

# 最小極大マッチング問題のニューラルネットワーク 並列解法の提案

梅谷 俊治<sup>†</sup> 船 曳 信 生<sup>†</sup> 西 川 清 史<sup>†</sup>

グラフ  $G = (V, E)$  の辺の部分集合  $M (\subseteq E)$  で、 $M$  のどの2辺も隣り合わないとき、 $M$  を  $G$  のマッチングと呼ぶ。本論文では、極大かつ要素数最小のマッチングを求める最小極大マッチング問題に対して、ニューラルネットワークを用いた並列解法を提案する。本問題は一般に NP 完全であることが知られている。本解法では、頂点数  $N$  のグラフに対し、 $N \times N$  の2次元ニューラルネットワークを用いる。頂点がマッチングに含まれないことを表す対角線上のニューロンが、制約条件を満たすときのみ発火するように働きかける項を導入することによって、収束率および解精度の向上を図っている。ランダムグラフを用いたシミュレーションにより、Greedy 解法、Simulated Annealing 解法より本解法の解精度が優れていることを示す。

## A Neural Network Parallel Algorithm for Minimum Maximal Matching Problems

SHUNJI UMETANI,<sup>†</sup> NOBUO FUNABIKI<sup>†</sup> and SEISHI NISHIKAWA<sup>†</sup>

For a given graph  $G(V, E)$ , a subset  $M$  of  $E$  is called a *maximal matching* of  $G$  if no pair of edges in  $M$  is adjacent to each other and no other matching  $M'$  such as  $M \subset M'$  exists for  $G$ . In this paper, we present a neural network parallel algorithm for the *minimum maximal matching problem*. This NP-complete problem requires to find a maximal matching composed of the least number of edges in  $G$ . The neural network consists of  $N \times N$  binary neurons for the  $N$ -vertex graph problem. In order to improve the solution quality, we introduce a heuristic term into the motion equation for a diagonal neuron which represents the exclusion of the vertex from the matching. We verify the performance through simulations using random graphs, where our neural network finds better solutions than the greedy algorithm and the simulated annealing.

### 1. はじめに

頂点の集合  $V$  および辺の集合  $E$  を持つグラフ  $G = (V, E)$  の辺の部分集合  $M (\subseteq E)$  で、 $M$  のどの2辺も隣り合わないとき、 $M$  はグラフ  $G$  のマッチング (Matching) と呼ばれる。グラフ  $G$  のあるマッチング  $M$  に対して、 $M \subset M'$  となるグラフ  $G$  のマッチング  $M'$  が存在しないとき、 $M$  はグラフ  $G$  の極大マッチングと呼ばれる。

グラフの極大マッチングを求める問題に対しては、複数の CREW PRAM 計算機モデルを用いた並列アルゴリズムが Lev<sup>1)</sup> によって提案されている。以後、Israeli<sup>2)</sup>, Luby<sup>3)</sup>, Kelsen<sup>4)</sup> 等によって本アルゴリ

ズムの改良が行われている。さらに、Chen<sup>5)</sup> によって Luby のアルゴリズムを基に  $m^3 \Delta$  ( $m$  はグラフの辺数、 $\Delta$  はグラフの頂点の最大次数) 個のプロセッサを用いて、 $O(\log^2 N)$  ( $N$  はグラフの頂点数) の計算時間で解くアルゴリズムが提案されている。

本論文で対象とするグラフ  $G$  の要素数最小の極大マッチングを求める問題は、最小極大マッチング (Minimum Maximal Matching Problem)<sup>6)</sup> と呼ばれる。最小極大マッチング問題は、一般のグラフについて NP 完全であることが知られている<sup>6)</sup>。特にすべての頂点の次数が3以下となる2部グラフや平面グラフに対しても NP 完全となることが知られている。逆に、要素数最大の極大マッチングを求める問題は最大マッチング問題と同義であり、多項式時間で解くアルゴリズムが Edmonds<sup>7)</sup> によって与えられている。

本論文では、グラフ  $G = (V, E)$  の最小極大マッチングを求める問題について、Hopfield 型ニューラル

<sup>†</sup> 大阪大学大学院基礎工学研究科情報数系専攻  
Department of Informatics and Mathematical Science  
Osaka Graduated School of Engineering Science, Osaka  
University

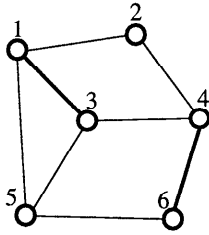


図1 最小極大マッチングの例

Fig. 1 An instance of a minimum maximal matching.

		→ j					
	$V_{ij}$	1	2	3	4	5	6
↓ i	1						
	2						
	3						
	4						
	5						
	6						

図2 図1に対応するニューラルネットワークの出力

Fig. 2 The neural network output for the instance in Fig.1.

ネットワーク<sup>8)</sup>を用いた並列解法を提案する。本解法では、すべてのニューロンで並列に計算を行うために、すべてのニューロンの状態を同時に更新する同期式更新を採用している<sup>9)</sup>。最小マッチング数が明らかなグラフおよびランダムグラフを用いたシミュレーションにより、本解法の性能評価を行う。

2. 最小極大マッチング問題の定義

最小極大マッチング問題は、以下に示す制約条件および目的条件を満たすグラフ  $G = (V, E)$  の辺  $E$  の部分集合  $M$  を見つける問題である。

● 制約条件

(1) マッチング条件

$M$  の各要素の辺が互いに隣り合わない。

(2) 極大条件

$M \subset M'$  となるマッチング  $M'$  が存在しない。

● 目的条件

制約条件を満たす  $M$  の要素数を最小にする。

図1に最小極大マッチングの例を示す。ここで、最小極大マッチングは  $M = \{\overline{13}, \overline{46}\}$  である。辺  $\overline{13}$  と辺  $\overline{46}$  は隣接していないので  $M$  はマッチング条件を満たす。また、 $M$  の要素でないどの辺を  $M$  に加えた場合でもマッチング条件を満たすことはできないので、 $M$  は極大条件を満たす。図1の6頂点グラフでは要素数1のマッチングは極大条件を満たさない。よって、 $M$  は最小極大マッチングとなる。

3. ニューラルネットワーク解法

3.1 ニューラルネットワーク表現

本解法では、対象となる  $N$  頂点グラフ  $G$  に対して、 $N \times N$  個のニューロンを持つ2次元ニューラルネットワークを使用する。 $i$  行  $j$  列のニューロン ( $i = 1 \sim N, j = 1 \sim N$ ) は入力  $U_{ij}$  と2値出力  $V_{ij}$  を持つ。本ニューラルネットワークでは、非対角線上のニューロン出力と対角線上のニューロン出力では異なる意味

を与えている。非対角線上のニューロン出力  $V_{ij} (i \neq j)$  は、

$$V_{ij} = \begin{cases} 1 & \text{辺 } \overline{ij} \text{ がマッチングに含まれる} \\ 0 & \text{辺 } \overline{ij} \text{ がマッチングに含まれない} \end{cases}$$

を表すこととする。ただし、 $V_{ij}$  と  $V_{ji}$  は同一の辺に対する出力なので、 $i > j$  となるニューロンの計算は行わず、出力  $V_{ij}$  には  $V_{ji}$  の値を代入することとする。また、対角線上のニューロン出力  $V_{ii}$  は、

$$V_{ii} = \begin{cases} 1 & \text{頂点 } i \text{ がマッチングに含まれない} \\ 0 & \text{頂点 } i \text{ がマッチングに含まれる} \end{cases}$$

を表すこととする。図2は図1のマッチングに対応するニューラルネットワークの出力を表している。すなわち、辺  $\overline{13}$  と  $\overline{46}$  がマッチングに含まれ、頂点2、5がマッチングに含まれないことを表している。

3.2 ニューロン関数

ニューロンの出力  $V_{ij}$  は、ニューロン関数と呼ばれる非減少関数により入力  $U_{ij}$  から求められる。本ニューラルネットワークでは、すべてのニューロン出力を同時に更新する同期式更新を用いるため、複数のニューロンが同時に状態変化を起こし、振動的になりやすい。さらに、ニューロン出力がつねに  $V_{ij} = V_{ji}$  となるため、状態変化を起こすニューロンの数が非常に多くなり、ニューラルネットワークが安定状態に収束しにくくなる。そのため、本ニューラルネットワークでは、ニューロンの振動現象を抑制する効果のあるヒステリシスバイナリ関数<sup>10)</sup>を採用する。ヒステリシスバイナリ関数を図3に示す。

ヒステリシスバイナリ関数は、

$$\begin{aligned} &\text{if } U_{ij} > \text{UTP then } V_{ij} = 1 \\ &\text{if } U_{ij} < \text{LTP then } V_{ij} = 0 \end{aligned}$$

で与えられる。文献 10) に従いニューロン入力の範囲を  $-20 \sim 20$  としたうえで、UTP, LTP の値は  $(\text{UTP}, \text{LTP}) = (15, -15)$  に設定する。シミュレシヨ

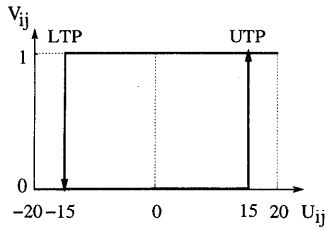


図3 ヒステリシスパイナリ関数  
Fig. 3 The hysteresis binary function.

ンより  $(UTP, LTP) = (5, -5), (10, -10)$  の場合には収束に要する更新回数は減少するものの、収束率が悪化することを確かめている。

### 3.3 エネルギー関数

エネルギー関数は、問題の制約条件および目標条件を表す項の線形和で表される非負関数である。

最小極大マッチング問題の制約条件であるマッチング条件は、各頂点につきマッチング  $M$  の要素の辺が1本だけ接続するか、またはまったく接続しないかのいずれかのときに満たされる。よって、マッチング条件を表すエネルギー関数  $E_1$  には、各行・各列で1つだけニューロンが発火しているときにのみ  $E_1 = 0$  となり、それ以外では  $E_1 > 0$  となる関数を選択する。ここで、出力が1となるニューロンの状態を発火と呼んでいる。マッチング条件のエネルギー関数  $E_1$  は式(1)となる。

$$E_1 = \sum_{i=1}^N \left( \sum_{k=1}^N d_{ik} V_{ik} - 1 \right)^2 + \sum_{j=1}^N \left( \sum_{k=1}^N d_{kj} V_{kj} - 1 \right)^2 \quad (1)$$

ここで、 $d_{ij}$  はグラフ  $G = (V, E)$  の隣接行列の  $i$  行  $j$  列の要素を表し、

$$d_{ij} = \begin{cases} 1 & \text{頂点 } i \text{ と頂点 } j \text{ を結ぶ辺がある} \\ 0 & \text{頂点 } i \text{ と頂点 } j \text{ を結ぶ辺がない} \end{cases}$$

である。ただし、対角線上の要素は  $d_{ii} = 1$  とする。また、グラフ  $G$  は無向グラフなので  $d_{ij} = d_{ji}$  となる。

極大条件は、マッチングに含まれていない頂点の任意の組合せについて、それらの頂点間を結ぶ辺が存在しないときに満たされる。極大条件のエネルギー関数  $E_2$  は式(2)となる。

$$E_2 = \sum_{i=1}^N \left( \sum_{\substack{k=1 \\ k \neq i}}^N d_{ik} V_{ik} V_{kk} \right) \quad (2)$$

目的条件は、マッチング  $M$  の要素数の最小化であるが、本ニューラルネットワークではマッチング  $M$  に含まれない頂点数を最大化することで、目的条件を実現する。よって、対角線上で  $V_{ii} = 1$  となるニューロンの数が最大のときに、最小値をとる関数を目的条件のエネルギー関数  $E_3$  とする。

$$E_3 = \sum_{i=1}^N (1 - V_{ii})^2 \quad (3)$$

式(1), (2), (3)より、最小極大マッチング問題のエネルギー関数は式(3)となる。各項の係数  $A, B, C$  は正の定数である。

$$E = \frac{A}{2} \sum_{i=1}^N \left( \sum_{k=1}^N d_{ik} V_{ik} - 1 \right)^2 + \frac{A}{2} \sum_{j=1}^N \left( \sum_{k=1}^N d_{kj} V_{kj} - 1 \right)^2 + \frac{B}{2} \sum_{i=1}^N \left( \sum_{\substack{k=1 \\ k \neq i}}^N d_{ik} V_{ii} V_{kk} \right) + \frac{C}{2} \sum_{i=1}^N (1 - V_{ii})^2 \quad (4)$$

### 3.4 動作方程式

動作方程式は、エネルギー関数の最小化を目的として各ニューロン入力の時変化を与える式である。本解法では最急降下法に基づく解の探索を行うため、動作方程式はエネルギー関数  $E$  をニューロン出力  $V_{ij}$  で偏微分することで与えられる。動作方程式を式(5)に示す。

if ( $i = j$ ) then

$$\frac{dU_{ii}}{dt} = -A \left( \sum_{k=1}^N d_{ik} V_{ik} - 1 \right) - A \left( \sum_{k=1}^N d_{ki} V_{ki} - 1 \right) - B \sum_{\substack{k=1 \\ k \neq i}}^N d_{ik} V_{kk} + C(1 - V_{ii})$$

else

$$\frac{dU_{ij}}{dt} = -A \left( \sum_{k=1}^N d_{ik} V_{ik} - 1 \right) - A \left( \sum_{k=1}^N d_{kj} V_{kj} - 1 \right) \quad (5)$$

### 3.5 動作方程式の改善

式(4)のエネルギー関数において目的条件を表す  $C$  項は通常解で 0 とならない。そのため、ニューラルネットワークの状態が解の近辺にあるときに式(5)の動作方程式の  $C$  項がニューラルネットワークが解に収束するのを妨げる働きをする。この改善として、本論文では  $C$  項が極大条件が満たされる ( $B$  項が 0 とする) ときのみ働くようにする。また、本動作方程式ではニューロン出力を 0 にする働きが強いためニューロンがまったく発火しない行や列が現れ、解に収束しない。そこで、ヒルクライミング項を  $D$  項として動作方程式に加えて、ニューロンがまったく発火しない行や列が現れたときにそれらのニューロンを強制的に発火させる。ただし、全ニューロン状態の同期式更新にともなう振動現象の抑制のため、本ニューラルネットワークでは  $D$  項に与える係数を各ニューロンごとにランダムに与えて、ニューロンの発火タイミングをずらしている。

if ( $i = j$ ) then

$$\begin{aligned} \frac{dU_{ii}}{dt} = & -A \left( \sum_{k=1}^N d_{ik} V_{ik} - 1 \right) \\ & -A \left( \sum_{k=1}^N d_{ki} V_{ki} - 1 \right) \\ & -B \sum_{\substack{k=1 \\ k \neq i}}^N d_{ik} V_{kk} \\ & + C(1 - V_{ii}) h \left( \sum_{\substack{k=1 \\ k \neq i}}^N d_{ik} V_{kk} \right) \\ & + Dh \left( \sum_{k=1}^N d_{ik} V_{ik} + \sum_{k=1}^N d_{ki} V_{ki} \right) \end{aligned}$$

else

$$\begin{aligned} \frac{dU_{ij}}{dt} = & -A \left( \sum_{k=1}^N d_{ik} V_{ik} - 1 \right) \\ & -A \left( \sum_{k=1}^N d_{kj} V_{kj} - 1 \right) \\ & + Dh \left( \sum_{k=1}^N d_{ik} V_{ik} + \sum_{k=1}^N d_{kj} V_{kj} \right) \quad (6) \end{aligned}$$

ここで、 $h(x)$  は

$$h(x) = \begin{cases} 1 & x = 0 \\ 0 & x \neq 0 \end{cases}$$

となる関数である。本解法では、各項の係数は  $A = 1$ ,  $B = 1$ ,  $C = 10$  とし、 $D$  は 1~6 を各更新回ごとにランダムに与える。ここで、シミュレーションにより  $C = 1$ , 5 の場合には若干解の精度が悪化し、 $C = 15$  の場合には更新回数が増加することを確かめている。また、各ニューロンの動作方程式およびニューロン関数の計算を並列に実行し、更新回数の上限は 3000 回とする。

## 4. シミュレーション結果と考察

### 4.1 ニューラルネットワーク解法の最適解探索能力

提案するニューラルネットワーク解法が最適解を探索可能であることを示すために、最小マッチング数が明らかなグラフを作成し、シミュレーションを行った。本グラフでは、図 4 に示すように初めに解となるマッチングの要素を選んでおき、マッチングに含まれない任意の頂点の間を結ぶ辺を除いてランダムに辺を生成する。このため、初めに選んだマッチングは作成したグラフの最適解の 1 つとなる。

頂点数  $N = 100, 150, 200$  の最小マッチング数が明らかなグラフを作成し、ニューラルネットワーク解法によるシミュレーションを異なる初期ニューロン入力を用いて各々 100 回ずつ行った。このとき、すべてのシミュレーションでニューラルネットワークは制約条件を満たす解に収束した。各シミュレーションにおいて最適解が得られた回数 (正答率) と平均マッチング数 (平均値) を表 1 に示す。表 1 から、本論文で提案するニューラルネットワーク解法により、つねに最適解もしくは最適解に非常に近い解が得られることが分かる。

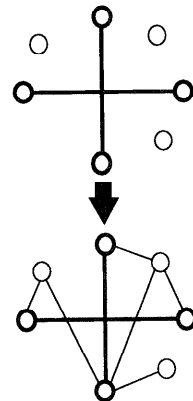


図 4 最小マッチング数が明らかなグラフ  
Fig. 4 A graph with a given minimum maximal matching.

表1 最小マッチング数が明らかなグラフに対するシミュレーション結果

Table 1 Simulation results for graphs with given minimum maximal matchings.

頂点数	最小解	正答率 (%)	平均値
100	11	86	11.14
150	16	93	16.07
200	21	93	21.07

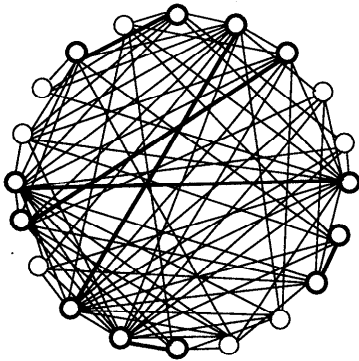


図5 20頂点グラフ問題の解の例

Fig. 5 A minimum maximal matching solution for a 20-vertex graph problem.

#### 4.2 ニューラルネットワーク解法と他解法の性能比較

頂点数  $N = 20 \sim 500$  で、辺を完全グラフに対して発生率 10%, 50% の 2 通りでランダムに発生させたグラフを作成し、本論文で提案するニューラルネットワーク解法のシミュレーションを行った。

本解法の性能比較のため、Greedy 解法、Simulated Annealing 解法もあわせて実装し、それらのシミュレーションも実施した。ここで、Greedy 解法では、現在得られているグラフ  $G = (V, E)$  のマッチング  $M$  の要素の辺とそれらに隣接する辺からなる集合を  $S$  とする。このとき、集合  $S$  に含まれない各辺について、隣接する辺から集合  $S$  に含まれる辺を除いた数をその辺の cover 数とし、cover 数の大きい辺から順次マッチングの要素に加えていく。

解の一例として、頂点数 20、辺の発生率 50% のランダムグラフにおいてニューラルネットワーク解法で得られた最良解を図 5 に示す。ニューラルネットワーク解法と Greedy 解法は各々 100 回ずつ、Simulated Annealing 解法は 1 回のみ SONY/NEWS5000 計算機上で、シミュレーションを行った。本ニューラルネットワーク解法が、各シミュレーションで解に収束した割合 (%) および収束時の平均更新回数を表 2 に示

表2 ランダムグラフに対するニューラルネットワーク解法の収束性能

Table 2 The convergence performance of the neural network algorithm for random graphs.

辺の発生率	頂点数	収束率 (%)	平均更新回数
10 %	20	100	234
	50	87	558
	100	98	565
	200	99	711
	300	97	843
	400	98	815
50 %	500	99	783
	20	98	142
	50	99	284
	100	99	343
	200	93	406
	300	90	390
400	84	436	
500	87	450	

す。表 2 から分かるように、ニューラルネットワーク解法は、頂点数  $N = 20 \sim 500$  のグラフに対してつねに 80% 以上の確率で解を得ることができる。また、収束までにかかる平均更新回数も、グラフの頂点数に関係なく定数で抑えられることが分かる。

次に、表 3 にランダムグラフに対して、各解法のシミュレーションを行ったときの解のマッチング数を示す。表 3 の N/A は、Simulated Annealing 解法の計算時間が非常に長く、解を得ることができなかったことを表す。表 3 より、ニューラルネットワーク解法が最小マッチング数、平均マッチング数ともに他の解法に比べ優れていることが分かる。しかし、図 6 に示すように辺発生率が高くなるに従って、ニューラルネットワーク解法と Greedy 解法の最小マッチング数の差は小さくなる傾向が見られる。これは、最小極大マッチング問題ではマッチング数の上限が辺の発生率に関係なく定義よりグラフの頂点数の半分である一方で、辺発生率が大きくなるに従って最適解のマッチング数が増加するためである。そこで、頂点数 200 辺発生率 10%, 50% の場合のニューラルネットワーク解法と Greedy 解法との平均マッチング数の差について、それぞれ有意水準 5% で検定を行い<sup>11)</sup>、辺発生率 10% では差が有意であり、辺発生率 50% では差が有意でないことを確かめた。さらに詳しく検討を行うため頂点数 200 辺発生率 10%, 50% の場合の解の分布を図 7、図 8 に示す。図 7 に比べ図 8 の方が全体的に解のばらつきが小さく、そのためニューラルネットワーク解法と Greedy 解法との平均マッチング数の差が有意でなかったと考えられる。また図 8 では差は小さいものの、ニューラルネットワーク解法の方が Greedy 解法

表3 ランダムグラフに対する各解法の解のマッチング数  
Table 3 Simulation results by three algorithms for random graphs.

辺の発生率	頂点数	Neural Network		Greedy		SA
		最小	平均	最小	平均	最小
10%	20	5	5.0	5	5.5	5
	50	14	15.0	16	17.1	18
	100	34	36.1	38	40.5	43
	200	80	82.4	86	89.4	N/A
	300	128	130.2	134	137.2	N/A
	400	177	179.2	183	186.5	N/A
50%	20	6	6.5	7	7.0	8
	50	22	22.2	22	22.8	24
	100	46	46.7	46	47.8	N/A
	200	95	96.0	96	97.5	N/A
	300	145	145.8	146	147.4	N/A
	400	195	195.7	196	197.0	N/A
500	244	245.5	245	247.2	N/A	

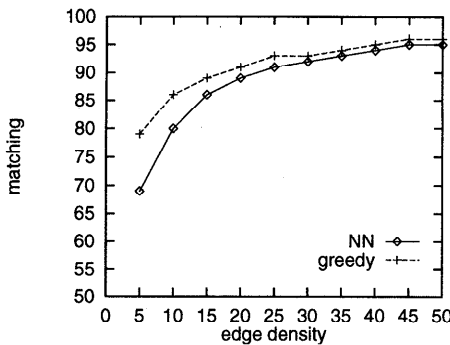


図6 辺発生率に対する最小マッチング数 ( $N = 200$ )  
Fig. 6 Minimum solutions for a 200 vertex 10 ~ 50% edge density random graph.

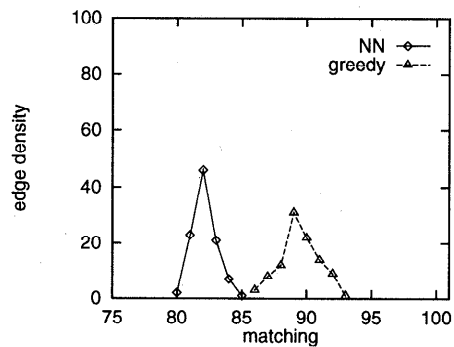


図7 解の分布 ( $N = 200$ , 辺発生率 10%)  
The distribution of solutions for a 200 vertex 10% edge density random graph.

より優れているといえる。なお、辺発生率 30%まではニューラルネットワーク解法と Greedy 解法の差が有意水準 5%で有意であることを検定により確認している。Simulated Annealing 解法は制約条件を満たすように組替えを行うため、組替えの前後でマッチング数の変化がたかだか  $\pm 1$  しかない。そのため、初期状態からの変化がほとんど起こらず、他の解法と比べて結果が悪くなると考えられる。

最後に、図 9、図 10 に、各解法の計算時間を示す。グラフ  $G = (V, E)$  に対して、ニューラルネットワーク解法では各更新ごとに頂点数  $|V|$  の 2 乗の数のニューロン計算を行い、また平均更新回数は定数で抑えられているので、全体の計算量は  $O(|V|^2)$  となる。Greedy 解法では、マッチングの要素を 1 つ決定するたびに集合  $S$  に含まれないすべての辺について cover 数を計算する。マッチング数は頂点数  $N$  で抑えられるので、Greedy 解法の計算量は  $O(|V|^3)$  となる。シミュレーションでの計算時間もこの計算量の評価と一致してい

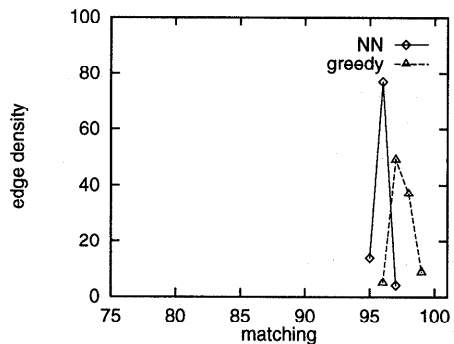


図8 解の分布 ( $N = 200$ , 辺発生率 50%)  
The distribution of solutions for a 200 vertex 50% edge density random graph.

る。すなわち、辺の発生率が 10%のときには、Greedy 解法が他の解法に比べて速い。しかし、辺の発生率が 50%のときには、Greedy 解法はニューラルネットワーク解法に比べ非常に遅くなっている。

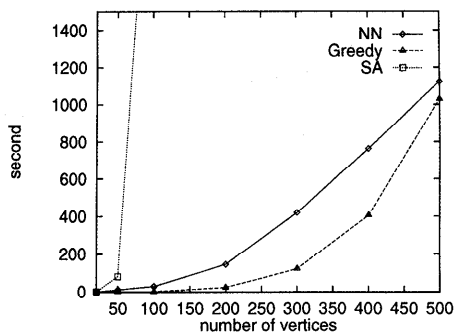


図9 辺発生率10%のランダムグラフに対する計算時間  
Fig. 9 Computation time by three algorithms for 10% edge-density random graphs.

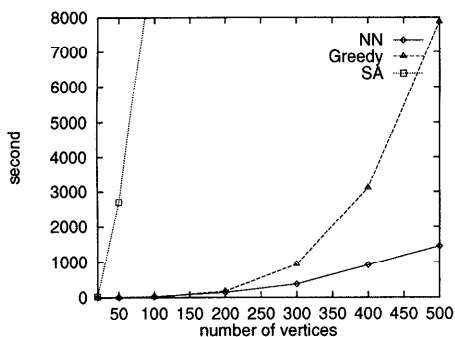


図10 辺発生率50%のランダムグラフに対する計算時間  
Fig. 10 Computation time by three algorithms for 50% edge-density random graphs.

## 5. まとめ

本論文では、最小極大マッチング問題に対してニューラルネットワークを用いた並列解法を提案した。本解法では、頂点数  $N$  のグラフに対して  $N \times N$  の2次元ニューラルネットワークを用いた。

本ニューラルネットワーク解法の特長は、頂点がマッチングに含まれないことを表す対角線上のニューロンが極大条件を満たすときのみ発火するように働きかけていることである。これにより、解への収束率を高め、より精度の高い解を得ることを可能にしている。

ニューラルネットワーク解法の性能評価を行うため、計算機上でシミュレーションを行った。始めに、最小マッチング数が明らかなグラフに対してシミュレーションを行い、本解法がつねに最適解に非常に近い解を求めることが可能なことを示した。次に、辺をランダムに発生させたグラフに対して Greedy 解法, Simulated Annealing 解法のシミュレーションをあわせて行い、大規模な問題に対してはニューラルネットワーク解法がこれらの解法より優れた求解性能を持つことを示

した。

今後は、幾何学的グラフなど一様ランダムグラフ以外を対象としたシミュレーションを行い、本解法の有効性を明らかにする予定である。

## 参考文献

- 1) Lev, G.: Size bounds and parallel algorithms for networks, Technical report, Dept. Computer Science, Univ. Edinburgh (1980).
- 2) Israeli, A. and Shiloach, Y.: An improved parallel algorithm for maximal matching, *Inform. Process. Lett.*, Vol.22, pp.57-60 (1986).
- 3) Luby, M.: A simple parallel algorithm for the maximal independent set problem, *SIAM J. Comput.*, Vol.15, pp.1036-1053 (1986).
- 4) Kelsen, P.: An optimal parallel algorithm for maximal matching, *Inform. Process. Lett.*, Vol.52, pp.223-228 (1994).
- 5) Chen, Z.Z.: A fast and efficient NC algorithm for maximal matching, *Inform. Process. Lett.*, Vol.55, pp.303-307 (1995).
- 6) Garey, M.R. and Johnson, D.S.: *Computers and intractability: A guide to the theory of NP-completeness*, W.H. Freeman and Company (1991).
- 7) Bondy, J.A. and Murty, U.S.: *Graph theory with applications*, The Macmillan Press LTD (1976).
- 8) Hopfield, J.J. and Tank, D.W.: Neural computation of decisions in optimization problems, *Biol. Cybern.*, Vol.52, pp.141-152 (1985).
- 9) 由雄宏明, 船曳信生, 西川清史: N-Queen 問題を対象としたニューラルネットワーク解法の並列アルゴリズムに関する研究, 信学技報, Vol.NC95-84, pp.75-82 (1995).
- 10) Takefuji, Y. and Lee, K.C.: An artificial hysteresis binary neuron: A model suppressing the oscillatory behaviors of neural dynamics, *Biol. Cybern.*, Vol.64, pp.353-356 (1991).
- 11) 浅井 晃, 村上正康: 初等統計学, 培風館 (1963).

(平成8年12月24日受付)

(平成9年12月1日採録)

**梅谷 俊治 (学生会員)**

平成 8 年大阪大学基礎工学部情報工学科中退。平成 8 年同大大学院博士前期課程入学。現在博士前期課程に在学。ニューラルネットワーク、組合せ最適化問題に関する研究に従事。

IEEE, 電子情報通信学会, 人工知能学会各会員。

**船曳 信生 (正会員)**

昭和 59 年東京大学工学部計数工学科卒業。同年住友金属工業(株)入社。平成 3 年米国ケースウエスタンリザーブ大学大学院修士課程了。平成 6 年大阪大学基礎工学部情報工

学科講師。現在, 同助教授。工学博士。ニューラルネットワーク, 組合せ最適化問題, プロセス制御等に関する研究に従事。IEEE, 電子情報通信学会各会員。

**西川 清史 (正会員)**

昭和 40 年東京大学工学部電子工学科卒業。昭和 45 年同大大学院工学系研究科博士課程修了。工学博士。同年日本電信電話公社(現 NTT)入社。同社基礎研究部, 情報通信処理研究所, コミュニケーション科学研究所所長等を経て, 平成 5 年より大阪大学基礎工学部情報工学科教授となり現在に至る。ハードウェア設計法, 並列アーキテクチャ, 最適化問題の解法等の研究に従事。電子情報通信学会, IEEE, ACM 各会員。