SCIENCE & TECHNOLOGY

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

Hybrid Discrete Hopfield Neural Network based Modified Clonal Selection Algorithm for VLSI Circuit Verification

Saratha Sathasivam[1], Mustafa Mamat[2], Mohd. Asyraf Mansor[3]* and Mohd Shareduwan Mohd Kasihmuddin[1]

[1]School of Mathematical Sciences, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia
[2]Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, 21300 UniSZA, Kuala Terengganu, Terengganu, Malaysia
[3]School of Distance Education, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia

ABSTRACT

Clonal selection algorithm and discrete Hopfield neural network are extensively employed for solving higher-order optimization problems ranging from the constraint satisfaction problem to complex pattern recognition. The modified clonal selection algorithm is a comprehensive and less iterative immune-inspired searching algorithm, utilized to search for the correct combination of instances for Very large-scale integrated (VLSI) circuit structure. In this research, the VLSI circuit framework consists of Boolean 3-Satisfiability instances with the different complexities and number of transistors are considered. Hence, a hybrid modified clonal selection algorithm with discrete Hopfield neural network is well developed to optimize the configuration of VLSI circuits with different number of electronic components such as transistors as the instances. Therefore, the performance of the developed hybrid model was assessed experimentally with the standard models, HNNVLSI-3SATES and HNNVLSI-3SATGA in term of circuit accuracy, sensitivity, robustness and runtime to complete the verification process. The results have demonstrated the developed model, HNNVLSI-3SATCSA produced a minimum error (consistently approaching 0), better accuracy (more than 80%) and faster computational time (less than 125 seconds) against changes in the complexity in term of the number of transistors. Furthermore, the developed hybrid model is able to minimize the computational burden and configurational noises for the variant of VLSI circuits.

Keywords: 3-Satisfiability problem, clonal selection algorithm, genetic algorithm; Hopfield neural network; VLSI circuit

ARTICLE INFO

Article history:
Received: 09 May 2019
Accepted: 19 July 2019
Published: 13 January 2020

E-mail addresses:
saratha@usm.my (Saratha Sathasivam)
must@unisza.edu.my (Mustafa Mamat)
asyrafman@usm.my (Mohd. Asyraf Mansor)
shareduwan@usm.my (Mohd Shareduwan Mohd Kasihmuddin)
* Corresponding author

ISSN: 0128-7680
e-ISSN: 2231-8526

INTRODUCTION

The unprecedented growth of hybrid computational approach combining neural network and the nature-inspired algorithm has benefited various applications such as in the circuit verification, face recognition, path optimization and many more (Erdener & Ozoguz, 2016; Jain et al., 2018; Elhoseny et al., 2018). The bombardments of hybrid verification model are fueled by the complexities of the problem as the industrial demand is challenging in this era. The conventional production of binary transistor units in Very Large-Scale Integration (VLSI) circuit requires an effective hybrid model to verify for any early fault due to power dissipation during the production (Constantinescu, 2003). Hence, the VLSI design with early verification will improve the performance of the VLSI configuration itself as the complexities are dependent on the number of transistors embedded in the system. We will propose a VLSI verification model by hybridizing the modified clonal selection algorithm (CSA) and Hopfield neural network (HNN) with the different number of bipolar transistors combination. The VLSI verification process tends to be tedious, due to the extensive searching process in the response to complexities (Kumar et al., 2018).

According to Mansor et al. (2016), the VLSI circuit can be configured into Boolean 2-Satisfiability (2-SAT) and its higher order counterpart, 3-Satisfiability (3-SAT) logic form by representing the literals as a single unit of bipolar transistors. The results were encouraging with the circuit accuracy above 90 % for the different number of transistors. However, according to Global VLSI circuit perspective, it was observed that the method proposed by Mansor et al. (2016) required modification in terms of training algorithm to obtain a better result. The work of Zaruba et al. (2016) utilizing the nature-inspired algorithm which was an artificial bee colony (ABC) in optimizing VLSI design has been the motivation to venture the robust nature-inspired algorithm. In addition, Kumar et al. (2018) had successfully applied the adaptive particle swarm optimization in improving VLSI optimization. The 3-Satisfiability was chosen for this work due to the reducibility feature especially for the higher-order combinatorial problem. The work of Rai et al. (2018) discussed the reduction in polynomial 3-SAT for solving the Sudoku puzzle.

Recently, a renowned immune-inspired algorithm, called clonal selection algorithm (CSA) has been utilized in various optimization problems ranging from the social media metrics, routing problem and pattern recognition. Basically, the CSA serves as other meta-heuristic or searching approach, probably effective than the standard standalone evolutionary algorithm such as a genetic algorithm (GA). Nevertheless, CSA is apparently different than GA, with the normalization and hypermutation will take place. The pioneer work of CSA has been coined by Layeb et al. (2010). Therefore, the modified clonal selection is selected due to the capability to work in tandem with discrete Hopfield neural network to tackle the logic programming such as Maximum k-Satisfiability problem (Mansor et al., 2017). The recent work by Zhang et al. (2019) had highlighted the ability of CSA with modified combinatorial recombinant in solving the various numerical optimization problem. The results obtained were acceptable to support the effectiveness of CSA in solving the optimization problem. Since the VLSI circuit verification can be regarded as an optimization problem, this work motivates us to venture this approach. Then, Avatefipour and Nafisian (2018) proposed the modified CSA as a feature selection paradigm to predict the load consumption with minimum error and iterations. The work has demonstrated better performance metrics when CSA is deployed as a feature selection approach. The effectiveness of CSA has been a motivation in modeling cell formation and verification problems (Karoum & Elbenani, 2017). Pursuing that, Schmidt et al. (2017) had applied CSA in internet traffic classification, which was a common problem in computer sciences. Cai et al. (2015) utilized CSA in order to detect the community in a complex network. In addition, CSA is applied widely in higher scale hydrothermal scheduling problem (Swain et al., 2011). Since most of the work focus on the implementation of CSA in solving optimization directly, there is limited work on combining CSA with a neural network as a single model. The effectiveness of CSA will be able to boost the capability of HNN during training and retrieval stage. Thus, we will combine CSA with HNN in verifying VLSI circuit in 3-SAT form.

In order to test the capability of CSA with HNN, we compare with the standard genetic algorithm (GA) and exhaustive search (ES). The genetic algorithm is a standard nature inspired searching algorithm, inspired by the Darwin theory. The standard algorithm of GA being used in this work is based on the studies done by Aiman and Asrar (2015) and Kasihmuddin et al. (2016). The effectiveness of GA in optimizing the weight in Multi-Criteria recommender system. This work has been coined by Kaur and Ratnoo (2019). Additionally, the exhaustive search is a primitive searching algorithm by deploying “enumerate and test” procedure in attaining the solution. In this work, the basis of exhaustive search is based on Mansor et al. (2016) and Kasihmuddin et al. (2017a). The ability of Hopfield neural network as a dynamic network especially to store the important information is the motivation of this research. Theoretically, Hopfield Neural Network is a class of recurrent neural network with sturdy capability in learning, acceptable memory, storage and mimics our biological brain system (Rojas, 2013). HNN was proposed by Hopfield (1982) to be utilized as a tool to solve notable combinatorial optimization problem and constraint satisfaction problem. In fact, HNN is an approach in artificial intelligence that demonstrates high-level learning behavior such as effective learning and retrieval mechanism. Since traditional HNN is prone to a few drawbacks (Gee et al., 1993), logic programming was embedded in HNN as a single intelligent unit (Abdullah, 1992). The effectiveness of HNN in VLSI verification has been demonstrated in the work of Mansor et al. (2016). The results were generally good, but the modifications need to be made to make it better. The main weaknesses of the work are the training method should be effective to truncate any circuit miss and errors especially if the number of transistors gets higher. In order to overwhelm the problem in complex VLSI circuit verification, the effectiveness of the proposed hybrid model will be simulated by using VLSI circuit with a different combination of transistors.

The work is organized as follows. The materials and method discuss about VLSI circuit, Boolean 3 Satisfiability representation, 3SAT Programming in Hopfield Neural Network, Clonal Selection Algorithm in VLSI Configuration and Implementation. In the following section, the results and discussions are enclosed briefly. In the final section, the concluding remarks are included to summarize the output of the work.

MATERIALS AND METHOD

Very Large-Scale Integration (VLSI) Circuit

The Boolean logic is the building block of the Boolean circuit units, utilized in various electronic components in the market. In theory, Very Large-Scale Integration (VLSI) can be defined as an amalgamation of an array of bipolar transistors to form an integrated circuit (IC) to be utilized in various devices (Kumar et al., 2018). The transistor plays an integral role as an automatic switch or controller for a specific IC. Due to the complexities of the devices, the VLSI circuit verification became tedious as the circuit structural configuration loss and defect might occur without any early alarm.

According to Mansor et al. (2016), the conventional paradigm to configure the VLSI circuit by translating then circuit structure (transistors configuration) into a Conjunctive Normal Form (CNF) instances. The weakness of this method is the circuit structural configuration loss, specifically if the circuit components are getting higher in number. Therefore, a VLSI circuit inspired by the Boolean circuit is suggested by considering the 3-SAT instances. The HNNVLSI-3SATCSA model shall magnify the early error or fault in the VLSI circuit.

According to Figure 1, the configuration of the bipolar transistor was constructed by representing its Boolean 3-SAT representation. The single clauses of 3-SAT logic are

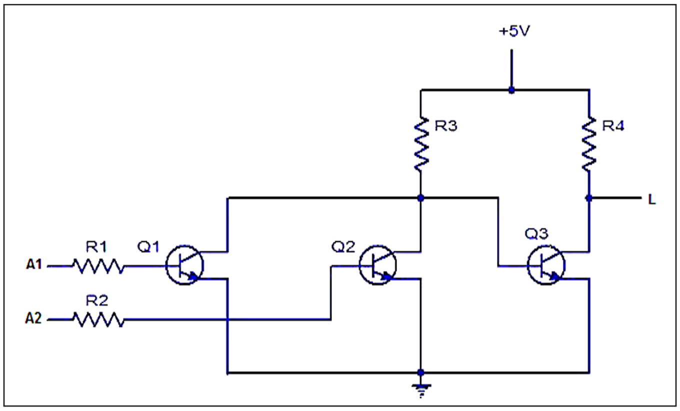

Figure 1. Schematic diagram of VLSI circuit (Mansor et al., 2016)

represented by a unit consisting of 3 bipolar transistors. Hence, the task of verifying the correct output is basically dependent on the number of transistors. The full implementation has been coined by Mansor et al. (2016). However, we verified the capability of CSA algorithm during the training phase to speed up the process per execution especially when dealing with more transistors.

Boolean 3 Satisfiability Representation

Implicit knowledge is hard to be represented in standard mathematical formulation (Sun et al., 2007). The conversion from implicit information to explicit representation can be done efficiently by formulating Boolean Satisfiability (SAT). SAT has been applied in different areas of electronic automation and functional verification (Kanj et al., 2017). With respect to VLSI application, any abstract circuit can be represented in terms of SAT formulation for which the output value needs to be validated. The resulting formulation will be mapped onto an instance of SAT. Given a set of assignment that represents the state of each component, the aim is to find the assignment of the component that satisfy the output circuit. Boolean Satisfiability (SAT) is a problem of deciding if there is a truth assignment that makes the Boolean function to be true. Any n-SAT problem with $n > 2$ where n is the number of variable, the problem can be reduced to 3-SAT (Shazli & Tahoori, 2010). In this paper, the systematic form of SAT will be formulated. The properties of SAT that ensembles 3-SAT logical rule are as follows:

The SAT formula comprises of an array of n variables, $z_1, z_2, \ldots., z_n, z \in \{-1,1\}$ inside each clause. Since $n = 2$, any SAT clauses will strictly consist of 3 variable/clause.

A set of k clauses connected by AND $(\wedge)$ in a 3-SAT formula as follows: $\exists k: F = c_1 \wedge c_2 \wedge \ldots \wedge c_k$.

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

The literals can be the variable itself or the negation of the variable. $\forall 1 \leq k \leq m, \leq i \leq 3$: $l_{k,i} = z_p$ or $l_{k,i} = \neg z_p$ for $1 \leq p \leq n$.

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

$$P = (\neg A \vee B \vee C) \wedge (\neg D \vee E \vee F)(\neg G \vee H \vee I) \tag{1}$$

If the state of each variable reads $A = D = E = F = 1$, $B = C = -1$, P becomes unsatisfiable. Several studies formulated (Prasad et al., 2005) various methods to find the consistent assignment that makes P became satisfiable. The problem with the proposed method is the complexity of the backtracking algorithm. This algorithm demands more conflict analysis towards the SAT formulation before the correct assignment can be generated. Thus, the complexity of the algorithm increases with the number of 3-SAT variables. These limitation motivates researchers (Martinez-Rios, 2017, Kasihmuddin et al., 2017b & Layeb, 2012) to emplore intelligent metaheuristic method to find the correct assignments. In this paper, the clonal selection metaheuristics is proposed to find consistent interpretation of 3-SAT formulaton.

3SAT Programming in Hopfield Neural Network

There is a number of ways of organizing bipolar input data. Generally, the element will be arranged in a single layer of N neurons that influence each other with external bias. One of the most powerful single layer neural network is Hopfield Neural Network (HNN). In HNN, each neuron represents the solution of the constraint optimization problem and the quality of the solution increase with the decrease of the energy function. In this case, if the energy function of HNN decreased to absolute minima, HNN reached optimal solution. Several studies implemented (Wang & Hong, 2019) Hebbian learning during the learning phase of HNN. Ideally, this learning rule produced optimal synaptic weight that asynchronously update the state of the neurons. In reality, conventional HNN prone to several weaknesses such as low storage capacity (Agliari et al., 2013) and easy to be trapped in local minima solution (Yang et al., 2016). Several studies indicated that the usage of logical rule during learning phase of HNN could increase the accuracy of the models (Sathasivam, 2010). In this paper, transistor configuration is tranformed to 3SAT logical rule.

$$(T_1, T_2, T_3, T_4, \dots\dots, T_N) \rightarrow P_{3SAT}, T_i \in \{-1,1\} \tag{2}$$

$T_i \in \{-1,1\}$ signifies “off” and “on” of the transistor. The cost function $E_{P_{Circuit}}$ of the logical rule in HNN is given by

$$E_{P_{Circuit}} = \sum_{i=1}^{NC} \prod_{j=1}^{k} M_{ij}, k = 3 \tag{3}$$

where NC is denoted by the number of clause containing transistors and M_{ij} is the inconsistency of the clause C_i given by

$$M_{ij} = \begin{cases} \frac{1}{2}(1 - T_x), if \neg x \\ \frac{1}{2}(1 + T_x), Otherwise \end{cases} \tag{4}$$

where $\neg x$ is the negation of literal in 3SAT clause. Generally, the local field of the HNN-3SAT is given as follows

$$T_i = \begin{cases} 1, & \sum_{k=1,i\neq j\neq k}^{N} Q_{ijk}^{(3)} T_j T_k + \sum_{j=1,i\neq j}^{N} Q_{ij}^{(2)} T_j + Q_i^{(1)} \geq \xi \\ -1, & \sum_{k=1,i\neq j\neq k}^{N} Q_{ijk}^{(3)} T_j T_k + \sum_{j=1,i\neq j}^{N} Q_{ij}^{(2)} T_j + Q_i^{(1)} < \xi \end{cases} \tag{5}$$

where Q_{ij} is the synaptic weight from unit j to i. $Q_i^{(1)}$, $Q_{ij}^{(2)}$, $Q_{ijk}^{(3)}$ are the first, second and third order neuron connection. T_j is the state of unit j and ξ is the threshold of unit i. The connection in HNN-3SAT has no connection with itself $Q_{ii} = Q_{jj} = Q_{kk} = Q_{iii} = Q_{jjj} = Q_{kkk} = 0$. It is necessary to examine the quality of the neuron state produced in equation (5). The Lypunov energy function for the $P_{Circuit}$ is given as follows

$$H_{P_{Circuit}} = -\frac{1}{3} \sum_{i=0,i\neq j\neq k}^{N} \sum_{j=0,j\neq i\neq k}^{N} \sum_{k=0,k\neq i\neq j}^{N} Q_{ijk}^{(3)} T_i T_j T_k - \frac{1}{2} \sum_{i=0,i\neq j}^{N} \sum_{j=0,j\neq i}^{N} Q_{ij}^{(2)} T_i T_j - \sum_{i=0}^{N} Q_i^{(1)} T_i \tag{6}$$

Synaptic weight of HNN-3SAT will be obtained by comparing equation (3) and (6). Due to the symmetrical property of HNN, Lyapunov energy function always converge to minimum energy. Since the energy value of each clause in 3SAT is always constant, the separation of global and local minimum energy is defined with energy threshold y.

$$\left| H_{P_{Circuit}}^{min} - H_{P_{Circuit}} \right| \leq \psi \tag{7}$$

Plotting the final state of neuron in HNN-3SAT as heights on a 2D state-space place creates a landscape of hills and valleys. Lypunov energy function will develop a neuron state that are locally stable. In this case, global solution corresponds to the correct transistor configuration.

Clonal Selection Algorithm in VLSI Configuration

Creating a functional VLSI model is important before it can be physically manufactured. Thousands of transistors will be simulated inside a single circuit board before it is ready for experimentation. Due to mathematical complexity in deciding the valid VLSI model such as Castañeda et al. (2018), the usage of metaheuristics will find the optimal solution in acceptable time range. Unfortunately, the inherent problems of VLSI configuration simulation are exacerbated in an exponential manner as the number of transistors increases linearly. Only a few works has attempted to simulate the VLSI configuration by using evolutionary metaheuristics algorithm (Laudis et al., 2018 & Kumar et al., 2018). These metaheuristics have been converted to binary representation. Pursuing that, the most popular evolutionary algorithm used is binary genetic algorithm. Although binary GA has a successful theoretical a practical history that arguably stretches further back compared to recent metaheuristics, binary GA prone to limitation such as initial solution convergence. Kasihmuddin et al. (2017b) argued that during the first few hundreds generation of crossover, only mutation would reduce the similarity of the candidate solution. This finding shows that the solution quality of GA (during the first few generation) is almost similar to conventional exhaustive search method.

This poses an important question, what if we could fully utilize the directional mutation behavior of the GA? Artificial Immune System (AIS) algorithm has evolved as a prolific metaheuristic technique that improve the main weaknesses in binary GA. AIS was introduced by Farmer et al. (1986) by systematically model the solution search according to Jerne's Immune network theory. Due to the nature of the immune system, AIS can be described as a distributed solution network which consist of functional B-Cell. Any massive and diverse population in B-Cells represent a massive space search of solutions that tends to global solutions. In this paper, each B-Cell is represented with a bipolar string which is a possible configuration of transistors. Bipolar string of 1 and -1 will be represented as "on" and "off" respectively. Our approach is to create a functional VLSI model that has the following objective function:

$$E_{P_{Circuit}} = 0 \tag{8}$$

The following steps represent the algorithm of AIS embedded to HNN:

Step 1: B-cells Initialization. 100 B-cells, B_{ij} are initialized. Each B-cell contains bipolar value that represent the configuration of transistors in VLSI. The formulation of initialization is as follows

$$B_{ij} = \begin{cases} 1 & ,rand(0,1) \geq 0.5 \\ -1 & ,otherwise \end{cases} ,1 \leq i \leq N, 1 \leq j \leq 100 \tag{9}$$

Step 2: B-cell Affinity Computation. The affinity of every B-cells would be computed. The affinity measures the sum of satisfied $P_{Circuit}$ that contains set of transistors.

$$aff_i = \sum_{i=1,j=1}^{NC} C_{ij} \tag{10}$$

$$C_{ij} = \begin{cases} 1, & E_{B_{i,j} \wedge B_2 \wedge B_3} = 0 \\ -1, & E_{B_{i,j} \wedge B_2 \wedge B_3} \geq 0 \end{cases} \tag{11}$$

where $E_{B_{i,j} \wedge B_2 \wedge B_3}$ is the cost function of clause C_{ij}.

Step 3: Cloning of B-cells. Based on step 2, top 5 B-cells with the (highest affinity value) were chosen. Roulette wheel selection (Goldberg & Deb, 1991) would identify B-cell with the most affinity value in order to proceed with the cloning phase.

$$N_{B_i} = \frac{aff_i}{\sum aff_i} \times \beta \tag{12}$$

where N_{B_i} is a number of newly produced B-cells population and β is the predefined number of clone population.

Step 4: Normalization of B-Cells. The antibodies exist in a memory response achieve a higher average affinity than those of the initial primary response (maturation of the immune response). Hence the normalized affinity of each B-cell, $aff\ B_i$ will be calculated based on the following:

$$aff_{B_i} = \frac{aff_{B_i} - \min aff_{B_i}}{\max aff_{B_i} - min\ aff_{B_i}} \tag{13}$$

where $\max aff_{B_i} \neq min\, aff_{B_i}$ because B-cell that achieved $aff_{B_i} = m\text{ax}\, aff_{B_i}$ will automatically exit the algorithm via step 2. Normalization of B-Cell is crucial in AIS because it defines the rigorousness of mutation during the next step.

Step 5: Somatic Hypermutation. Local maxima (non-improving B-cell) is potentially disrupted the local search process. In this step, somatic hypermutation is implemented by flipping the state of B-Cell based on the following formula

$$NB_i = \left(\frac{1}{\eta}\right)(aff\ N_i) + (1 - aff\ N_i)(0.01) \tag{14}$$

where NB_i denotes the number of mutation in B-cell i and η refers to the number of variables. The affinity of the B-cells was calculated by using equation (10). The best B-cells were selected as the candidate cell and stored into the memory cell. In this case, this memory cell would be retrieved to combat pathogenic attacks. This step is required to reduce the similarity index among the antibodies. In our context, any satisfied VLSI configuration would be stored in CAM to be recalled by the network.

Implementation

The implementation of hybrid models in VLSI reconfiguration are:

Step 1. The 3-SAT clauses were translated and transformed into Boolean algebra. Basically, the clauses would form a formula that would determine the overall satisfiability. In VLSI circuits, the clauses denote the different set of bipolar transistors.

Step 2. Identify a neuron to each ground neuron.

Step 3. Initialize the entire synaptic weights to zero.

Step 4. Derive a cost function that is related with negation of all 3-SAT clauses. For instance, $X=\frac{1}{2}\left(1+S_X\right)$ and $\bar{X}=\frac{1}{2}\left(1-S_X\right)$. $S_X=1$ (True) and $S_X=-1$ (False). Multiplication represents CNF and addition represents DNF.

Step 5. Compare the cost function with energy function to attain the values of synaptic weight. (Abdullah, 1992).

Step 6. Check clauses satisfaction by using ES (Mansor et al., 2016), GA (Kasihmuddin et al, 2017a) and the modified CSA. Hence, the satisfied clauses will be stored. In VLSI circuits, the satisfied transistors configuration will be stored as content addressable memory.

Step 7. Randomize the states of the neurons. The network undergoes sequences of network relaxation via Sathasivam method (Sathasivam, 2010).

Step 8. Find the corresponding local field of the state. If the final state is stable for 5 runs, we consider it as final state.

Step 9. Compute the corresponding final energy of the final state by using Lypunov equation. Validate whether the final energy obtained is a global minimum energy or local minima. In VLSI circuit, the final energy will determine correct configuration of the circuit.

Step 10. The RMSE, MAE, SSE, circuit accuracy and circuit runtime are calculated the VLSI circuit with different number of transistors per execution.

The implementation of VLSI verification models, HNNVLSI-3SATCSA, HNNVLSI-3SATES and HNNVLSI-3SATGA was carried out via Microsoft Visual Basic C++ 2013 for Windows 10. Similar processing system and CPU would be used in every execution to avoid possible bad sector. In addition, it would make the comparison to be fair and square.

RESULTS AND DISCUSSIONS

As compared to the previous VLSI verification method by utilizing HNN as coined by Mansor et al. (2016), this simulation had been developed by using modified clonal selection algorithm (CSA) as the training algorithm. The comparison would be made with the other algorithms such as exhaustive search (ES) and genetic algorithm (GA). The hybrid VLSI verification model proposed is HNNVLSIA-3SATCSA and would be compared with HNNVLSI-3SATES and HNNVLSI-3SATGA. The simulation had been restricted until the number of transistors was 108 for simplicity.

The root mean square error (RMSE) and mean absolute error (MAE) recorded by the developed model, HNNVLSI-3SATCSA are presented in Figure 2 and Figure 3 to

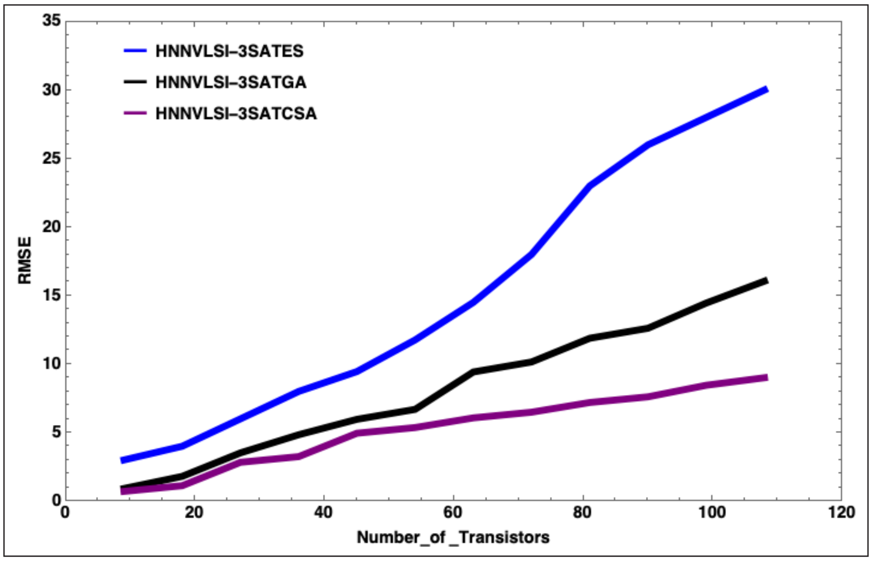

Figure 2. RMSE for the HNNVLSI-3SAT models

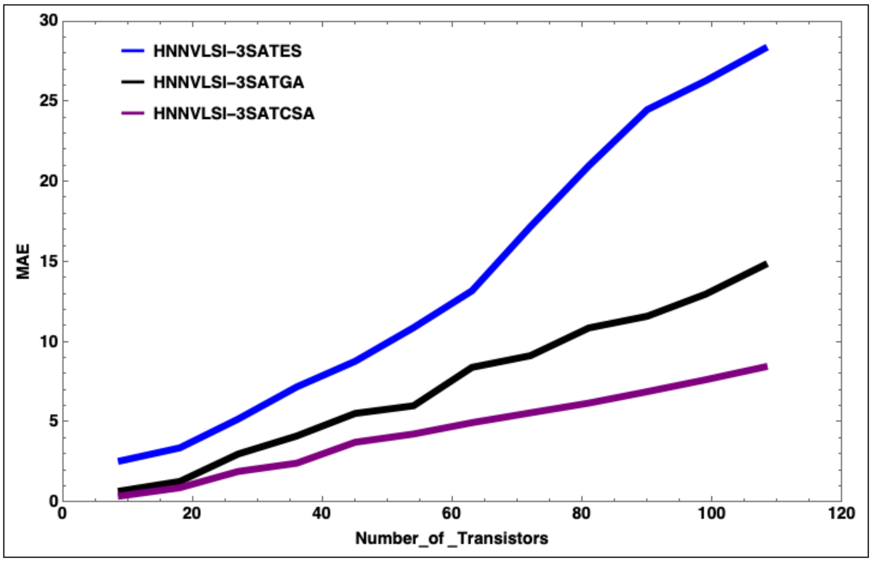

Figure 3. MAE for the HNNVLSI-3SAT models

be compared to other two counterparts, HNNVLSI-3SATES and HNNVLSI-3SATGA. Figure 2 and Figure 3 demonstrate the ability of our proposed model in verifying the circuit configuration during training phase without consuming the additional iterations and errors in generating fit strings as VLSI combinations. Thus, this emphasizes the supremacy of the somatic hypermutation operator in CSA that allows the VLSI circuits to be verified accurately without undergoing multiple unnecessary iterations and processes. The solutions will be improved directly via the effective flipping mechanism (Layeb, 2012) to obtain the feasible VLSI circuit combinations. Hence, fewer iterations will allow the model to attain faster convergence, resulting in minimum RSME and MAE obtained by HNNVLSIA-3SATCSA. HNNVLSI-3SATES is performed apparently poor due to the "trial and enumerate" procedure in attaining the correct VLSI circuit combinations. Additionally, HNNVLSI-3SATGA is still acceptable for the lower number of transistors as the non-fit strings need to be improved before undergoing the mutation operator (Aiman & Asrar, 2015).

Figure 4 shows the sum of squared error (SSE) recorded by the models in VLSI verification. Generally, it can be deduced that as the number of transistors increases, the accumulation of errors also increases. The accumulation of the errors can be magnified by observing the SSE for different execution. The significant differences can be seen when the number of transistors is between 63 to 108. From that point, the developed model, HNNVLSI-3SATCSA outperforms HNNVLSI-3SATES and HNNVLSI-3SATGA in term of sensitivity towards any incoming errors. Thus, the error during the training phase of the VLSI verification can be reduced by the optimization operators employed by CSA such as normalization of affinity and somatic hypermutation. In fact, the effectiveness of somatic hypermutation in improving the solutions has been coined by Layeb (2012) and Mansor et al. (2017). On the contrary, the ES algorithm deploys the tedious iteration processes before attaining the feasible solutions. Then, GA worked well but required early fitness and undergoing crossover and mutation.

Figure 5 manifests the circuit accuracy when the simulation is carried out by using different models. The circuit accuracy is calculated by the number of correct configurations of VLSI circuit after the retrieval phase. The proposed model, HNNVLSI-3SATCSA has recorded the accuracy between 95%-100% for the different number of transistors. The results delineate the performance of the developed model as compared with HNNVLSI03SATES and HNN-3SATGA in generating the correct VLSI circuits at the end of every execution. The reason lies in the efficiency of CSA in modeling and verifying the VLSI circuits without the interferences of massive errors and iterations. Apart from that, the proposed model is apparently worked well when the number of transistors was 108 compared to the other counterparts. Relatively high circuit accuracy demonstrates that HNNVLSI-3SATCSA has a greater stability when more transistors are introduced. In addition, better accuracy

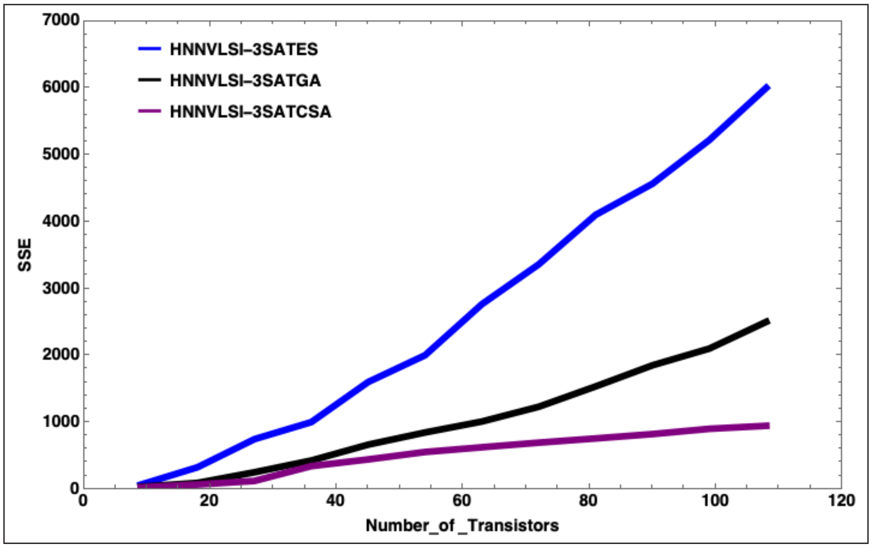

Figure 4. SSE for the HNNVLSI-3SAT models

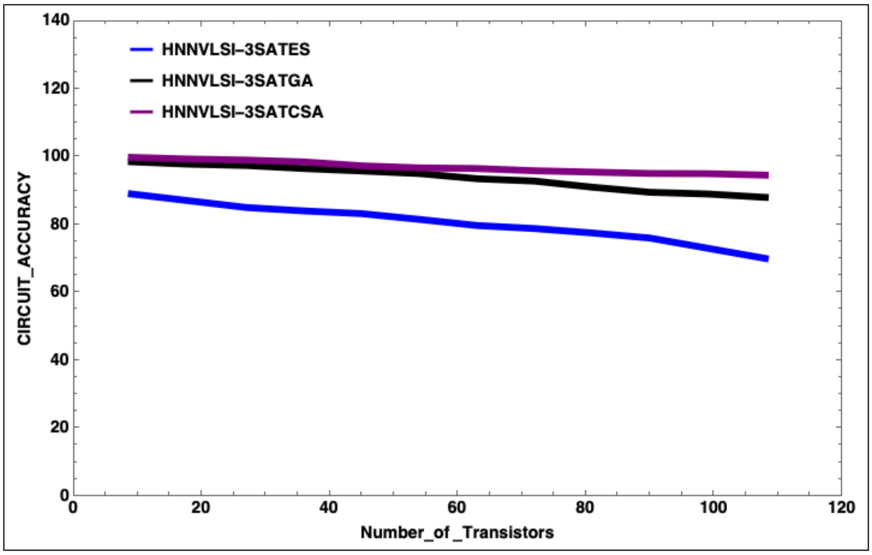

Figure 5. Circuit accuracy (%) for the HNNVLSI-3SAT models

will improve the functionality of normal VLSI circuit as the complexity increases. The robust training algorithm will exponentially lower the complexity of the whole VLSI verification processes. The conventional approach, HNNVLSI-3SATES has recorded the lowest accuracy due to the ineffectiveness of generating the fit VLSI combinations and reduced the number of correct VLSI configurations generated at the end of the execution.

Figure 6 demonstrates the circuit runtime recorded by HNNVLSI-3SATCSA, HNNVLSI-3SATGA, and HNNVLSI-3SATES by using different combinations of the transistor. Based on the circuit runtime, HNNVLSI-3SATCSA completes the VLSI verification faster than HNNVLSI-3SATGA and HNNVLSI-3SATES. Theoretically, the training process by ES requires extra training time due to the trial and error process in attaining the correct VLSI configurations. Therefore, the entire non-fit strings that resemble the VLSI combinations will collapse if any one of the clauses is not satisfied. On the contrary, the GA will enhance the training process but require early population adjustment together with crossover and mutation. The procedure will reduce the time taken to check the VLSI circuit as compared to ES. However, when compared with CSA, the correct non-fit VLSI circuit can be improved with the robust hypermutation operator without the need to reset the whole combinations. Thus, HNNVLSI-3SATCSA experienced less computation burden during the training processes as compared to the other two approaches. Thus, the developed model is more robust than the other two counterparts. The faster circuit runtime together with better accuracy is essentially needed in VLSI circuit verification in order to avoid any miss and power dissipation. The results have improved the VLSI verification paradigm done by Mansor et al. (2016) and Kumar et al. (2018).

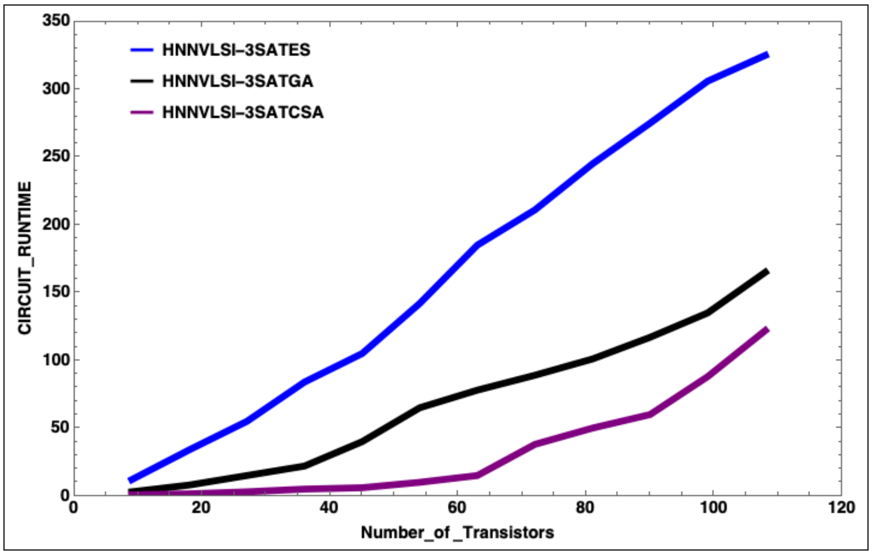

Figure 6. Circuit runtime for the HNNVLSI-3SAT models

CONCLUSION

This work was largely motivated from industrial application point of view where we clearly identified a need for model verification by using neural network ensembles as decision support system. We have presented our proposed algorithms, namely HNNVLSI-3SATCSA model and the conventional models, HNNVLSI-3SATGA and HNNVLSI-3SATES to check the VLSI circuit accuracy if the number of transistors gets higher. It had been shown by the computer simulations that both models that incorporated with HNN were able to retrieve the desired output as the traditional VLSI models did. However, it was identified that HNNVLSI-3SATCSA outperformed the other models in terms of accuracy and robustness of the techniques in VLSI configuration. Hence, the proposed models are supported by the solid agreement of RMSE, MAE, SSE, circuit accuracy and circuit runtime obtained. Thus, our hybrid paradigm can be further integrated to solve and model more complicated electronics problem. Future research topics are to deal with other variant of HNN such as Mutation HNN (Hu et al., 2011), memristor HNN (Liu et al., 2018) and genetic optimized HNN (Jayashree & Kumar, 2019).

ACKNOWLEDGEMENT

This research is supported by Universiti Sains Malaysia and Fundamental Research Grant Scheme (FRGS) (203/PMATHS/6711689) by Ministry of Higher Education Malaysia.

REFERENCES

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

Agliari, E., Barra, A., De Antoni, A., & Galluzzi, A. (2013). Parallel retrieval of correlated patterns: From Hopfield networks to Boltzmann machines. *Neural Networks, 38*, 52-63.

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

Avatefipour, O., & Nafisian, A. (2018). A novel electric load consumption prediction and feature selection model based on modified clonal selection algorithm. *Journal of Intelligent and Fuzzy Systems, 34*(4), 2261-2272.

Cai, Q., Gong, M., Ma, L., & Jiao, L. (2015). A novel clonal selection algorithm for community detection in complex networks. *Computational Intelligence, 31*(3), 442-464.

Castañeda, O., Goldstein, T., & Studer, C. (2018). VLSI designs for joint channel estimation and data detection in large SIMO wireless systems. *IEEE Transactions on Circuits and Systems I: Regular Papers, 65*(3), 1120-1132.

Constantinescu, C. (2003). Trends and challenges in VLSI circuit reliability. *IEEE Micro, 23*(4), 14-19.

Elhoseny, M., Tharwat, A., & Hassanien, A. E. (2018). Bezier curve based path planning in a dynamic field using modified genetic algorithm. *Journal of Computational Science, 25*, 339-350.

Erdener, Ö., & Ozoguz, S. (2016). A new neuron and synapse model suitable for low power VLSI implementation. *Analog Integrated Circuits and Signal Processing, 89*(3), 749-770.

Farmer, J. D., Packard, N. H., & Perelson, A. S. (1986). The immune system, adaptation, and machine learning. *Physica D: Nonlinear Phenomena, 22*(1-3), 187-204.

Gee, A. H., Aiyer, S. V., & Prager, R. W. (1993). An analytical framework for optimizing neural networks. *Neural Networks, 6*(1), 79-97.

Goldberg, D. E., & Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. In G. J. E. Rawlins (Ed.), *Foundations of genetic algorithms* (Vol. 1, pp. 69-93). San Mateo, CA: Morgan Kaufmann Publishers, Inc.

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.

Hu, L., Sun, F., Xu, H., Liu, H., & Zhang, X. (2011). Mutation Hopfield neural network and its applications. *Information Sciences, 181*(1), 92-105.

Jain, N., Kumar, S., Kumar, A., Shamsolmoali, P., & Zareapoor, M. (2018). Hybrid deep neural networks for face emotion recognition. *Pattern Recognition Letters, 115*, 101-106.

Jayashree, J., & Kumar, S. A. (2019). Evolutionary correlated gravitational search algorithm (ECGS) with genetic optimized hopfield neural network (GHNN)–A hybrid expert system for diagnosis of diabetes. *Measurement, 145*, 551-558.

Kanj, I., Thilikos, D. M., & Xia, G. (2017). On the parameterized complexity of monotone and antimonotone weighted circuit satisfiability. *Information and Computation, 257*, 139-156.

Karoum, B., & Elbenani, Y. B. (2017). A clonal selection algorithm for the generalized cell formation problem considering machine reliability and alternative routings. *Production Engineering, 11*(4-5), 545-556.

Kasihmuddin, B. M., Shareduwan, M., Mansor, B., Asyraf, M., & Sathasivam, S. (2016). Genetic Algorithm for Restricted Maximum k-Satisfiability in the Hopfield Network. *International Journal of Interactive Multimedia and Artificial Intelligence, 4*(2), 52-60.

Kasihmuddin, M. S. M., Mansor, M. A., & Sathasivam, S. (2017a). Hybrid Genetic Algorithm in the Hopfield Network for Logic Satisfiability Problem. *Pertanika Journal of Science and Technology, 25*(1), 139-152.

Kasihmuddin, M. S. M., Sathasivam, S., & Mansor, M. A. (2017b). Hybrid genetic algorithm in the Hopfield network for maximum 2-satisfiability problem. In U. Wagenknecht, P. Potschke, S. Wiessner & M. Gehde (Eds.), *AIP Conference Proceedings* (Vol. 1870, No. 1, p. 050001). Melville, NY: AIP Publishing.

Kaur, G., & Ratnoo, S. (2019). Adaptive genetic algorithm for feature weighting in multi-criteria recommender systems. *Pertanika Journal of Science & Technology, 27*(1), 123-141.

Kumar, S. V., Rao, P. V., Sharath, H. A., Sachin, B. M., Ravi, U. S., & Monica, B. V. (2018, In Press). Review on VLSI design using optimization and self-adaptive particle swarm optimization. *Journal of King Saud University-Computer and Information Sciences*.

Laudis, L. L., Shyam, S., Jemila, C., & Suresh, V. (2018). MOBA: multi objective bat algorithm for combinatorial optimization in VLSI. *Procedia Computer Science, 125*, 840-846.

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

Layeb, A., Deneche, A. H., & Meshoul, S. (2010). A new artificial immune system for solving the maximum satisfiability problem. In N. García-Pedrajas, F. Herrera, C. Fyfe, J. M. Benítez & M. Ali (Eds.), *Trends in Applied Intelligent Systems* (pp. 136-142). Berlin, Heidelberg: Springer.

Liu, S., Yu, Y., Zhang, S., & Zhang, Y. (2018). Robust stability of fractional-order memristor-based Hopfield neural networks with parameter disturbances. *Physica A: Statistical Mechanics and its Applications, 509*, 845-854.

Mansor, M. A. B., Kasihmuddin, M. S. B. M., & Sathasivam, S. (2017). Robust Artificial Immune System in the Hopfield network for Maximum k-Satisfiability. *International Journal of Interactive Multimedia and Artificial Intelligence, 4*(4), 63-71.

Mansor, M. A., Kasihmuddin, M. S. M., & Sathasivam, S. (2016). VLSI circuit configuration using satisfiability logic in Hopfield network. *International Journal of Intelligent Systems and Applications (IJISA), 8*(9), 22-29.

Martinez-Rios, F. (2017). A new hybridized algorithm based on population-based simulated annealing with an experimental study of phase transition in 3-SAT. *Procedia Computer Science, 116*, 427-434.

Prasad, M. R., Biere, A., & Gupta, A. (2005). A survey of recent advances in SAT-based formal verification. *International Journal on Software Tools for Technology Transfer, 7*(2), 156-173.

Rai, D., Chaudhari, N. S., & Ingle, M. (2018). Polynomial 3-SAT Reduction of Sudoku Puzzle. *International Journal of Advanced Research in Computer Science, 9*(3), 194-197.

Rojas, R. (2013). *Neural networks: A systematic introduction*. Heidelberg, Germany: Springer-Verlag.

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

Schmidt, B., Al-Fuqaha, A., Gupta, A., & Kountanis, D. (2017). Optimizing an artificial immune system algorithm in support of flow-Based internet traffic classification. *Applied Soft Computing, 54*, 1-22.

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

Sun, R., Zhang, X., Slusarz, P., & Mathews, R. (2007). The interaction of implicit learning, explicit hypothesis testing learning and implicit-to-explicit knowledge extraction. *Neural networks, 20*(1), 34-47.

Swain, R. K., Barisal, A. K., Hota, P. K., & Chakrabarti, R. (2011). Short-term hydrothermal scheduling using clonal selection algorithm. International *Journal of Electrical Power and Energy Systems, 33*(3), 647-656.

Wang, H., & Hong, M. (2019). Supervised Hebb rule based feature selection for text classification. *Information Processing and Management, 56*(1), 167-191.

Yang, G., Wu, S., Jin, Q., & Xu, J. (2016). A hybrid approach based on stochastic competitive Hopfield neural network and efficient genetic algorithm for frequency assignment problem. *Applied Soft Computing, 39*, 104-116.

Zaruba, D., Zaporozhets, D., & Kureichik, V. (2016). Artificial bee colony algorithm-a novel tool for VLSI placement. In A. Abraham, S. Kovalev, V. Tarassov, & V. Snášel (Eds.), *Proceedings of the First International Scientific Conference "Intelligent Information Technologies for Industry"(IITI'16)* (pp. 433-442). Cham, Switzerland: Springer.

Zhang, W., Gao, K., Zhang, W., Wang, X., Zhang, Q., & Wang, H. (2019). A hybrid clonal selection algorithm with modified combinatorial recombination and success-history based adaptive mutation for numerical optimization. *Applied Intelligence, 49*(2), 819-836.