

# ニューラルネットワークの 組合せ最適化への応用

武藤 佳恭

## まえがき

McCulloch と Pitts 等 [1] が動物や人間の頭脳 (神経回路網) の数学モデルを 1943 年に発表して以来, Hebb [2], Widrow [3], Rosenblatt [4] 等によって 1960 年代にニューラル・コンピューティング研究は最初の全盛期を迎える。ところがニューラル・コンピューティングの研究者であった Minsky 等 [5] は, Rosenblatt の提案する Perceptrons モデルをきびしく非難したため, 1970 年代と 1980 年代前半には Amari, Cooper, Fukushima, Grossberg, Anderson, Kohonen 等は研究規模において衰退期を迎える。1985 年に Hopfield 等 [6, 7] が組合せ最適化へのニューラル・ネットワークを示し, 多くの若手研究者を生み出すが, Wilson [8], Paielli [9] 等は Hopfield モデルをきびしく非難したため, 米国政府は組合せ最適化へのニューラル・コンピューティングの研究費を完全に削除した。本論文の筆者は政府の取り決め以前に 7 万ドルの研究費を NSF (National Science Foundation) から運良く受賞したが, 筆者の知る限り, 組合せ最適化への米国政府研究費は現在もゼロである。ニューラル・コンピューティングの NSF の総研究費は年間 100 万ドルにすぎない\*。したがって本論文では筆者等の研究成果を中心に報告する。

これまでに取り組んできた最適化および組合せ理論問題には, 四色および K 色問題 [10], グラフの平面性判定および平面埋め込み問題 [11, 12], タイル問題 [13], ソートおよび文字列のパターンマッチ問題 [14, 15], サークル・グラフの最大独立集合問題 [16], RNA の 2 次構造予測問題 [16, 17], クロスバ・スイッチ・スケジューリング問題 [18], 冗長配置問題 [19], TDM クロスバ・スイッチ・システムのタイム・スロット・配属問題 [20], チャネル配線問題 [21], スルー・ホール最小

化問題 [22], 最大クリーク問題 [23, 24], 2 部グラフ問題 [25], 最大グラフ切断問題 [26], グラフ分割問題 [27], モジュール・オリエンテーション問題 [28], N・クイーン問題 [29], Hip ゲーム問題 [30], 蜂の巣問題 [31], BIBD 問題 [32], 巡回騎士問題 [33], ジョブ・ショップ問題 [34] 等がある。これらの問題は, グラフ理論, ゲーム理論, 離散数学, コンピュータ・サイエンス, 分子生物学, VLSI CAD, 信頼性工学, マネジメント・サイエンス, コンピュータ通信工学等の幅広い分野に属する。本論文では, ニューラル・コンピューティングの原理を簡単に紹介し, 紙面の許す限りいくつかの問題例を具体的に説明する。

## ニューラル・コンピューティングの原理

脳の研究は長年にわたってかなり行なわれているが, 脳の動作原理についてはほとんど解析されていない。われわれの脳は約  $10^{11}$  個のニューロン (プロセッサ) と  $10^{15}$  個のシナプス・リンクから成り立っている。実際には多種類のニューロンが存在するが, 本論文で述べる人工ニューラル・ネットワークは 1 種類の多ニューロンがシナプス・リンクを介して接続されていると単純に考える。組合せあるいは最適化問題を解くに当たって重要なことは, ニューロンをどのように使って問題を表現するかである。学習とはニューロン間のシナプス・リンクの強さを決定することである。最適化問題では, あらかじめシナプス・リンクの接続を固定したニューラル・ネットワークを用いる。つまり, 与えられた問題を解くに当たってニューラル表現を決定することは, ニューロン間のシナプス・リンクを決定することに等しい。ニューラル・ネットワークあるいはニューラル表現は一般にニューロン動作式で与えられる。

ニューロン  $i$  の出力  $V_i$  は入力  $U_i$  の関数である。

$$V_i = f(U_i)$$

たけふじ よしやす Case Western Reserve University, 慶応義塾大学

\* 米国では研究費が出ないテーマは研究しづらい環境にある。

ここで  $f$  は非線形増加関数である。たとえば McCulloch-Pitts ニューロンの入出力関数 (図 1 参照) は

$$V_i = 1 \text{ if } U_i > 0, \text{ 0 otherwise}$$

シグモイド・ニューロンの入出力関数 (図 2 参照) は

$$V_i = 1/2(\tanh(\lambda_0 U_i) + 1)$$

ここで  $\lambda_0$  はゲイン定数。

振動現象を軽減するヒステリシス・McCulloch-Pitts ニューロン [18] の入出力関数 (図 3 参照) は

$$V_i = 1 \text{ if } U_i > UTP, \text{ 0 if } U_i < LTP, \\ \text{unchange otherwise.}$$

マキシマム・ニューロン [25] の入出力関数は

$$V_{mi} = 1 \text{ if } U_{mi} = \max\{U_{m1}, \dots, U_{mn}\}, \\ \text{0 otherwise.}$$

ニューロンの一般動作式は次式で表現される。

$$\frac{dU_i}{dt} = -\frac{\partial E}{\partial V_i}$$

ここで  $E = E(V_1, V_2, \dots, V_n)$  はエネルギー関数と呼ばれ、最適化の目標は、問題の制約条件を満たしながら  $E$  を最小化することである。  $f$  が非線形連続増加関数の場合 (図 2), エネルギー関数  $E$  が常に減少か増加しないことを次に示す。  $f$  が非線形非連続増加関数の場合に関しては参考文献 [35] を参照せよ。

$$\frac{\partial E}{\partial t} = \sum_i \frac{dV_i}{dt} \cdot \frac{\partial E}{\partial V_i} = \sum_i \frac{dU_i}{dt} \cdot \frac{\partial V_i}{\partial U_i} \cdot \frac{\partial E}{\partial V_i}$$

ここでニューロン動作方程式を導入すると

$$= \sum_i \left( -\frac{\partial E}{\partial V_i} \right) \cdot \frac{\partial V_i}{\partial U_i} \cdot \frac{\partial E}{\partial V_i} = -\sum_i \left( \frac{\partial E}{\partial V_i} \right)^2 \cdot \frac{\partial V_i}{\partial U_i} \leq 0$$

証明終了。

組合せ最適化問題を解くにあたって、いちばん重要なことは、いかにして正しいニューロン動作式を導出するかである。専門家の中でもエネルギー関数  $E$  を示さないと納得しない人がいるが、ニューロン動作式はニューラル表現およびニューロン間のシナプス・リンク結合情報もすべて含んでいる。ニューロン動作式からエネルギー

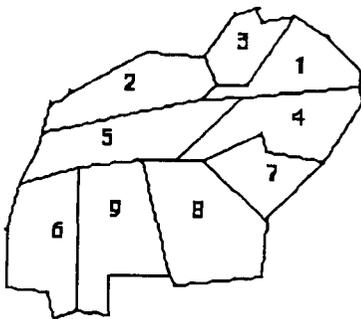


図 4 9-領域問題

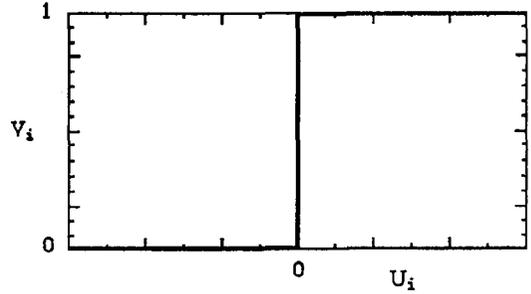


図 1 McCulloch-Pitts ニューロン入出力関数

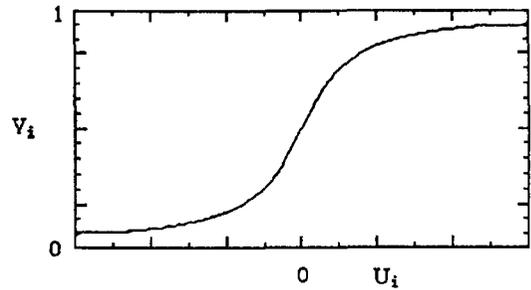


図 2 シグモイドニューロン入出力関数

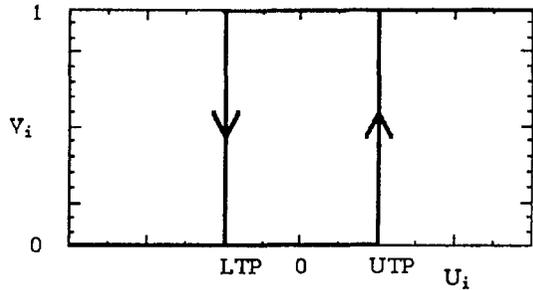


図 3 ヒステリシス McCulloch-Pitts ニューロン入出力関数

関数  $E$  は簡単に導びける。  $E = \int \partial E = - \int \frac{\partial U_i}{\partial t} \cdot \partial V_i$  つまり動作式を  $\partial V_i$  に関して積分してやればよい。

組合せ理論問題例：3色問題 (NP (Nondeterministic Polynomial time) 問題) 地図の3色塗り問題とは、3色だけを使って互いに隣接する領域が同色にならないように塗り分ける問題である。図 4 に9領域から成る地図問題を示す。

まず3色のどの色を塗るかを表現するために 100, 010, 001 をそれぞれ赤色, 黄色, 青色塗りとする。この問題を解くのに  $9 \times 3 = 27$  個のニューロンを使えばよい。図 5 において領域 1 を黄色に塗ると、隣接領域 2, 3, 4, 5 は黄色を塗ることはできない。図 6 の3色塗りの解は図 5 のシステム状態に対応している。ニュー

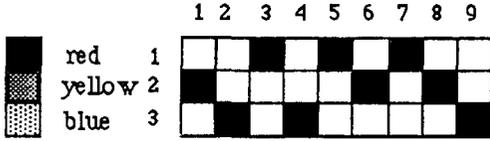


図 5 9×3 ニューラル表現

ラル・ネットワークの行なう仕事は2つで、1つは各領域に1つの色を塗る作業、もう1つは隣接する領域どうしが同色にならないようにする作業である。X領域のi番目の色ニューロンの動作式は次式ようになる。

$$\frac{dU_{Xi}}{dt} = -\left(\sum_{j=1}^9 V_{Xj} - 1\right) - \sum_{\substack{Y=1 \\ Y \neq X}}^9 d_{XY} \cdot V_{Yi}$$

ここで  $d_{XY}$  は隣接行列を表わし、 $d_{XY}=1$  とはX領域とY領域は隣接することを示す。たとえばX領域に色が塗られていないと第1項は+1となり  $U_{Xi}$  を増加させ、 $U_{Xi}$  が  $U_{Xi} > 0$  となると  $V_{Xi}=1$  となり、ニューロン  $X_i$  が発火したと呼ぶ。X領域に2色を使っていると、第1項は-1となり  $U_{Xi}$  を減少させる。X領域とY領域が隣接して同色  $i$  を塗った場合は  $U_{Xi}$  を減少させる。先ほどの動作式を正規化すると  $n$  領域3色問題は次式で簡単に解決できることが実験的に知られている。

$$\frac{dU_{Xi}}{dt} = -\left(\sum_{j=1}^n V_{Xj} - 1\right) - \sum_{\substack{Y=1 \\ Y \neq X}}^n d_{XY} V_{Yi} \sum_{k=1}^n d_{Yk} \\ + h\left(\sum_{j=1}^n V_{Xj}\right) \left( \sum_{k=1}^n d_{Xk} + \frac{\sum_{k=1}^n \sum_{Y=1}^n d_{XY} d_{Yk}}{\sum_{k=1}^n d_{Xk}} \right)$$

ここで第3項は hill-climbing 項と呼ばれ、システム状態を常に解に導いてくれる。  $h(X)=1$  if  $X=0$ , 0 otherwise.  $K$  色塗り問題の解は常に  $E=0$  となる。

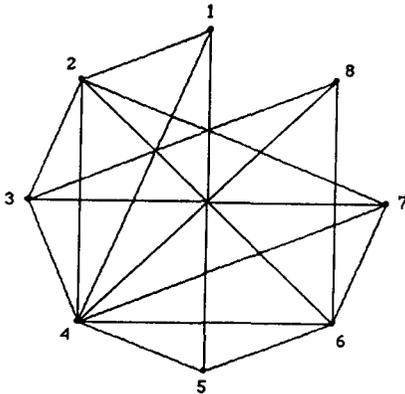


図 7 8-節点17辺グラフ

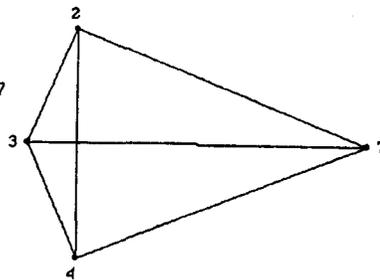


図 8 図7のマキシマム・クリーク

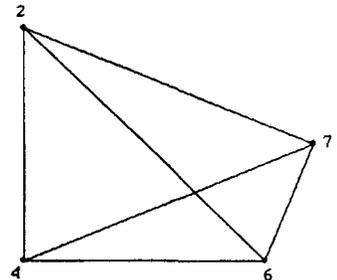


図 9 図7のマキシマム・クリーク

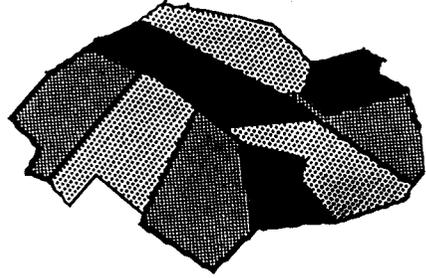


図 6 三色塗りの解

エネルギー関数  $E$  に興味のある読者は動作方程式を積分して算出せよ。

答えは参考文献 [35] を参照せよ。

**最適化問題例：最大クリーク問題 (NP問題)**

グラフ  $G(V, E)$  と正整数  $J \leq |V|$  が与えられた時、 $G$  には大きさ  $J$  以上のクリークが存在するか、すなわち  $V' \subseteq V$ ,  $|V'| \geq J$ , かつ  $V'$  のどの節点も  $E$  の辺で結ばれているような  $V'$  が存在するか？

たとえば図7に示すグラフ  $G$  は8節点17辺から成る。図8と図9に2つの最大クリークを示す。本論文では最大クリークあるいは最大クリークに近い解を得るためのニューラル表現を2つ紹介する。

**ニューラル表現1：**

McCulloch-Pitts ニューロンを用いて  $n$  節点グラフ問題を解くには  $n$  個のニューロンを使えば必要十分である。つまりそれぞれのニューロンは各節点がクリークに所属するかしないかを単純に示す。たとえば  $V_i=1$  というのは節点  $i$  がクリーク・グラフに所属し、 $V_i=0$  は所属しないものとする。最大クリーク問題 ( $n$  節点グラフ) のニューロン動作式は次式で与えられる。われわれの実験結果では従来の最善方式に比べて1000倍以上の高速化が可能となった。

$$\frac{dU_i}{dt} = -\sum_{j=1}^n (1-d_{ij}) + Bh \left( \sum_{j=1}^n (1-d_{ij}) V_j + V_i \right)$$

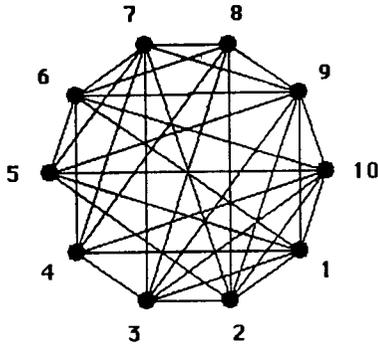


図 10 10節点34辺グラフG

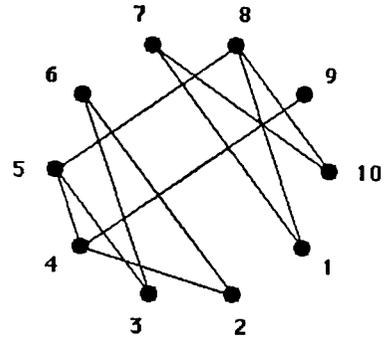


図 11 図10のグラフGの補グラフ

ここで、節点  $i$  と  $j$  が辺で結ばれていると  $d_{ij}=1$  となり、それ以外は  $d_{ij}=0$  となる。また  $d_{ij}=d_{ji}$ ,  $d_{jj}=1$  は常に満たされる。係数  $B$  は定数で次式によりあらかじめ導出しておく。

$$B = \frac{\text{隣接している節点の数}}{n \times (\text{グラフ密度})}$$

$h(X)$  は hill-climbing 項で  $h(X)=1$  if  $X=0$ , 0 otherwise.

ニューラル表現 2:

マキシマム・ニューロンを用いるとクリーク問題のエネルギー関数  $E$  は次式で表現される。

$$E = \frac{1}{2} \sum_{X=0}^n \sum_{Y=0}^n \sum_{i=1}^n W(X, Y) V_{Xi} V_{Yi}$$

したがって、

$$\frac{dU_{Xi}}{dt} = -\frac{\partial E}{\partial V_{Xi}} = -\sum_{Y=0}^n W(X, Y) V_{Yi}$$

for  $i=1$  or  $2$  となる。

このモデルには物理学上の Ising モデルを用いており、むずかしい理論を説明するかわりに次の例題を使っ

て説明する。

10節点34辺グラフ問題を図10に示す。まず最初にやる作業は補グラフを構築する。図11に図10の補グラフを示す。次に #0 節点を補グラフに加える。(図12参照)

次にウェイト・マトリックスを計算する。このウェイト・マトリックスは先ほどのニューロン動作式の係数である。計算法は

$$W_{ij} = W_{ji} = \frac{1}{4} q_{ij}$$

$$W_{0i} = W_{i0} = \frac{1}{4} \left( \sum_{j=1}^n q_{ij} - 1 \right) \text{ for } i, j=1 \text{ to } 10$$

ここで補グラフで  $i$  と  $j$  節点間に辺が存在すると  $q_{ij}=1$  それ以外は  $q_{ij}=0$  とする。たとえば  $W_{24}=0.25$ 。

このようにして計算したウェイト・マトリックスを図13に示す。

ニューロン動作式を使って得られたクリークを図14に示す。

組合せ最適化のためのニューラルネットワーク・シミュレータの作り方

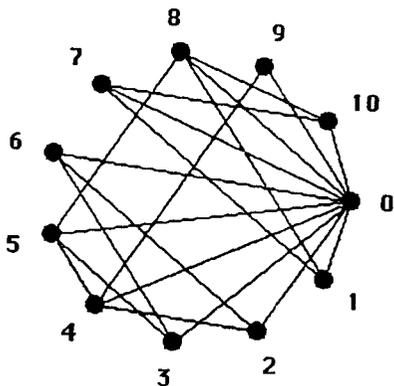


図 12 グラフ  $G_M$

	0	1	2	3	4	5	6	7	8	9	10
0	0.00	0.25	0.25	0.25	0.50	0.50	0.25	0.25	0.50	0.00	0.25
1	0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.25	0.25	0.00	0.00
2	0.25	0.00	0.00	0.00	0.25	0.00	0.25	0.00	0.00	0.00	0.00
3	0.25	0.00	0.00	0.00	0.00	0.25	0.25	0.00	0.00	0.00	0.00
4	0.50	0.00	0.25	0.00	0.00	0.25	0.00	0.00	0.00	0.25	0.00
5	0.50	0.00	0.00	0.25	0.25	0.00	0.00	0.00	0.25	0.00	0.00
6	0.25	0.00	0.25	0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.25	0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.25
8	0.50	0.25	0.00	0.00	0.00	0.25	0.00	0.00	0.00	0.00	0.25
9	0.00	0.00	0.00	0.00	0.25	0.00	0.00	0.00	0.00	0.00	0.00
10	0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.25	0.25	0.00	0.00

図 13 グラフ  $G_M$  のウェイト・マトリックス

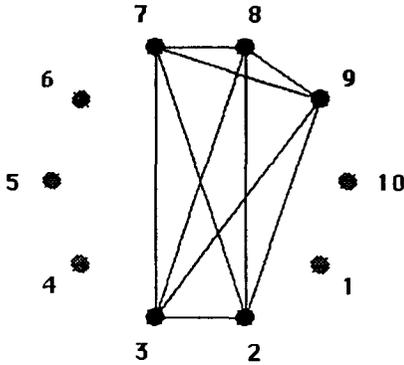


図 14 図10のグラフのマキシマム・クリーク

ニューロン動作式を導出することができれば、シミュレータは簡単に構築でき、1次 Euler 法を用いて数値解析できる。

$$U_i(t+1) = U_i(t) + \Delta U_i \cdot \Delta t \text{ for } i=1, \dots, n$$

$n$  個のニューロンを用いるニューラル・ネットワークのシミュレータは  $n$  個の動作式を持つ。  $\Delta t=1$  とすれば

$$U_i(t+1) = U_i(t) + \Delta U_i \text{ for } i=1, \dots, n$$

ここで  $\Delta U_i$  はニューロン  $i$  の動作式を示す。

入力  $U_i$  を変更した後出力  $V_i$  を計算する。

$$V_i = f(U_i)$$

シミュレータには逐次式と並列式が考えられる。

図15に逐次式シミュレータ、図16に並列式シミュレータを示す。

今までに、30以上のアプリケーション問題をわれわれは解いているが、すべて整数計算のみでシミュレータは構築されており、Macintosh と Unix コンピュータ上で稼働する。

質問があれば下記の住所か Email で連絡してください

```

Program1 sequential-simulator
begin
initialization of Uij and Vij for i,j:=1 to N;
/**** Main Program ****/
while (a set of conflicts is not empty) do
for i:=1 to N
for j:=1 to N
begin
Uij:=Uij+ ΔUij;
If Uij>0 then Vij:=1 else Vij:=0;
end;
end;
/**** Main Program end ****/

```

図 15 逐次式シミュレータ

い。

〒252 藤沢市遠藤5322 慶応義塾大学環境情報学部  
武藤佳恭 (takefuji @ aa. cs. keio. ac. jp)

Prof. Y. Takefuji, Dept. of Electrical Engineering & Applied Physics, Case Western Reserve University, Cleveland, OH 44106 (takefuji @ axon. CWRU. edu)

## 今後の展望

エンジニアリング問題ばかりでなく、数学上あるいは物理学上の未解決問題を近い将来においてニューラル・コンピューティング技術は解く可能性を持っている。

ニューラル・コンピューティングは将来において並列計算手法の主流となるであろう\*。アルゴリズムばかりでなく、ニューラルネットワークのハードウェア・デバイス技術のブレイクスルーが今後期待される。

\* 従来の並列計算アルゴリズムで要求されるむずかしい同期機構をいっさい必要としないのがニューラル・コンピューティングの特長である。従来の並列計算アルゴリズムでは、どこをどう並列化するかが重要であるが、ニューラル・コンピューティングではニューラル表現を考えることが並列化の表現である。

[感謝] この論文を書く機会を与えてくださった安西教授に感謝する。われわれのプロジェクトは NSF (MIP-8902819), 住友金属, 富士電機, AIWARE (OHIO) の支援による。

```

Program2 synchronous-parallel-simulator
begin
initialization of Uij and Vij for i,j:=1 to N;
/**** Main Program ****/
while (a set of conflicts is not empty) do
begin
/**** The first loop ****/
for i:=1 to N
for j:=1 to N
Uij:=Uij+ΔUij;
/**** End of the first loop ****/
/**** The second loop ****/
for i:=1 to N
for j:=1 to N
If Uij>0 then Vij:=1 else Vij:=0;
/**** End of the second loop ****/
end;
end;
/**** Main Program end ****/

```

図 16 並列式シミュレータ

## 参 考 文 献

- [1] McCulloch W. S., and Pitts W. H., (1943), A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5, 115.
- [2] Hebb D. O., (1949), *The organization of behavior* (New York : Wiley).
- [3] Widrow B, and Hoff M. E., (Sept. 1960), Adaptive switching circuits. *Proc. of IREWESCON Convention Record*.
- [4] Rosenblatt F., (1962), *Principles of neurodynamics* (New York : Spartan).
- [5] Minsky M., and Papert S., (1969), *Perceptrons* (MIT Press).
- [6] Hopfield J. J., and Tank D. W., (1985), Neural Computation of Decisions in Optimization Problems. *Biological Cybernetics*, 52, 141-152.
- [7] Tank D. W., and Hopfield J. J., (1986), Simple Neural Optimization Networks : An A/D Converter, Signal Decision Circuit, and a Linear Programming Circuit, *IEEE Trans. on Circuits and Systems*, Vol. CAS-33, No.5, pp.533-541.
- [8] Wilson G. V. and Pawley G. S., (1988), On stability of the Travelling Salesman Problem Algorithm of Hopfield and Tank. *Bio. Cybern.*, 58, 63-70.
- [9] Paielli R. A., (1988), Simulation tests of the optimization method of Hopfield and Tank using neural networks, *NASA Technical Memorandum* 101047.
- [10] Takefuji Y., and Lee K. C., (1991), Artificial neural networks for four-coloring problems and k-colorability problems, *IEEE Trans. on Circuits and Systems*, 38, 3, pp.326-333.
- [11] Takefuji Y., and Lee K. C., (1989), A near-optimum parallel planarization algorithm, *Science*, 245, 1221-1223.
- [12] Takefuji Y., Lee K. C., and Cho Y. B., (1991), Comments on "O(n<sup>2</sup>) algorithms for graph planarization," *IEEE Trans. on CAD*, 10m 12, 1582-1583.
- [13] Takefuji Y. and Lee K. C., (1990), A parallel algorithm for tiling problems, *IEEE Trans. on Neural Networks*, 1, 1, 143-145.
- [14] Takefuji Y. and Lee K. C., (1990), A super parallel sorting algorithm based on neural networks, *IEEE Trans. on Circuits and Systems*, 37, 11, 1425-1429.
- [15] Takefuji Y., Tanaka T., and Lee K. C., (1992), to appear in *IEEE Trans. on System, Man, and Cybernetics*.
- [16] Takefuji Y., Chen L. L., Lee K. C., and Huffman J., 1990 b, Parallel algorithms for finding a near-maximum independent set of a circle graph. *IEEE Trans. on Neural Networks*, 1, 3, 263-267.
- [17] Takefuji Y., Lin C. W., and Lee K. C., (1990), A parallel algorithm for estimating the secondary structure in Ribonucleic Acids. *Biological Cybernetics*, 63, 337-340.
- [18] Takefuji Y., Lee K. C., (1991), An hysteresis binary neuron : a model suppressing the oscillatory behaviors of neural dynamics. *Biological Cybernetics*. 64, 353-356.
- [19] Funabiki N. and Takefuji Y., (1991), A parallel algorithm for spare allocation problems, *IEEE Trans. on Reliability*, 40, 3, 338-346.
- [20] Funabiki N. and Takefuji Y., A parallel algorithm for time slot assignment problems in TDM hierarchical switching systems, to appear in *IEEE Trans. on Communications*.
- [21] Funabiki N. and Takefuji Y., (1992), A parallel algorithm for channel routing problems, to appear in *IEEE Trans. on CAD*.
- [22] Funabiki N. and Takefuji Y., (1992), A parallel algorithm for via-minimization problems, to appear in *IEEE Trans. on CAD*.
- [23] Funabiki N. and Takefuji Y., (1992), A neural network model for finding a near-maximum clique, to appear in *Journal of Parallel Distributed Computing*.
- [24] Lee K. C. and Takefuji Y. (1991), A ma-

- thematical model fusing the Ising model and the artificial neural network for the maximum clique problem, *CAISR technical report*.
- [25] Lee K. C., Funabiki N., and Takefuji Y., (1992), A parallel improvement algorithm for the bipartite subgraph problem, *IEEE Trans. on Neural Networks*, 3, 1, 139-145.
- [26] Lee K. C. and Takefuji Y., (1991), A maximum neural network for the max cut problem, *Proc. of IJCNN91-Seattle*.
- [27] Funabiki N., Takefuji Y., Lee K. C., Cho Y. B., (1992), A neural network parallel algorithm for clique vertex-partition problems, to appear in *International Journal of Electronics*.
- [28] Lee K. C., Takefuji Y., (1992), A generalized maximum neural network for the module orientation problem, to appear in *International Journal of Electronics*.
- [29] Takefuji Y., (1992), Chapter 1 entitled N-queen problems in *Neural Network Parallel Computing*, Kluwer Academic Publishers.
- [30] Funabiki N., Takefuji Y., (1991), A parallel algorithm for solving the Hip games, *Neurocomputing*, 3, 97-106.
- [31] Rofkar J., Takefuji Y., (1992), A parallel algorithm for solving unfriendly beehive problems, to appear in *Neurocomputing*.
- [32] Kurokawa T., Takefuji Y., (1992), A parallel algorithm for BIBD problems, to appear in *IEEE Trans. on Circuits and Systems*.
- [33] Takefuji Y., (1992), Chapter 7 entitled Knight's tour problems in *Neural Network Parallel Computing*, Kluwer Academic Publishers.
- [34] Foo S., Takefuji Y., (1988), Stochastic neural networks for solving job-shop scheduling, *Proc. of IEEE IJCNN '88*, 275-290; and Integer-linear programming neural networks for job-shop scheduling, *Proc. of IEEE IJCNN '88*, 341-348.
- [35] Takefuji Y., (1992), *Neural Network Parallel Computing*, Kluwer Academic Publishers'

平成4年度役員・支部長名簿

理事会 長	伊理 正夫(東京大学)	ビス(株)	
" 副会長	斎藤 嘉博(武蔵野美術大学)	" "	栗原 宏文(東燃システム研究所)
" "	高井 英造(三菱石油(株))	" "	藤井 進(神戸大学)
" "	権藤 元(近畿大学)	" "	伏見 正則(東京大学)
" 庶務	小池 清(日本アイ・ピー・エム(株))	監 事	三平 武男(川鉄システム開発(株))
" "	田口 東(中央大学)	" "	高橋 磐郎(日本大学)
" 会計	山田 郁夫(三菱電機(株))	支 部 長	
" 研究普及	中野 文平(東京工業大学)	北海道支部	関口 恭毅(北海道大学)
" "	香田 正人(日本アイ・ピー・エム(株))	東北支部	中澤 博司(東北電力(株))
" 編集	若山 邦紘(法政大学)	中部支部	田中 庸平(中部電力)
" "	茨木 俊秀(京都大学)	関西支部	藤井 進(神戸大学)
" 国際	腰塚 武志(筑波大学)	中国・四国支部	尾崎 俊治(広島大学)
" 無任所	山本 保(東北コンピュータ・サー	九州支部	岩本 誠一(九州大学)