Title: “Satisfiability Problem Solver by Hysteresis Neural Networks”

Name: Toshiya NAKAGUCHI*, Kenya JIN’NO** and Mamoru TANAKA*

Affiliation: * M.Tanaka Lab.,
Department of Electrical and Electronics Engineering,
Sophia University, Tokyo, 102-8554, Japan.

** Department of Electrical and Electronics Engineering,
Nippon Institute of Technology, Saitama, 345-8501, Japan.

Tel.: +81–3–3238–3878
Fax.: +81–3–3238–3321
Email: nakaguchi@ieee.org

Category: (2.3) Neural network architectures
Satisfiability Problem Solver
by Hysteresis Neural Networks

Toshiya Nakaguchi*, Kenya Jin’no** and Mamoru Tanaka*

* Sophia University, Tokyo, 102-8554, Japan
** Nippon Institute of Technology, Saitama, 345-8501, Japan
nakaguchi@ieee.org

Abstract
In this paper, we propose novel systems to solve “Satisfiability problem.” Our proposing system is one of the artificial neural networks named “Hysteresis Neural Networks.” Though this system has applied to some problems so far, these problems are not so complicated and have few applicable fields. We propose the way to apply this system into the satisfiability problem and give considerations to the phenomena.

1 Introduction

Hysteresis Neural Networks (HNNs) are one of the artificial neural networks and are proposed in 1991[1]. HNNs are continuous time neural networks and have binary outputs with continuous internal states. Since HNNs are described by differential equations, they have possibilities of parallel computing. It is very important that the only stable output vector corresponds with an important information of the problem and vibration phenomena of output vector have a meaningless information. HNNs do not guarantee that the energy function decreases monotonously. However by constructing the system properly, the output vector becomes stable only when it corresponds with a solution of the problem[2].

So far, HNNs have applied into some combinatorial optimization problems, N-Queens Problem[2], Traveling Salesman Problem[3], Box-constraint Problem[4] and so on. Especially, results of N-Queens Problem solver are quite effective considering hardware implementations. However this problem has few applicable fields in the actual environments.

Satisfiability problem (SAT) is fundamental to solve many practical problems in mathematical logic, inference, machine learning, constant satisfaction and VLSI engineering. Theoretically, SAT is a core of a large family of computationally intractable NP-complete problems[5]. Several such NP-complete problems have been identified as central to a variety of areas in computing theory and engineering. Therefore, methods to solve SAT play an important role in the development of computing theory and systems.

SAT has two kinds of basically algorithms, complete ones and incomplete ones. The complete algorithms are that they are able to verify satisfiability as well as unsatisfiability. The incomplete algorithms can only find one random solution for certain CNF formulas and give no answer if the CNF formula is not satisfiable. The incomplete algorithms include the local search algorithms[6] and so on. And also, our algorithm by using HNNs is one of the incomplete algorithms.

In this study, we propose novel algorithms to solve SAT by using the HNN. When the HNN is applied with first order systems, it exhibits wrong solutions called local minimum. To overcome this problem, we propose higher order systems. After giving consideration to phenomena of these systems, we conclude this study.

2 Hysteresis Neural Network (HNN)

An implementation example of HNNs is shown in Figure 1. Applying KCL (Kirchoff’s Current Law) at node $N_A$, we obtain the circuit equations as
between outputs. h(x) is switched from 0 to 1 if x reaches the right threshold 1, and vice versa. We call the region between $0 < x < 1$ in this figure, Hysteresis Region.

The equilibrium points $p_i$ are given by

$$p_i = -\sum_{j=1}^{N} w_{ij} y_j + d_i,$$

(3)

where $p_i$ depends on the output vector. About the system’s stability, we can define as follows,

**Definition:**

the system is said to be **stable**

if $(2p_i - 1)(2y_i - 1) > -1$ $(\forall i)$

the system is said to be **unstable**

if $(2p_i - 1)(2y_i - 1) \leq -1$ $(\exists i)$.

### 3 Satisfiability Problem (SAT)

The satisfiability problem (SAT) has three components:

- A set of $n$ boolean variables: $V_1, V_2, \ldots, V_n$.
- A set of literals. A literal is a variable or a negation of a variable.
- A set of $m$ distinct clauses: $L_1, L_2, \ldots, L_m$. Each clause consists of only literals combined by just logical or ($\lor$) connectors.

The goal of the complete SAT is to determine whether there exists an assignment of truth values to variables that makes the following boolean formula satisfiable:

$$L_1 \land L_2 \land \cdots \land L_m,$$

(5)
where ∨ is a logical or connector. The formulas where Equation (5) equals True are called conjunctive normal form (CNF) formulas. Since our proposing system is one of the incomplete algorithm, it can only find one random solution for certain CNF formulas.

3.1 First order systems

We adapt this problem into the HNN. Variables of SAT, \( V_i \) correspond with output vector of the HNN \( (y_i) \). When \( V_i = \text{True}, y_i = 1 \), and vice versa. The value of clauses using in the HNN are given as follows,

\[
l_i = \begin{cases} 
1 & \text{if } L_i = \text{True}, \\
0 & \text{if } L_i = \text{False}, 
\end{cases}
\]

where \( l_i \) is the one-dimensional function of \( y_i \). We give a constant table \( e_{ij} \) as follows,

\[
e_{ij} = \begin{cases} 
1 & \text{if } j\text{-th clause includes } V_i, \\
0 & \text{if } j\text{-th clause includes } \overline{V_i}, \\
-1 & \text{if } j\text{-th clause includes } \overline{V_i}, 
\end{cases}
\]

Using \( e_{ij} \) and \( l_i \), the dynamics of the HNN can be given as follows,

\[
\lambda_i \frac{d}{dt} x_i = -x_i + \sum_{j=1}^{m} e_{ij}(1 - l_j) + 0.5. 
\]

When CNF formula is satisfied, \( \sum_{j=1}^{m} e_{ij}(1 - l_j) \) becomes 0. Therefore, final term in this equation \((+0.5)\) places the equilibrium point on the center of hysteresis region at satisfying CNF formula.

Equation (8) has only one-dimensional function of \( y_i \). Such systems are called “First order systems.” First order systems are more simple and easier to implement than higher order systems.

3.2 Results of first order systems

First, we provide a simple example problem with 4 variables and 10 clauses as follows,

\[
(\overline{V_1} \lor V_2 \lor V_3) \land (\overline{V_1} \lor V_2 \lor V_3) \\
\land (V_1 \lor \overline{V_2} \lor \overline{V_3}) \land (V_1 \lor V_2 \lor V_3) \\
\land (V_1 \lor \overline{V_2} \lor V_3) \land (V_1 \lor \overline{V_2} \lor V_3) \\
\land (V_1 \lor V_2 \lor \overline{V_3}) \land (V_1 \lor V_2 \lor V_3) \\
\land (V_1 \lor V_2 \lor V_3) \land (V_1 \lor V_2 \lor V_3) = \text{True.}
\]

This CNF formula is randomly generated by the exact 3-SAT model. One of the feasible solutions is \( \{V_1, V_2, V_3, V_4\} = \{\text{False, True, False, True}\} \).

We simulate this problem for 10,000 times with randomly set initial values of \( x_i \) and \( y_i \). The value of time constants(\( \lambda_i \)) are uniform, 1. This simulation result is shown in Table 1.

In this table, “Average of convergence time” means the average of estimated hardware calculating time which can be obtained by calculating \( \tau \) in equations of the HNN. This numerical calculating method is important characteristics of HNNs. Generally, calculating time is very important to estimate and compare with other many algorithms. However, because our convergence time is given by parallel computing on analog hardware, it is impossible to compare with calculating time of other algorithms given by sequential computing on digital hardware.

When the system converges with wrong solutions, they are denoted in “Fail rate” in Table 1. This phenomenon is called a “Local minimum.” As one of very important characteristics of HNNs, when the system is given properly, it does never exhibit local minimum. Then this result means that this system is not properly designed to solve SAT.

3.3 Higher order systems

To overcome the local minimum, we propose higher order systems. Instead of the constant table \( e_{ij} \), new constant tables \( z_{ij} \), \( b_{ij} \) are given as follows,

\[
z_{ij} = \begin{cases} 
1 & \text{if } j\text{-th clause includes } V_i, \\
0 & \text{if } j\text{-th clause includes } \overline{V_i}, 
\end{cases}
\]

\[
b_{ij} = \begin{cases} 
1 & \text{if } j\text{-th clause includes } \overline{V_i}, \\
0 & \text{if } j\text{-th clause includes } V_i, 
\end{cases}
\]

Using \( z_{ij}, b_{ij} \) and \( l_i \), the dynamics of the HNN can be given as follows,

\[
\lambda_i \frac{d}{dt} x_i = -x_i + (1 - y_i) \sum_{j=1}^{m} z_{ij}(1 - l_j) - y_i \sum_{j=1}^{m} b_{ij}(1 - l_j) + 0.5. 
\]

This equation has two-dimensional function of \( y_i \). Such systems are called “Higher order systems.”

3.4 Results of higher order systems

We simulate CNF formula (9) for 10,000 times as shown in Table 2, where \( \lambda_i \) is uniform, 1.

<table>
<thead>
<tr>
<th>Success rate</th>
<th>Fail rate</th>
<th>Average of convergence time</th>
</tr>
</thead>
<tbody>
<tr>
<td>82.1%</td>
<td>17.9%</td>
<td>0.3727</td>
</tr>
</tbody>
</table>

Table 1 : 10,000 times simulations result of CNF formula (9) with first order systems.
As an effect of higher order systems, all values of “Fail rate” become 0%. It means that the system exhibits no local minimum and the system is properly given. Since the system becomes more complicated than first order systems, the average of convergence time is much more than first order systems. This is not so important phenomena comparing with the existence of the local minimum.

Next, we treat more complicated problems which are provided by SATLIB[7]. They provide many kinds of SAT problem to benchmark algorithms. In this study, we use class “UF20-91” of them. Problems of this class have 20 variables and 91 clauses. Against from problem No.1 to No.9, 10,000 times simulations results are shown in Table 3.

This table shows that the system exhibits no local minimum for every problems. However, the system exhibits very important phenomena, “Limit cycles”. The limit cycle means vibration phenomena, and that the system searches feasible solutions permanently. If the problem has no feasible solution called “Unsatisfiable problem”, the limit cycle must be exhibited. But since these problems(No.3, No.5 and No.7) exactly have feasible solutions, the system must never exhibit limit cycles. In the next section, we give considerations of limit cycles.

We focus on the average of convergence time in Table 3. These values depend on the problem and are independent from the rate of limit cycles. For example, problem No.7 which has limit cycles takes a little convergence time. On the other hand, though problem No.9 takes a lot of convergence time, this problem has no limit cycle. In the future, we have to analyze about this phenomena theoretically.

### 4 Limit cycle

In this section, we focus on the limit cycle. This phenomenon means that the system searches feasible solutions permanently and is one of the important undesirable characteristics of HNNs. We calculate the number of limit cycles for 10,000 times simulations against many UF20-91 problems as shown in Figure 3.

This result shows that problems No.22, No.59 and No.93 have quite large number of limit cycles. So, we consider the way to reduce the number of limit cycles against these three problems.

HNNs have important parameter, “Time constant (λi)". These values control only changing speed of internal states and have no efficiency to equilibrium points. In order to reduce the number of limit cycles, the attractor should be scrambled. To scramble the attractor, we add some noise to time constants. Figure 4 shows the results of this examination.

This graph shows that after adding some noise to time constants, the number of limit cycles is reduced. However too much amount of noise increases the number of limit cycle. The best amount of noise depends on the problem, 0.1% noise is the best for No.22, 50% is for No.59 and 5% is for No.93, respectively. In future problem, we have to analyze the relation between the best amount of noise and the problems. And we have to produce novel algorithms to remove limit cycles perfectly against every prob-

<table>
<thead>
<tr>
<th>Problem number</th>
<th>Success rate</th>
<th>Fail rate</th>
<th>Limit cycle</th>
<th>Average of conv. time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100.0%</td>
<td>0%</td>
<td>0%</td>
<td>61.825</td>
</tr>
<tr>
<td>2</td>
<td>100.0%</td>
<td>0%</td>
<td>0%</td>
<td>24.999</td>
</tr>
<tr>
<td>3</td>
<td>96.9%</td>
<td>0%</td>
<td>3.1%</td>
<td>212.089</td>
</tr>
<tr>
<td>4</td>
<td>100.0%</td>
<td>0%</td>
<td>0%</td>
<td>276.624</td>
</tr>
<tr>
<td>5</td>
<td>82.8%</td>
<td>0%</td>
<td>17.2%</td>
<td>241.37</td>
</tr>
<tr>
<td>6</td>
<td>100.0%</td>
<td>0%</td>
<td>0%</td>
<td>165.002</td>
</tr>
<tr>
<td>7</td>
<td>97.9%</td>
<td>0%</td>
<td>2.1%</td>
<td>32.503</td>
</tr>
<tr>
<td>8</td>
<td>100.0%</td>
<td>0%</td>
<td>0%</td>
<td>176.836</td>
</tr>
<tr>
<td>9</td>
<td>100.0%</td>
<td>0%</td>
<td>0%</td>
<td>454.837</td>
</tr>
</tbody>
</table>

Table 3: 10,000 times simulations results of SATLIB UF20-91 problems from No.1 to No.9 with higher order systems.

Figure 4: The relation between an amount of noise adding to time constants and the number of limit cycles against UF20-91 No.22, No.59 and No.93

5 Conclusions

We propose a novel system to solve satisfiability problems. First order systems of HNNs exhibit the local minimum. To overcome this problem, we propose higher order systems of HNNs. However higher order systems exhibit the limit cycles. To reduce the number of limit cycles, we propose the method using time constants. In future problem, we have to produce novel algorithms to remove limit cycles perfectly against every problems.

References