

Bezier Curves Satisfiability Model in Enhanced Hopfield Network

Mohd Shareduwan M. Kasihmuddin

School of Mathematical Sciences, Universiti Sains Malaysia, 11800 USM, Penang Malaysia

E-mail: iwanmaidin@gmail.com

Mohd Asyraf Mansor

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

E-mail: asyrafalvez@live.com

Saratha Sathasivam

School of Mathematical Sciences, Universiti Sains Malaysia, 11800 USM, Penang Malaysia

E-mail: saratha@usm.my

Abstract—Bezier curve is one of the most pragmatic curves that has vast application in computer aided geometry design. Unlike other normal curves, any Bezier curve model must follow the properties of Bezier curve. In our paper, we proposed the reconstruction of Bezier models by implementing satisfiability problem in Hopfield neural network as Bezier properties verification technique. We represent our logic construction to 2-satisfiability (2SAT) clauses in order to represent the properties of the Bezier curve model. The developed Bezier model will be integrated with Hopfield neural network in order to detect the existence of any non-Bezier curve. Microsoft Visual C++ 2013 is used as a platform for training, testing and validating of our proposed design. Hence, the performance of our proposed technique is evaluated based on global Bezier model and computation time. It has been observed that most of the model produced by HNN-2SAT are Bezier curve models.

Index Terms—Bezier curve, Hopfield network, 2-satisfiability, Logic programming, Wan Abdullah's method.

I. INTRODUCTION

Computer Aided Geometric Design (CAGD) has emerged as one of the most eminent fields in computer graphic, numerical computation, and geometrical studies. The renaissance of CAGD is inaugurated by the various form of Bezier curves that has attracted a prolific number of research [1]. Hence, Bezier curves are one of the most intensively used in CAGD [2]. Popularized by Bezier [3], Bezier curves are used to construct smooth curves at any scale by considering its own properties to be applied in the design. The main impetus of this research is to reconstruct the Bezier curves according to the correct properties by using neural network approach. Hence, the Bezier properties verification process requires a robust and stable algorithm in order to construct more complex

Bezier curves. Thus, we choose to translate the Bezier curves properties into 2-Satisfiability problem and integrate it in Hopfield neural network by using logic programming (HNN-2SAT). Basically, 2-Satisfiability (2SAT) is the prominent counterpart of the Boolean satisfiability (SAT) optimization problem, that is denoted in Conjunctive Normal Form (CNF) form [4, 8]. Without a doubt, there have been a lot of applications of 2-Satisfiability such as the previous work by Femer [5], Papadimitriou [6], Patreschi and Simeone [7]. These related works emphasized on the 2-Satisfiability as an optimization problem. On the separate note, we can transform any real dataset or mathematical properties into 2SAT form with the assistance of logic programming [9]. In this paper, we emphasized on the Bezier curve properties as 2SAT combinatorial optimization problems in logic programming.

The Hopfield neural network plays an important role in the field of artificial intelligence and mathematical computation. Recurrent Hopfield neural networks are principally dynamical schemes that feedback signals to themselves. The network was inaugurated by Hopfield and Tank [10]. One of the interesting features is the network possess a dynamical system with stable states with each own basin and attraction [11]. Moreover, the Hopfield neural network minimizes Lyapunov energy because of physical spin of the neuron states. On top of that, the network produced global output by minimizing the network energy. Gadi Pinkas and Wan Abdullah [9, 12] demarcated a bi-directional mapping between logic and energy function of a symmetric neural network. Besides, both related works are the building blocks for a corresponding logic program. The work of Sathasivam [13] presented that the optimized recurrent Hopfield network could be possibly used to do logic programming. Above all, the ability of learning by using Hopfield network is the main priority in the reconstruction of the Bezier curves. The memory will be stored in Hopfield's brain as content addressable memory (CAM) [14]. Hence, we can reconstruct the correct Bezier curves by retrieving

the stored memory based on the properties.

Logic Programming refers to an auspicious field, and widely used to resolve numerous constraint optimization problem [15]. Additionally, the logic programming can be demarcated as an optimization problem [13, 16]. In this paper, the Bezier properties will be represented by the clauses in 2SAT. After that, 2SAT problem will be translated into logic programming integrated with Hopfield neural network. The impetus of our study is the well-known Wan Abdullah's technique [9, 12]. Wan Abdullah presented a model of doing logic programming in Hopfield network by opt for the Horn clauses as the problem [12, 17]. Theoretically, the Hopfield network can minimize the logical inconsistency in any order of logic programs [18]. Thus, we develop a network called HNN-2SAT to be tested with the Bezier satisfiability problem. The implementation of Hopfield neural network, 2-Satisfiability and logic programming as a hybrid model (HNN-2SAT) in the reconstruction of various Bezier curves is the main contribution of the study.

The rest of the paper is organized as follows. Section II contains the fundamental conceptual background of satisfiability problems, especially the 2-satisfiability (2SAT) problem. In Section III, we discuss the core theory of Bezier satisfiability and the important properties of Bezier curves. In addition, we emphasize the different type of Bezier curves being used in our proposed Bezier satisfiability problem. Section IV accentuates the neuro-logic paradigm, consists of Hopfield neural network and logic programming. This will include the implementation of Hopfield network and logic programming in reconstructing the Bezier curves. Section V describes the theory implementation for our proposed algorithm in doing Bezier satisfiability. Section VI presents the result and analysis. In this section, the discussion will point on the trend of global Bezier curve and the running time recorded for different type of Bezier curves. Section VII then highlights the future work and encloses the conclusion of this work.

II. SATISFIABILITY (SAT) PROBLEM

Satisfiability (SAT) is a significant problem in computer science and mathematics [4]. SAT problem helps researcher to deal with constraint optimization problem such as circuit and pattern reconstruction. One way to learn via SAT is by embedding our required information inside and SAT problem and solve it optimally. In Bezier reconstruction, learning and verifying is Bezier curve model work identically with SAT problem.

In general, a Boolean formula is satisfiable if there exists an assignment of values true and false that makes the entire expression true [5]. The easiest way to solve SAT problem is by utilizing exhaustive search method, where SAT will try out every possible truth assignment. For example, given a problem size n , there will be 2^n such assignments and l literals to set for each assignment [13, 27]. At this point, this method involves

$O(1.2^n)$ operations [28] and it was proven by many that SAT is an NP-complete problem.

SAT problem normally represented in Boolean variables or expressions in conjunctive normal form (CNF). CNF is defined as conjunction of clauses, where the clauses are disjunction of literal [35]. Literal is a variable or its negation. For example:

$$(x_1 \vee x_2) \wedge (\neg x_2 \vee x_3 \vee \neg x_5) \wedge (\neg x_1 \vee x_4) \quad (1)$$

Based on (1), x_1, x_2, x_3, x_4 are Boolean variables to be assigned, \neg means negations (logical NOT), \vee means negations (logical OR), \wedge means negations (logical AND). We can satisfy formula (1) by taking $x_1 = true, x_2 = false, x_3 = false, x_4 = true$. However, if any formula is not satisfiable, it will be termed as unsatisfiable.

In Bezier model reconstruction, we integrate the foundation of satisfiability in order to obtain the correct Bezier model. The literals in every clause will be represented the properties of the Bezier curve model.

A. 2SAT

2SAT is a subset of SAT problem. It is a classical NP-problem that determine the satisfiability of sets of clauses with at most two literals per clause (2-CNF formulas) [8]. Besides, it is popular and highly regard problem for general Boolean satisfiability which can involve constraints on two variables [31]. In addition, the variables can allow two possibilities for the value of each variable. 2SAT problem can be expressed as 2-CNF (2-Conjunctive Normal Form). Randomized 2SAT problem is considered as NP problem or non-deterministic problem [6]. The three fundamental components of 2SAT are summarized as follows:

1. A set of m variables, x_1, x_2, \dots, x_m
2. A set of literals. A literal is a variable or a negation of a variable.
3. A set of n distinct clauses: C_1, C_2, \dots, C_n . Each clause consists of only literals combined by just logical OR (\vee). Each clause must consist of 2 variables.

The Boolean values are $\{1, -1\}$. Researchers have replaced F and T in the neural networks by 1 and -1, respectively to emphasized false and true. Since each variable can take only two values, a statement with n variables requires a table with 2^n rows. The goal of the 2SAT problem is to determine whether there exists an assignment of truth values to variables that makes the following Boolean formula P satisfiable.

$$P = \bigwedge_{i=1}^n C_i \quad (2)$$

Where \wedge is a logical AND connector. C_i is a clausal

form of DNF with 2 variables. Each clause in 2SAT has the following form

$$C_i = \bigvee_{i=1}^n (x_i, y_i) \quad (3)$$

$x_i \in \{k_i, \neg k_i\}$ and $y_i \in \{r_i, \neg r_i\}$ $\neg k_i$ and $\neg r_i$ are negations of the literals.

III. BEZIER MODEL SATISFIABILITY

Popularized by French engineer, Pierre Bezier, Bezier curve is a parametric curve that frequently used in computer graphics and design automobile bodies. Currently, Bezier curves are extensively used to model smooth curves and other applications [20]. A Bezier curve is defined by a set of control points P_0 through P_n , where n is the order of the curve. In essence, the first and the last control point are always the end point of the curve. The other point between the first control point to the last do not lie on the curve [21]. Since the curve is completely enclosed in the convex hull of its control points, the points can be manipulated graphically.

A. Properties of Bezier Curve

Bezier curve is a parametric curve that uses the Bernstein polynomials as a basis. Generally, a Bezier curve of degree n (Order $n+1$) is represented by

$$B(t) = \sum_{i=0}^n b_{i,n}(t) P_i, \quad 0 \leq t \leq 1 \quad (4)$$

where

$$b_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad n = 0, 1, 2, 3, \dots, n \quad (5)$$

The shape of the curve can be determined by using the coefficient P_i which are the control points and $B_{i,n}(t)$ are the basis of the function [21]. Lines can be drawn between the control points of the curve in order to form control polygon. Bezier curve possessed the following properties

1) Geometry invariance property:

Partition of unity of the Bernstein polynomial satisfy any of the Bezier curve although the curve undergoes translation and rotation of its control point [22].

The basis function of any Bezier curve can be represented by using the following graph:

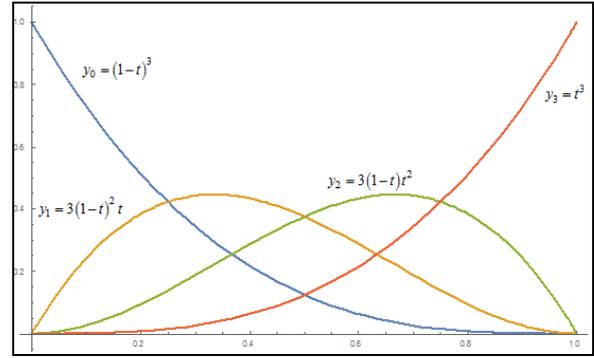


Fig.1. Basis function for cubic Bezier

2) Geometric point property:

(i) The first and the last control points are the endpoints of the curve. Moreover, $b_0 = B(0)$ and $b_n = B(1)$.

(ii) The curve is a tangent to the control polygon at the endpoints [21]. This can be verified by taking first derivatives of a Bezier curve

$$B'(t) = \frac{dB(t)}{dt} = n \sum_{i=0}^{n-1} (P_{i+1} - P_i) b_{i,n-1}(t), \quad 0 \leq t \leq 1 \quad (6)$$

In detail, we acquire $P'(0) = n(P_1 - P_0)$ and $B'(1) = n(P_n - P_{n-1})$. The above equation can further simplified by setting $\Delta P_i = P_{i+1} - P_i$:

$$B'(t) = n \sum_{i=0}^{n-1} \Delta P_i B_{i,n-1}(t), \quad 0 \leq t \leq 1 \quad (7)$$

3) Convex hull property:

A domain D is considered as a convex if any two points P_1 and P_2 in the domain, the segment $\overline{P_1 P_2}$ is entirely embodied in the domain D [23]. The convex hull of a set of points P is the boundary of the smallest convex domain containing P . The entire curve is contained within the convex hull of the control point.

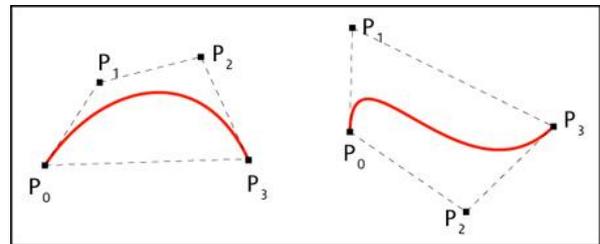


Fig.2. Convex hull built from the control point

B. Type of Bezier Curve

A Bezier curve is defined based on their control points. For our studies, we will examine the following curve

1) Linear Bezier curves

Given points P_0 and P_1 , a linear Bezier curve is a straight line between those two points. The curve can be formulated as followed

$$B(t) = (1-t)P_0 + tP_1 \quad (8)$$

Where $0 \leq t \leq 1$. The equation normally similar to linear interpolation.



Fig.3. Linear Bezier curve

2) Quadratic Bezier curves

Quadratic Bezier curve is a curve that traced by the function $B(t)$ given points P_0, P_1 and P_2 . The curve can be formulated as followed:

$$B(t) = (1-t)^2 P_0 + 2(1-t)tP_1 + t^2 P_2 \quad (9)$$

Where $0 \leq t \leq 1$. The equation is interpreted as the linear interpolant of corresponding points on the linear Bezier curves from P_0 to P_1 and from P_1 to P_2 [24].

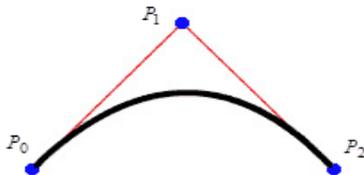


Fig.4. Quadratic Bezier curve

3) Cubic Bezier curves

Cubic Bezier curve is a curve that traced by the function $B(t)$ given points P_0, P_1, P_2 and P_3 . Cubic Bezier curve can be formulated as followed:

$$B(t) = (1-t)^3 P_0 + 3(1-t)^2 tP_1 + 3(1-t)t^2 P_2 + t^3 P_3 \quad (10)$$

Where $0 \leq t \leq 1$. The curve will not pass through P_1 or P_2 because these points only to provide directional information. The cubic Bezier curve can be defined as a linear combination of two quadratic Bezier curves [25].

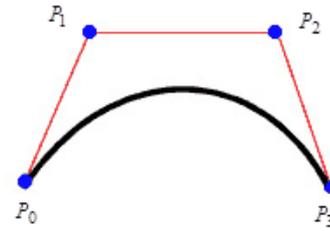


Fig.5. Cubic Bezier curve

C. Bezier-SAT

Table 1. Bezier-2SAT clause representation

Clause	Properties	Descriptions
C_1	Partition of unity $\sum_{i=0}^n B_{i,n}(t) = 1$	Linear Bezier $C_{11} = (1-t)$ $C_{12} = t$ Total clauses: 2
		Quadratic Bezier $C_{11} = (1-t)^2$ $C_{12} = 2(1-t)t$ $C_{13} = t^2$ Total clauses: 3
C_2	Geometric point where P are the control points of the curve. The curve must touch the first and the endpoint.	Quadratic Bezier $C_{11} = (1-t)^3$ $C_{12} = 3(1-t)t^2$ $C_{13} = 3(1-t)t^2$ $C_{14} = t^3$ Total clauses: 4
C_3	Convex hull property	$\{P_0, P_1, P_2, P_3, \dots, P_i\} \in D_C$ Where D_C is the domain of the convex hull Total clause: $i-1$

The properties of Bezier curve with different orders can be represented by using logic. In this case, 2SAT logic is utilized to implant the important properties of Bezier curve. If any of the random curve followed all the important properties of Bezier curve [25], the curve model is considered as Satisfiable (It is a Bezier curve). On contrary, if any of the curve model does not abide with properties, 2SAT logic will consider the curve as

unsatisfiable (It is not a Bezier curve) [26]. The task of representing the important properties of Bezier curve in 2SAT will be used in training phase of neural network. It was proven by previous researchers that the hybridization between 2SAT logic and neural network is able to solve constraint optimization problem such as pattern reconstruction. Table 1 shows the representation of the properties of the Bezier curve and 2SAT clause during training phase.

IV. BEZIER-SAT IN HOPFIELD NEURAL NETWORK

Neuro-logic is the blend of neural network and logic programming as a single hybrid network. Specifically, the neuro-logic methods can be applied in a various field such as constraint optimization problem, pattern recognition, circuit and curve reconstruction [33]. In this paper, we apply the Hopfield neural network and logic programming to reconstruct the Bezier curves according to the properties needed based on 2SAT instances. The effectiveness of our proposed paradigm (HNN-2SAT) will be discussed briefly in Section VI.

A. Hopfield Model

In Bezier curve reconstruction, we select the Hopfield neural network because it is well distributed. Hence, Hopfield neural network is easier to be integrated with any paradigms to solve satisfiability problem [31].

Technically, we incorporate the 2SAT with 3 clauses in Hopfield neural network so that we can relate it with the Bezier curve properties. The correctness of Bezier curve that will be reconstructed totally rely on the effectiveness our proposed network. In addition, the Hopfield neural network can be demarcated as a model of content addressable memory (CAM) [15, 16]. In layman's term, we call it as Hopfield's brain. Hence, Hopfield's brain replicates our biological brain function to store and process the memories. Besides, the learning and retrieving data in Hopfield neural network are the fundamental aspect of content addressable memory (CAM) [11, 14].

Specifically, the Hopfield neural network is a class of recurrent auto-associative network [33]. The units in Hopfield models are predominantly binary threshold unit [34]. Hence, the Hopfield nets will yield a binary value such as 1 and -1. The definition for unit i 's activation, a_i are given as follows:

$$a_i = \begin{cases} 1 & \text{if } \sum_j W_{ij} S_j > \xi_i \\ -1 & \text{Otherwise} \end{cases} \quad (11)$$

where W_{ij} is the connection strength from unit j to i .

S_j is the state of unit j and ξ_i is the threshold of unit i . The connection in Hopfield net typically has no connection with itself $W_{ii} = 0$ and connections are symmetric or bidirectional [12, 13].

Hopfield network work asynchronously with each neuron updating their state deterministically. The system consists of N formal neurons, each is described by an Ising variable. Neurons are bipolar $S_i \in \{1, -1\}$ obeying the dynamics $S_i \rightarrow \text{sgn}(h_i)$ where the local field h_i . The connection model can be generalized to include higher order connection. This changes the field to

$$h_i = \sum_j W_{ij}^{(2)} S_j + J_i^{(1)} \quad (12)$$

The weight in Hopfield network is always symmetrical. The weight in Hopfield network denotes to the connection strength between the neurons. The updating rule maintains as follows:

$$S_i(t+1) = \text{sgn}[h_i(t)] \quad (13)$$

This property guarantee the energy will decrease monotonically even though following the activation system [14, 30]. However, it will drive the network to search for the possible minimum energy. The following equation represents energy for Hopfield network.

$$E = \dots - \frac{1}{2} \sum_i \sum_j W_{ij}^{(2)} S_i S_j - \sum_i W_i^{(1)} S_i \quad (14)$$

This energy function is vital to improve the degree of convergence of the proposed network [18]. Thus, the energy value is important to obtain global Bezier curves. The power of reconstructing the curves depends on how the synaptic weights are computed. Hence, our proposed networks are able to update the weights and proceed the Bezier curves reconstruction effectively.

B. Logic Programming in Hopfield Network

Fundamentally, logic programming can be treated as a problem in constraint optimization outlook [22]. Hence, it can be carried out in the Hopfield neural network to obtain global solutions. This can be done by using the neurons to store the truth values of the literal and writing a cost function which is minimized when all clauses are satisfied [13, 29]. In other words, the core mission is to discover 'models' corresponding to the given logic program. The important of Hopfield network in doing logic programming was brought up because of its exceptional content addressable memory properties during learning process.

Implementation of HNN-2SAT in Logic Programming.

i. The 2SAT clauses are transformed into Boolean algebra. Thus, the clauses will form a formula that will be used to check the properties of Bezier curves. In this paper, each clause represents the properties of the Bezier curve.

ii. Identify a neuron to each ground neuron. Then, initialize all the possible weights to zero.

iii. Derive a cost function with negation of all 2SAT clauses. As an illustration, we have the following cost functions, $X = \frac{1}{2}(1+S_X)$ and $\bar{X} = \frac{1}{2}(1-S_X)$. Hence, the states are represented as $S_X=1$ (True) and $S_X=-1$ (False). For this scenario, the multiplication signifies conjunction and addition symbolizes disjunction.

iv. Comparing the cost function with energy, E by obtaining the values of the connection strengths. (Wan Abdullah's Method) [9]

v. Check clauses satisfaction by using exhaustive search. Hence, the satisfied assignments will be stored. In this paper, the satisfied assignments for the Bezier curves (correct properties) will be stored as content addressable memory (CAM).

vi. The states of the neurons are randomized. The network undergoes sequences of network relaxation [14]. Calculate the resultant local field $h_i(t)$ of the state. Let say if the final state is stable for 5 runs, we consider it as final state.

vii. Compute the corresponding final energy E of the final state via implementing Lypunov equation. Authenticate whether the final energy achieved is a global minimum energy or local minima. In Bezier curves reconstruction, the final energy depicts the correct properties of the Bezier curves. Finally, the global Bezier model and running time are computed for every Bezier models.

V. IMPLEMENTATION

In this exploration, we require methodical procedures. The procedures would work by using Microsoft Visual C++ 2013 as a platform to simulate our logic program. First of all, we generate a random program based on 2SAT clauses. In Bezier curve reconstruction, we represent each of the properties of Bezier model as the clauses. Each of the clauses form randomized 2SAT formula. Each of the clauses consist of the randomized state. The states of the 2SAT clauses (representing properties of the Bezier) will be verified and the correct clauses will be retained (training stage). After undergo HNN-2SAT, the network reached the final states. Equation (13) is vital to ensure the network achieve stable states. Stable final states will be achieved when the state remains unchanged for 5 consecutive runs. Pinkas [18] emphasized that by permitting an ANN to evolve in time shall lead to the stable state where the energy function obtained does not change further. In this case, the corresponding final energy for the stable state will be calculated. If the difference between the final energy and the global minimum energy is within the termination criteria, then consider the solution as global minimum energy. 0.001 was chosen as a termination criterion since this value gave us better output accuracy. We run 100 training and 100 combinations of neurons in order to reduce statistical error. Since the network will produce 10000 Bezier models, we calculate the percentage of

global Bezier model. Running time will be recorded from the start to the end of the program. In this paper, the global Bezier model is considered as the global solution.

VI. RESULT AND DISCUSSION

In order to test the performance of HNN-2SAT in reconstructing the Bezier curve model, we evaluated the proposed paradigm based on global Bezier model and running time.

A. Global Bezier model

Table 2. Global Bezier model for reconstructed linear Bezier curves

Number of Reconstructed Linear Bezier Curves	Global Bezier Model (%)
20	100
40	100
60	98.3
80	96.7
100	96.2
120	94.9
140	93.7

Table 3. Global Bezier model for reconstructed quadratic Bezier curves

Number of Reconstructed Quadratic Bezier Curves	Global Bezier Model (%)
20	100
40	99.9
60	98.6
80	96.6
100	95.7
120	94.8
140	92.3

Table 4. Global Bezier model for reconstructed cubic Bezier curves

Number of Reconstructed Cubic Bezier Curves	Global Bezier Model (%)
20	100
40	98.9
60	98.0
80	96.1
100	95.3
120	92.5
140	90.8

Table 2, 3 and 4 illustrate the global Bezier model configuration for the different type of curve. Based on the table, we successfully reconstruct the correct Bezier model for all type curves. HNN-2SAT are able to reconstruct 20 Bezier curves (Global Bezier model) for all order of the curve without encountering any non-Bezier model (Local Bezier model). The proposed paradigm is able to train the network with the properties of the Bezier via 2SAT and successfully retrieved the correct clause during the reconstruction phase. As the number of reconstructed Bezier curve increased, the

complexity of the network increased. Local Bezier models are expected to emerge because the solution of HNN-2SAT trapped at the suboptimal solution. This is due to spurious minima occurred during the retrieval phase and the network recalled the wrong states for the clauses (Bezier properties) [14, 13]. Consequently, the existence of local Bezier model will create a curve which does not follow the properties of the Bezier curve. Comparatively, cubic Bezier model produced the highest number of local Bezier curve compared to linear and quadratic Bezier curve. This is because cubic Bezier has more basis and control points compared to linear and quadratic Bezier curve. By the same token, the cubic Bezier constraints embedded in HNN-2SAT will be larger and more complex compared to linear and quadratic Bezier curve. Local Bezier curves are expected to increase if the order of the Bezier curve increased. For example, quartic Bezier curves are expected to produce more local Bezier model compared to cubic Bezier. In general, almost 90% of the curve (linear, quadratic and cubic) produced by HNN-2SAT are global Bezier curve.

B. Running Time

Table 5. Running time for reconstructed linear Bezier curves

Number of Reconstructed Linear Bezier Curves	Running Time (s)
20	0.0100
40	0.2600
60	0.5238
80	1.263
100	1.640
120	2.050
140	2.503

Table 6. Running time for reconstructed quadratic Bezier curves

Number of Reconstructed Quadratic Bezier Curves	Running Time (s)
20	0.1500
40	0.2900
60	1.040
80	1.113
100	1.530
120	2.440
140	3.450

Table 7. Running time for reconstructed cubic Bezier curves

Number of Reconstructed Cubic Bezier Curves	Running Time (s)
20	0.2772
40	0.6600
60	1.318
80	1.790
100	2.175
120	3.326
140	4.100

Running time is another measure or indicator to check the effectiveness of our proposed network [15]. Table 5, 6 and 7 depict the running time for our proposed network, HNN-2SAT to reconstruct different Bezier curves correctly. A closer look at the running time indicates that the HNN-2SAT has successfully reconstructed the curves within the stipulated time frame. The running time obtained in this study provides convincing evidence that the time taken for the HNN-2SAT to learn and retrieve the correct curves slightly varies for the different type of Bezier curves. As a matter of fact, the training process consumed most of the running time as our proposed network will be checking the properties of the correct Bezier curves. Strictly speaking, in order to generate the correct Bezier curves, three main properties (constraints) must be fulfilled. Hence, the verification requires particular time to complete the entire process. Theoretically, as the number of Bezier curves model increases, the running time will also increase. In the previous measures, the issue under scrutiny is the existence of local Bezier in any complex model. Table 5, 6 and 7 delineates the significant differences in the running time obtained for the linear, quadratic and cubic Bezier model if the number of curves is ranging from 100 to 140. According to the results, the running time for linear Bezier model is faster than quadratic Bezier, especially when more curves are introduced. Conversely, the cubic Bezier model required more running time. This is due to the cubic Bezier comprised of more basis and control points compared to linear and quadratic Bezier curve. Thus, the training process in order to reconstruct the correct Bezier curve became much slower. There is overwhelming evidence corroborating the notion that more running time required for highly complex Bezier model. A clear evidence is the computation burden to reconstruct higher order curves. Hence, better optimization technique can be applied to improve the running time for highly complex Bezier models. For the time being, HNN-2SAT is still the best network.

VII. CONCLUSION

We have presented our proposed paradigm, namely HNN-2SAT network to reconstruct various Bezier curves model. It had been presented by the computer simulations that our proposed model that incorporated with Hopfield neural networks were able to retrieve and reconstruct correct curves model within the exceptional time frame. Information from the various Bezier models can be stored in the clausal form in Hopfield and most of the retrieved model are exact Bezier models. Hence, the proposed models are supported by the strong agreement of global Bezier model obtained and running time. This early work may have some limitations which will be addressed in future work. Finding satisfied interpretations which have been a building block for clause during simulation can be very complex and detrimental. We have noted that optimized and effective searching technique should be drawn in order to find the correct clausal state. This requires further investigation. For future work,

approximation algorithm such as heuristic methods can be implemented in order to find the correct interpretation for clauses. On the separate note, we can consider variety satisfiability logic such as 3-Satisfiability to represent the clauses in Hopfield network.

REFERENCES

- [1] G. Farin, *A history of curves and surfaces. Handbook of Computer Aided Geometric Design*, Elsevier, 2002.
- [2] A. Riškus, & G. Liutkus, An Improved Algorithm for the Approximation of a Cubic Bezier Curve and its Application for Approximating Quadratic Bezier Curve. *Information Technology and Control*, 42(4), pp. 303-308, 2013.
- [3] B. T. Bertka, An Introduction to Bezier Curves, B-Splines, and Tensor Product Surfaces with History and Applications. *University of California Santa Cruz, May 30th*, 2008.
- [4] R.A. Kowalski, *Logic for Problem Solving*. New York: Elsevier Science Publishing, 1979.
- [5] T. Feder, Network flow and 2-satisfiability, *Algorithmica*, 11(3), pp. 291-319, 1994.
- [6] C. H. Papadimitriou, On selecting a satisfying truth assignment. In *Foundations of Computer Science*, 1991. Proceedings., 32nd Annual Symposium, pp. 163-169, 1991.
- [7] R. Petreschi, & B. Simeone, Experimental comparison of 2-satisfiability algorithms. *Revue française d'automatique, d'informatique et de recherche opérationnelle, Recherche opérationnelle*, 25(3), pp. 241-264, 1991.
- [8] W. Fernandez, Random 2-SAT: Result and Problems, *Theoretical Computer Science*, 265, pp. 131-146, 2001.
- [9] W.A.T. Wan Abdullah, Logic Programming on a Neural Network. *Malaysian Journal of computer Science*, 9 (1), pp. 1-5, 1993.
- [10] J. J. Hopfield, D. W. Tank, Neural computation of decisions in optimization problem, *Biological Cybernetics*, 52, pp. 141-152, 1985.
- [11] S. Haykin, *Neural Networks: A Comprehensive Foundation*, New York: Macmillan College Publishing, 1999.
- [12] G. Pinkas, R. Dechter, Improving energy connectionist energy minimization, *Journal of Artificial Intelligence Research*, 3, pp. 223-15, 1995.
- [13] S. Sathasivam, Upgrading Logic Programming in Hopfield Network, *Sains Malaysiana*, 39, pp. 115-118, 2010.
- [14] S. Sathasivam, Energy Relaxation for Hopfield Network with the New Learning Rule, *International Conference on Power Control and Optimization*, pp. 1-5, 2009.
- [15] S. Sathasivam, P.F. Ng, N. Hamadneh, Developing agent based modelling for reverse analysis method, 6 (22), pp. 4281-4288, 2013.
- [16] S. Sathasivam, Learning in the Recurrent Hopfield Network, *Proceedings of the Fifth International Conference on Computer Graphics, Imaging and Visualisation*, pp. 323-328, 2008.
- [17] R. Rojas, *Neural Networks: A Systematic Introduction*. Berlin: Springer, 1996.
- [18] G. Pinkas, *Propositional non-monotonic reasoning and inconsistency in symmetric neural networks*. Washington University, Department of Computer Science, 1991.
- [19] K. G. Jolly and R. S. Kumar, A Bezier curve based path planning in a multi-agent robot soccer system without violating the acceleration limits, *Robotics and Autonomous System*, 57(1), pp. 23-33, 2009.
- [20] G. Farin, *Curves and Surfaces for Computer-Aided Geometric Design: A Practical Guide*, Elsevier, 2014.
- [21] Patterson and R. Richard, Projective Transformations of the Parameter of a Bernstein-Bezier Curve, *ACM Transactions on Graphics, TOG*, 4(4), pp. 276-290, 1985.
- [22] F. A. Robin, Interactive Interpolation and Approximation by Bezier Polynomials, 15(1), pp. 71-79, 1972.
- [23] Bohm, Wolfgang, G. Farin, J. Kahmann, A survey of curve and surface methods in CAGD: *Computer Aided Geometric Design* 1, 1, pp. 1-60, 1984.
- [24] A. Marc, Linear Combination of Transformation, *ACM Transaction on Graphics, TOG* 21(3), pp. 380-387, 2002.
- [25] S.M. Hu, G. Z. Wang, T.G. Jin, Properties of two types of generalized Ball curves: *Computer-Aided Design* 28, 2, pp. 125-133, 1996.
- [26] C.H. Chu, H. S. Carlo, Developable Bezier patches: properties and design: *Computer-Aided Design* 34, 7, pp. 511-527, 2002.
- [27] J. Gu, Local Search for Satisfiability (SAT) Problem, *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23 pp. 1108-1129, 1993.
- [28] R. Puff, J. Gu, A BDD SAT solver for satisfiability testing: An industrial case study, *Annals of Mathematics and Artificial Intelligence*, 17 (2), pp. 315-337, 1996.
- [29] A. Cimatti, M. Roveri, Bertoli. P, Conformant planning via symbolic model checking and heuristic search, *Artificial Intelligence*, 159 (1), pp. 127-206, 2004.
- [30] A. D. Pwasong and S. Sathasivam, Forecasting Performance of Random Walk with Drift and Feed Forward Neural Network Models, *International Journal of Intelligent System and Application*, vol. 23 pp. 49-56, 2015.
- [31] D. Vilhelm, J. Peter, & W. Magnus, Counting models for 2SAT and 3SAT formulae. *Theoretical Computer Science*, 332 (1), pp. 265-291, 2005.
- [32] B. Sebastian, H. Pascal and H. Steffen, Connectionist model generation: A first-order approach, *Neurocomputing* 71(13) (2008) 2420-2432.
- [33] H. Asgari, Y. S. Kaviani, and A. Mahani, A Systolic Architecture for Hopfield Neural Networks. *Procedia Technology*, 17 (2014) 736-741.
- [34] Y. L. Xou and T. C. Bao, Novel global stability criteria for high-order Hopfield-type neural networks with time-varying delays, *Journal of Mathematical Analysis and Applications* 330 (2007) 144-15.
- [35] K. Iwama, "CNF Satisfiability Test by Counting and Polynomial Average Time," *SIAM, Journal of Computer*, vol. 4, no. 1, pp. 385-391, 1989.

Authors' Profiles



Mohd Shareduwan bin M. Kasihmuddin received his MSc (2014) and BSc(Ed) (2013) from Universiti Sains Malaysia. He is currently pursuing Ph.D degree with School of Mathematical Science, Universiti Sains Malaysia Penang Malaysia. His current research interests include neuro-heuristic method, constrained optimization, neural network and logic programming.



Mohd Asyraf Mansor was born in Sarawak, Malaysia in 1990. He obtained his MSc (2014) and BSc(Ed) (2013) from Universiti Sains Malaysia. He is currently pursuing Ph.D degree at School of Mathematical Science, Universiti Sains Malaysia. His current research interests include evolutionary algorithm, satisfiability problem, neural networks, logic programming and heuristic method especially Artificial Immune System.



Saratha Sathasivam is a lecturer in the School of Mathematical Sciences, Universiti Sains Malaysia. She received her MSc and BSc(Ed) from Universiti Sains Malaysia. She received her Ph.D at Universiti Malaya, Malaysia. Her current research interest are neural networks, agent based modeling and constrained optimization problem.

How to cite this paper: Mohd Shareduwan M. Kasihmuddin, Mohd Asyraf Mansor, Saratha Sathasivam, "Bezier Curves Satisfiability Model in Enhanced Hopfield Network", *International Journal of Intelligent Systems and Applications (IJISA)*, Vol.8, No.12, pp.9-17, 2016. DOI: 10.5815/ijisa.2016.12.02