Global convergence and suppression of spurious states of theHopfield neural networks

Abe, Shigeo

(Citation)
IEEE transactions on circuits and systems. I,40(4):246-257

(Issue Date)
1993-04

(Resource Type)
journal article

(Version)
Version of Record

(URL)
https://hdl.handle.net/20.500.14094/90000207

Global Convergence and Suppression of Spurious States of the Hopfield Neural Networks

Shigeo Abe, *Senior Member, IEEE*

***Abstract*— Assuming that the output function of neurons is monotonic and differentiable at any interior point in the output range, we clarify the condition necessary for a vertex of a hypercube to become a local minimum of the Hopfield neural networks and the form of the convergence region to that minimum. Based on this, we derive a method to analyze and suppress spurious states in the networks. Finally we show that all the spurious states of the TSP for the Hopfield original energy function can be suppressed by our method, and we demonstrate validity of the method by computer simulations.**

I. Introduction

SINCE Hopfield showed that neural networks can give the near optimal solution for the traveling salesman problem (TSP), which belongs to a class of the NP complete problems, much effort has been made to apply them to several combinatorial optimization problems [1]–[23]. (Hereafter, we refer to the Hopfield neural networks as the Hopfield model.) In the Hopfield model, the combinatorial optimization problem that minimizes a discrete objective function is converted into a continuous optimization problem that minimizes an energy function, that is, a weighted sum of constraints and an objective function. From it, a set of differential equations, each of which specifies the behavior of a neuron, is derived. Its integration with respect to time decreases, or at least it does not increase the value of the energy function. Thus we can obtain a local minimum solution where the state of each neuron is fired or not fired, namely its output is one or zero. (Hereafter we use a local minimum and a stable solution interchangeably.)

The major problem of the Hopfield model, however, is how to determine the energy function. As for the TSP, several energy functions have been proposed because the Hopfield original formulation does not seem to work properly. Evaluation, however, is based on trial and error because there are no systematic analysis approaches when we obtain an infeasible solution or when the solution is not so good.

Contrary to the heuristic approach, some eigenvalue analyses have been reported to clarify the convergence of the Hopfield model [18]–[22]. In [20], by eigenvalue analysis around an equilibrium point, we clarified the condition that a vertex of a hypercube becomes a local minimum of the Hopfield model, and based on the condition we derived a general algorithm to determine the weights in the energy function, which guarantees that all the feasible solutions become local minima [21]. The algorithm was successfully applied to an LSI module placement problem [23]. In [22], approximating the hyperbolic tangent function by a piecewise linear function, and neglecting the energy term concerning distance, modification of the energy function was proposed to obtain feasible solutions.

Manuscript received October 8, 1991; revised August 5, 1992. This paper was recommended by Associate Editor G. E. Ford.

The author is with Hitachi Research Laboratory, Hitachi, Ltd., 7-1-1 Omika, Hitachi, Japan.

IEEE Log Number 9208186.

Our paper aims at establishing the method of analyzing the convergence of the Hopfield model. In this paper, we first show the condition necessary for a vertex of a hypercube to become a local minimum of the Hopfield model and the form of the convergence region to that minimum. Then, based on the condition, we derive a method to analyze and suppress spurious states in the Hopfield model. Finally, we show that all the spurious states of the TSP can be suppressed if the weights in the energy function proposed by Hopfield are set according to the criteria derived by our method, and we show validity of the method by computer simulations.

II. The Hopfield Model

Consider minimizing the following objective function:

$$E_1 = \frac{1}{2}x^t P\boldsymbol{x} + \boldsymbol{q}^t\boldsymbol{x} \tag{1}$$

under the linear equality constraints:

$$\boldsymbol{r}_i^t\boldsymbol{x} = s_i \qquad \text{for } i = 1, \cdots, l \tag{2}$$

where

$\boldsymbol{x} = (x_1, \cdots, x_n)^t$: the variable vector $x_i = 1$ or $0, i = 1, \cdots, n$;

P: the $n \times n$ symmetric constant matrix;

q: the n-dimensional constant vector; and

$\boldsymbol{r}_i = (r_{1i}, \cdots, r_{ni})^t$: the constant vector.

Now we can formulate the energy function E as follows:

$$E = AE_1 + BE_2 \tag{3}$$

where A and B are weights and E_2 is the energy corresponding to the equality constraints (2), and is derived by adding the square of (2), divided by two, and subtracting the constant terms as follows:

$$E_2 = \frac{1}{2}\boldsymbol{x}^t R\boldsymbol{x} + \boldsymbol{s}^t\boldsymbol{x} \tag{4}$$

where

$$R = \sum_{i=1}^{l} \boldsymbol{r}_i\boldsymbol{r}_i^t; \tag{5}$$

$$s = -\sum_{i=1}^{l} s_i \boldsymbol{r}_i. \tag{6}$$

Now let

$$T = AP + BR; \quad \text{and}$$
$$\boldsymbol{b} = A\boldsymbol{q} + B\boldsymbol{s} \text{ where } T \text{ is symmetric.} \tag{7}$$

Thus the energy E given by (3) becomes

$$E = \frac{1}{2}\boldsymbol{x}^t T\boldsymbol{x} + \boldsymbol{b}^t\boldsymbol{x}. \tag{8}$$

Then the Hopfield model for the energy function E is given by introducing the internal variable vector $\boldsymbol{u}$, and extending the range of $x_i (i = 1, \cdots, n)$ to [0, 1]:

$$x_i = f(u_i) \qquad \text{for } 1 \geq x_i \geq 0 \tag{9}$$
$$\frac{d\boldsymbol{u}}{dt} = -\frac{\partial E}{\partial \boldsymbol{x}} = -T\boldsymbol{x} - b \tag{10}$$

where $f(u_i) = 1/2(1 + \tanh u_i)$: a hyperbolic tangent function; and

$$\boldsymbol{u} = (u_1, \cdots, u_n)^t, \quad \infty > u_i > -\infty, \quad i = 1, \cdots, n.$$

It is easy to see that $dE/dt \leq 0$ holds [19]. Thus integrating (9) and (10) for an arbitrary initial value of $\boldsymbol{u}$, we obtain a local minimum solution in the sense that the energy function E is locally minimized.

III. Convergence Region of Local Minima

Local Minima

The square term $T_{ii}x_i^2/2$ in the energy function (8) can be replaced by the linear term $T_{ii}x_i/2$ since the energy at any vertex in the n-dimensional hypercube does not change. Thus we delete the square terms as follows [3], [19], [20]:

$$b_i \leftarrow b_i + T_{ii}/2 \quad \text{and} \quad T_{ii} \leftarrow 0 \qquad \text{for } i = 1, \cdots, n.$$

Therefore, in the following study, we assume that all T_{ii} are zero.

In the original Hopfield model, $f(u_i)$ in (9) is assumed to be a hyperbolic tangent function. But we need not restrict $f(u_i)$ to the hyperbolic tangent function so long as it is monotonic and differentiable for $0 < x_i < 1$ and it saturates to 1 or 0 as u_i approaches $\pm\infty$, for example a piecewise-linear function proposed in [22] (see Fig. 1). Using the piecewise-linear function, (9) and (10) become a linear system that operates on the hypercube as follows [24]:

$$\frac{d\boldsymbol{x}}{dt} = -T\boldsymbol{x} - \boldsymbol{b}, \; 0 \leq x_i \leq 1, \qquad \text{for } i = 1, \cdots, n. \tag{11}$$

Therefore, in the following we assume that $f(u_i)$ is monotonic and differentiable for $0 < x_i < 1$ and it saturates to 1 or 0 as u_i approaches $\pm\infty$.

Now let the hypercube and the interior of the hypercube be defined by

$$H = \{\boldsymbol{x} \mid 0 \leq x_i \leq 1 \text{ for } i = 1, \cdots, n\} \tag{12}$$

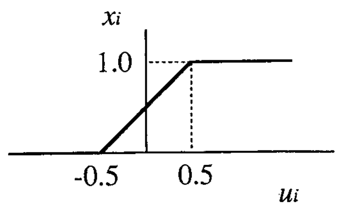


Fig. 1. Piecewise-linear function.

and

$$H^- = \{\boldsymbol{x} \mid 0 < x_i < 1 \text{ for } i = 1, \cdots, n\} \tag{13}$$

respectively. And for the vertex $\boldsymbol{c} = (c_1, \cdots, c_n)$ in H we define n adjacent vertexes $\boldsymbol{c}(i) = (c_1, \cdots, c_{i-1}, 1 - c_i, c_{i+1}, \cdots, c_n)$ where $c_i = 1$, or 0 for $i = 1, \cdots, n$.

The solutions of (9) and (10) (local minima) are stable equilibrium points among all the equilibrium points of (9) and (10) given by

1. vertex $\boldsymbol{c} = (c_1, \cdots, c_n)$, $c_i = 1$, or 0 for $i = 1, \cdots, n$;
2. the solution of $T\boldsymbol{x} + \boldsymbol{b} = 0$; and
3. the combination of 1 and 2

where for the equilibrium points 2 and 3, if $\boldsymbol{x}$ satisfies $1 \geq x_i \geq 0$, for $i = 1, \cdots, n$. The equilibrium points 3 are on the surface of H. In the following, stability of the above equilibrium points is discussed.

Stability of Vertexes

To investigate the stability of vertexes, we first consider the behavior of the solution for (9) and (10) on the edge between $\boldsymbol{c}$ and $\boldsymbol{c}(i)$, namely, at

$$\boldsymbol{x} = \gamma\boldsymbol{c} + (1 - \gamma)\boldsymbol{c}(i) \qquad \text{for } 0 \leq \gamma \leq 1. \tag{14}$$

From (10) and (14),

$$\frac{du_j}{dt} = 0 \qquad j \neq i, \qquad j = 1, \cdots, n, \quad \text{and}$$
$$\frac{du_i}{dt} = -T_i\boldsymbol{x} - b_i \tag{15}$$

hold where T_i is the ith row vector of T. Also from $T_{ii} = 0$,

$$T_i\boldsymbol{c} + b_i = T_i\boldsymbol{c}(i) + b_i = T_i\boldsymbol{x} + b_i \tag{16}$$

holds. This means that the velocity of u_i is the same for $0 < \gamma < 1$. The difference in the piecewise-linear function and the hyperbolic tangent function is, therefore, that for the former x_i moves to 0 or 1 with the same speed, but for the latter x_i slows down as it approaches 0 or 1. Equation (16) also means that hyperplane $T_i\boldsymbol{x} + b_i = 0$ is parallel to the x_i axis.

It is easy to see that

$$E_{\boldsymbol{c}} - E_{\boldsymbol{c}(i)} = (2c_i - 1)(T_i\boldsymbol{c}' + b_i) \tag{17}$$

holds where E_c is the energy E evaluated at $\boldsymbol{c}$ and

$$\boldsymbol{c}' = \left(c_1, \cdots, c_{i-1}, \frac{1}{2}, c_{i+1}, \cdots, c_n\right).$$

Therefore, (i) for $E_{\boldsymbol{c}} = E_{\boldsymbol{c}(i)}$, all the values of the right-hand side of (15) are zero, and $\boldsymbol{x}$ given by (14) is an equilibrium point of (9) and (10), and (ii) for $E_{\boldsymbol{c}} < E_{\boldsymbol{c}(i)}$, $\boldsymbol{x}$ moves to $\boldsymbol{c}(e)$ and for $E_{\boldsymbol{c}} > E_{\boldsymbol{c}(i)}$, $\boldsymbol{x}$ moves to $\boldsymbol{c}(i)$.

Here we say vertexes $\boldsymbol{c}$ and $\boldsymbol{c}(i)$ are *degenerate* when energies $E_{\boldsymbol{c}}$ and $E_{\boldsymbol{c}(i)}$ are the same. This is equivalent to the condition that hyperplane $T_i\boldsymbol{x} + b_i = 0$ contains vertexes $\boldsymbol{c}$ and $\boldsymbol{c}(i)$.

Now we investigate the behavior when $\boldsymbol{x}$ is in the interior of the hypercube H^-. Let H^- be divided into three parts by the hyperplane $T_i\boldsymbol{x} + b_i = 0$ as follows:

$$D^+(i) = \{\boldsymbol{x} \mid 0 < x_j < 1, j = 1, \cdots, n, T_i\boldsymbol{x} + b_i > 0\},$$
$$D^-(i) = \{\boldsymbol{x} \mid 0 < x_j < 1, j = 1, \cdots, n, T_i\boldsymbol{x} + b_i < 0\}, \quad \text{and}$$
$$D^0(i) = \{\boldsymbol{x} \mid 0 < x_j < 1, j = 1, \cdots, n, T_i\boldsymbol{x} + b_i = 0\}. \quad (18)$$

For $\boldsymbol{x} \in D^+(i)$, as time elapses component x_i always decreases from (9) and (10) so long as $\boldsymbol{x}$ is in domain $D^+(i)$. Likewise for $\boldsymbol{x} \in D^-(i)$, x_i increases and for $\boldsymbol{x} \in D^0(i)$, x_i does not change. If either $D^+(i)$ or $D^-(i)$ is empty, x_i increases or decreases for $0 < x_i < 1$. Thus x_i converges to the same value 0 or 1 for any initial value satisfying $0 < x_i < 1$.

Then for each vertex $\boldsymbol{c}$ in hypercube H we define domain $D_{\boldsymbol{c}}$ as follows:

$$D_c = \bigcap_{i=1,\cdots,n} Z(i) \quad (19)$$

where

$$Z(i) = \begin{cases} D^+(i) & \text{for } T_i\boldsymbol{c} + b_i > 0 \\ D^-(i) & \text{for } T_i\boldsymbol{c} + b_i < 0\,. \\ H^- & \text{for } T_i\boldsymbol{c} + b_i = 0 \end{cases} \quad (20)$$

If $\boldsymbol{c}$ and $\boldsymbol{c}(i)$ are degenerate, namely $T_i\boldsymbol{c}+b_i = 0$, any point on the edge between $\boldsymbol{c}$ and $\boldsymbol{c}(i)$ given by (14) may become stable. Therefore, to confine our discussion to the nondegenerate component of $\boldsymbol{c}$, we set $Z(i) = H^-$ for $T_i\boldsymbol{c}+b_i = 0$. According to the definition it is easy to see that for any $\boldsymbol{x} \in D_{\boldsymbol{c}}$, component x_i, in which $\boldsymbol{c}$ and $\boldsymbol{c}(i)$ are not degenerate, moves monotonically so long as $\boldsymbol{x}$ remains in $D_{\boldsymbol{c}}$. Now we classify vertex $\boldsymbol{c}$ as follows.

i) $E_{\boldsymbol{c}} \leq E_{\boldsymbol{c}(i)}$ holds for $i = 1, \cdots, n$ and the strict inequality holds for at least one i.

Component x_i, in which $\boldsymbol{c}$ and $\boldsymbol{c}(i)$ are not degenerate, moves monotonically to c_i so long as $\boldsymbol{x}$ remains in $D_{\boldsymbol{c}}$. If the strict inequalities hold for all i, i.e., vertex $\boldsymbol{c}$ and all $\boldsymbol{c}(i)$ $i = 1, \cdots, n$ are not degenerate, then we say the vertex is *strongly stable* and the domain is *strongly convergent*. Otherwise, we say that the vertex is *weakly stable* and the domain is *weakly convergent*.

ii) $E_{\boldsymbol{c}} \geq E_{\boldsymbol{c}(i)}$ holds for $i = 1, \cdots, n$.

So long as $\boldsymbol{x}$ remains in $D_{\boldsymbol{c}}$, component x_i, in which $\boldsymbol{c}$ and $\boldsymbol{c}(i)$ are not degenerate, monotonically goes away from c_i as long as $\boldsymbol{x}$ remains in $D_{\boldsymbol{c}}$. We say that the vertex is *unstable* and the domain is *divergent*.

iii) $E_{\boldsymbol{c}} \geq E_{\boldsymbol{c}(i)}$ or $E_{\boldsymbol{c}} \leq E_{\boldsymbol{c}(i)}$ holds for $i = 1, \cdots, n$ where at least the strict inequality holds for one i for each type of inequalities.

So long as $\boldsymbol{x}$ remains in $D_{\boldsymbol{c}}$, component x_i, in which $E_{\boldsymbol{c}} < E_{\boldsymbol{c}(i)}$ holds, monotonically moves to c_i and component x_i, in which $E_{\boldsymbol{c}} > E_{\boldsymbol{c}(i)}$ holds, monotonically goes away from c_i so long as $\boldsymbol{x}$ remains in $D_{\boldsymbol{c}}$. We say that the vertex is a *saddle* and the domain is *transit*.

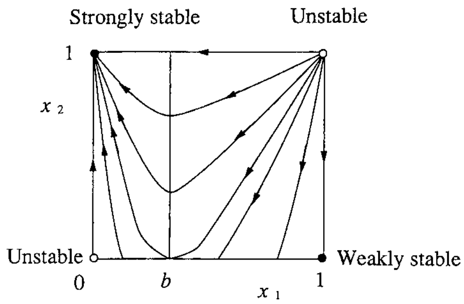


Fig. 2. Loci of a two-neuron network.

Now for a strongly stable vertex $\boldsymbol{c}$ we define convergent domain $D'_{\boldsymbol{c}}$ as follows:

$$D'_c = \{\boldsymbol{x} \mid \boldsymbol{x} \in D_c \text{ and } (x_1, \cdots, x_{i-1}, x'_i, x_{i+1}, \cdots, x_n) \in D_{\boldsymbol{c}}$$
$$x'_i = \gamma c_i + (1-\gamma)x_i, 0 \leq \gamma < 1 \text{ for } i = 1, \cdots, n\}. \quad (21)$$

The condition that $(x_1, \cdots, x_{i-1} x'_i, x_{i+1}, \cdots, x_n) \in D_{\boldsymbol{c}}$ means that a line segment connecting $\boldsymbol{x}$ and $(x_1, \cdots, x_{i-1}, c_i, x_{i+1}, \cdots, x_n)$ excluding the latter point, which is parallel to the x_i axis, does not intersect with hyperplanes $T_i\boldsymbol{x} + b_i = 0 (i = 1, \cdots, n)$. Also it is easy to see that $D'_{\boldsymbol{c}}$ is a connected domain. Therefore, so long as $\boldsymbol{x}$ is in $D'_{\boldsymbol{c}}$, $\boldsymbol{x}$ does not go back to $H^- - D'_{\boldsymbol{c}}$.

When $n = 2$, convergent domain $D'_{\boldsymbol{c}}$ coincides with domain $D_{\boldsymbol{c}}$ as will be seen later (see Fig. 2). When n is larger than two, a case exists that convergent domain $D'_{\boldsymbol{c}}$ is included in domain $D_{\boldsymbol{c}}$. But it is nonempty since hyperplanes $T_i\boldsymbol{x} + b_i = 0 (i = 1, \cdots, n)$ do not include a strongly stable vertex $\boldsymbol{c}$.

From the above definitions it is easy to see that the following theorem holds.

Theorem 1: Let vertex $\boldsymbol{c}$ be strongly stable. Then for any $\boldsymbol{x} \in D'_{\boldsymbol{c}}$, $\boldsymbol{x}$ converges monotonically to $\boldsymbol{c}$.

Proof: Since $E_{\boldsymbol{c}} < E_{\boldsymbol{c}(i)}$ and also from (17), $\boldsymbol{c}$ does not satisfy $T_i\boldsymbol{x}+b_i = 0$ for $i = 1, \cdots, n$. Thus for $\boldsymbol{x} \in D'_{\boldsymbol{c}}$, $\boldsymbol{x}$ is corrected towards $\boldsymbol{c}$. Since hyperplane $T_i\boldsymbol{x}+b_i = 0$ is parallel to the x_i axis, the hyperplane does not exist in the corrected direction. Also since $(x_1, \cdots, x_{j-1}, x'_j, x_{j-1}, \cdots, x_n) \in D_{\boldsymbol{c}}$ where $x'_j = \gamma c_j + (1-\gamma)x_j$, $0 \leq \gamma < 1$ for $j = 1, \cdots, n$, hyperplanes $T_j\boldsymbol{x}+b_j = 0$ for $j \neq i$, $j = 1, \cdots, n$ do not exist in $D'_{\boldsymbol{c}}$ in the corrected direction. Thus while the corrected $\boldsymbol{x}$ remains in $D_{\boldsymbol{c}}$, $\boldsymbol{x}$ monotonically converges to $\boldsymbol{c}$. When $\boldsymbol{x}$ reaches the surface of H, namely $\boldsymbol{x}$ is in the closure of $D'_{\boldsymbol{c}}$, it is clear, from the definition of the convergent domain, that $\boldsymbol{x}$ is still in the convergent domain on that surface. Thus the theorem holds (end of proof).

In [20] we proved a similar theorem linearizing (9) and (10) around a vertex and investigating the eigenvalues (see Appendix A). The above theorem not only validates the previous approach but also clarifies global convergence to a local minimum. Theorem 1 also means that two adjacent vertexes having different energies cannot be stable solutions of the Hopfield model at the same time.

The following corollary is evident.

Corollary 1: Let vertex $\boldsymbol{c}$ be weakly or strongly stable. Then for $E_{\boldsymbol{c}(i)} < E_{\boldsymbol{c}(i)}$, $x_i(\boldsymbol{x} \in D_{\boldsymbol{c}})$ monotonically converges to c_i so long as $\boldsymbol{x}$ remains in $D_{\boldsymbol{c}}$.

If vertex $\boldsymbol{c}$ is weakly stable, there is a possibility of converging to an analog value, even if the diagonal elements of T are set to zero. Fig. 2 shows an example of a two-neuron network with

$$T\boldsymbol{x} + \boldsymbol{b} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 0 \\ -b \end{bmatrix} = \begin{bmatrix} x_2 \\ x_1 - b \end{bmatrix}$$

where $0 < b < 1$. Vertexes $(0,0)$ and $(1,0)$ are degenerate and

$$D_{(0,\,0)} = D_{(0,\,1)},\ D_{(1,\,0)} = D_{(1,\,1)}.$$

Vertexes $(0,0)$ and $(1,1)$ are unstable, vertex $(0,1)$ is strongly stable, and vertex $(1,0)$ is weakly stable. Clearly convergent domain $D'_{(0,\,1)}$ coincides with $D_{(0,\,1)}$. As seen from the figure, points on the edge connecting $(0,0)$ and $(1,0)$ are equilibrium points and become stable when $b < x_1 \leq 1$.

Stability of Non-Vertex Equilibrium Points

As for the stability of the equilibrium point which satisfies $\boldsymbol{Tx} + \boldsymbol{b} = 0$ the following theorem holds.

Theorem 2: Let $\boldsymbol{s}$ be the solution of $\boldsymbol{Tx}+\boldsymbol{b} = 0$ and satisfy $0 < s_i < 1$ for $i = 1, \cdots, n$. Then $\boldsymbol{s}$ is an unstable equilibrium point.

Proof: Assume that there is a strong stable vertex $\boldsymbol{c}$ and take $\boldsymbol{x} \in D_{\boldsymbol{c}}$ in the neighborhood of $\boldsymbol{s}$. According to the definition of $D_{\boldsymbol{c}}$, $\boldsymbol{x}$ moves towards $\boldsymbol{c}$, namely goes away from $\boldsymbol{s}$. Thus $\boldsymbol{s}$ is unstable.

If there is no strongly stable vertex, we can at least take a weakly stable vertex $\boldsymbol{c}$ in which for some i, $E_{\boldsymbol{c}} < E_{\boldsymbol{c}(i)}$ holds. To prove this, suppose the energies of all the vertices are the same. Then for $\boldsymbol{c} = (0, \cdots, 0)$, $E_{\boldsymbol{c}} = 0$. Thus for $\boldsymbol{c}(i)$, $E_{\boldsymbol{c}(i)} = b_i = 0$. Also for $\boldsymbol{c}(i,\,j)$, in which only ith and jth elements are 1, $E_{\boldsymbol{c}(i,\,j)} = T_{ij} = 0$. Thus the Hopfield model reduces to a trivial case where all the elements of T_{ij} and b_i are 0.

For a weakly stable vertex $\boldsymbol{c}$, take $\boldsymbol{x} \in D_{\boldsymbol{c}}$ in the neighborhood of $\boldsymbol{s}$. Thus x_i, in which $E_{\boldsymbol{c}} < E_{\boldsymbol{c}(i)}$ holds, moves towards c_i, namely goes away form s_i. Thus $\boldsymbol{s}$ is unstable (end of proof).

This theorem is a sophisticated version of [20, Theorem 7], which only states that the energy $E_{\boldsymbol{s}}$ is equal to or greater than $E_{\boldsymbol{c}}$.

From Theorem 2, if for a strongly stable vertex, $D'_{\boldsymbol{c}} = D_{\boldsymbol{c}}$ holds, this vertex can be reached from the neighborhood of $\boldsymbol{s}$ if $0 < s_i < 1$ for $i = 1, \cdots, n$ holds. Even if $D'_{\boldsymbol{c}} \neq D_{\boldsymbol{c}}$, it is conjectured that vertex $\boldsymbol{c}$ can be reached from the neighborhood of $\boldsymbol{s}$. Namely,

Conjecture: Let vertex $\boldsymbol{c}$ be strongly stable. Then for any $\boldsymbol{x} \in D_{\boldsymbol{c}}$, $\boldsymbol{x}$ converges to $\boldsymbol{c}$.

If this holds, it seems to be a good policy to select the initial value of $\boldsymbol{x}$ in the neighborhood of $\boldsymbol{s}$. Or if we do not want to solve $\boldsymbol{Tx}+\boldsymbol{b} = 0$, at least we should not select the initial value in a strongly convergent domain. The second best choice is in a divergent domain.

The remaining equilibrium points are on the surface of H. Let S_k include k integers selected from $\{1, \cdots, n\}$. Then the equilibrium points on the surface are the solution of

$$\begin{aligned} T_i\boldsymbol{x} + b_i &= 0 \text{ for } i \in S_k \\ x_i &= 0 \text{ or } 1 \text{ for } i \notin S_k, \qquad i = 1, \cdots, n \end{aligned} \tag{22}$$

which satisfies $0 < x_i < 1$ for $i \in S_k$, $k = 1, \cdots, n$. We can reduce the dimension of (9) and (10) from n to k by substituting variables x_j fixed to 0 or 1 to the ith ($i \in S_k$) component of (10) and deleting the remaining $n - k$ components corresponding to the fixed variables. Now for the reduced system, the diagonal elements of the coefficient matrix are still zero. Thus Theorem 2 holds for the reduced system, and the non-vertex equilibrium point given by (22) is an unstable equilibrium point. Thus the following corollary holds.

Corollary 2: If solution $\boldsymbol{s}$ of (22) satisfies $0 < s_i < 1$ for $i = 1, \cdots, \boldsymbol{n}$, $\boldsymbol{s}$ is an unstable equilibrium point.

IV. Suppression of Spurious States

According to the discussion in the previous section, the non-vertex equilibrium points in H are unstable. Thus they cannot be spurious states (infeasible solutions). Then all we need is to suppress spurious states at vertexes.

Accordingly, we only need to suppress infeasible solutions at the corner of the n-dimensional space H. Using Theorem 1 we can: i) suppress spurious states adjacent to feasible solutions; and ii) suppress spurious states not adjacent to feasible solutions.

Step i) is relatively easy. From Theorem 1 if all the feasible solutions are strongly stable, there are no spurious states adjacent to feasible solutions. Take an arbitrary, feasible solution. The solution can be strongly stable if its energy is the smallest among those of the solutions that are adjacent to it. Thus all we have to do is to select weights A and B in the energy function (8) so that for any feasible solution $\boldsymbol{c}$, its energy becomes the smallest among those of the solutions which are generated by reversing the state of any one neuron in the feasible solution from 1 to 0 or 0 to 1. Namely, weights are selected so that

$$E_{\boldsymbol{c}(i)} > E_{\boldsymbol{c}}$$

holds for an arbitrary vertex $\boldsymbol{c}$, which satisfies the constraints, and corresponding adjacent vertexes $\boldsymbol{c}(i)(i = 1, \cdots, n)$.

Step ii) is not so easy since we need to investigate, using Theorem 1, what kind of vertex becomes a local minimum for a given energy function. If spurious states are found that are not adjacent to feasible solutions, we check whether they can be suppressed by changing weights. If they cannot but should be, we need to add suitable constraints to suppress those types of spurious states.

V. Traveling Salesman Problem

Problem Formulation

Original formulation: The traveling salesman problem is to travel to n cities within the minimum distance with the constraints that all the cities must be visited only once.

Hopfield assigned n neurons for each city and stated that if the output of the ith neuron of the city is one, the city is visited at the ith order. He defined the energy function as follows [3]:

$$E = \frac{A}{2}\sum_x\sum_i\sum_{j\neq i} V_{xi}V_{xj} + \frac{B}{2}\sum_i\sum_x\sum_{y\neq x} V_{xi}V_{yi} + \frac{C}{2}\left(\sum_x\sum_i V_{xi} - n\right)^2 + \frac{D}{2}\sum_x\sum_{y\neq x}\sum_i d_{xy}V_{xi}(V_{y,\,i+1} + V_{y,\,i-1}) \tag{23}$$

where

V_{xi}	ith neuron for city x, and $0 \le V_{xi} \le 1$,
d_{xy}	distance between city x and city y,
A, B, C	weights and subscripts $i-1$ and $i+1$ are calculated by modulo n.

We call this the original formulation. The first term of the energy function specifies that for city x, the number of subscripts i that satisfy $V_{xi} = 1$ is at most one. The second term specifies that for subscript i, the number of cities x that satisfy $V_{xi} = 1$ is at most one. And the third term specifies that the total number of neurons that fire is n. The first three terms are constraints.

Now consider determining weights to suppress spurious states (invalid tours). If more than n neurons fire, the energy always increases. Thus spurious states only occur when n neurons or less fire.

To suppress spurious states adjacent to feasible solutions, we assume an arbitrary feasible solution (valid tour), and change the state of any one neuron from one to zero. (We need not consider changing one neuron from 0 to 1 since it will fire $n+1$ neurons.) With this change the values of the first two terms in (23) are still zero since that does not violate the corresponding constraints. But the value of the third term increases by

$$\frac{C}{2}$$

and the value of the fourth term decreases by

$$D(d_{xy} + d_{xz}).$$

Note that in the summation at cities x, y, and z, distances $d_{xy} + d_{xz}$, d_{yx}, and d_{zx} are added. Thus from Theorem 1, if

$$C > 2D \max_{x,\,y,\,z}(d_{xy} + d_{xz}) \tag{24}$$

holds and weights A and B are positive, all the feasible solutions become strongly stable. In order for the constraints to work equally we set

$$A = B = C. \tag{25}$$

The procedure to derive the condition to suppress spurious states not adjacent to feasible solutions is not so straightforward. Also, as will be seen from the simulation, they can be suppressed so long as (24) holds. So we leave detailed discussions to Appendix B. From the procedure, the spurious states not adjacent to feasible solutions can be suppressed if

$$C > 2D \max_x \sum_y^{(4)} d_{xy} \tag{26}$$

holds where the summation is added for the maximum to fourth maximum distances from city x. Therefore, from (24) if (26) holds all the spurious states are suppressed and stable states are feasible solutions.

Modified formulation: After Hopfield's formulation several energy functions have been proposed. Among them we consider the following energy function which is often used:

$$E = \frac{A}{2}\sum_x\left(\sum_i V_{xi} - 1\right)^2 + \frac{B}{2}\sum_i\left(\sum_y V_{yi} - 1\right)^2 + \frac{D}{2}\sum_x\sum_{y\neq x}\sum_i d_{xy}V_{xi}(V_{y,\,i+1} + V_{y,\,i-1}). \tag{27}$$

We call this the modified formulation. The first and the second terms impose the constraint that $V_{xi} = 1$ holds for one x or i while varying them from 1 to n.

First consider suppressing spurious states adjacent to feasible solutions. Assume an arbitrary feasible solution. If the state of any one neuron changes from one to zero, the energies corresponding to the first and the second constraints increase by $A/2$ and $B/2$. The energy of the third term decreases by

$$D(d_{xy} + d_{xz}).$$

Thus in order for all the feasible solutions to be strongly stable, the following inequality must hold:

$$\frac{1}{2}(A + B) > D \max_{x,\,y,\,z}(d_{xy} + d_{xz}). \tag{28}$$

For the two constraints to work equally, we set

$$A = B. \tag{29}$$

Thus (28) becomes

$$A > D \max_{x,\,y,\,z}(d_{xy} + d_{xz}). \tag{30}$$

When n is odd, the spurious states not adjacent to feasible solutions can be suppressed, if

$$A > D \max_x \sum_y^{(4)} d_{xy} \tag{31}$$

holds (see Appendix B). With the modified formulation all the spurious states cannot be suppressed if n is even. We can suppress them by constraining not more than one neuron to fire in each column. One example is to add the second constraint in (23).

Initial Value Selection

Since we do not have *a priori* knowledge about the optimal solution, we need to set the initial value of V_{xi} (or $\boldsymbol{x}$) at such a point as will lead to the optimal solution. For the original and modified formulations, the equilibrium point $\boldsymbol{s}$ exists in H^- as shown in Theorem A2 in Appendix C. Therefore, from Theorem 2 and the conjecture, if we set the initial value around $\boldsymbol{s}$, the probability to obtain the optimal solution may be increased compared to a random initial value. We can set the initial value around $\boldsymbol{s}$ without solving $T\boldsymbol{x} + \boldsymbol{b} = 0$. If we set the initial values at the center of the hypercube, or more generally if we set the initial values as

$$V_{xi} = 0.5 + w_x \qquad \text{for } i = 1, \cdots, n \tag{32}$$

where w_x takes a value in $(-0.5, 0.5)$, V_{xi} for $i = 1, \cdots, n$ behave in the same way, and the network is reduced from n^2 dimensions to n dimensions [3], [20]. Since the diagonal elements of the reduced matrix of T become large, the interior equilibrium point $\boldsymbol{s}$ becomes the only stable point in the hypercube as shown in Appendix C. Thus by selecting initial values by (32) we obtain the singular point $\boldsymbol{s}$. Therefore, setting the initial values around the equilibrium point $\boldsymbol{s}$ without solving $T\boldsymbol{x} + \boldsymbol{b} = 0$ can be achieved by adding noise to the initial values given by (32). Thus in our following study we set the initial values as follows:

$$V_{xi} = 0.5 + \alpha \times \text{RAND} \tag{33}$$

where α is a parameter and RAND is a uniformly distributed random variable ranging from -0.5 to 0.5.

Numerical Calculations

The purpose of this section is to demonstrate the validity of the criteria of the weights (24), (26), (30), and (31), and to investigate the quality of the solution of the Hopfield model. An accurate way to demonstrate the validity is to enumerate all the spurious states. But this is impractical since it would take roughly nine days in CPU time to enumerate them only for the six-city TSP using a 30 MIPS mainframe computer. Therefore, we repeatedly solved (9) and (10) setting the initial values by the uniformly distributed random variable in [0, 1]. To investigate the quality of the solutions we set the initial values around the center of the hypercube as discussed previously.

In the study we used the 10- and 30-city TSP's in [3]. Instead of the hyperbolic tangent function we used the piecewise linear function shown in Fig. 1. And we applied the Euler method to (11) to obtain combinatorial solutions, namely,

$$x_i' = x_i - \Delta t\{T_i\boldsymbol{x} + b_i\}$$

$$\text{new } x_i = \begin{cases} x_i' & \text{for } 0 < x_i' < 1 \\ 1 & \text{for } x_i' \geq 1 \\ 0 & \text{for } x_{i'} \leq 0 \end{cases} \tag{34}$$

$$\Delta t = \frac{\text{const}}{\max_i\left(\frac{1}{2}T\boldsymbol{1} + \boldsymbol{b}\right)_i} \tag{35}$$

where Δt is a time interval of integration; $\boldsymbol{1}$ denotes an n-dimensional vector whose elements are 1; j denotes the jth row of a vector; and const is selected as 0.3 in our study.

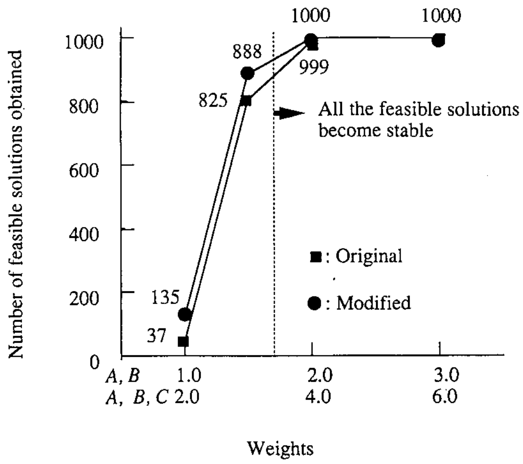


Fig. 3. Number of feasible solutions of 10-city TSP with random initial values in 0 to 1 (1000 trials).

Equation (35) limits the correlation at $x_i = 0.5$ to at most the value of the const so that the solution does not converge to the origin by over-correction of $\boldsymbol{x}$. The calculation of (34) was terminated when all the corrections of the variables x_i were within the specified limit.

Ten-city TSP: Fig. 3 shows the number of feasible solutions obtained for 1000 trials with the uniformly distributed random initial values ranging from 0 to 1, and changing weights. The conditions (24) and (30), that all the feasible solutions become strongly stable, are given by

$$A = B = C > 3.3 \tag{36}$$

and

$$A = B > 1.65 \tag{37}$$

and those for (26) and (31) are

$$A = B = C > 6.2 \tag{38}$$

and

$$A = B > 3.1 \tag{39}$$

respectively. So long as the weights satisfy (36) and (37), feasible solutions are almost always obtained for both formulations. The one infeasible solution for the original formulation with weights $A = B = C = 4.0$ is shown in Fig. 4, which is a special case of Fig. 12(c) and becomes unstable when weights $A = B = C = 6.0$. Also, the number of feasible solutions decreases drastically as the weights violate these conditions.

Fig. 5 shows the average tour length among the feasible solutions corresponding to Fig. 3. It is seen that the quality of solutions is improved as the weights become smaller and that the modified formulation performs better.

Figs. 6 and 7 show the number of feasible solutions obtained for 1000 trials, setting the initial values around the center with weights $A = B = C = 4.0$ for the original formulation and $A = B = 2.0$ for the modified formulation. The average tour length for Figs. 6 and 7 is shown in Fig. 8. The one infeasible solution for $\alpha = 10^{-1}$ in Fig. 6 is the same type as in Fig. 12(c).

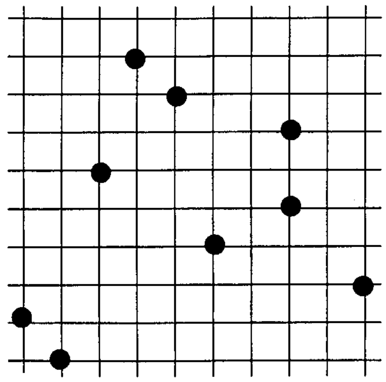


Fig. 4. Spurious state of the 10-city TSP for the original formulation (weights $A = B = C = 4.0$).

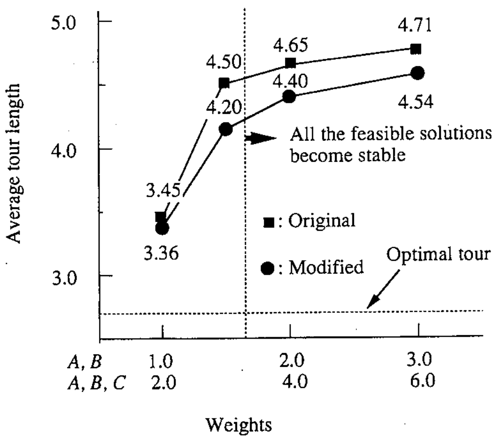


Fig. 5. Average tour length for 10-city TSP with uniform initial values in 0 to 1 (1000 trials).

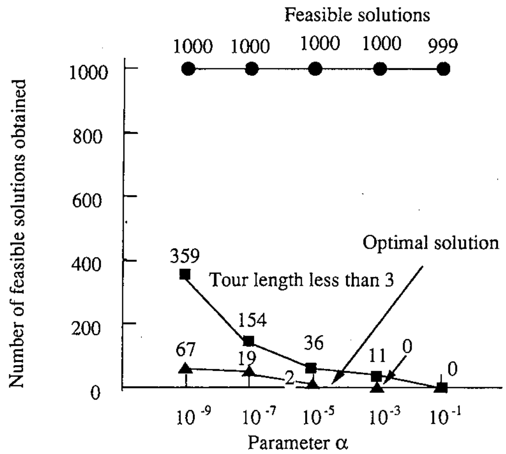


Fig. 6. Number of feasible solutions of 10-city TSP by the original formulation (1000 trials, initial values around the center, weight $A = B = C = 4.0$).

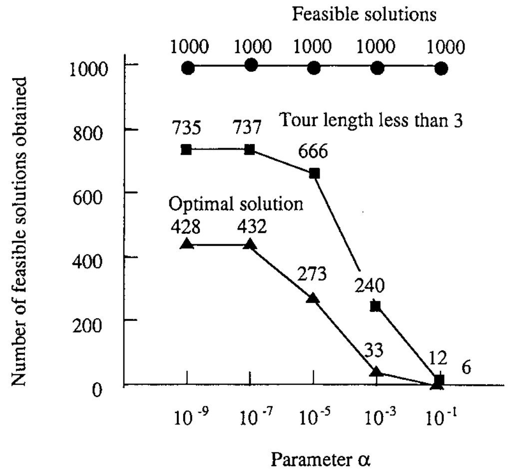


Fig. 7. Number of feasible solutions of 10-city TSP by the modified formulation (1000 trials, weights $A = B = 2.0$).

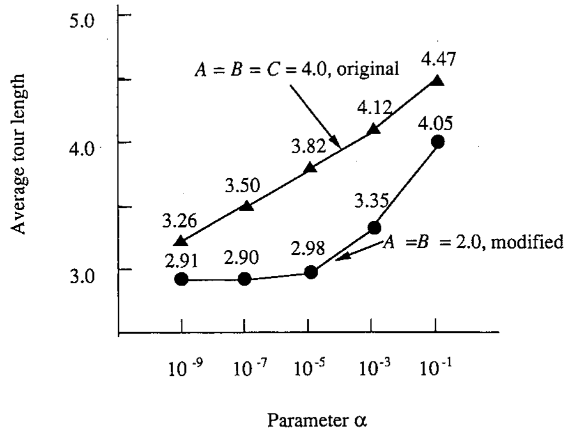


Fig. 8. Average tour length of 10-city TSP (1000 trials, initial values around the center).

For both formulations the quality of solutions improves as α approaches 0 or equivalently, the initial values are set as close as possible to the interior equilibrium point. But for the original formulation the optimal solutions are rarely obtained, only 6.7% of the 1000 trials for $\alpha = 10^{-9}$. For the modified formulation the best case occurs when $\alpha = 10^{-7}$. The optimal solution is obtained at a rate of 43% against 1000 trials. Also the solution with tour length less than 3 is obtained at a rate of 73% for $\alpha = 10^{-7}$.

From these results we can conclude for the TSP problem that:

1) We only need to suppress spurious state adjacent to feasible solutions.
2) From the standpoint of the quality of solutions, the weights should be the smallest values that satisfy 1), and the initial value should be set as close as possible to the interior equilibrium point.
3) The modified formulation is much better than the original formulation in the quality of solutions obtained.

In the following study we used only the modified formulation.

Thirty-City TSP: Fig. 9 shows the optimal solution for the 30-city TSP [3]. According to (30), if the weights satisfy

$$A = B > 2.18 \tag{40}$$

all the feasible solutions become strongly stable. Figs. 10 and 11 show the convergence characteristics for 1000 trials when

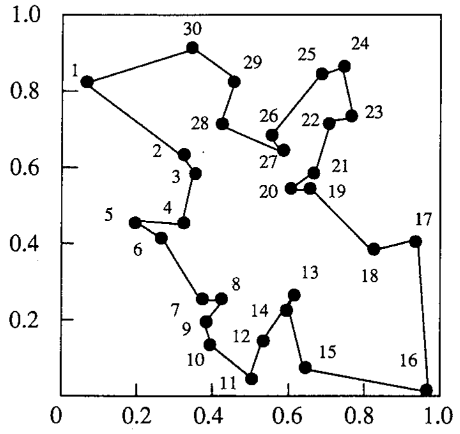


Fig. 9. Optimal tour of 30-city TSP with tour length=4.312.

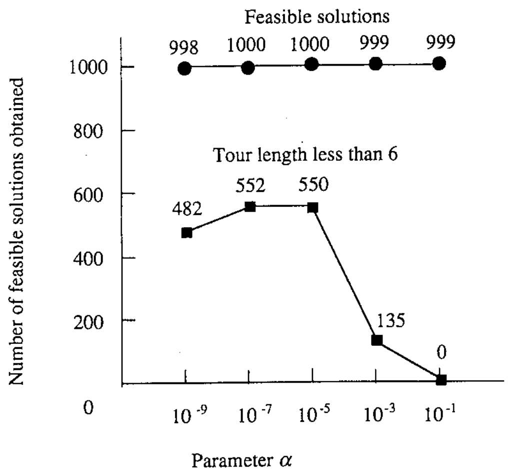


Fig. 10. Number of feasible solutions of 30-city TSP by the modified formulation (1000 trials, weights $A = B = 2.2$).

the initial values are set around the center of the hypercube. The difference with the former results is that it becomes difficult to obtain the near optimal solutions.

VI. Discussion of Results

When the weights in the energy function are set so that all the infeasible solutions adjacent to feasible solutions become unstable, feasible solutions are almost always obtained for the original and the modified formulations. One reason that the original formulation did not work in the previous study [5] is mal-selection of the weights. Another reason is slow convergence of the network due to the use of the hyperbolic tangent function instead of the piecewise-linear function [22]. With the hyperbolic tangent function, corrections of $\boldsymbol{x}$ become smaller as $\boldsymbol{x}$ approaches 1 or 0. Therefore, convergence slows down immensely and the calculation is terminated before it reaches a feasible solution.

Although the rate at which the feasible solutions is obtained is the same for two energy functions so long as we set the weights in the energy function according to our theory, the quality of the solutions depends heavily on the formulation of

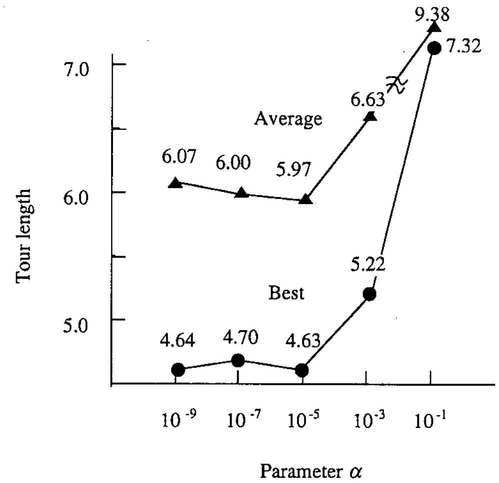


Fig. 11. Average tour length of 30-city TSP by the modified formulation (1000 trials, initial values around the center, weights $A = B = 2.2$).

the energy function. Further study requires clarification of a good formulation to improve the quality of solutions.

The selection of the initial values near the center of the hypercube performed better than that of the uniformly distributed random values in 0 and 1 as is a direct conclusion of our theory. For the 10-city TSP the quality of the solution is good. But for the 30-city TSP, performance is very poor. One way to improve the quality is to use simulated annealing [25]–[27]. Another way may be to combine the Hopfield network with other methods as discussed in [28]. In our future study, we need to clarify why the quality of the solution becomes inferior for a large size problem.

VII. Conclusions

Assuming that the output function of neurons is monotonic and differentiable at any interior point in the output range, we clarified the condition necessary for a vertex of a hypercube to become a local minimum of the Hopfield neural networks and the form of the convergence region to that minimum. Then based on this we derived a method to analyze and suppress spurious states in networks which are adjacent to feasible solutions and not adjacent to them. Finally we showed that all the spurious states of the TSP for the Hopfield original energy function could be suppressed by our method. According to computer simulations, feasible solutions were almost always obtained when all the infeasible solutions which were adjacent to feasible solutions were suppressed.

Appendix A

Stability of a Vertex by Eigenvalue Analysis

Investigate stability of a vertex when (9) is expressed by a hyperbolic tangent function. Eliminating $\boldsymbol{u}$ from (10) we obtain

$$\frac{d\boldsymbol{x}}{dt} = -\begin{pmatrix} 2(1-x_1)x_1 & & 0 \\ & \ddots & \\ 0 & & 2(1-x_n)x_n \end{pmatrix} T'\boldsymbol{x} + \boldsymbol{b}. \quad \text{(A1)}$$

The linearized equation of (A1) around vertex $\boldsymbol{c}$ in the n-dimensional hypercube is given by substituting

$$\boldsymbol{x} = \boldsymbol{x}' + \boldsymbol{c} \tag{A2}$$

into (A-1):

$$\frac{d\boldsymbol{x}'}{dt} = \begin{pmatrix} 2(2c_1 - 1)(T_1'\boldsymbol{c} + b_1) & & 0 \\ & \ddots & \\ 0 & & 2(2c_n - 1)(T_n'\boldsymbol{c} + b_n) \end{pmatrix} \boldsymbol{x}'. \tag{A3}$$

Thus the eigenvalues of (A3) are given by

$$\lambda_{c,\,i} = 2(2c_i - 1)(T_i'\boldsymbol{c} + b_i) \qquad \text{for } i = 1, \cdots, n. \tag{A4}$$

The condition that vertex $\boldsymbol{c}$ becomes a stable equilibrium point is given by the following theorem.

Theorem A1: Vertex $\boldsymbol{c}$ which satisfies

$$E_{\boldsymbol{c}}' < E_{\boldsymbol{c}(i)} \qquad \text{for } i = 1, \cdots, n$$

is a stable equilibrium point of (A-3).

Proof: From (8) and T being symmetric,

$$\begin{aligned} &E_{\boldsymbol{c}} - E_{\boldsymbol{c}(i)} \\ &= \boldsymbol{c}^t T\boldsymbol{c} + \boldsymbol{b}^t\boldsymbol{c} - \boldsymbol{c}(i)^t T\boldsymbol{c}(i) - \boldsymbol{b}^t\boldsymbol{c}(i) \\ &= \boldsymbol{c}^t T\boldsymbol{c} - \boldsymbol{c}(i)^t T\boldsymbol{c} + \boldsymbol{c}^t T\boldsymbol{c}(i) \\ &\quad - \boldsymbol{c}(i)^t T\boldsymbol{c}(i) + \boldsymbol{b}^t\boldsymbol{c} - \boldsymbol{b}^t\boldsymbol{c}(i) \\ &= (0, \cdots, 0,\, 2c_i - 1,\, 0, \cdots, 0)\{T\boldsymbol{c} + T\boldsymbol{c}(i)\} \\ &\quad + (2c_i - 1)b_i \\ &= (2c_i - 1)\{T_i\boldsymbol{c} + T_i\boldsymbol{c}(i) + 2b_i\} \\ &= \lambda_{\boldsymbol{c},i} - \lambda_{c(i),\,i}. \end{aligned} \tag{A5}$$

From (A4) and $T_{ii} = 0$,

$$\begin{aligned} \lambda_{\boldsymbol{c}(i),\,i} &= -2(2c_i - 1) \\ &\quad \cdot \{T_i(c_1, \cdots, c_{i-1}, 1 - c_i, \cdots, c_n)^t + b_i\} \\ &= -\lambda_{\boldsymbol{c},\,i}. \end{aligned}$$

Therefore, from (A5),

$$\lambda_{\boldsymbol{c},\,i} < 0 \qquad \text{for } i = 1, \cdots, n$$

holds and thus vertex $\boldsymbol{c}$ is stable.

APPENDIX B

Suppression of Spurious States for the Original Formulation

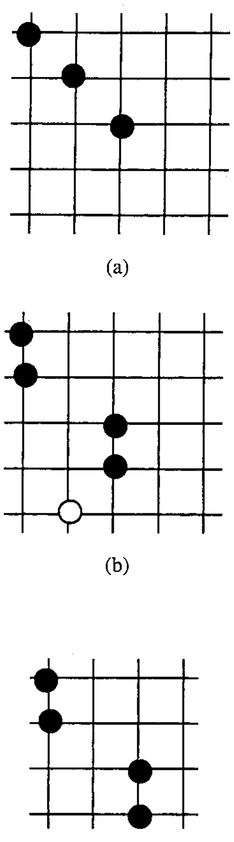
Consider suppressing spurious states not adjacent to feasible solutions as follows (see Fig. 12).

i) State which only violates the third constraint.

By changing one neuron from 0 to 1 without violating the first and second constraints, the energy corresponding to the third term decreases by at least $C/2$ and the fourth term increases by $D(d_{xy} + d_{xz})$ at most, thus from (24) this state cannot be a spurious state.

ii) State which only violates the first and second constraints.

By changing one neuron that violates constraints from 1 to 0, the first, second, and fourth terms decrease by at least A in total and the third term increases by $C/2$. Thus from (25), this state is not stable.



(a)

(b)

(c)

Fig. 12. Analysis of spurious states for the original formulation.

iii) In the same column $k(\geq 2)$ neurons fire at most and $n - l$ neurons fire in total.

By changing one neuron in the column from 1 to 0, the first and second constraints and the fourth term at least decrease by

$$\frac{k(k-1)}{2}A - \frac{(k-1)(k-2)}{2}A = (k-1)A \tag{A6}$$

in total, and the third constraint increases by

$$\left\{\frac{(l+1)^2}{2} - \frac{l^2}{2}\right\}C = \left(l + \frac{1}{2}\right)C. \tag{A7}$$

Therefore, spurious states may exist when

$$\left(l + \frac{1}{2}\right)C > (k-1)A \tag{A8}$$

holds. From (25), (A8) becomes

$$l > k - \frac{3}{2}. \tag{A9}$$

Also by changing the state of one neuron from 0 to 1 that is in the same column as the k neurons and that belongs to the row in which no neurons fire, the first and second constraints and the fourth term increase by at least kA in total, and the third term decreases by $(l - 1/2)C$. Thus if

$$kA > \left(l - \frac{1}{2}\right)C \tag{A10}$$

holds, spurious states may exist. From (25), (A10) becomes

$$l < k + \frac{1}{2}. \tag{A11}$$

Thus from (A9) and (A11), spurious states may exist when

$$k = l, \quad \text{or} \quad k = l + 1 \tag{A12}$$

holds.

Now change the state of one 0-state neuron that does not violate the first and second constraints when the state is changed to one, and which is adjacent to a 1-state neuron in the column. (Here we mean adjacent in the modulo n sense i.e., neurons in the first and $(n+1)$st are adjacent.) The first and second constraints and fourth term increase at most by (see Fig. 12(c))

$$D \max_x \sum_y^{(2k)} d_{xy} \tag{A13}$$

in total where the summation is added for the maximum to $2k$th maximum distances from city x, and the third constraint decreases by $(l - 1/2)C$. From (A12) the maximum value of (A13) is obtained when $k = l + 1$. Thus if

$$C > \frac{D}{\left(l - \frac{1}{2}\right)} \max_x \sum_y^{(2l+2)} d_{xy} \tag{A14}$$

holds, we can suppress this kind of spurious states. Since

$$\frac{1}{4} \max_x \sum_y^{(4)} d_{xy} \geq \frac{1}{2l+2} \max_x \sum_y^{(2l+2)} d_{xy} \tag{A15}$$

it is easy to see that

$$2D \max_x \sum_y^{(4)} d_{xy} \geq \frac{l+1}{2l-1} D \max_x \sum_y^{(4)} d_{xy} \geq \frac{D}{l - \frac{1}{2}} \max_x \sum_y^{(2l+2)} d_{xy} \tag{A16}$$

holds. Thus we need to check (A14) for $l = 1$, namely (26). Therefore, from (24) if (26) holds all the spurious states are suppressed and stable states are all feasible solutions.

Suppression of Spurious States for the Modified Formulation

Consider suppressing spurious states not adjacent to feasible solutions as follows.

i) At most one neuron fires in any row or column (Fig. 12(a)).

If the number of neurons that fire is less than n, we can select a 0-state neuron which does not violate constraints when the state is changed to 1. With this change the first and second constraints decrease at least by $(A + B)/2$ in total and the third term increases at most by $D(d_{xy} + d_{yz})$. Thus so long as (30) holds this state is not stable.

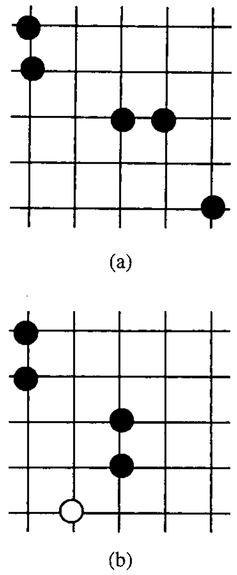


Fig. 13. Analysis of spurious states for the modified formulation.

ii) k neurons ($k \geq 3$) in a row or column fire.

By changing the state of the above one neuron from 1 to 0, the energy always decreases. Thus the state that satisfies ii) is not stable.

iii) Two neurons fire in a row or column.

If V_{xi}, V_{xj}, and $V_{yi} = 1$, the energy always decreases by changing V_{xi} to 0. Thus we need not consider this situation. Also, we exclude the situation in which a 1-state neuron is adjacent to the above neuron(s), since, by changing the state of one of the two neurons next to the adjacent neuron, the third term decreases although the total energy of the first and the second constraints is the same.

According to the above discussion the spurious states may exist for the situation in which two neurons fire in a row or column and there are no 1-state neurons adjacent to them as follows.

iv) Two neurons fire in columns (n neurons in total) and no 1-state neurons are adjacent to them.

Since by changing any 1-state neuron to zero the energy does not change, thus this state is weakly stable, and cannot be suppressed by changing the weights. This spurious state occurs when n is even (See Fig. 13 (i)).

v) Two neurons fire in at least two columns.

Consider the most critical case in which two neurons fire at columns i and $i+2$. Then all the neurons at column $i+1$ are zero. By changing the state of one neuron at column $i+1$ to 1, which does not violate the constraints, the first and the second constraints decrease by $(A+B)/2$ in total and the third term increases at most by

$$D \max_x \sum_y^{(4)} d_{xy}. \tag{A17}$$

Therefore, from (30) if (31) holds this state is unstable (see Fig. 13(b)).

Appendix C

Stability of the Reduced System of the TSP

The reduced matrices and coefficient vectors for the original and modified formulations are given by (A18) and (A19) at the bottom of the page, and

$$T = \begin{bmatrix} (n-1)A & A+2D\,d_{12} & \cdots & A+2D\,d_{1n} \\ A+2D\,d_{21} & (n-1)A & \cdots & A+2D\,d_{2n} \\ & \cdots & & \\ A+2D\,d_{n1} & \cdots & & (n-1)A \end{bmatrix} \tag{A20}$$

$$\boldsymbol{b} = -(A,\cdots,A)^t \tag{A21}$$

respectively. Thus T is symmetric with positive elements, $\boldsymbol{b}$ has negative elements, and from (25) and (29)

$$T_{ii} = T_{jj}, \quad b_i = b_j$$
$$T_{ii} > T_{ij} \quad T_{ij} > -b_i \quad \text{for } i \neq j,\ i, j = 1,\cdots,n. \tag{A22}$$

Now we can show the following.

Theorem A2: Let T and $\boldsymbol{b}$ satisfy (A22). Then a set of linear equations $T\boldsymbol{x} + \boldsymbol{b} = 0$ has a unique solution which satisfies $0 < x_i < 1$ for $i = 1,\cdots,n$.

Proof: We prove the theorem by reduction.

i) For $n = 2$ the solution is

$$x_1 = x_2 = \frac{-b_1}{T_{11} + T_{12}}. \tag{A23}$$

Therefore, from (A22) the theorem holds.

ii) Let the theorem hold for $n = k$ and prove that it also holds for $n = k + 1$.

According to (A22) the value at which hyperplane $T_i\boldsymbol{x} + b_i = 0$ intersects the x_i axis is the smallest among the remaining hyperplanes. Thus two hyperplanes $T_{k+1}\boldsymbol{x}+b_{k+1} = 0$ and $T_i\boldsymbol{x}+b_i = 0 (i \neq k+1)$ intersect in H^- since hyperplane $T_i\boldsymbol{x} + b_i = 0$ is on different sides of $T_{k+1}\boldsymbol{x} + b_{k+1} = 0$ at $x_i = 1$ on the x_i axis and at $x_{k+1} = 1$ on the x_{k+1} axis (see Fig. 14).

According to the assumption of reduction, the solution of $T_i\boldsymbol{s} + b_i = 0 (i = 1,\cdots,k)$ for $s_{k+1} = 0$ satisfies $0 < s < 1 (i \neq k)$. Thus the intersection of the k hyperplanes $T_i\boldsymbol{x}+b_i = 0 (i = 1,\cdots,k)$ is on different sides of $T_{k+1}\boldsymbol{x} + b_{k+1} = 0$ at $x_{k+1} = 1$ on the x_{k+1} axis and at $(\boldsymbol{s}, 0)$. Thus the intersection of $k + 1$ hyperplanes exists in the interior of the hypercube.

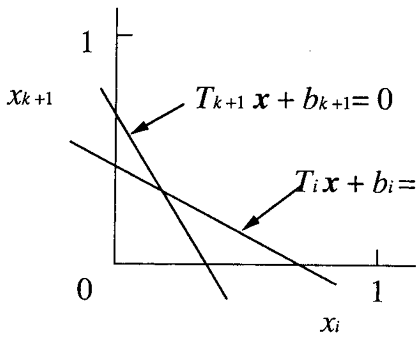


Fig. 14. Intersections of hyperplanes.

Thus from i) and ii) the theorem holds (end of proof).

Now let's consider the stability of equilibrium points (22). According to (A22) any x_i cannot be 1 in H. Thus the equilibrium points (22) are obtained by solving

$$\begin{aligned} T_i\boldsymbol{x} + b_i &= 0 \quad \text{for } i \in S_k \\ x_i &= 0 \quad \text{for } i \notin S_k,\ i = 1,\cdots,n \end{aligned} \tag{A24}$$

which satisfies $0 < x_i < 1$ for $i \in S_k$ because for (A24) Theorem A2 holds.

Since hyperplane $T_i\boldsymbol{x} + b_i = 0$ is not parallel to the x_i axis but intersects it, and since for $\boldsymbol{x} \in D^-(i)$ the component x_i increases and for $\boldsymbol{x} \in D^+(i)$ it decreases, at any point $\boldsymbol{x} \in H^-$ except $T\boldsymbol{x} + \boldsymbol{b} = 0$ the component x_i is corrected so that $\boldsymbol{x}$ moves to the hyperplane. If $\boldsymbol{x}$ is included in the space spanned by k x_i axes where $i \in S_k$, $\boldsymbol{x}$ converges to the solution of (A24), but if $\boldsymbol{x}$ is not in that space, $\boldsymbol{x}$ moves away from the solution towards the hyperplane $T_i\boldsymbol{x} + b_i = 0 i \notin S_k$. Thus the equilibrium points given by (A24) are saddle points. Likewise, all the vertexes are unstable. Thus for any $\boldsymbol{x}$ in H^-, $\boldsymbol{x}$ converges to the solution of $T\boldsymbol{x}+\boldsymbol{b} = 0$ which is the only stable solution.

Acknowledgment

The author wishes to thank S. V. B. Aiyer, M. Niranjan, and F. Fallside for suggesting the usefulness of the piecewise-linear function instead of the hyperbolic tangent function. He also thanks K. Nakanishi for showing that the stability condition for the hyperbolic tangent function also holds for the piecewise-linear function.

$$T = \begin{bmatrix} (n-1)(C+\frac{A}{2}) & nC+\frac{B}{2}+2D\,d_{12} & \cdots & nC+\frac{B}{2}+2D\,d_{1n} \\ nC+\frac{B}{2}+2D\,d_{21} & (n-1)(C+\frac{A}{2}) & \cdots & nC+\frac{B}{2}+2D\,d_{2n} \\ & \cdots & & \\ nC+\frac{B}{2}+2D_{\,n1} & \cdots & & (n-1)(C+\frac{A}{2}) \end{bmatrix} \tag{A18}$$

$$\boldsymbol{b} = -\begin{bmatrix} (n-\frac{1}{2})C \\ \vdots \\ (n-\frac{1}{2})C \end{bmatrix} \tag{A19}$$

REFERENCES

[1] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci. USA*, vol. 79, pp. 2554–2558, Apr. 1982.

[2] ——, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proc. Nat. Acad. Sci. USA*, vol. 81, pp. 3088–3092, May 1984.

[3] J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Bio. Cybern.*, vol. 52, pp. 141–152, 1985.

[4] R. P. Lippman, "An introduction to computing with neural nets," *IEEE ASSP Magazine*, pp. 4–22, Apr. 1987.

[5] V. Wilson and G. S. Pawley, "On the stability of the TSP problem algorithm of Hopfield and Tank," *Bio. Cybern.*, vol. 58, pp. 63–70, 1988.

[6] B. W. Lee and B. J. Sheu, "An investigation on local minima of Hopfield network for optimization circuits," *Proc. ICNN-88*, pp. 45–51, July 1988.

[7] D. J. Burr, "An improved elastic net method for the traveling salesman problem," in *Proc. ICNN-88*, pp. 69–76.

[8] G. J. Hueter, "Solution of the traveling salesman problem with an adaptive ring," in *Proc. ICNN-88*, pp. 85–92.

[9] H. Szu, "Fast TSP algorithm based on binary neuron output and analog neuron input using the zero-diagonal interconnect matrix and necessary and sufficient constraints of the permutation matrix," in *Proc. ICNN-88*, pp. 259–266.

[10] S. U. Hedge, J. L. Sweet, and W. B. Levy, "Determination of parameters in a Hopfield/Tank computational network," in *Proc. ICNN-88*, pp. 291–298.

[11] D. E. Van den Bout and T. K. Miller, "A traveling salesman objective function that works," in *Proc. ICNN-88*, pp. 299–303.

[12] E. Wacholder, J. Han, and R. C. Mann, "An extension of the Hopfield–Tank model for solution of the multiple traveling salesman problem, in *Proc. ICNN-88*, pp. 305–324.

[13] J. Ramanujam and P. Sadayappan, "Optimization by neural networks," in *Proc. ICNN-88*, pp. 325–332.

[14] R. D. Brandt, Y. Wang, A. J. Laub, and S. K. Mitra, "Alternative networks for solving the traveling salesman problem and the list-matching problem," in *Proc. ICNN-88*, pp. 333–340.

[15] Y. S. Foo and Y. Takefuji, "Integer linear programming neural networks for job-shop scheduling," in *Proc. ICNN-88*, pp. 341–348.

[16] B. Kamgar-Parsi *et al.*, "On problem solving with Hopfield neural networks," *Bio. Cybern.*, vol. 62, pp. 415–423, 1990.

[17] E. Mjolsness, "Control of attention in neural networks," in *Proc. ICNN-87*, vol. 2, pp. 567–574, 1987.

[18] A. Culhane, M. Peckerar, and C. Marrian, "A neural net approach to discrete Hartley and Fourier transforms," *IEEE Trans. Circuits Syst.*, vol. 36, May 1989.

[19] Y. Uesaka, "On the stability of neural network with the energy function induced from a real-valued function of binary variables," *IEICE Tech. Rep.* PRU-86-6, June 1988.

[20] S. Abe, "Theories on the Hopfield neural networks," in *Proc. IJCNN-89*, vol. 1, pp. 557–564, June 1989.

[21] S. Abe and J. Kawakami, "Theories on the Hopfield neural networks with inequality constraints," in *Proc. IJCNN-WASH-DC-90*, vol. 1, pp. 349–352, Jan. 1990.

[22] S. V. B. Aiyer, M. Niranjan, and F. Fallside, "A theoretical investigation into the performance of the Hopfield model," *IEEE Trans. Neural Networks*, vol. 1, 2, June 1990.

[23] H. Date *et al.*, "LSI module replacement methods using neural computation networks," in *Proc. IJCNN-89*, vol. III, pp. 831–836, June 1990.

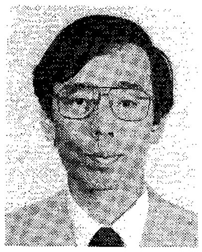
[24] J.-H. Li, A. N. Michel, and W. Porod, "Analysis and synthesis of a class of neural networks: Linear systems operating on a closed hypercube," *IEEE Trans. Circuits Syst.*, vol. CAS-36, pp. 1405–1422, Nov. 1989.

[25] G. E. Hinton *et al.*, "Boltzmann machines: Constraint satisfaction networks that learn," *CMU-CS-84-119*, CMU, 1984.

[26] H. Szu, "Fast simulated annealing," *Physics Letters A*, vol. 122, pp. 157–162, June 1987.

[27] Y. Akiyama *et al.*, "Combinatorial optimization with Gaussian machines," in *Proc. IJCNN-89*, vol. I, pp. 533–540, June 1989.

[28] X. Xu and W. T. Tsai, "Effective neural algorithms for the traveling salesman problem," *Neural Networks*, vol. 4, pp. 193–205, 1991.



Shigeo Abe (M'79–SM'83) received the B.S. degree in electronics engineering, the M.S. degree in electrical engineering, and the Doctor of Engineering degree all from Kyoto University, Kyoto, Japan, in 1970, 1972, and 1984, respectively.

Since 1972, he has been with Hitachi Research Laboratory, Hitachi, Ltd. and has been engaged in power system analysis, development of a vector processor, a Prolog processor, and neural network theories. From 1978 to 1979 he was a visiting research associate at the University of Texas at Arlington.

Dr. Abe was awarded an outstanding paper prize from the Institute of Electrical Engineers of Japan in 1984. He is a member of the International Neural Network Society, the Institute of Electrical Engineers of Japan, the Information Processing Society of Japan, and the Institute of Electronics, Information and Communication Engineers of Japan.