論　文

ホップフィールド型ニューラルネットによる最適化問題の効率的な解法

正　員　田中　敏雄[†]　　正　員　樋口　哲也[†]　　非会員　古谷　立美[†]

An Efficient Algorithm for Solving Optimization Problems on Hopfield-Type Neural Networks

Toshio TANAKA [†], Tetsuya HIGUCHI [†], *Members and* Tatsumi FURUYA [†], *Nonmember*

あらまし　ホップフィールドネットは最適化問題に適用したとき極小値に陥り，良い解が得られない．極小値から脱出するためにボルツマンマシン等が提案されているが，計算量が増大せざるを得ない．本論文では，これらの問題を解決するために，ニューロンの出力が2値のホップフィールドネットを用いた効率的な最適化問題解法アルゴリズムを提案している．このアルゴリズムはネットワークのエネルギーを効率的に減少させるために，エネルギー減少量が最大となるニューロンのみを選択的に状態変化させる方法である．10都市の巡回セールスマン問題に適用した結果，従来の方法と比較し，より低いエネルギー状態が得られることから，準最適解が高い割合で得られた．またエネルギーが低い状態に速く到達できることから，計算量が約1けた減少できた．

キーワード　ホップフィールドネット，最適化問題，巡回セールスマン問題，極小値

1. まえがき

ホップフィールドらは対称な相互結合をもつニューラルネットがエネルギー関数を極小化するダイナミックスをもつことを示し，この性質を利用して巡回セールスマン問題(TSP)が高速に解けることを示した[1]．これを契機として，種々の組合せ最適化問題をホップフィールドネットを用いて解くことが盛んに試みられている．

一般に，ホップフィールドネットを用いて組合せ最適化問題を解くには，まずエネルギーが最も低くなった状態が問題の解であるようにエネルギー関数を設計する．そして，適当な初期状態を設定し，ニューロンをランダムに1個選択する．選択したニューロンの出力を状態遷移規則に基づき変化させると，ネットワークのエネルギーは必ず減少する．これを繰り返すことにより，やがてエネルギーが減少しなくなる平衡状態に達し，最適化問題を解くことができる．

しかし，エネルギー関数は複数の極小値をもつため，極小値に陥り，必ずしも最小値に収束するとは限らない．極小値から脱出するために，これまでに種々の方法が提案されている．例えば，ホップフィールドネットの状態遷移規則を確率化したボルツマンマシン[2]，ガウスノイズを印加するガウシアンマシン[3]，ニューロンの複数素子の同時変化を許す多段遷移ニューラルネット[4]等が提案されている．しかし，これらの方法は，いずれもニューロンをランダムに選択しており，解を得るまでの計算量が多いという問題がある．

そこで本論文では，従来の極小値に陥る問題や計算量が多いという問題を解決するために，ニューロンの出力が2値のホップフィールドネットを用いた効率的な最適化問題解法アルゴリズムを提案する[5]．このアルゴリズムは，ネットワークのエネルギーを効率的に減少させるために，ニューロンをランダムに選択するのではなく，エネルギー減少量が最大のニューロンのみを選択的に状態変化させる方法である．本方法は，従来のニューロンをランダムに選択する方法と比較すると，次の二つの利点がある．一つは，エネルギーが低い状態により速く到達できることから，計算量が少ない．もう一つは，より低いエネルギー状態に収束することから，準最適解が得られる割合が高い．

本論文では，本方法の有効性を検証するために，2

† 電子技術総合研究所，つくば市
Electrotechnical Laboratory, Tsukuba-shi, 305 Japan

値モデルでは良い解を得るのが難しいとされているTSPへ適用し，従来のランダム選択法との比較を行い，性能を評価する．また，ホップフィールドネットでは，必ず解が得られるという保証はないため，本方法に基づく修復アルゴリズムを提案する．以下**2.**では，従来のランダム選択法と本方法による最適化問題の解法について述べる．**3.**では，本方法によるTSPの解法について述べる．**4.**では，本方法と従来のランダム選択法との性能比較を行う．**5.**では，本方法が従来のランダム選択法と比較し，高い割合で準最適解が得られる理由について検討し，両者の計算量の比較を行う．**6.**では，得られた知見と残された課題について述べる．

2. 最適化問題の解法

本章では，最初に2値モデルホップフィールドネットを用いた従来の最適化問題の解法について述べる．次に，本方法による最適化問題の解法について述べ，従来の方法との違いを明確化する．

2.1 2値モデルホップフィールドネット

ホップフィールドネットは，相互結合型のネットワークで，ニューロン間の結合は対称結合になっている．また自分自身への結合はない．ニューロンの総数をNとすると，i番目のニューロンの入力の総和U_iは次式で与えられる．

$$U_i=\sum_{i\neq j}^{N} W_{ij}V_j+I_i \qquad (1)$$

ここで，W_{ij}はニューロンjからニューロンiへの結合荷重，I_iはニューロンiの入力バイアスである．ニューロンの出力V_iは，次の状態遷移規則で与えられる．

$U_i>0$ならば，$V_i=1$

$U_i<0$ならば，$V_i=0$

$U_i=0$ならば，状態を変えない　　(2)

次式で与えられるエネルギー関数

$$E=-\frac{1}{2}\sum_{i}^{N}\sum_{j\neq i}^{N} W_{ij}V_iV_j-\sum_{i}^{N} I_iV_i \qquad (3)$$

を考えると，ホップフィールドネットが式(2)の状態遷移規則に従い変化すると，式(3)のエネルギー関数の値は必ず減少することが知られている[1]．この性質を利用すると，最適化問題を解くことができる．その手順を以下に示す．

(a) 最小化すべき関数をN個の2値変数に関する2次形式で表現する．

(b) 最小化すべき2次形式と式(3)のエネルギー関数を係数比較し，両者を一致させる係数W_{ij}，I_iの値を求める．

(c) 得られたW_{ij}，I_iの値をもつホップフィールドネットを構成する．

(d) 適当な初期状態を設定する．

(e) ニューロンをランダムに1個選択し，そのニューロンの入力の総和を式(1)で計算する．

(f) 選択したニューロンの出力を式(2)の状態遷移規則に基づき決定する．

(g) (e)，(f)の処理をネットワークが平衡状態に達するまで行う．すなわち，各ニューロンの出力が変化しなくなるまで行う．

(h) 出力結果を読み出し，解の判定を行う．

本論文では，このような解法をニューロンをランダムに選択するという理由で，ランダム選択と呼ぶことにする．

2.2 提案するアルゴリズム

本方法は，**2.1**で述べたランダム選択法による最適化問題の解法手順のうち，ホップフィールドネットを構成する(a)から(c)までは同一であるが，(d)以降が異なる．(d)以降の手順は次の5ステップで構成される．

(ステップ1) 初期状態の設定

ニューロンの出力はすべて1に設定する．

(ステップ2) ニューロンの選択

出力が1のニューロンの中で，ニューロンの出力を0に変化させたときのエネルギー減少量が最大のニューロンを1個選択する．これは，出力が1のニューロンの中で式(1)で定義される入力の総和が最小のニューロンを選択することに相当する(付録参照)．なお，入力の総和が最小のニューロンが複数個存在する場合には，この中の1個をランダムに選択する．

(ステップ3) 状態遷移規則の適用

状態遷移規則は二つ(A，B)あり，状態遷移規則Aを適用した後で，状態遷移規則Bを適用する．状態遷移規則Aは，ステップ2で選択されたニューロンの出力を0にする．状態遷移規則Bは，状態遷移規則Aで生じた矛盾を解消するものである．すなわち，出力が0のニューロンの中にステップ2で求めたニューロンの入力の総和の最小値よりも大きいものが存在するならば，それらのニューロンの中で入力の総和が最大のニューロンを1個選択し，そのニューロンの出力を1にする．

(ステップ4) 終了条件の判定

終了条件(例えばTSPでは，各行，各列に1が1個以下)を満足したら停止する(終了条件の判定は**2.2.1**で

述べる)．そうでないならば，ステップ2へ戻る．
(ステップ5) 解の判定

解(例えばTSPでは，各行，各列に1が1個)ならば終了．そうでないならば，修復アルゴリズムに入る(修復アルゴリズムについては**2.2.2**で述べる)．

2.2.1 終了条件の判定

ステップ4の終了条件の判定について述べる．最適化問題の解法では，制約条件(例えば，次章で述べるTSPでは，各行，各列に1が1個)を満足したときが解であるように表現するのが一般的である．従って，終了条件の判定は制約条件を満足したとき停止することが考えられる．しかし，ホップフィールドネットで解く場合，必ず解が得られるという保証はない．そのため，解が得られなかったときのことも考慮し，TSPを解く場合には，各行，各列に1が1個以下を満足したら停止するようにしている．このような終了条件の判定は，TSP以外の最適化問題にも一般性を失うことなく適用できる．

2.2.2 修復アルゴリズム

ステップ5で解が得られなかった場合，修復アルゴリズムに入る．修復アルゴリズムの基本的な考え方は，制約条件を満足しているニューロンの状態はそのまま残し，制約条件を満足していないニューロンの状態を修復する．制約条件を満足していないニューロンの出力を1に再設定し，この状態を初期状態とし，ステップ2へ戻り，再度同一アルゴリズムを繰り返すことにより解を得る．修復アルゴリズムの具体例については，**3.3**で述べる．

3. 本方法によるTSPの解法

本章では，組合せ最適化問題の一つであるTSPを**2.2**で述べたアルゴリズムを用いて解く方法について述べる．

TSPは，1人のセールスマンがN個の都市を1度ずつ訪問し，出発した都市に戻るような経路のうちで，最短の経路を求める問題である．TSPを解くには，最初に，エネルギーが最も低くなった状態がTSPの解であるようにエネルギー関数を設計する．次に，そのエネルギー関数からニューロン間の結合荷重とバイアスを求め，ホップフィールドネットを構成する．そして，**2.2**で述べた手順に従いニューラルネットを動作させると，TSPを解くことができる．

以下**3.1**では，TSPの解の表現について述べる．**3.2**では，TSPを解くためのエネルギー関数の設計について述べる．**3.3**では，解が得られなかったときの修復アルゴリズムの具体例について述べる．

	1	2	3	4	5
A	0	0	1	0	0
B	1	0	0	0	0
C	0	1	0	0	0
D	0	0	0	0	1
E	0	0	0	1	0

図1 TSPを解くための経路の表現
Fig. 1 Representation of a tour.

3.1 TSPの解の表現

TSPの解は都市数をNとすると，$N\times N$個のニューロンを用いて，各都市を行に，列を巡回順序に対応させて，各行，各列に1が一つだけあるニューロンの状態により表すことができる．例えば図1の例は，経路が$B\to C\to A\to E\to D$であることを示している．

3.2 エネルギー関数の設計

TSPを解くためのエネルギー関数は，ホップフィールドのオリジナルなエネルギー関数を改良した以下のようなエネルギー関数を用いている[6]．

$$E=\frac{\alpha}{2}\left(\sum_x\left(\sum_i V_{xi}-1\right)^2+\sum_i\left(\sum_x V_{xi}-1\right)^2\right)+\frac{\beta}{2}\sum_x\sum_{y\neq x}\sum_i\frac{d_{xy}}{d_{norm}}\frac{V_{xi}(V_{y,i+1}+V_{y,i-1})}{2}+\alpha\sum_x\sum_i V_{xi}(1-V_{xi})\qquad(\alpha\geqq 0,\ \beta\geqq 0)\quad(4)$$

第1項は，各都市が1度だけ訪問されるという行方向の制約と同時には一つの都市しか訪問できないという列方向の制約を表している．第2項は，短い距離を指向させるためのコスト項である．第3項は，自己結合削除項と呼ばれ，ホップフィールドネットの条件$W_{xi,xi}=0$を満たすために付加される．

x，yは都市，iは訪問順，d_{xy}は都市x，y間の距離，αは制約項の係数，βはコスト項の係数を表す．d_{norm}は，すべての異なる都市間の距離データ((N^2-N)個)の平均値である．

式(4)のTSPのエネルギー関数から結合荷重とバイアスを求め，ホップフィールドネットを構成すると，x行i列の入力の総和は次式で表される．

$$U_{xi}=-\alpha\left(\sum_{j\neq i}V_{xj}+\sum_{y\neq x}V_{yi}\right)$$

$$-\beta\sum_{y\neq x}^{N}\frac{d_{xy}}{d_{norm}}\frac{(V_{y,i+1}+V_{y,i-1})}{2}+\alpha \qquad (5)$$

式(5)より，**2.2** で述べた手順に従いニューラルネットを動作させると，TSP を解くことができる．しかし，必ず解が得られるとは限らない．その場合には，次に述べる修復アルゴリズムに入る．

3.3 修復アルゴリズムの具体例

修復アルゴリズムは，制約条件を満足しているニューロンの状態はそのまま残し，制約条件を満足していないニューロンの状態を修復する．制約条件を満足していない行および列のニューロンの出力は，**2.2.1** で述べた終了条件の判定から，すべて 0 になる．そのため，暫定的に得られている都市の訪問順序はそのまま残し，ニューロンの出力が 0 の行および列の修復を図る．ニューロンの出力が 0 の行と列の交点のニューロンの出力をすべて 1 に設定し，この状態を初期状態として **2.2** で述べたアルゴリズムを再度繰り返すことにより解を得る．例えば，図 2(a)は解が得られなかったニューロンの状態を示し，図 2(b)は修復アルゴリズムのための初期状態を示している．

	1	2	3	4	5
A	0	0	0	0	0
B	1	0	0	0	0
C	0	0	1	0	0
D	0	0	0	0	1
E	0	0	0	0	0

(a) A case where solutions can not be obtained.

	1	2	3	4	5
A	0	1	0	1	0
B	1	0	0	0	0
C	0	0	1	0	0
D	0	0	0	0	1
E	0	1	0	1	0

(b) Initial state for a restoration algorithm.

図 2　修復アルゴリズム
Fig. 2　The restoration algorithm.

修復アルゴリズムでは，制約充足を強くし，すなわちコスト係数 β の値を制約項の係数 α よりも十分小さな値に設定することにより，暫定的に得られている都市の訪問順序を破壊することなく，残りの部分を修復できる．

4. 実験結果

本章では，本方法の有効性を検証するために，都市数 10 の三つの都市配置について，本方法とランダム選択法との比較を行う．最初に実験条件について述べ，次に実験結果について述べる．

4.1 実験条件

図 3 に，実験に使用した 3 種類の都市配置を示す．MAP A は，ベンチマークとして多くの研究者が採用している都市配置である．MAP B と C は，MAP A よりも最適解が得にくい都市配置である．

ランダム選択法のニューロンの初期値は，0 と 1 が 50 % ずつになるように乱数を用いて設定している．また，ランダム選択法では，終了判定を行う必要がある．ランダム選択法での終了判定は，繰返し回数の上限を設け，その繰返し回数に到達したときに終了するのが一般的である．ここでは，秋山[6]が示した方法を用いている．すなわち，N^2 個のニューロンのうちからランダムに 1 個選び，状態遷移規則に基づきニューロンの状態を変える．これを N^2 回行うことを処理のステップと呼び，100 ステップ(およそ $10N$)で終了する．

4.2 結　　果

本節では，まず **4.2.1** で本方法とランダム選択法でのコスト係数の影響について述べ，本方法の性能は，コスト係数の値に大きく依存し，都市配置により最適なコスト係数の値が存在することを示す．**4.2.2** では，最適なコスト係数の値に設定すれば，いずれの都市配置

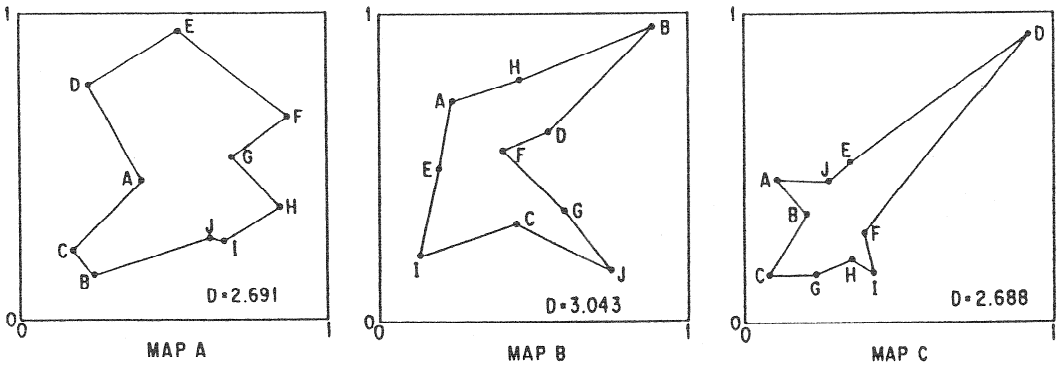
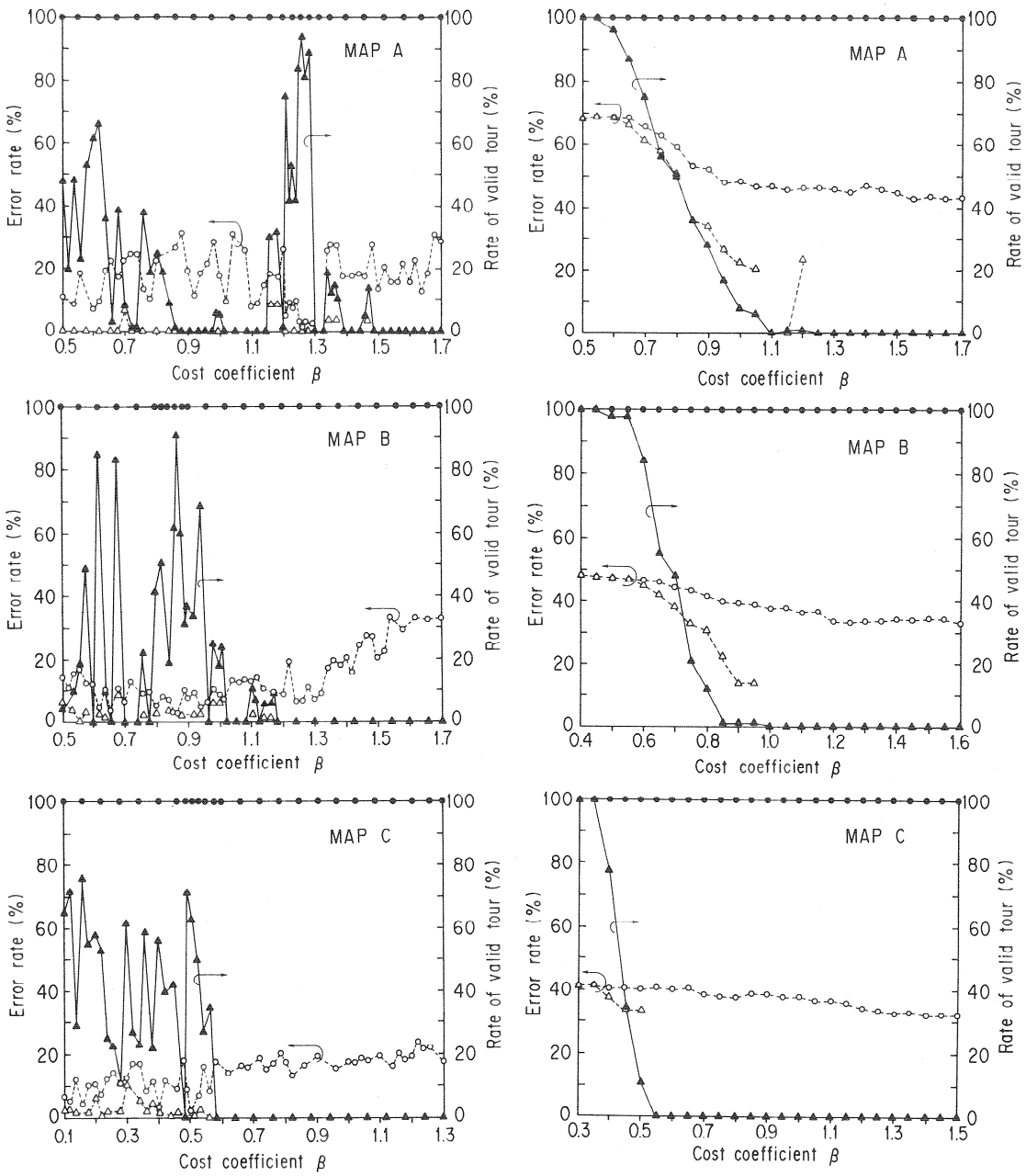


図 3　実験に使用した都市配置
Fig. 3　Maps of city allocations.



(a) Proposed method　　(b) Random selection method

△ is the error rate without restoration.
▲ is the succes rate in getting tour without restoration.
○ is the error rate with restoration.
● is the succes rate in getting tour with restoration.

図4　コスト係数に対するエラーレートと解が得られる割合の変化
Fig. 4　Cost-coefficient value vs. error rate and rate of valid tour.

でも高い割合で準最適解が得られることを示す．

4.2.1　コスト係数の影響

図4は，エネルギー関数中の制約項の係数 α を一定（$\alpha=1$）とし，コスト係数 β の値を変えたときの解が得られる割合とエラーレートを調べたものである．同図（a）は，本方法での結果を示し，（b）は，ランダム選択法での結果を示している．エラーレートは，各コスト係数の値で乱数の種を変えて100回行ったときの平均

値である．エラーレートは，次式で与えられる．

エラーレート＝(経路長－最小経路長)/最小経路長 (6)

図中△と▲は，修復アルゴリズムなしのときのエラーレートと解が得られる割合を示し，○と●は，修復アルゴリズムありのときのエラーレートと解が得られる割合をそれぞれ示している．なお，ランダム選択法でも解が得られなかった場合，同一条件で性能を比較するため，3.3で述べた修復アルゴリズムを用いている．修復アルゴリズムでのコスト係数は$\beta=0.1$に固定している．本方法，ランダム選択法とも，修復アルゴリズムによりすべて解が得られていることがわかる．

図4(b)より，ランダム選択法では修復アルゴリズムなしの場合，いずれの都市配置でもコスト係数βの値が小さいと，巡回経路は得られるが局所解に陥りエラーレートが悪くなる．一方，βの値を大きくすると，エラーレートは改善されるが巡回経路が得られる割合が低下することがわかる．

これに対して本方法では，図4(a)に示すように修復アルゴリズムなしの場合，コスト係数βの値により解が得られる割合とエラーレートが大きく変化することがわかる．また，いずれの都市配置でも，コスト係数βの値が大きく，なおかつ解が得られる割合が高いところでエラーレートが良くなることがわかる．このことから，最適なコスト係数の値は，制約条件を満足する最大のコスト係数に設定すれば良いことがわかる．以上は，修復アルゴリズムなしの場合であるが，次に，修復アルゴリズムありの場合について述べる．

図4(a)より，本方法での修復アルゴリズムありの場合，エラーレートが最も小さくなるのは，いずれの都市配置でも，修復アルゴリズムなしの場合のコスト係数βの値が大きく，なおかつ解が得られる割合が高い付近にあることがわかる．例えば，エラーレートが最も小さくなる最適なコスト係数の値は，MAP Aでは，1.30～1.32，MAP Bでは0.86，MAP Cでは0.50になる．

これに対して，図4(b)のランダム選択法での修復アルゴリズムありの場合のエラーレートは，修復アルゴリズムなしの場合の解が得られなくなるコスト係数の値付近で小さくなり，コスト係数の値がこれ以上大きくなってもエラーレートはあまり改善されないことがわかる．最適なコスト係数の値はMAP Aでは1.60，MAP Bでは1.20，MAP Cでは1.30になる．

このような最適なコスト係数の設定法は，理論的には確立されておらず，試行錯誤や経験的に求めているのが実状である．経験的にコスト係数βの値を設定する方法として，$\beta=\alpha$に設定する方法がある[6]．なお，このように設定できるのは，式(4)において，距離データを平均値で正規化しているためである．表1に，$\beta=\alpha(=1)$に設定したときの本方法とランダム選択法でのエラーレートの比較を示す．また表2に，本方法，ランダム選択法とも最適なコスト係数の値に設定したときのエラーレートの比較を示す．

表1 $\beta=\alpha$に設定したときのエラーレートの比較

	MAP A	MAP B	MAP C
Proposed method (%)	17.5	8.0	17.8
Random selection method (%)	48.2	37.4	37.8

表2 最適なコスト係数に設定したときのエラーレートの比較

	MAP A	MAP B	MAP C
Proposed method (%)	0.0	2.15	1.92
Random selection method (%)	43.2	33.6	32.2

表1と表2から本方法のエラーレートは，ランダム選択法と比較し，いずれの都市配置でも改善されることがわかる．表1の$\beta=\alpha$に設定したときは，3都市の平均では2.85倍改善され，表2の最適なコスト係数の値に設定したときは，3都市の平均では26.7倍と大幅に改善される．このことから，本方法の性能を引き出すためには，最適なコスト係数の設定法を開発する必要があることがわかる．本方法での最適なコスト係数の値は，制約条件を満足するような最大のコスト係数の値に設定すればよいことを述べた．そこで我々は最適なコスト係数の設定法として，制約条件を段階的に変えていき，その制約条件下でコスト係数の値が最大値をとるように制御する方法を提案している[7]．この方法の具体的なアルゴリズムについては，紙数の関係から稿を改めて発表する予定である．

4.2.2 巡回経路の分布

解の質を調べるために，図5(a)と(b)に$\beta=\alpha$に設定したときの本方法とランダム選択法での巡回経路の分布をそれぞれ示す．また図6(a)と(b)に最適なコスト係数に設定したときの本方法とランダム選択法での

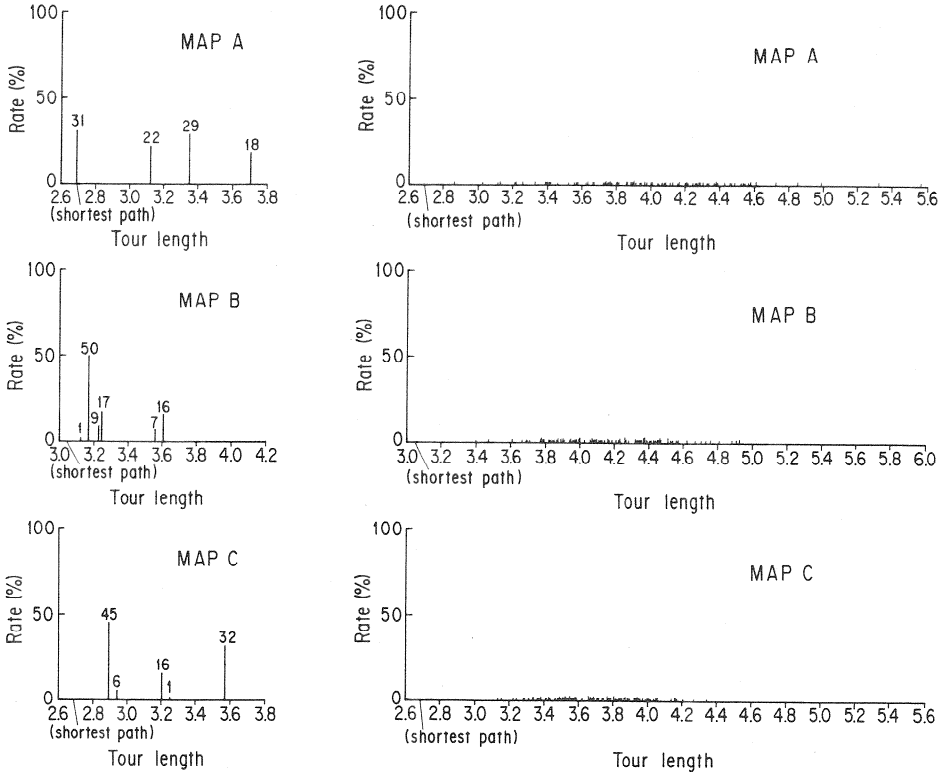


図5　$\beta=\alpha$に設定したときの巡回経路の分布
Fig. 5　The histogram of the route path when the cost coefficient β is set to be α.

巡回経路の分布をそれぞれ示す．

図5(b)と図6(b)より，ランダム選択法での巡回経路は，いずれの都市配置でもほぼ一様に分布し，極小値に陥る割合が高いことがわかる．これに対して本方法では，図5(a)と図6(a)に示すように巡回経路は，いずれの都市配置でも最小経路の近くに多く分布することがわかる．特に最適なコスト係数の値に設定した場合には，エラーレートが3％以内に収束する割合は，MAP Aでは100％，MAP Bでは84％，MAP Cでは85％が収束する．これらの結果から本方法は，いずれの都市配置でも高い割合で準最適解が得られることがわかる．次章では，この理由について検討する．

5. 検　　討

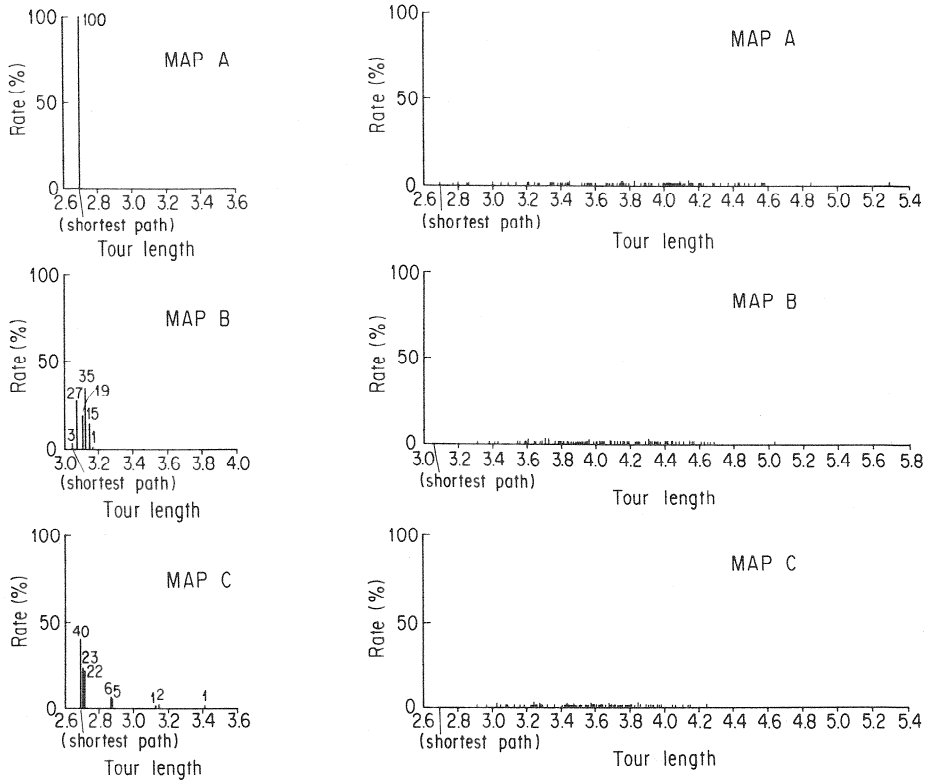
本章では，最初に本方法がランダム選択法と比較し，高い割合で準最適解が得られる理由について検討する．次に，両者の計算量を比較し，本方法はランダム選択法より約1けた計算量が減少することを示す．

5.1 高い割合で準最適解が得られる理由

本方法がランダム選択法と比較し，高い割合で準最適解が得られるのは，次のような理由であると考えられる．

ランダム選択法では，ニューロンをランダムに1個選択する．そして，選択したニューロンの入力の総和を計算し，その総和が正ならば出力を1にし，負ならば0にする．これを繰り返すと，出力が1のところは入力の総和が正になり，出力が0のところは入力の総和が負になり，平衡状態に達する．平衡状態では，たとえエネルギーが最小でなくてもこれ以上状態遷移できなくなる．そのため，極小値に陥る割合が高くなる．

これに対して本方法では，状態遷移規則Bを除き，出力が1のニューロンの中からエネルギー減少量が最大のニューロンを1個選択し，その出力を0にする．これを終了条件を満足するまで繰り返すことにより解を得る．従って，平衡状態で停止することはない．各ニューロンは，ネットワークのエネルギーが最も低い状態になるように状態遷移し，終了条件を満足したときにはランダム選択法よりも，より低いエネルギー状態に達する可能性が高い．このため，高い割合で準最適解が得られたものと考えられる．



(a) Proposed method　　(b) Random selection method

図6　最適なコスト係数に設定したときの巡回経路の分布
Fig. 6　The histogram of the route path when the cost coefficient β is set to be the optimum value.

5.2　計算量の比較

本節では，本方法とランダム選択法との計算量の比較を行う．計算量は，状態遷移する1個のニューロンの計算量と解を得るまでの繰返し回数の積で表される．

都市数を N とすると，ランダム選択法での1個のニューロンの入力の総和の計算量は，式(5)より $4N$ (行と列の計算にそれぞれ N，距離の計算に $2N$ で，合計 $4N$) になり，これを N^2 回繰り返すと1ステップになる．従って，1ステップの計算量は $4N^3$ になる．これを100ステップ(およそ $10N$) 繰り返すことにより解を得る．従って全計算量は，およそ $40N^4$ になる．

これに対して本方法では，**2.2** で述べたように，出力が1のニューロンの中で入力の総和が最小のニューロンを1個探すのに N^2，また出力が0のニューロンの中で入力の総和が最大のニューロンを1個探すのに N^2 で，合計 $2N^2$ になる．選択された1個のニューロンの入力の総和の計算量は $4(1+\varepsilon)N$ になる．ここで，ε は状態遷移規則Bにより0から1へ状態遷移する割合であり，実験に使用した3都市の平均では，$\varepsilon=0.14$ になる．従って，状態遷移する1個のニューロンの計算量は $(2N^2+4(1+\varepsilon)N)$ になる．解を得るまでの繰返し回数は，ニューロンが1から0へ状態遷移する数になる．N^2 個のニューロンの初期状態は，すべて1である．そして解が得られるのは，各行，各列に1が1個になったときである．すなわち，解のときの1のニューロンの数は N 個になる．従って状態遷移規則Aにより，(N^2-N) 回0へ状態遷移すれば解が得られることになる．ところが，状態遷移規則Bにより0から1へ状態遷移するニューロンがあるため，繰返し回数が $(1+\varepsilon)$ 倍増加する．そのため，解を得るまでの繰返し回数は，$(1+\varepsilon)(N^2-N)$ になる．従って全計算量は，およそ $2N^4$ になる．

両者の計算量を比較すると，本方法は，ランダム選択法より約1けた計算量が減少することがわかる．この理由は，本方法では，状態遷移する1個のニューロンの計算量は多いが，解を得るまでの繰返し回数が少ないためである．すなわち，出力が1のニューロンの中でエネルギー減少量が最大のニューロンを選択的に0へ状態変化させることが繰返し回数の減少に大きく寄与している．

6. むすび

本論文では，ホップフィールドネットの極小値へ陥る問題や計算量が多いという問題を解決するために，ニューロンをランダムに選択するのではなく，エネルギー減少量が最大となるニューロンのみを選択的に状態変化させる方法を提案した．また，解が得られなかったときの修復アルゴリズムを提案した．本方法の性能を評価するために，10都市のTSPへ適用し，以下のような知見が得られた．

（1） 本方法は，従来のランダム選択法と比較し，エネルギーが低い状態に速く到達できることから計算量が約1けた減少できた．

（2） 本方法の解の質は，エネルギー関数中の制約項の係数を一定とすると，コスト係数の値に大きく依存し，都市配置により最適なコスト係数の値が存在することを示した．

（3） 本方法の性能を評価するために，二つの方法で比較した．一つは，コスト係数の値を経験的に設定する方法であり，もう一つは，最適なコスト係数の値に設定する方法である．前者では，従来のランダム選択法と比較し，エラーレートが3種類の都市配置で平均2.85倍改善でき，後者では，26.7倍と大幅に改善できることがわかった．

（4） 本方法の解の質は，最適なコスト係数の値に設定すれば，いずれの都市配置でも，エラーレートが3％以内に84％以上の割合で得られた．

今後の課題として以下の点が挙げられる．

（1） 最適なコスト係数の値を自動的に決定する方法を開発する必要がある．この方法は，既に提案[7]しているが，具体的なアルゴリズムについては，稿を改めて発表する予定である．

（2） 本方法を連続値モデルに拡張することにより，解の質が向上すると考えられる．従って，連続値モデルに拡張し，2値モデルとの性能比較を行う．

（3） 都市数を増加したときの本方法の性能評価を行う．また，他の最適化問題へ適用することも今後の検討課題として挙げられる．

謝辞 本研究の機会を与えて下さった太田情報アーキテクチャ部長，討論に参加して頂いた研究室諸氏に感謝します．

文　献

（1） Hopfield J. J. and Tank D. W. : "Neural computation of decisions in optimization problems", Biol. Cybern., **52**, pp. 141-152 (1985).

（2） Ackley D. H., Hinton G. E. and Sejnowski T. J. : "A learning algorithm for Boltzmann machines", Congnitive Science, **9**, pp. 147-169 (1985).

（3） 秋山　泰："ガウシアンマシンとそのアナログ/ディジタル専用アーキテクチャ"，信学技報，**CPSY88**-16 (1988).

（4） 村島定行，渕田孝康："多段遷移ニューラルネットによる最小値探索"，信学論(D-II)，**J73-D-II**，12，pp. 2012-2021 (1990-12).

（5） Tanaka T., Higuchi T. and Furuya T. : "An efficient algorithm for solving Hopfield-type neural networks", WCNN'93, **4**, pp. 337-341 (1993).

（6） 秋山　泰："ホップフィールド型ニューラルネットにおけるエネルギー最小状態への収束性を向上させる3つの技法"，信学技報，**NC90**-40 (1990).

（7） Tanaka T., Higuchi T. and Furuya T. : "Cost coefficient control method for solving optimization problems on Hopfield-type neural networks", IJCNN'93, **2**, pp. 1528-1532 (1993).

付　録

ニューロンの出力を1から0へ変化させたときのエネルギー減少量が最大のニューロンを選択することは，出力が1のニューロンの中で入力の総和が最小のニューロンを探すことに相当することを説明する．

ホップフィールドネットにおいて，m番目のニューロンの出力が1から0へ変化したとすると，式(1)，(2)より，

$$U_m=\sum_{j\neq m}^{N} W_{mj}V_j+I_m<0 \qquad (A\cdot1)$$

であることを意味している．m番目のニューロンが出力を変える前と変えた後のエネルギー変化量ΔEを求めると，変化前のエネルギー関数E_1は，$V_m=1$であるから

$$\begin{aligned}E_1&=-\frac{1}{2}\sum_{i}^{N}\sum_{j\neq i}^{N}W_{ij}V_iV_j-\sum_{i}^{N}I_iV_i\\&=-\frac{1}{2}\sum_{i\neq m}^{N}\sum_{j\neq i,m}^{N}W_{ij}V_iV_j-\sum_{i\neq m}^{N}I_iV_i\\&\quad-\sum_{j\neq m}^{N}W_{mj}V_mV_j-I_mV_m\\&=-\frac{1}{2}\sum_{i\neq m}^{N}\sum_{j\neq i,m}^{N}W_{ij}V_iV_j-\sum_{i\neq m}^{N}I_iV_i\\&\quad-\sum_{j\neq m}^{N}W_{mj}V_j-I_m \qquad (A\cdot2)\end{aligned}$$

であり，一方，変化後のエネルギー関数E_2は，$V_m=0$であるから

$$E_2=-\frac{1}{2}\sum_{i\neq m}^{N}\sum_{j\neq i,m}^{N}W_{ij}V_iV_j-\sum_{i\neq m}^{N}I_iV_i \qquad (A\cdot3)$$

である．従って，エネルギー変化量 ΔE は，

$$\Delta E=E_2-E_1$$

$$=\sum_{j\neq m}^{N}W_{mj}V_j+I_m \qquad (A\cdot4)$$

になる．ここで式(A・1)から，式(A・4)は負であるからエネルギーは減少し，減少量は入力の総和と同一になる．従って，エネルギー減少量が最大のニューロンを選択することは，入力の総和が負で絶対値が最大のニューロンを探すことになる．すなわち，出力が1のニューロンの中で入力の総和が最小のニューロンを探すことに相当する．

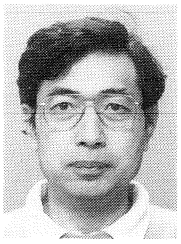
(平成5年11月25日受付，6年1月27日再受付)

田中　敏雄

昭46日大・理工・電気2部卒．昭42電気試験所(現，電子技術総合研究所)入所．光磁気記録，垂直磁気記録，ニューラルネットの研究に従事．現在，情報アーキテクチャ部計算機構研究室主任研究官．情報処理学会会員．

樋口　哲也

昭57慶大・大学院工学研究科博士課程了．同年，電子技術総合研究所入所．工博．超並列連想システムアーキテクチャ，遺伝的アルゴリズムの研究に従事．現在，情報アーキテクチャ部計算機構研究室長．第25回市村学術賞受賞．情報処理学会会員．



古谷　立美

昭48成蹊大大学院工学専攻修士課程了．同年電子技術総合研究所入所．工博．平5新情報処理開発機構出向．現在超並列ニューロ部長．IEEE，情報処理学会各会員．