial gray codes via exhaustive search with SAT solvers. *Information Theory, IEEE Transactions on, 54*(4), 1819–1823.