

マキシマムニューロンを用いた N-Queen 問題のニューラルネット解法の提案

竹 中 要 一[†] 船 曳 信 生[†] 西 川 清 史[†]

本論文では、マキシマムニューロンを用いた N-Queen 問題の解法を提案する。マキシマムニューロンは、ニューラルネットワークによる組合せ最適化問題の効率的解法を目的として、Takefuji らによって提案されたニューロンモデルである。マキシマムニューロンでは、解空間を構成するニューロンをグループに分割し、各グループ内でただ 1 つのニューロンのみが発火する。これにより、探索空間の大幅な縮小、ニューロン状態更新に必要な計算量の減少を実現している。本論文では、シミュレーションにより、マキシマムニューロンによる解法が、従来のニューラルネットワーク解法より優れた求解性能を有することを示す。特に、本解法が準同期式並列計算に非常に適した方法であることを明らかにする。

Maximum Neural Network Algorithms for N-Queen Problems

YOICHI TAKENAKA,[†] NOBUO FUNABIKI[†] and SEISHI NISHIKAWA[†]

This paper presents maximum neural network algorithms for N-queen problems using the maximum neuron model. The maximum neuron model is proposed by Takefuji et al. in order to provide efficient neural network solutions for combinatorial optimization problems. In this model, one and only one neuron is always fired in each group of neurons, which can not only reduce the searching space drastically but also save the computation load. The simulation results show that the performance of our maximum neural network surpasses the existing neural network for the same problem. Particularly, the maximum neural network is shown to be much suitable for the semi-synchronous computation.

1. ま え が き

組合せ最適化問題は、LSI 自動設計、ネットワーク制御等、工学分野への応用上重要な多くの問題を含んでいる。しかし、それらの多くは NP 完全のクラスに属しているため、多項式時間で解を求めるアルゴリズムの存在は期待し難い¹⁾。そこで、実用上、膨大な時間を掛けて厳密解を得る代わりに、短時間で近似解を得たい場合が生じてくる。ニューラルネットワークは、このような組合せ最適化問題の近似解を得る有力な手法のひとつとして研究が進められている²⁾。

ニューラルネットワークの特長は、汎用性と並列性にある。ニューラルネットワークでは、エネルギー関数と呼ばれる評価関数（コスト関数）の変更により、種々の組合せ最適化問題に適用可能である。また、ニューラルネットワークは、ニューロンと呼ばれる演算ユニットの集合体であるため、並列処理が可能である³⁾。こ

れらの性質が、今後発展の予想されるマルチプロセッサ環境への対応を可能としている。

マキシマムニューロンは、ニューラルネットワークによる組合せ最適化問題の効率的解法の実現のために Takefuji, Lee らによって提案されたニューロンモデルである^{4)~6)}。マキシマムニューロンでは、解空間を構成するニューロンをグループに分け、各グループ内でただ 1 つのニューロンのみが発火する。グラフ彩色問題⁷⁾、N-Queen 問題^{8),9)}等、ニューラルネットワークの対象としてきた組合せ最適化問題には、ある特定のグループから 1 つの状態を選択するという制約条件（グループ選択条件）を有する問題が多数存在する。マキシマムニューロンは、ニューロンレベルでグループ選択条件を満足させることにより、探索空間の大幅な縮小を実現している。

マキシマムニューロンは、Takefuji, Lee らによりこれまでに総配線長が最短となるモジュール配置方向を求める Module Orientation 問題⁵⁾、グラフの最大 2 部グラフを求める Bipartite Subgraph 問題⁴⁾に適用されている。この 2 つの組合せ最適化問題は、1 種

[†] 大阪大学基礎工学部

Faculty of Engineering Science, Osaka University

類のグループ選択条件と最適化すべき目的関数とから構成される。グループ選択条件はマキシмумニューロンによって完全に実現できるため、ニューラルネットワークの探索機能は目的関数の最適化のみを行えばよいこととなる。たとえば Bipartite Subgraph 問題では、各頂点を2部グラフのどちらかに分割する制約条件（グループ選択条件）をマキシмумニューロンで実現し、カット辺の最大化（目的関数の最大化）にニューラルネットワークの探索機能を用いるのである。

しかし、巡回セールスマン問題や N-Queen 問題のように、グループ選択条件を2種類以上有する組合せ最適化問題も数多く存在する。たとえば巡回セールスマン問題のニューラルネットワーク解法では、各都市について訪問順を1つ選択する条件と、各訪問順について都市を1つ選択する条件を同時に満たさなければならない²⁾。これらの問題では、 N 都市より1都市を選択するため、定数個より1個を選択していた従来のマキシмумニューロンの手法と同義的に扱うことはできない。本論文では、N-Queen 問題を対象として、 N 個より1個を選ぶ選択条件を2種類有する組合せ最適化問題に対するマキシмумニューロンの適用方法を提案し、シミュレーションによりその評価を行う。

N-Queen 問題は、 $N \times N$ 個の罫目のチェス盤上に N 個の Queen をそれぞれが他の Queen の利き筋に当たらないように配置する問題である。N-Queen 問題のバイナリー型ニューロンを用いたニューラルネットワーク解法は、Takefuji¹⁰⁾や Mandziuk⁸⁾によって提案されている。このうち、Takefuji の解法が求解性能で優れていることが明らかにされている⁹⁾。本論文では従来の代表的解法である Takefuji の解法を基にして、マキシмумニューロンを用いた解法を提案する。シミュレーションにより、本解法が Takefuji の解法より優れていることを示す。

2. 従来の解法 (Takefuji の解法)

2.1 N-Queen 問題の定義

N-Queen 問題とは、 $N \times N$ の罫目のチェスボード上に、 N 個の Queen を互いに攻撃しあわないように配置する問題である。Queen は図1に示すように縦、横、斜めに移動できるため、本問題では次の4つの制約条件を同時に満足することが要求される。

- (1) 各行に必ず1個の Queen を配置する
- (2) 各列に必ず1個の Queen を配置する
- (3) 右斜めに2個以上の Queen を配置しない
- (4) 左斜めに2個以上の Queen を配置しない

N-Queen 問題の目標は、この4つの制約条件を同

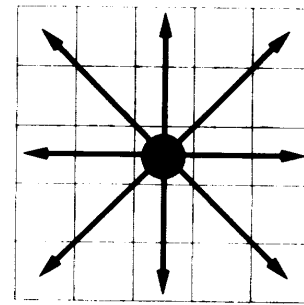


図1 Queen の動作
Fig.1 Motion of a Queen.

時に満たす解を見つけることである。

2.2 ニューロンモデルとニューロン表現

各ニューロンは、入力値 U と出力値 V を有する。ニューロンの出力値に0または1の値をとるバイナリー関数型ニューロンが採用されている。出力値 V は、入力値 U の閾値判定により決定される。ニューロンが出力値 V に1を持つ状態を、通常ニューロンの発火と呼んでいる。従来の解法では、後述するニューロンの状態更新方法によって、閾値が1つであるバイナリーニューロンと、閾値が2つあるヒステリシスバイナリーニューロンが使い分けられている(図2)。

$N \times N$ のチェスボードにおける N-Queen 問題に対しては、 $N \times N$ の2次元ニューラルネットワークを採用している。 i 行 j 列のニューロン出力 $V_{ij} = 1$ は、チェスボードの i 行 j 列の罫目に Queen を配置することを、 $V_{ij} = 0$ は、配置しないことを表現するものと約束している。

i 行 j 列のニューロンの入出力の関係は、バイナリーニューロンでは、

$$\text{if } U_{ij} > 0 \text{ then } V_{ij} = 1 \text{ else } V_{ij} = 0 \quad (1)$$

で与えられ、ヒステリシスバイナリーニューロンでは、

$$\text{if } U_{ij} > \text{UTP} \text{ then } V_{ij} = 1$$

$$\text{if } U_{ij} < \text{LTP} \text{ then } V_{ij} = 0 \quad (2)$$

で与えられる。UTP, LTP は、2種類の閾値を表し、UTP = 3, LTP = 0 が使用されている。

2.3 エネルギー関数

エネルギー関数は、組合せ最適化問題のすべての制約条件を表す非負関数である。N-Queen 問題の4つの制約条件に対し、4つの項からなるエネルギー関数 E を採用している。

$$E = \frac{A_1}{2} \sum_{i=1}^N \left(\sum_{k=1}^N V_{ik} - 1 \right)^2$$

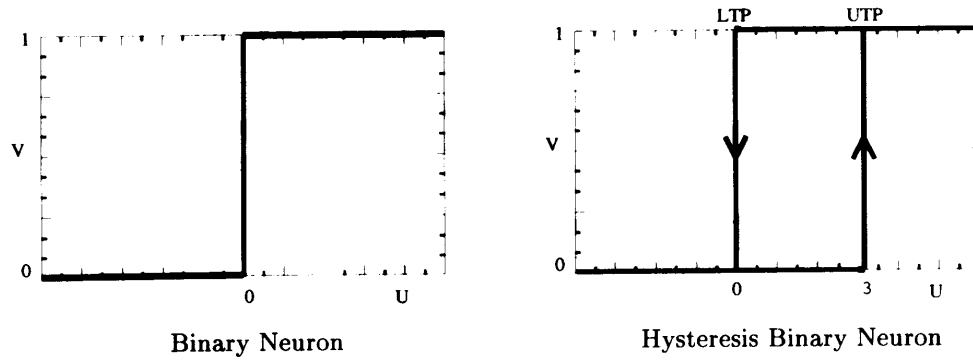


図2 2種類のバイナリー型ニューロンモデル
Fig. 2 Two binary neuron models.

$$\begin{aligned}
 & + \frac{A_2}{2} \sum_{j=1}^N \left(\sum_{k=1}^N V_{kj} - 1 \right)^2 \\
 & + \frac{B_1}{2} \sum_{i=1}^N \sum_{j=1}^N V_{ij} \sum_{k \neq 0, 1 \leq i+k, j+k \leq N} V_{i+k, j+k} \\
 & + \frac{B_2}{2} \sum_{i=1}^N \sum_{j=1}^N V_{ij} \sum_{k \neq 0, 1 \leq i+k, j-k \leq N} V_{i+k, j-k}
 \end{aligned} \quad (3)$$

A_1 項は、各行に必ず1つ Queen を配置する条件を表している。各行の N 個のニューロンのうち、1つのニューロンが出力1を持つ場合にのみ0となる。 A_2 項は同様に各列に必ず1つ Queen を配置する条件を表している。 B_1 項は右斜めに2つ以上 Queen を配置しない条件を表している。右斜め方向の樹目に対応したニューロンのうち、出力1を持つニューロンの個数が1つ以内の場合0となる。同様に B_2 項は、左斜めに2つ以上 Queen を配置しない条件を表している。エネルギー関数の値が0であるときに、ニューロンの出力値は制約条件をすべて満たした解を表現している。

2.4 動作方程式

エネルギー関数の値を0とするニューロン出力値の探索のために、最急降下法である動作方程式を用いる。すなわち、エネルギー関数式(3)の右辺を、ニューロン出力 V_{ij} により偏微分することにより、 i 行 j 列のニューロンにおける動作方程式を求める。

$$\begin{aligned}
 \frac{dU_{ij}}{dt} &= -\frac{\partial E}{\partial V_{ij}} \\
 &= -A_1 \left(\sum_{k=1}^N V_{ik} - 1 \right) - A_2 \left(\sum_{k=1}^N V_{kj} - 1 \right)
 \end{aligned}$$

$$\begin{aligned}
 & -B_1 \left(\sum_{k \neq 0, 1 \leq i+k, j+k \leq N} V_{i+k, j+k} \right) \\
 & -B_2 \left(\sum_{k \neq 0, 1 \leq i+k, j-k \leq N} V_{i+k, j-k} \right)
 \end{aligned} \quad (4)$$

ここで、 A_k, B_k は $A_k = B_k = 1$ ($k = \{1, 2\}$) を満たす係数である。

ニューラルネットワークでは、局所解への収束が問題となる。そこで Takefuji は、局所解脱出のために、Hill-Climbing 項を動作方程式に追加している。 i 行 j 列のニューロンに対する Hill-Climbing 項は、

$$+ C_1 h \left(\sum_{k=1}^N V_{ik} \right) + C_2 h \left(\sum_{k=1}^N V_{kj} \right) \quad (5)$$

で与えられる。ここで、関数 $h(x)$ は、

$$h(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

で定義される。

C_1 項は各行に1つも Queen が配置されていない場合に Queen を配置するように働きかける項である。 C_2 項は各列に対して同様の働きかけを行う項である。

C_k ($k = 1, 2$) は、

$$\begin{aligned}
 & \text{if } (t \bmod 20 < 5) \\
 & \text{then } C_k = 4 \text{ else } C_k = 1
 \end{aligned} \quad (7)$$

で与えられる係数である。 t は、動作方程式を一次オイラー法で解く場合の更新回数を表す。

これをまとめると動作方程式は式(8)のように表される。

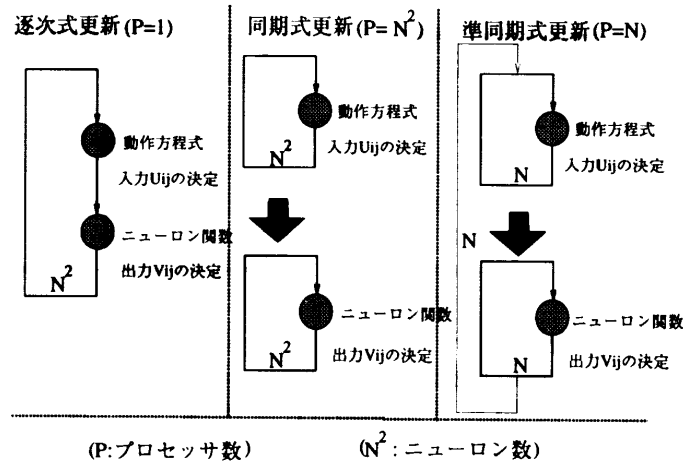


図3 3種類の状態更新方法

Fig. 3 Three methods of updating neuron states.

$$\begin{aligned}
 \frac{dU_{ij}}{dt} = & -A_1 \left(\sum_{k=1}^N V_{ik} - 1 \right) - A_2 \left(\sum_{k=1}^N V_{kj} - 1 \right) \\
 & - B_1 \left(\sum_{k \neq 0, 1 \leq i+k, j+k \leq N} V_{i+k, j+k} \right) \\
 & - B_2 \left(\sum_{k \neq 0, 1 \leq i+k, j-k \leq N} V_{i+k, j-k} \right) \\
 & + C_1 h \left(\sum_{k=1}^N V_{ik} \right) + C_2 h \left(\sum_{k=1}^N V_{kj} \right)
 \end{aligned} \quad (8)$$

ただし、ニューロン入力値には、上下制限を行っている。これは、ニューラルネットワークの状態変化を生じやすくすることにより、解への収束能力を高めるためである。

if $U_{ij} > U_{\max}$ then $U_{ij} = U_{\max}$

if $U_{ij} < U_{\min}$ then $U_{ij} = U_{\min}$ (9)

ここで、 $U_{\max}$ および $U_{\min}$ は、それぞれニューロン入力値の最大値、最小値である。

2.5 状態更新方法

ニューラルネットワークでは、動作方程式によりニューロン入力値を更新し、ニューロン関数により出力値を更新する。ニューロン入力値に乱数を用いて初期値を与えた後、入出力値の更新を繰り返すことにより、ニューラルネットワークは解の探索を行う。このとき、同時に更新を行うニューロン数の違いによって、逐次式、同期式、準同期式の3種類の更新方法が存在する(図3)。

逐次式更新では、同時に1つのニューロンの状態のみを更新する。同期式更新では、同時にすべてのニューロンの状態を更新する。すなわち、すべてのニューロ

ンの入力値の更新後、すべてのニューロンの出力値を更新する。準同期式更新では、 N^2 個のニューロンを行ごとに N 個のグループに分け、各グループに属する N 個のニューロン状態を同時に更新し、グループ間は逐次的に更新する。同期式並列計算機の最大プロセッサ数に対応させた場合、逐次式はプロセッサ数1に、同期式は N^2 、準同期式では、 N にそれぞれ対応する。

従来の解法では、更新方法によって、ニューロンモデルおよびニューロン入力値の上下限值を使い分けている。すなわち、逐次式更新では、バイナリーニューロンを用い、上下限值を $U_{\max} = 15, U_{\min} = -5$ 、としている。同期式、準同期式更新では、ヒステリシスバイナリーニューロンを用い、上下限值を $U_{\max} = 15, U_{\min} = -20$ としている。

3. マキシマムニューロンを用いた解法

3.1 マキシマムニューロンモデル

マキシマムニューロンモデルでは、解空間を構成するニューロン全体を互いに素なグループに分け、グループ内で入力値最大のニューロンのうち1つのみが出力値1をとる。N-Queen問題では、 N^2 個のニューロンを各行ごとに N 個のグループに分け、各グループに所属する N 個のニューロンの中で、入力値最大のニューロンが出力値1をとるものとする。これを式で表すと、

$$V_{ij} = \begin{cases} 1 & \text{if } U_{ij} = \max(U_{i1}, U_{i2}, \dots, U_{iN}) \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

である。ただし、関数 $\max()$ は、値が最大の引数を返すものとする。

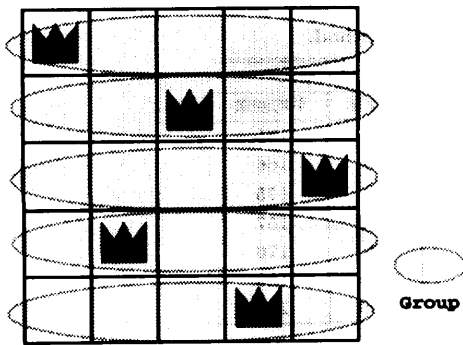
図4 グループの構成例 ($N=5$)

Fig. 4 An example of neuron group configuration.

入力値最大のニューロンが2つ以上存在する場合でも、マキシマムニューロンの定義より、1つのニューロンのみが選択されなければならない。我々は以下の手続きに沿って選択するニューロンを決定した。

- (1) 直前に出力値1であったニューロンを選択する。
- (2) 直前に出力値1のニューロンのニューロンが選択候補にない場合、添字の小さいニューロンを選択する。

図4に5-Queen問題におけるマキシマムニューロンの構成例を示す。

3.2 動作方程式と状態更新方法

マキシマムニューロン導入による求解性能への影響を明らかにするため、動作方程式、3種類の状態更新方法、およびニューロン入力値の上下制限は、基本的に従来の解法のものを採用する。ただし、マキシマムニューロンを用いることにより、式(8)のうち、行の制約条件を表す A_1 項、 C_1 項は不要となるため、式(11)の動作方程式を用いる。

$$\begin{aligned} \frac{dU_{ij}}{dt} = & -A_2 \left(\sum_{k=1}^N V_{kj} - 1 \right) \\ & -B_1 \left(\sum_{k \neq 0, 1 \leq i+k, j+k \leq N} V_{i+k, j+k} \right) \\ & -B_2 \left(\sum_{k \neq 0, 1 \leq i+k, j-k \leq N} V_{i+k, j-k} \right) \\ & +C_2 h \left(\sum_{k=1}^N V_{kj} \right) \end{aligned} \quad (11)$$

3.3 マキシマムニューロンの特徴

従来の解法と比較した場合、マキシマムニューロンを用いた本解法の長所として以下の2つがあげられる。

- (1) 各ニューロンの入力値の更新1回あたりの計算量を約30%削減できる。
- マキシマムニューロンを用いることにより、動

作方程式の A_1 項、 C_1 項を削減できているためである。ただし、マキシマムニューロンの更新には、同一グループ N 個のニューロン間の出力値の比較が必要なため、逐次式更新方法では計算量が多くなってしまふ。

- (2) 探索空間を大幅に縮小可能である。

マキシマムニューロンを用いることにより、発火しているニューロン数が固定されるためである。従来のニューラルネットワークでは、各ニューロンが独立に発火できるため、探索空間の大きさは 2^{N^2} であったが、マキシマムニューロンを用いた場合、各行に1つしか発火できないため、探索空間は N^N に縮小される。これを8-Queen問題にあてはめてみると、従来の探索空間が 2^{64} であったものが $8^8 = 2^{24}$ と、格段に減少していることが分かる。

しかしながら、マキシマムニューロンを用いることにより以下の2点が懸念される。

- (1) 行と列の制約条件に関する対称性の崩れにともない、収束性能が悪化する。

N-Queen問題では行と列の制約条件間に対称性がある。しかし、本解法では、行の制約条件をマキシマムニューロンによって実現し、列の制約条件を動作方程式で実現するため、行と列との対称性が崩れてしまふ。そのために、ニューラルネットワークの解探索能力に支障をきたす恐れがある。

- (2) ニューロン間競合の減少により、解探索能力が低下する。

ニューラルネットワークでは、多くのニューロンが互いに競合しながら、解の探索を行う。マキシマムニューロンでは、発火しているニューロンの個数が固定されるため、競合が不十分となる恐れがある。

本論文では、シミュレーションにより、探索空間の縮小が非常に効果的であり、対称性の崩れやニューロン間競合の減少は求解性能の悪化をともしないことを確認した。

4. シミュレーション結果と考察

4.1 シミュレーション条件と結果

従来の解法、および、提案するマキシマムニューロンを用いた解法の性能比較を行うため、 $N=20 \sim 300$ の7種類のN-Queen問題に対するシミュレーションを行った。各問題に対し、異なるニューロン入力値の初期値を用いて100回の試行を行い、収束率、および

表 1 従来の解法のシミュレーション結果
Table 1 Simulation results of Takefuji's method.

N	逐次式		同期式		準同期式	
	収束率	平均回数	収束率	平均回数	収束率	平均回数
20	100%	78 (回)	57	169	55	185
30	100	68	54	381	76	166
50	100	56	62	107	95	175
100	100	61	44	82	100	157
150	100	61	13	103	98	176
200	100	71	0	-	100	189
300	100	77	0	-	100	201

表 2 本解法のシミュレーション結果
Table 2 Simulation results of our method.

N	逐次式		同期式		準同期式	
	収束率	平均回数	収束率	平均回数	収束率	平均回数
20	24%	43 (回)	84	129	100	122
30	36	43	96	172	100	113
50	63	55	97	154	100	64
100	92	55	98	199	100	47
150	100	45	99	245	100	39
200	100	48	93	271	100	40
300	100	47	94	335	100	51

平均更新回数を求めた。収束率とは、100 回の試行のうち、収束した割合を表す。平均更新回数とは収束した場合の更新回数の平均値である。ただし、最大更新回数を 1000 回とし、それまでに各ニューロンが収束しなければ収束失敗と見なした。また、ニューロン入力値の初期値には $0 \sim U_{\min}$ の間の一様乱数を用いた。状態更新方法は、逐次式、同期式、準同期式の 3 種類の方法で行った。準同期式において、各行を同時に、各列を逐次的に更新を行った。表 1 に従来の解法のシミュレーション結果を、表 2 に本解法のシミュレーション結果を示す。

4.2 従来の解法と本解法の性能比較

逐次式更新では、収束率に関し、 $N \leq 100$ において、従来の解法の方が優れている。しかしながら、平均更新回数に関しては、本解法は従来の解法の約 6 割となり、本解法の方が優れている。

同期式更新では、従来の解法が $N \geq 200$ においてまったく収束しなかったのに対し、本解法はつねに高い割合で収束している。

準同期式更新では、収束率、平均更新回数共に、本解法は従来の解法より優れている。また、マキシマムニューロンを用いた解法は従来の解法よりも、特に同期式や準同期式の並列計算において優れていることが分かった。これらは、探索空間の縮小が非常に効果的であり、逆に対称性の崩れやニューロン間競合の減少が性能悪化要因とならなかったことを示している。

4.3 本解法の 3 種類の更新方法における性能比較

本解法の 3 種類の更新方法による比較では、準同期式更新が最も良い性能を示している。すなわち、マキシマムニューロンを用いた本解法は、同一グループ内では同期式更新、グループ間では逐次式更新を行う準同期式並列計算に最も適した解法であるといえる。

4.4 考 察

N-Queen 問題において、従来の解法では逐次式が、マキシマムニューロンを用いた本解法では準同期式が、最も優れた性能を示す原因について考察を行う。従来の解法において、逐次式では各ニューロンの更新前後で、その他のニューロンに状態変化は生じない。そのため、各ニューロンは、その時点の他のニューロンの状態に基づいた最適な状態更新を行うことができる。すなわち、最急降下法が実現されているのである。しかし、同期式、準同期式では、更新の前後で他のニューロンの状態も同時に更新される可能性があるため、必ずしも最適な状態更新が行われるとは限らないと考えられる。これが、従来の解法において逐次式が、最も良い性能を得る理由であると考えられる。

一方、マキシマムニューロンを用いた本解法では、グループ内のニューロンの状態は、グループ外のニューロンの状態により更新される。そのため、状態更新の最適化のためには、グループの更新前後で他のグループの状態が変化しないことが望ましい。つまり、同期式よりも逐次式、準同期式を用いることが望ましいの

である。このことは、シミュレーションの結果にも合致している。

さらに、準同期式と異なり逐次式では、グループ内に状態更新が行われたニューロンと、更新されていないニューロンが混在する状態が生じる。このため、ニューロンの選択時に、更新前の状態と更新後の状態が比較されてしまう。このことが、準同期式が逐次式よりも良い性能を得ている原因と考えられる。すなわち、マキシマムニューロンでは、グループ内を同期式に、グループ間を逐次式に更新する、準同期式が最も適した更新方法である。

5. む す び

本論文では、従来の Takefuji の解法をもとに、マキシマムニューロンを用いた N-Queen 問題の解法を提案した。3 種類の更新方法におけるシミュレーションにより、本解法は並列計算において従来の解法より優れていることを明らかにした。特に、マキシマムニューロンを用いた本解法は、準同期式において効率的な並列計算を実現するのみにとどまらず、収束に要する更新回数、収束率の点においても非常に優れていることを示した。

参 考 文 献

- 1) Garey, R. and Johnson, S.: *Computers and Intractability*, Freeman and Company (1991).
- 2) Hopfield, J. and Tank, D.: Neural Computation of Decisions in Optimization Problems, *Biol. Cybern.*, Vol.52, pp.141-152 (1985).
- 3) Kurokawa, T. and Yamashita, H.: Bus Connected Neural Network Hardware System, *Electronics Letters*, Vol.30, No.12, pp.979-980 (1994).
- 4) Lee, K., Funabiki, N. and Takefuji, Y.: A Parallel Improvement Algorithm for the Bipartite Subgraph Problem, *IEEE Trans. Neural Networks*, Vol.3, No.1, pp.139-145 (1992).
- 5) Takefuji, Y. and Lee, K.: An Artificial Maximum Neural Network: a Winner-Take All Neuron Model Forcing the State of the System in a Solution Domain, *Biol. Cybern.*, Vol.67, pp.243-251 (1992).
- 6) Lee, K. and Takefuji, Y.: A Generalized Maximum Neural Network for the Module Orientation Problem, *Int. J. Electronics*, Vol.72, pp.331-355 (1992).
- 7) Takefuji, Y. and Lee, K.: Artificial Neural Networks for Four-Coloringmap Problems and K-colorability Problems, *IEEE Trans. Circuits Systems*, Vol.38, pp.326-333 (1991).
- 8) Mandziuk, J.: Solving the N-Queens Problem with a Binary Hopfield-type Network, *Biol. Cybern.*, Vol.66, pp.375-379 (1995).
- 9) 由雄宏明: ニューラルネットワークによる N-Queen 問題の解法, 電子情報通信学会総合大会, Vol.D-17, p.17 (1995).
- 10) Takefuji, Y.: *Neural Network Parallel Computing*, Kluwer Academic Publishers (1992).

付 録

A.1 アルゴリズム

マキシマムニューロンを用いた場合の、N-Queen 問題の準同期式アルゴリズムを示す。本アルゴリズムは、 $\Delta t = 1$ として、一次のオイラー法により動作方程式を解いている。

Step 1 $t = 0$, $U_{\min} = -20$, $A_2 = B_1 = B_2 = C_2 = 1$ に設定。

Step 2 $U_{ij}(t)$ ($i, j = 1, \dots, N$) の初期値を, $U_{\min} \sim 0$ の負の整数値に設定。

Step 3 マキシマムニューロンにより, 各ニューロン出力値 $V_{ij}(t)$ ($i, j = 1, \dots, N$) を以下のように設定する。

同一グループのニューロン入力値 $\{U_{i1}, \dots, U_{iN}\}$ が最大のニューロンの出力値を, $V_{ij} = 1$ に設定。ただし, 最大値をとるニューロンが 2 つ以上存在する場合, 出力値が, $V_{ij} = 1$ であるニューロンを, $V_{ij} = 1$ に, 出力値が, $V_{ij} = 1$ であるニューロンが存在しなければ, 添字の最も小さいニューロンの出力値を $V_{ij} = 1$ に設定。その他のニューロンの出力値を $V_{ij} = 0$ に設定する。

Step 4 $i = 1$ に設定。

Step 5 $j = 1, \dots, N$ に対して,

$$\begin{aligned} \Delta U_{ij}(t) = & -A_2 \left(\sum_{k=1}^N V_{kj} - 1 \right) \\ & -B_1 \left(\sum_{k \neq 0, 1 \leq i+k, j+k \leq N} V_{i+k, j+k} \right) \\ & -B_2 \left(\sum_{k \neq 0, 1 \leq i+k, j-k \leq N} V_{i+k, j-k} \right) \\ & +C_2 h \left(\sum_{k=1}^N V_{kj} \right) \end{aligned} \quad (12)$$

を算出。

Step 6 $j = 1, \dots, N$ に対して, $U_{ij}(t+1) = U_{ij}(t) + \Delta U_{ij}(t)$. 上下制限の実施。

Step 7 マキシマムニューロンにより, 各ニューロン出力値 $V_{ij}(t)$ ($j = 1, \dots, N$) を Step 3 と同様に

設定する.

Step 8 $i = N$ を満たすまで i を 1 増加させ, Step 5 へ.

Step 9 エネルギー関数の値が,

$$\begin{aligned}
 E &= \frac{A_2}{2} \sum_{j=1}^N \left(\sum_{k=1}^N V_{kj} - 1 \right)^2 \\
 &+ \frac{B_1}{2} \sum_{i=1}^N \sum_{j=1}^N V_{ij} \sum_{k \neq 0, 1 \leq i+k, j+k \leq N} V_{i+k, j+k} \\
 &+ \frac{B_2}{2} \sum_{i=1}^N \sum_{j=1}^N V_{ij} \sum_{k \neq 0, 1 \leq i+k, j-k \leq N} V_{i+k, j-k} \\
 &= 0 \quad (13)
 \end{aligned}$$

になるか, $t = 1000$ に達すれば, 手続きを終了. そうでなければ, t を 1 増加して Step 4 へ.

(平成 8 年 3 月 4 日受付)

(平成 8 年 7 月 4 日採録)



竹中 要一 (正会員)

昭和 48 年生. 平成 4 年大阪大学基礎工学部情報工学科入学. 平成 7 年同大学院博士課程前期課程入学. 現在同博士課程前期課程に在学中. ニューラルネットワークに関する研

究に従事. 電子情報通信学会会員.



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, プロセス制御など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 年同大学院工学系研究科博士課程了. 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などを経て, 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, 最適化問題の解法などの研究に従事. 電子情報通信学会, IEEE, ACM 各会員.