

ホップフィールド型ニューラルネットのコスト係数の制御

正員 田中 敏雄[†] 正員 樋口 哲也[†] 非会員 古谷 立美[†]

Cost Coefficient Control Method for Solving Optimization Problems on Hopfield-Type Neural Networks

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

あらまし 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際には、エネルギー関数中の制約項の係数とコスト項の係数を適切に設定しなければならない。従来、これらの係数の値は、経験や試行錯誤的に決められていた。このために制約条件を満足する解が得られなかったり、コスト値の良い解が得られない等の問題が生じた。本論文では、これらの問題を解決するために制約項の係数を一定とし、コスト項の係数を適切に制御する方法を提案する。この方法では、解を得るための制約条件を段階的に変えていき、最後の制約条件をすべて満足したときが解になるようにする。その際、各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ことにより、高品質の解が得られる。この方法を評価するために10から30都市の巡回セールスマン問題に適用した結果、従来の制約項の係数を制御する方法と比較し、エラーはいずれの都市数でも改善でき、計算時間は約1けた短縮できた。

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、制約項係数、コスト係数

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。

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、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。一般にエネルギー関数は、二つのパラメータからなる。すなわち、解に対する制約条件を表す係数(制約項係数)と最小化したいコスト係数から成り立っている。従来、これらのパラメータの値は、経験や試行錯誤的に決められていた。このために制約条件を満足する解が得られなかったり、コスト値の良い解が得られない等の問題が生じた。これらの解決のためには、この二つのパ

ラメータの値を適切に設定する必要がある。

最近、これらのパラメータの値を適切に設定する方法として、五十嵐⁽²⁾は、制約項係数の値をアニーリング操作を行いながら制御する方法を提案したが、確率的な動作であるために計算量の点で難があった。また、大堀⁽³⁾らは、決定論的動作を行うホップフィールドネットに対して、制約条件を満足する最小の制約項係数を2分探索法により求める方法やラグランジュ乗数法を提案した。しかし、全制約式のバランスをとりながらラグランジュ乗数を更新しなければならず、更新に膨大な時間を要した。このように、これらの方法は、いずれも解を得るまでの計算量が多いという問題があった。また、いずれの方法もコスト係数を一定とし、制約項係数を制御する方法であった。

これに対して、本論文で提案する方法⁽⁴⁾は、制約項係数を一定とし、コスト係数を制御する方法であり、計算量を低減することができる。この方法では、解を得るための制約条件を段階的に変えていき、最後の制約条件をすべて満足したときが解になるようにする。その際、各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ことにより、高品質の解が得られる。提案する方法でのホップフィールドネットは決定論的

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

に動作し、しかも従来のホップフィールドネットのようにニューロンをランダムに選択するのではなく、全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し、状態変化させる。これによってネットワークのエネルギーは、効率的に減少し、従来のニューロンをランダムに選択する方法と比較し、計算量が少ない利点が生ずる⁽⁵⁾。

本論文では、この方法を10から30都市のTSPへ適用し、近似解が効率良く得られることを示す。以下2.では、ホップフィールドネットによるTSPの解法について述べる。3.では、提案するアルゴリズムによるTSPの解法について述べる。4.では、提案するアルゴリズムの有効性を検証するために、コスト係数を制御した場合と、しない場合との性能比較を行う。5.では、得られた結果を検討し、いずれの都市配置でもコスト係数の値が適切に制御されることを示す。6.では、提案した方法を評価するために従来手法との性能比較を行う。7.では、得られた知見と残された課題について述べる。

2. ホップフィールドネットによるTSPの解法

本章では、まずホップフィールドネットについて説明する。次に、ホップフィールドネットによるTSPの解法について述べる。

2.1 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 U_i は次式で与えられる。

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

また、ニューロンの出力 V_i は次式で与えられる。

$$V_i = f(U_i) = \frac{1}{1 + \exp(-U_i/U_0)} \quad (2)$$

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、 U_0 は式(2)で定義されるシグモイド関数のこう配を決定するパラメータである。

ホップフィールドネットは、次式で与えられるエネルギー関数を極小化する性質がある⁽¹⁾。

$$E(v) = -\frac{1}{2} \sum_{i,j=1}^N W_{ij} V_i V_j - \sum_{i=1}^N I_i V_i$$

	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.

$$+ \sum_{i=1}^N \int_0^{v_i} f^{-1}(v) dv \quad (3)$$

すなわち、ニューロンをランダムに1個選択し、そのニューロンの出力を式(2)に基づき変化させると、式(3)のネットワークのエネルギーは必ず減少することが保証されている。この性質を利用すると、組合せ最適化問題を解くことができる。次節では、組合せ最適化問題の中でも良い解を得るのが難しいとされているTSPの解法について述べる。

2.2 TSPの解法

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。以下2.2.1では、ニューラルネットによるTSPの解の表現について述べる。2.2.2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。

2.2.1 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 $B \rightarrow C \rightarrow A \rightarrow E \rightarrow D$ であることを示している。

2.2.2 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⁾。

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

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。

$$U_{xi} = -\alpha \left(\sum_{j \neq i}^N V_{xj} + \sum_{y \neq x}^N V_{yi} \right) - \beta \sum_{y \neq x}^N \frac{d_{xy}}{d_{norm}} \frac{(V_{y,i+1} + V_{y,i-1})}{2} + \alpha \quad (5)$$

従って、適当な初期状態を設定し、ニューロンをランダムに1個選択し、そのニューロンの出力を式(2)に基づき変化させると、式(4)の TSP のエネルギーは必ず減少することから TSP を解くことができる。

しかし、式(4)の TSP のエネルギー関数は複数の極小値をもつため、何らかの工夫をしない限り、最適解や準最適解を効率良く求めることはできない。

そこで我々は、極小値に陥る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、全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のみを選択的に状態変化させる方法を提案し、いずれの都市配置でも準最適解が極めて高い割合で得られることを示した⁽⁵⁾。しかしながら高い割合で準最適解が得られるのは、式(4)の制約項係数 α を一定とすると、コスト係数 β の値を適切に設定した場合であり、適切なコスト係数の設定法が問題点として残されていた。

準最適解を効率良く得るための適切なコスト係数の設定法については、制約項係数 α を一定とすると、制約条件(各行, 各列に1が1個)を満足するような最大のコスト係数の値に設定する必要があることが明らかとなっている⁽⁵⁾。しかし、制約条件を満たす最大のコスト係数の値を解析的に求めるのは困難である。そこで我々は、適切なコスト係数の設定法として、解を得るための制約条件を段階的に変えていき、各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考案した。次章では、この方法による TSP の解法について述べる。

3. 提案するアルゴリズムによる TSP の解法

本章では、まず 3.1 で、本アルゴリズムの二つの特徴と基本的アイデアについて述べ、それらを踏まえて 3.2 で本アルゴリズムの全体について詳細に述べる。

3.1 本アルゴリズムの特徴

我々が提案する TSP を解くためのアルゴリズムは、次の二つが従来の TSP アルゴリズムと異なる。

(1) 解を得るための制約条件を段階的に変えていくながら、最終的に制約条件を満足する TSP の解を見つける。

(2) 各制約条件下でコスト係数の値が最大値をとるようにすることにより、高品質の解を得る。

以下、これら二つの特徴と基本的アイデアについて詳細に述べる。

3.1.1 制約条件の段階的設定

制約条件は段階的に設定し、最後の制約条件をすべて満足したときが TSP の解になるようにするが、その制約条件の数は都市数を N とすると、以下のように $(N-1)$ 個になる。なお、ニューロンの初期状態はすべて1に設定する(図2(a)参照)。このようにニューロンの初期状態をすべて1に設定するのは、初期状態では TSP の解を含む必要があるためと、以下に述べるように制約条件を段階的に設定するためである。

(制約条件1) 状態変化するニューロンの数が各行, あるいは各列方向1個に設定する。

(制約条件2) 状態変化するニューロンの数が各行, あるいは各列方向2個に設定する。

(制約条件 $(N-1)$) 状態変化するニューロンの数が各行, あるいは各列方向 $(N-1)$ 個に設定する。すなわち、1のニューロンの数が各行, あるいは各列方向1個に設定する。

制約条件を変えるのは、以下の二つの状態のうち、どちらかの状態を満足したときに次の制約条件へ移る(具体例については、3.1.2で述べる)。

(イ) その制約条件をすべて満足したとき。

(ロ) コスト係数の値が巡回経路が得られるコスト係数の最小値 β_{min} (β_{min} の設定は付録1参照) よりも小さくなったとき。

その際コスト係数の値は、(イ)の場合には、そのときのコスト係数の値を引き継ぐ。(ロ)の場合には、コスト係数の初期値 β_0 に再設定する。

このように制約条件を段階的に設定すると、最後の制約条件をすべて満足したときが TSP の解になる。そ

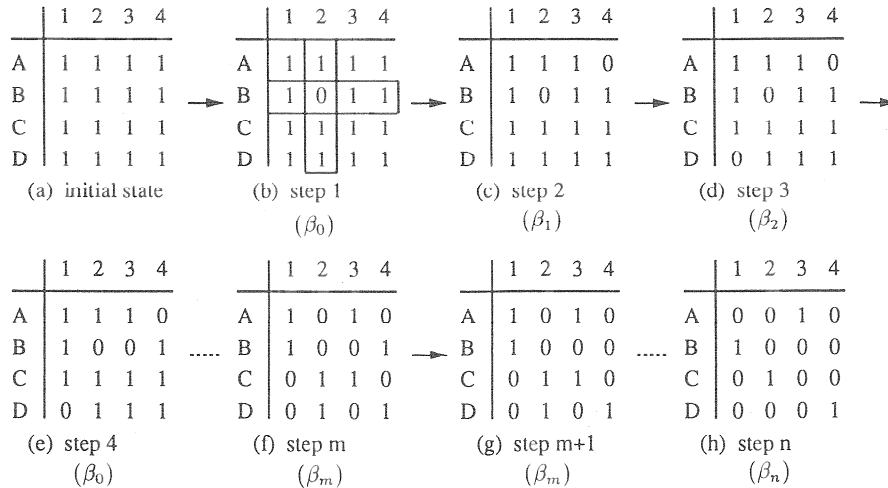


図2 コスト係数制御の模式図
Fig. 2 A schematic diagram of the cost coefficient control method.

して、各制約条件下でコスト係数が最大値をとるようにすることにより、高品質の解が得られる。次節では、このようなコスト係数の制御方法について述べる。

3.1.2 コスト係数の制御方法

以下では、説明を簡単化するためにニューロンの出力は2値で、都市数 N は4であるとする。また、B都市を2番目に訪問することを示しているニューロンを N_{B2} のように表すとする。図2に、コスト係数制御の模式図を示す。

最初に、状態変化するニューロンの数が各行、あるいは各列方向に1個の制約条件下でコスト係数が最大値をとる制御方法について説明する。なお、コスト係数の初期値 β_0 は、制約項の係数 α よりも十分大きな値に設定する。

本方法では、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が1個選択され、0へ状態変化する。これは、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選択することに相当する⁽⁶⁾。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、その中の1個をランダムに選択する。例えばそのようなニューロンとして、最初に図2(b)に示すように、 N_{B2} のニューロンが選択され、0に変化したとする。2番目に状態変化するニューロンは、コスト係数の値が大きいために図2(b)に示した横長の四角で囲まれた1のニューロンの一つ(例えば、 N_{B4} のニューロン)が選択される。しかし、四角で囲まれたニューロンは、既に制約条件を満足しているので、これ以外のニューロンを選択しなければならない。

そのためにはコスト係数の値を減少させる。コスト係数の値を減少させていくと、やがて四角で囲まれたニューロン以外のニューロン(制約条件を満足していないニューロン)が選択される。これは、出力が1のニューロンの中で四角で囲まれたニューロンよりも、そうでないニューロンの方が制約項の総和が小さいために、コスト係数の値を減少させていくと、四角で囲まれたニューロンよりも入力の総和が小さくなり、選択される。例えばそのようなニューロンとして、図2(c)に示すように N_{A4} のニューロンが選択され、0に変化したとする。このときのコスト係数の値が、コスト係数の最大値になる。

同様に3番目に状態変化するニューロンは、制約条件を満足していないニューロンが選択されるようにコスト係数の値を減少させる。そうすると例えば、図2(d)に示すように N_{D1} のニューロンが選択され、0に変化する。以上のようなやり方に従って、状態変化するニューロンの数が各行、あるいは各列方向に1個の制約条件下でコスト係数が最大値をとるように制御する。なお、コスト係数の最大値は、コスト係数の初期値 β_0 から徐々に減少させて求めるのではなく、計算により求めている(付録2.参照)。

このようにしてコスト係数の最大値を求めて、ニューロンの状態を0に変化させていくと、やがてネットワークがこの制約条件をすべて満たす状態になるか、あるいはコスト係数の値が巡回経路が得られるコスト係数の最小値 β_{min} よりも小さくなる。

この二つの状態のうち、どちらかの状態に達すれば3.1.1で述べた方法で制約条件を変える。すなわち、状

状態変化するニューロンの数が各行、あるいは各列方向 2 個になるように設定する。そのとき、コスト係数の値は制約条件をすべて満足していれば、そのときのコスト係数の値を引き継ぐ。コスト係数の値が β_{\min} よりも小さくなったときは、コスト係数の初期値 β_0 に再設定する。

例えば図 2(e)は、コスト係数の初期値 β_0 に再設定され、状態変化するニューロンの数が各行、あるいは各列方向 2 個の制約条件に変化し、 N_{B3} のニューロンが選択され、その出力が 0 に変化した状態を示している。また図 2(f)は、状態変化するニューロンの数が各行、あるいは各列方向 2 個の制約条件をすべて満足した状態を示している。従って、次の図 2(g)では、状態変化するニューロンの数が各行、あるいは各列方向 3 個の制約条件に設定され、コスト係数の値は、図 2(f)のときの値を引き継いでいる。そして、図 2(g)に示すように N_{B4} のニューロンが選択され、その出力が 0 に変化している。

このようなやり方に従って、制約条件の設定とコスト係数の制御を状態変化するニューロンの数が各行、あるいは各列方向に $(N-1)$ 個、すなわち、1 のニューロンの数が各行、あるいは各列方向 1 個になるまで繰り返すことにより解を得る(図 2(h))。

3.2 全体のアルゴリズム

TSP を解くための全体のアルゴリズムは、以下の 11 ステップで構成される。なお、ニューロンの出力は、連続値をとるとする。

(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) コスト係数の初期値の設定

コスト係数の初期値 β_0 は、制約項の係数 α よりも十分大きな値に設定する。

(ステップ 3) 制約条件の設定

3.1.1 で述べたように制約条件は段階的に設定する。

(ステップ 4) ニューロンの選択

出力が 1 のニューロンの中で式(5)で定義される入力総和が最小のニューロンを 1 個選択する。なお、入力総和が最小のニューロンが複数個存在する場合には、その中の 1 個をランダムに選択する。

(ステップ 5) 制約条件の判定

選択したニューロンが制約条件を満足しているかを調べる。満足していないならば、ステップ 8 へいく。満足していれば、満足していないニューロンを選択するためにステップ 6 へいく。もし制約条件をすべて満

足していれば、3.1.1 で述べた方法でステップ 3 の制約条件を変える。そして、そのときのコスト係数の値を引き継いでステップ 4 へ戻る。

(ステップ 6) コスト係数の最大値の設定

付録 2. に示した方法でコスト係数の最大値を求めて、ステップ 7 へいく。

(ステップ 7) コスト係数値の判定

求めたコスト係数の最大値が巡回経路が得られるコスト係数の最小値 β_{\min} (β_{\min} の設定は付録 1. 参照) よりも大きいかを調べる。大きいならば、そのコスト係数の値をもってステップ 4 へ戻る。そうでないならば、コスト係数の初期値 β_0 に再設定するためにステップ 2 へ戻り、3.1.1 で述べた方法でステップ 3 の制約条件を変える。

(ステップ 8) 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、選択されたニューロンの出力を式(2)に従い決定する。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、状態変化したニューロンの中にステップ 4 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 9) 終了条件の判定

終了条件(各行、各列に 1 が 1 個以下)を満足したら停止する(終了条件の判定については 3.2.1 で述べる)。そうでないならば、ステップ 4 へ戻る。

(ステップ 10) 出力の 2 値化

ニューロンの出力が 1 ならば 1、そうでないならば 0 にする。

(ステップ 11) 解の判定

解(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 3.2.2 で述べる)。

ここで述べたアルゴリズムは、従来のニューラルネットの動作と大きく異なる。特に、ステップ 4 のニューロンの選択では、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 1 個選択する必要がある。この部分は直列処理となるが、コンパレータを並列に配置すれば入力の総和の最小値を求める部分は並列計算が可能になる。ステップ 8 の状態遷移規則 B についても同様にコンパレータを並列に配置すれば並列計算

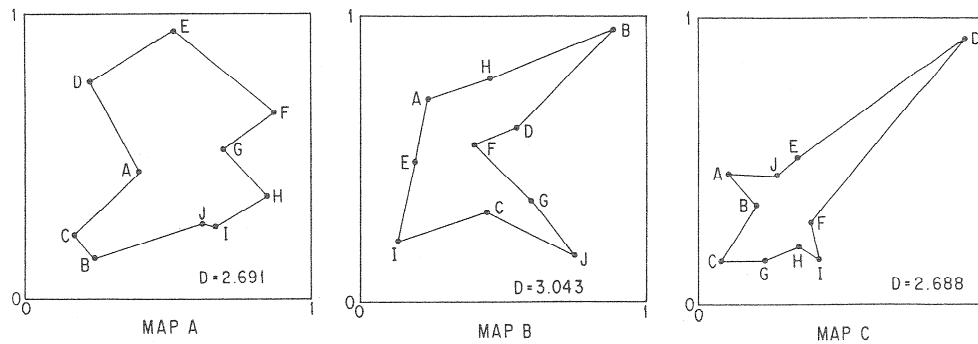


図3 実験に使用した都市配置
Fig. 3 Maps of city allocations.

が可能になる。

3.2.1 終了条件の判定

ステップ9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。

3.2.2 修復アルゴリズム

ステップ11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行と列の交点のニューロンをすべて1に再設定し、この状態を初期状態とし、ステップ4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。ただし、修復アルゴリズムでは、コスト係数の制御は行わない。そのためにステップ5から7はスキップし、コスト係数の値は、制約項の係数 α よりも十分小さな値に設定する。

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⁽⁵⁾。

4. 実験結果

本章では、提案した方法を評価するために、都市数10の三つの都市配置について、コスト係数を制御した場合としない場合(コスト係数固定)との性能比較を行う。以下4.1では、実験条件について述べ、4.2では、得られた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。マップ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。マップBとCは、マップAよりも最適解が得にくい都市配置である。

制約項の係数は $\alpha=2$ に設定し、シグモイド関数のこの配は $U_0=8$ に設定している。また、修復アルゴリズムでのコスト係数の値は、 $\beta=0.1$ に設定している。

コスト係数を制御したときのコスト係数の初期値は $\beta_0=10,000$ に設定している。また、巡回経路が得られるコスト係数の最小値 β_{min} は、式(A・1)より、マップA, B, Cではそれぞれ1.17, 0.97, 0.71になる。

コスト係数を固定したときの値は、秋山の論文⁽⁶⁾で与えられている経験値 $\beta=\alpha(=2)$ に設定している。なお、このように設定できるのは、式(4)において、距離データを正規化しているためである。

4.2 結果

表1に、コスト係数を制御したときとコスト係数固定のときのエラーの比較を示す。エラーは、乱数の種を変えて100回行ったときの平均値である。エラーは、次式で与えられる。

$$\text{エラー} = (\text{経路長} - \text{最小経路長}) / \text{最小経路長} \quad (6)$$

表1より、コスト係数を制御することにより、最適解が得にくいマップBとCでは、エラーが改善されていることがわかる。マップAは、コスト係数固定法と

比較するとエラーが悪くなっているが、エラーの値自身は、十分小さな値 (1.59 %) が得られている。マップ A については、コスト係数固定のときのコスト係数の値が、エラーが小さくなる範囲に設定されたために良い解が得られたものと考えられる。これについては、次章で検討する。

解の質を調べるために、図 4 (a) と図 4 (b) にコスト係数を制御したときとコスト係数固定のときの巡回経路の分布をそれぞれ示す。図 4 から、コスト係数固定法と比較すると、コスト係数を制御することにより、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。例えば、エラーが 5 % 以

内に収束する割合は、コスト係数固定法では、マップ A は 99 %, マップ B は 15 %, マップ C は 2 % しか収束しない。これに対して、提案した方法では、マップ A は 89 %, マップ B は 57 %, マップ C は 41 % 収束する。

これらの結果から、コスト係数固定法と比較すると、コスト係数を制御することにより、近似解が効率良く得られることがわかる。特に、最適解が得にくいマップ B と C については、近似解が得られる割合が大幅に改善されている。次章では、この理由について検討する。

5. 検 討

4.2 では、コスト係数の値を制御することにより、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を示した。このことは、コスト係数の値が、エラーが小さくなるコスト係数の範囲に適切に制御されたためであると考えられる。それを調べたのが図 5 と図 6 である。

図 5 は、巡回経路が得られるまでのコスト係数値の

表 1 エラーの比較

	MAP A	MAP B	MAP C
コスト係数制御 (%)	1.59	5.68	9.45
コスト係数固定 (%)	0.04	11.15	11.94

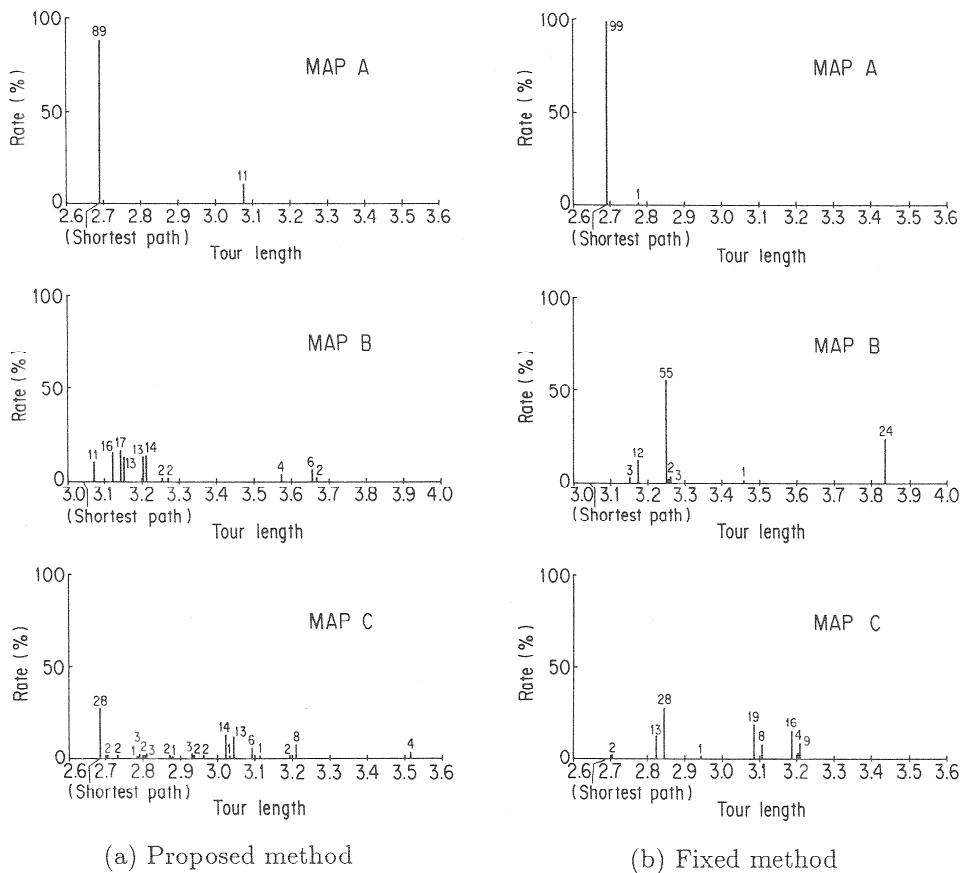


図 4 巡回経路の分布
Fig. 4 The analysis of the solution quality.

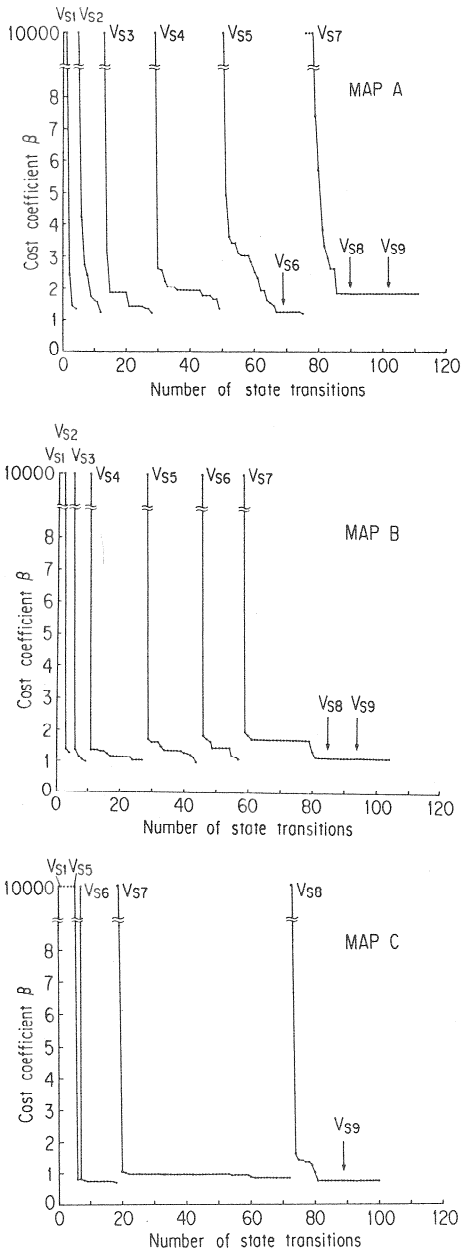


図5 コスト係数の変化

Fig. 5 Cost coefficient vs. number of state transitions.

変化の様子を示したものである。図中 V_{s1} から V_{s9} は、制約条件が変化したことを示している。すなわち、都市数を N とすると、 V_{s1} は、状態変化するニューロンの数が各行、あるいは各列方向に1個の制約条件を示し、 V_{s2} は、2個に変化したことを示している。そして、 V_{s9} は、状態変化するニューロンの数が各行、あるいは各列方向 $(N-1)$ 個に変化したことを示している。

図5から、コスト係数の値の主な動作範囲がわかる。例えばコスト係数の値は、マップA, B, Cでは、それぞれ1.4~2.1, 1.2~1.7, 0.75~0.95の範囲で主に動作

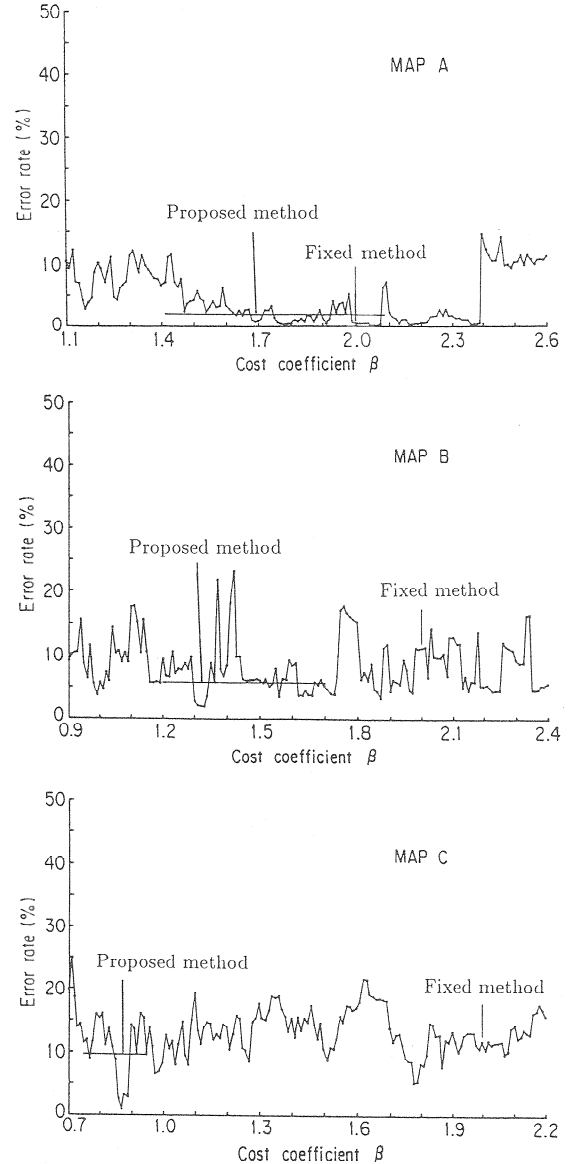


図6 コスト係数に対するエラーの変化
Fig. 6 Error vs. cost coefficient.

している。

図6は、コスト係数の値に対するエラーの変化を調べたものである。図6には、コスト係数の値を制御したときの主なコスト係数の動作範囲とエラーの値を実線(—)で重ねて示している。また、コスト係数固定 ($\beta=2$) のときのエラーの値も示している。

図6より、エラーが最も小さくなる最適なコスト係数の値が存在し、最適なコスト係数の値は都市配置により異なることがわかる。例えば最適なコスト係数の値は、マップAでは2.04~2.06、マップBでは1.33、マップCでは0.87になる。

コスト係数固定法では、マップAのみが最適なコス

ト係数の値近くに設定され、エラーが小さくなっていることがわかる。このためにマップ A は、良い解が得られたものと考えられる。しかし、他の都市配置では、いずれも最適なコスト係数の値近くには設定されていない。そのためにエラーが悪くなっている。これらの結果から、従来のコスト係数固定法では、最適なコスト係数の値が都市配置により異なることから、常に最適なコスト係数の値近くに設定されるわけではないことがわかる。

これに対して、提案した方法でのコスト係数の値は、いずれの都市配置でも最適なコスト係数の値を含むように動作していることがわかる。このために提案した方法では、コスト係数固定法と比較し、いずれの都市配置でも近似解が効率良く得られたものと考えられる。

6. 性能評価

本章では、提案した方法を評価するために従来手法との性能比較を行う。1.で述べたように、五十嵐⁽²⁾の方法は、確率的な動作であるためにエラーは改善されるが、計算時間が増大する欠点がある。ここでは、提案した方法と同様にネットワークが決定論的動作を行う大堀^{(3),(8)}らの方法と比較する。大堀らの方法による制約項係数の制御は、文献(8)に従い、2分探索法とラグランジュ乗数の更新により行った。文献(8)によると、式(2)のシグモイド関数のこう配 U_0 の値が大きいほど解の精度は向上するが、計算時間が増加することが示されている。ここでは、解の精度が向上する $U_0=0.3$ に設定した。

図7に、都市数を変えたときの本方法と大堀らの方法のエラーの比較を示す。大堀らの方法でのエラーの値は、ニューロンの初期値を50回変えて行ったときの平均値である。提案した方法でのエラーの値は、乱数の種を変えて100回行ったときの平均値である。図7より、両手法とも都市数が増加するに従ってエラーが大きくなる傾向があるが、提案した方法は、大堀らの方法と比較すると、いずれの都市数でも改善されていることがわかる。

図8に、都市数に対する両手法の1個の解を得るまでの計算時間の比較を示す。使用計算機は、Sun Sparc 2を用いた。提案した方法の計算時間は、大堀らの方法と比較すると、約1けた短縮できることがわかる。大堀らの方法は、2分探索法とラグランジュ乗数の更新により、制約条件を満たす最小の制約項係数を求める方法である。すなわち、巡回経路条件を満足する制約

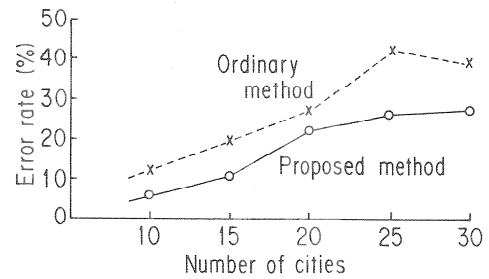


図7 都市数に対するエラーの変化
Fig. 7 The number of cities vs. error.
(a comparison between the proposed method and the ordinary method)

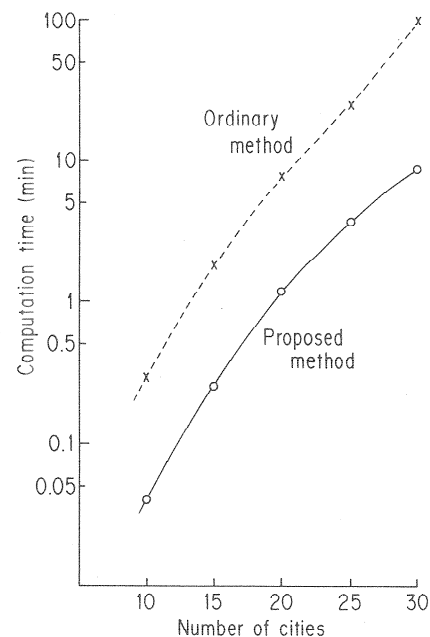


図8 都市数に対する計算時間の変化
Fig. 8 The number of cities vs. computation time.
(a comparison between the proposed method and the ordinary method)

項係数を上限とし、巡回経路条件を満足しない制約項係数を下限とし、上限と下限の差が十分小さくなるまで(実験では0.05)繰り返す。従って、計算の途中では制約項係数は一定であり、最終的な巡回経路を得るには、制約項係数を変えて上限と下限の差が十分小さくなるまで繰り返す必要がある。そのために計算時間が増加する。

これに対して提案した方法は、制約項係数を一定とし、コスト係数を制御する方法であり、コスト係数の値は、計算の途中で自動的に変化し、最後の制約条件を満足したときに巡回経路が得られる。すなわち、コスト係数の値を変えて何回も繰り返す必要はない。また、大堀らの方法はニューロンをランダムに選択して

いるが、提案した方法は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を選択しているためにネットワークのエネルギーが効率的に減少する。これら二つの方法により、大堀らの方法と比較し、計算時間が約1けた短縮できたと考えられる。

7. むすび

本論文では、ホップフィールドネットのコスト値の良い解が得られない問題を解決するために、エネルギー関数中の制約項の係数を一定とし、コスト係数の値を適切に制御する方法を提案した。この方法では、解を得るための制約条件を段階的に変えていき、最後の制約条件をすべて満足したときが解となる。その際、各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ことにより、高品質の解が得られる。提案した方法を評価するために、10から30都市のTSPへ適用し、以下のような知見が得られた。

(1) 提案した方法は、従来のコスト係数固定法と比較すると、コスト係数の値が適切に制御されることにより、近似解が効率良く得られた。

(2) 従来のコスト係数固定法では、エラーが小さくなる最適なコスト係数の値が都市配置により異なることから、常に最適なコスト係数の値近くに設定されるわけではないことがわかった。これに対して、提案した方法でのコスト係数の値は、いずれの都市配置でも最適なコスト係数の値を含むように動作することから、近似解が効率良く得られたものと考えられる。

(3) 提案した方法を評価するために、従来の制約項係数を制御する大堀⁽⁸⁾らの方法と比較し、エラーはいずれの都市数でも改善でき、計算時間は約1けた短縮できることがわかった。

提案した方法は、他の最適化問題へ適用できると考えられるので、他の最適化問題へ適用し、本方法の性能評価を行うことが今後の検討課題である。

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) 五十嵐治一: "シミュレーテッドアニーリング法におけるエネルギー関数中のパラメータ値の決定法", *信学論(D-II)*, **J75-D-II**, 7, pp. 1232-1240 (1992-07).
- (3) 大堀隆文, 山根厚志, 佐々木健一: "2値モデルニューラルネットによる巡回セールスマン問題の一解法", *信学論(D-II)*, **J74-D-II**, 12, pp. 1788-1793 (1991-12).
- (4) 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).
- (5) 田中敏雄, 樋口哲也, 古谷立美: "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", *信学論(D-II)*, **J77-D-II**, 6, pp. 1157-1166 (1994-06).
- (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) Abe S.: "Theories on the Hopfield neural networks", *IJCNN'89*, **1**, pp. 557-564 (1989).
- (8) 大堀隆文, 中村一敏, 渡辺一央: "アルゴリズム駆動形アナログニューラルネットによるTSPの解法", *信学論(D-II)*, **J75-D-II**, 12, pp. 2072-2074 (1992-12).

付 録

1. コスト係数の最小値の設定

巡回経路が得られるコスト係数の最小値の設定⁽⁷⁾について述べる。巡回経路が得られた状態を考えると、式(4)のエネルギーは、制約条件を満足しない状態のエネルギーよりも小さくしなければならない。もし、巡回経路が得られた状態から一つのニューロンが1から0へ変化したとすると、制約条件からはずれたことによるエネルギーの増分は、式(4)の第1項から α になる。一方、式(4)の第2項のエネルギーの減少分の最大値は、 $(\beta/2)[\max(d_{xy} + d_{xz})/d_{norm}]$ になる。ここで、 $\max(d_{xy} + d_{xz})/d_{norm}$ は、隣接する二つの経路の最大値であり、都市配置から求めることができる。従って、巡回経路を得るためには、次式を満足しなければならない。

$$\alpha > \frac{\beta}{2} \frac{\max(d_{xy} + d_{xz})}{d_{norm}} \quad (\text{A} \cdot 1)$$

式(A・1)から巡回経路が得られるコスト係数の最小値は、式(A・1)を満足する最大の β の値に設定すればよいことがわかる。

2. コスト係数の最大値の設定

3.1.2で述べたコスト係数の最大値の設定について述べる。制約条件を満足していない出力が1のニューロンの中で入力総和が最小のニューロンを選択するためには、制約条件を満足していない出力が1のニューロンの入力の総和の最小値 $(U_{xi,min})_ns$ が、制約条件を満足している出力が1のニューロンの入力の総和の最小値 $(U_{xi,min})_s$ よりも小さくなるようにコスト係数の値を設定しなければならない。すなわち、次式を満足す

るコスト係数の最大値に設定する必要がある。

$$(U_{xy,min})_{ns} < (U_{xy,min})_s \quad (A \cdot 2)$$

入力ノイズの総和は式(5)で与えられるから、式(A・2)を満足するコスト係数 β は、次式で表される。

$$\beta < \frac{2\alpha \left[\left(\sum_{j \neq i}^N V_{xj} + \sum_{y \neq x}^N V_{yi} \right)_{ns} - \left(\sum_{j \neq i}^N V_{xj} + \sum_{y \neq x}^N V_{yi} \right)_s \right]}{\left[\sum_{y \neq x}^N \frac{d_{xy}}{d_{norm}} (V_{y,i+1} + V_{y,i-1}) \right]_s - \left[\sum_{y \neq x}^N \frac{d_{xy}}{d_{norm}} (V_{y,i+1} + V_{y,i-1}) \right]_{ns}} \quad (A \cdot 3)$$

ここで、 $(*)_s$ は、制約条件を満足している制約項の総和を示し、 $(*)_{ns}$ は、制約条件を満足していない制約項の総和を示している。また、 $[*]_s$ は、制約条件を満足している距離項の総和を示し、 $[*]_{ns}$ は、制約条件を満足していない距離項の総和を示している。式(A・3)を満たす最大の β の値が、コスト係数の最大値になる。

(平成6年4月13日受付, 8月9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成蹊大学院修士課程了。昭48工業技術院電子技術総合研究所入所。平5新情報処理開発機構出向。平6東邦大学理学部情報科学科教授。情報処理学会、IEEE各会員。