人　工　知　能 94－2
（1994．6．20）

分業巡回セールスマン問題のニューラルネットワークによる解法

中村 友洋, 角田 達彦, 田中 英彦
{ hiro,tsunoda,tanaka} @mtl.t.u-tokyo.ac.jp
東京大学工学部

Abstract

組合せ最適化問題は解空間が広くアルゴリズミックに解くことが困難な問題であるが、高速近似解法としてニューラルネットワークを用いた解法が提案されている。しかしながら、この解法には問題点もあり、応用研究も多いとはいえない。そこで、本研究ではこの解法を用いて新しい組合せ最適化問題を解き、解法の問題点解決についての一考察を行なった。この新しい組合せ最適化問題は、「分業セールスマン問題 (nTSP)」と呼ぶことにし、スケジューリングなどの多方面での実用的なシステムの基礎技術として位置付けられる。

Solution of n–Traveling Salesmen Problem Using Neural Network Dynamics

Tomohiro Nakamura, Tatsuhiko Tsunoda, Hidehiko Tanaka,
University of Tokyo, Department of Engineering,
7–3–1 Hongou, Bunkyo–ku, Tokyo 113, Japan

The combinatorial optimization problem is very hard to solve by algorithmic solution, because of wide solution space. Recently, solutions using neural network have been proposed to solve it fast and approximately.However, these solutions have some disadvantages and there are few applications. This paper presents an application for a newly proposed combinatorial optimization problem using this solution. This newly proposed combinatorial optimization problem, named n–Traveling Salesmem Problem(nTSP), will be one of the basic techniques for scheduling in many practical systems.

1　はじめに

並列処理や分散処理がさまざま場面で行なわれるようになり、この種の処理に対する高速スケジューリングの要望が高まっている。しかしながら、並列処理や分散処理のスケジューリング問題は解空間が広く、従来のアルゴリズム的な手法を用いて高速に最適解を求めることは容易ではない。そこで、本研究では並列処理や分散処理のスケジューリング問題を扱うための基礎技術として「分業巡回セールスマン問題」を設定し、ニューラルネットワークを用いた組合せ最適化問題の解法に埋め込んだ。

この報告では、まず新しく設定した「分業巡回セールスマン問題 (nTSP)」の定義とその意味について述べる。この問題は有名な組合せ最適化問題である巡回セールスマン問題 (TSP) を包含する形の拡張であり、巡回セールスマン問題よりもその適用範囲が広い。よって、巡回セールスマン問題に投影することができなかった問題についても、分業巡回セールスマン問題に投影することができる。

次に分業巡回セールスマン問題のニューラルネットワークへの埋め込み方法について説明する。分業巡回セールスマン問題は、巡回セールスマン問題にくらべて制約条件が多く、解空間も広いために、ニューラルネットワークへの埋め込みも難しい。本報告では、分業巡回セールスマン問題のモデル化について詳しく述べ、この種の手法の研究課題であるエネルギー関数・その結合係数の設定法についても簡単な考察を行なった。

2　分業巡回セールスマン問題 (nTSP)

2.1　定義

nTSP:

複数 (n 人) のセールスマンが複数の都市 (N 都市) を各々の巡路ができるだけ短くなるように分業して巡る n コの巡路を求めることを目的とする。

例えば、3 人のセールスマンが 10 都市を分業して巡る問題を 10 都市 3TSP などと呼ぶことにする。よって N 都市の巡回セールスマン問題は、N 都市 1TSP ということになり、分業巡回セールスマン問題の一部と考えられる。

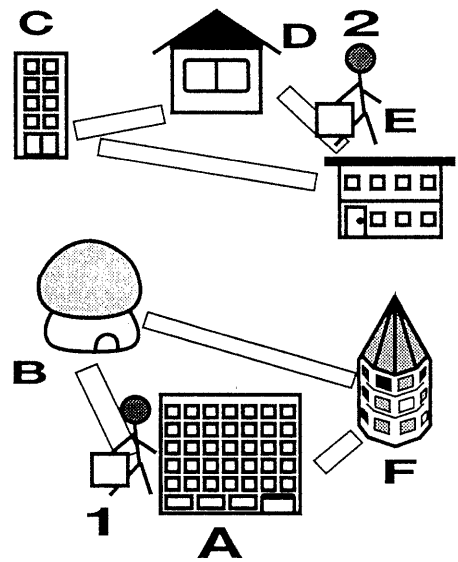

図 1: 6 都市 2TSP

2.2　背景と目的

現代は非常にシステムが複雑化しこれを運用・管理するためには非常に多くの人やコンピューターが必要になっている。そして各々の人間が分業をして並列的に仕事を行なうことがほとんどである。例えばコンピュータを例にとっても以前はシングルタスクのマシンが普通であったが、今日ではパソコンに至るまでマルチタスクが当たり前の時代となっている。このような場面では仕事の分業を含むスケジューリングが非常に重要になってくる。またこの種の問題は他にもさまざまな分野に存在する。運送業界ではトラック便のトラックの配車を「配車マン」と呼ばれる熟練の職人が行なっている。一流の「配車マン」になるまでには10〜20 年という非常に長い年月がかかると

いわれ、その後継者育成に苦労をしている。また列車やバスのダイヤや車両・乗務員運用についてもごく一部を除いて「スジ屋」と呼ばれる専門の職人がスジ引きをしているのが現状であり、非常に時間がかかる作業である。しかしながらこのような問題は非常に解空間が広く、その中から最適解を見つけることは、全解探索などの単純な方法では困難である。そこでこのような問題を一般化したものがnTSPであり、nTSPの高速近似解法がこの種の問題を解くためのシステムの基本技術と考えられる。

2.3 TSPとの相違と特徴

前節ではnTSPの定義とその背景について述べたが、ここではnTSPの特徴等を述べる。

nTSPの解は問題の定義からも明らかなように複数のループになる。このループ1つ1つはTSPの解と同様に各セールスマンの分担する都市を最短経路で巡るようになる。つまりnTSPの解は各セールスマンが巡るべき都市の分担と、各々の分担を最短経路で巡るという2つの問題を結合させたものであると見ることができる。そこでnTSPを都市の分担部分とTSP部分に分けることができればnTSPを解くということと同様の結果が得られる。しかし次のような点でこれは困難である。

- TSP自体がNP完全の問題であり、更に都市の分担もNP完全の問題である。よってこれらを「都市の分担→TSP」と逐次的に解くには時間的に問題がある。
- 都市の分担においては分担後の巡路を想定する必要がある。なぜならば最適な分担とは各セールスマンの巡路ができるだけ短くなるようなものであるからである。

つまり都市の分担とTSPというnTSPの2面は単純に分離できるものではなく、両方の問題を協調させて解く必要があると考えられる。そのためにTSPを使ってnTSP相当の問題を解くことは現実的ではなく新たにnTSPという問題を設定する必要がある。

次に挙げるのはnTSPとTSPの総巡路数の差を示す表である。総巡路数が多いほど解空間が広いことになるので、TSPにくらべnTSPはより困難な問題であることが分かる。

都市数	TSP	2TSP	3TSP
3	1	3	1
5	12	15	25
8	2,520	5,547	5,131
10	181,440	414,144	450,225
16	654×10^9	11384×10^9	2254×10^9

表1: TSP,2TSP,3TSPの総巡路数の比較

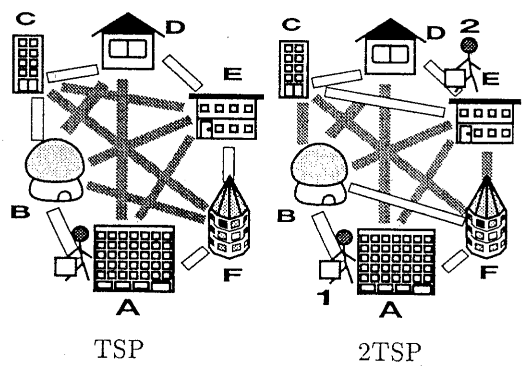

図2:

3 2TSPのモデル化

3.1 都市の訪問順行列

Hopfieldによって提案されたTSPの解法では、図3のように、都市の訪問順序を表現するものを使っている。

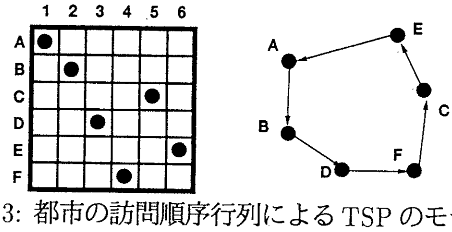

図3: 都市の訪問順序行列によるTSPのモデル

そこで、このモデルを使って2TSPの実現が可能かどうかの検討をおこなった。

このモデルを2TSPに当てはめると、図4のようにTSPのときの行列を2枚用意した形になる。

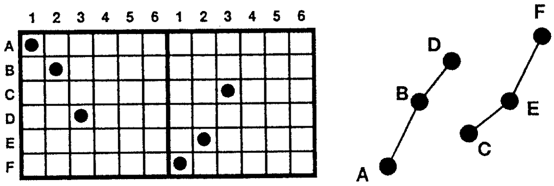

図 4: 都市の訪問順序行列による2TSPのモデル

このモデルでは、各々の行列が1人のセールスマンの訪問する都市と、その訪問順を表現している。よって全6都市の内、1人目のセールスマンが3都市だけを巡るというような場合が多いと思われるので、発火するニューロンは各行・各列に1つずつではなく、ある列以降はニューロンが1つも発火しないという状態が最終状態にならなければならない。このためにTSPのときのように、制約項(TSPの場合各行・各列に発火するニューロンが1つずつ)がうまく作れない。また評価関数であるが、TSPの場合はd_{ij}を都市i,j間の距離、x_{jk}はj行k列のニューロンの内部電圧とすると、次式で巡路の距離を表現できる。

$$E = \sum_i \sum_j \sum_k d_{ij} x_{ik} (x_{j\ k+1} + x_{j\ k-1}) \qquad (1)$$

しかし2TSPの場合には、途中の列でニューロンの発火が止まってしまうので、ループの巡路距離は求められず、片道の距離になってしまう。これは次のような点で不都合である。

- TSPの場合には巡路となるので、表現上どの都市が出発点になっても実際にはループなので、別の都市が出発点でも構わないが、片道の総距離が評価関数となるような、このモデルの2TSPでは出発点を限定することになる。つまり出発点が決まっているときには、それを与える手段が必要となり、そのための制約項がいる。
- 各列1つずつのニューロンが発火するわけではないので、非常に制約項が作りにくい。

以上の点から、このモデルを2TSPに適用することは断念した。つまり、問題に応じたモデル化というものの重要性がここでも明らかになったといえ、これはエネルギー関数の制約が1つの原因となっている。

3.2 接点接続行列

そこで、TSPのモデルとして研究されている接点接続行列を用いたモデルを検討した。このモデルは都市の隣接性に基づいた解法として知られている。

具体的には次のように2TSPのモデル化を行なった。

連続値型のホップフィールドモデルを採用し、[図??]のように各都市を節点と見た時の経路の節点接続行列を表現する。図5の右側は実際の都市配置とそれを2人のセールスマンが分業して巡る巡路を表している。つまり2人のセールスマンはA → B → C,D → E → Fと巡ることになる。左側はこれを2つの節点接続行列で表現した形になっている。つまり図5では、都市AB間に巡路があるので節点接続行列ではA行B列とB行A列のニューロンが発火することになる。2TSPの場合2人のセールスマンに対する巡路を各々表現する必要があるため各々に節点接続行列を用意してある。そのため、見かけ上TSPを解くためのニューラルネットワークモデルが2枚並んだ形になっている。

設定したエネルギー関数は以下の通りである。

$$E_1 = \sum_i^N \left(\sum_j^{2N} x_{ij} - 2 \right)^2 + \sum_j^N \left(\sum_i^N (x_{ij} + x_{i\ j+N}) - 2 \right)^2 \qquad (2)$$

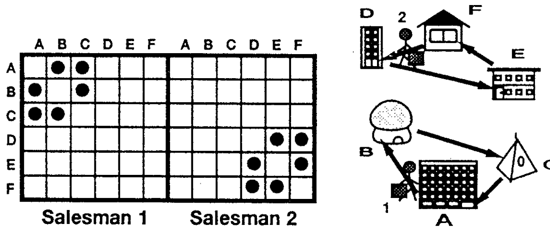

図 5: 2TSP のニューラルネットワークモデル

$$E_2 = \sum_{i}^{N}\sum_{j}^{N} d_{ij}\left(x_{ij} + x_{i\ j+N}\right) \tag{3}$$

$$E_3 = \sum_{i}^{N}\left\{\left(\sum_{j}^{N} x_{ij} - 2\right)^2 + \left(\sum_{j}^{N} x_{i\ j+N} - 2\right)^2\right\} \tag{4}$$

$$E_4 = \sum_{j}^{N}\left\{\left(\sum_{i}^{N} x_{ij} - 2\right)^2 + \left(\sum_{i}^{N} x_{i\ j+N} - 2\right)^2\right\} \tag{5}$$

$$E_5 = \sum_{i}^{N}\sum_{j}^{N}\left\{(x_{ij} - x_{ji})^2 + (x_{i\ j+N} - x_{j+N\ i})^2\right\} \tag{6}$$

$$\begin{aligned} E_6 &= \sum_{i}^{N}\sum_{j}^{N}\{u(d_{\alpha i} - d_{\beta i})u(d_{\alpha j} - d_{\beta j}) \\ &- u(d_{\beta i} - d_{\alpha i})u(d_{\beta j} - d_{\alpha j})\}\left(x_{ij} - x_{i\ j+N}\right) \end{aligned} \tag{7}$$

$$E_7 = \sum_{i}^{N}\left(\sum_{j}^{N} x_{ij}\sum_{j}^{N} x_{i\ j+N}\right) \tag{8}$$

$$E_8 = \sum_{j}^{N}\left(\sum_{i}^{N} x_{ij}\sum_{i}^{N} x_{i\ j+N}\right) \tag{9}$$

ただし x_{ij}:i 行 j 列のニューロンの発火量,d_{ij}:都市 i,j 間の距離,N:都市数,α β:2 人のセールスマンの各々の出発都市として示唆した都市番号,$u()$:単位ステップ関数 である。
各エネルギー関数の意味は次の通りである。

E_1 各行・列に発火したニューロンが 2 つずつの状態がエネルギー最小にするような制約関数

E_2 2 つのループの距離のそれぞれが出来るだけ小さくなるようにする評価関数

E_3 それぞれのネットワークで各行に発火したニューロンが 2 つずつの状態がエネルギー最小にするような制約関数

E_4 それぞれのネットワークで各列に発火したニューロンが 2 つずつの状態がエネルギー最小にするような制約関数

E_5 各ネットワーク内でニューロンが対称的に発火するような制約関数

E_6 始点として示唆した 2 都市との位置関係により各都市の分業方法を示唆する制約条件

E_7 各ネットの各行の和の積が 0 となる状態がエネルギー関数の最小になる状態に対応させたる制約条件

E_8 各ネットの各列の和の積が 0 となる状態がエネルギー関数の最小になる状態に対応させる制約条件

このエネルギー関数について、若干の説明をしておく。まず E_1,E_2,E_5 は、接点接続行列を用いた TSP の解法で使われているものを、そのまま 2TSP に拡張したものである。E_3,E_4,E_6 ~E_8 は 2TSP で新たに導入したエネルギー関数で、主にネットワークの状態を意味のあるものにするための制約関数である。

このように、2TSP では TSP に較べて多くのエネルギー関数を設定しなければならなかった。これは問題の複雑さのため、もしくはモデル化の方法によるものと思われるが、ここではこれらのエネルギー関数をその性質に応じて分類しておく。

強い制約条件 これに分類されるエネルギー関数は最終(収束)状態において各々に決まった値になっていなければ解が意味のあるものとならない条件である。よってこのエネルギー関数は他のエネルギー関数に優先されるべきものである。

弱い制約条件 弱い制約条件は最終状態においては不要な制約条件である。この制約条件はネットワークの状態遷移において初期状態では意味をもっても、最終状態では邪魔になるという性質をもつ。一見不要な条件のようにも思えるが、この制約条件によって初期状態においてネットワークを意味のある解へ向かわせるという効果がある。

評価関数 解の評価をする関数であり、この値が小さいほど良い解が得られたことになる。ただし強い制約条件を満たしているという前提が必要である。

2TSP のエネルギー関数では、E_2は2人のセールスマンの巡回距離を表す評価関数、その他は各行・各列の発火量などに関する制約条件であるが、絶対に満足されなければならない強い制約条件はE_1, E_5, E_7, E_8である。その他は弱い制約条件であり最終状態では満たされる必要がない。これらは大きく分けて2つの役割を果たしているエネルギー関数である。

- 2つのネットワークの発火状態が nTSP の解として意味のある状態になるように誘導する (E_3, E_4, E_7, E_8)
- 2人のセールスマンの始点を示唆して最適分業を行なえるように誘導する (E_6)

4 評価実験・考察

4.1 解の分類

2TSP では真の最適解とは2人のセールスマンが巡る各々の巡路ができるだけ短くなるように分業し、全都市を巡るという2つのループで表される巡路のことである。そこで 2TSP の評価のための解の分類を次のようにする。

最適解 真の最適解へ収束した場合

準最適解 真の最適解に近い状態に収束した場合（真の最適解と巡路の総距離の比が1.5倍以下の場合）

有意味解

- 最適解·準最適解に含まれないが2TSP の解へ収束した場合
- 一部解としては不備があるが簡単な修正で最適解・準最適解へなる場合
  - 「一部はずれ解」（後述）
  - 「他ループ所属解」（後述）

無意味解 上記いずれにも属さず「強い制約条件」を満たしていない場合

4.2 エネルギー関数に関する考察

組合せ最適化問題をニューラルネットワークを用いて解く際にはエネルギー関数の設定方法が非常に重要なのは明らかである。なぜならば組合せ最適化問題毎の特徴·差異は主にエネルギー関数の形にあらわれ、その他の部分は初期状態の設定法などを除けば問題によらないためである。よって問題に応じたエネルギー関数を正しく作るのはもちろんのことであるが、その調整が重要であり、また難問でもある。調整とはエネルギー関数の結合係数を調整することなどである。例えば特定のエネルギー関数の結合係数を大きくすれば、本来の組合せ最適化問題ではなくそのエネルギー関数のみを最小にすればよいような問題の解が求まってしまう。よってこれらの結合係数をうまく調整して本来の組合せ最適化問題を正確に表現するようにしなければならない。このことからも分かるようにこの調整は問題毎に異なり、一般的な調整方法も確立されていない。そのために試行錯誤して調整するのが普通である。2TSP においてもこの調整は思考錯誤によって決定をした。

次に 2TSP におけるエネルギー関数について考えてみる。エネルギー関数の一覧と意味については第3.2節の式（2）～（9）に示した

通りであり、3 種類に分類しておいた。この分類に基づいて考えると、強い制約条件である E_1, E_5, E_7, E_8 は最終状態においてエネルギー値が 0 に、評価関数 E_2 はできるだけ小さい値になることが望まれる。さて実際に 8 都市 2TSP において最適解が得られた場合（図 6）と最適解へ収束しなかった場合（図 7）のエネルギー関数の値の変化を以下に示す。

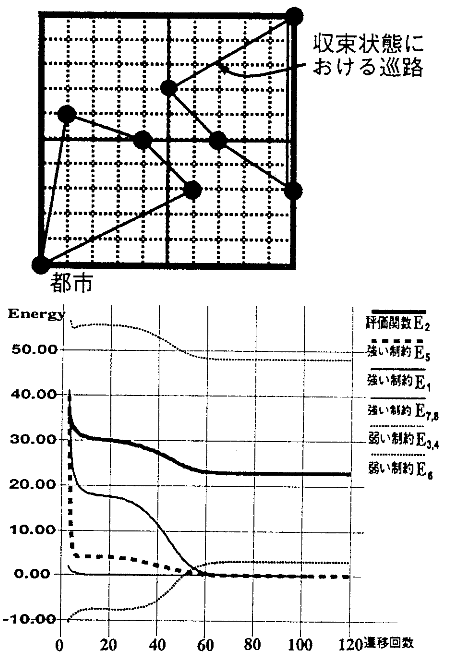

図 6: 最適解へ収束した場合の解とエネルギー関数の値の変化

これらを見ると強い制約条件の値に違いが表れている。つまり 2TSP では強い制約条件が最終状態で 0 にならなければならないのに、失敗例ではこれが 0 まで落ちていない。そこで強い制約条件を満たすためにエネルギー関数の調整をしなければならないと考えられる。つまりエネルギー関数について次のようなことがいえる。

- 問題の差異（都市配置の違いなど）に応じて調整をする必要がある。
- 状態遷移の段階により調整する必要がある。

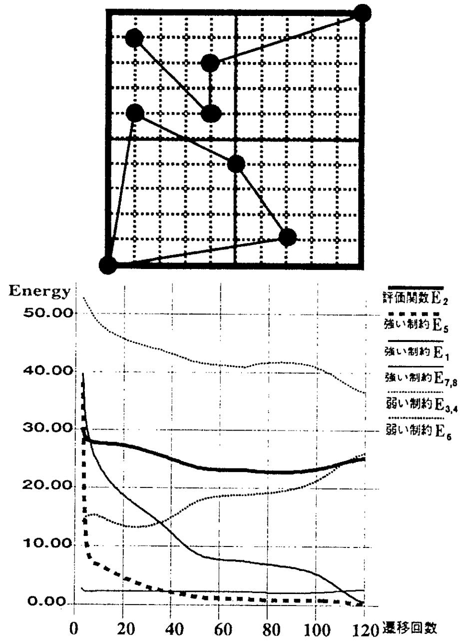

図 7: 最適解へ収束なかった場合の解とエネルギー関数の値の変化

これは問題によって制約条件をより強くしなければならない場合があるということと、状態遷移が最終（収束）段階に近付いているにも関わらず、強い制約条件が満たされていないときにこの制約条件を強くしなければならない場合があるということを意味している。しかし、問題の差異による調整は問題の種類をグループ化することが nTSP では難しく（つまり都市配置の違いをグループ化して分類することが難しい）、定性的にも扱うことが困難であるし、問題による差異は状態遷移の段階でエネルギー関数の値に見い出されると考えられるので、ここでは状態遷移の段階に応じた調整を考えることにする。

さて状態遷移の段階に応じた調整ということになるとまず適応制御が考える。しかしエネルギー関数の種類や値の状態に応じた適応制御の方法を考えることは変数が多過ぎて困難である。そこで次の 3 点に的を絞って制御を考えた。

1. 状態変化の初期の段階で弱い制約条件が他のエネルギー関数に比べて優先され過ぎている場合には弱い制約条件を表すエネルギー関数の結合係数を相対的に減少させる。

2. ある程度状態遷移が進んだ時に強い制約条件があまり満たされていない場合にはその制約条件を表すエネルギー関数の結合係数を相対的に増加させる。

3. 最終状態に近付いた場合には弱い制約条件を表すエネルギー関数の結合係数を 0 として、最終的な収束への障害を取り除く。

このうち 1 番目の制御は弱い制約条件というものの扱いが難しいことに加え、優先され過ぎているという基準が問題によりことなることもあって実現ができなかった。2 番目の制御は最終状態が意味を持つ解となるように誘導するものであるが、あまり強く誘導し過ぎてしまうと強い制約条件は満たすが、評価値は良くない解となってしまうことがあり、この制御のタイミングと程度を決めるのが難しい。しかしどのような都市配置の問題についても遷移回数が 50〜70 回程度でネットワークの状態が落ち着いてくることから、これを基準に制御を行なえば良いと思われる。3 番目の制御は最終状態では不要な弱い制約条件を取り除くなどし、収束速度を上げる目的で行なう。このタイミングは強い制約条件がほぼ満たされた時とすることで良いであろう。以上のことから次のような制御を加えて 8 都市 2TSP においてシミュレーションを行なった。

1. 遷移回数 70 回目に強い制約条件を表すエネルギー関数の値を調べ、要望値（2TSP では 0）に近付いていない（具体的にはエネルギー関数値が 1 以上）場合には強いエネルギー関数の結合係数を 50 %増加する。

2. 強い制約条件を表すエネルギー関数の値が要望値に近付いた場合には弱い制約条件を表すエネルギー関数の結合係数を 0 にし、その後は強い制約条件と評価関数を表すエネルギー関数のみで状態遷移を行なう。

この結果最終状態と収束速度（最終状態に収束するまでにかかった遷移回数）が表 2, 図 8 のように改善された。最終状態の表中の各解の意味は第 4.1 節で分類した通りである。最終状態については無意味解が有意味解に改善されたという場合が見られた。これは上記の 1 番目の制御によって強い制約条件を強くした結果が表れたものと考えられる。また収束速度についても、2 番目の制御によって状態遷移の最終段階における速度の改善があり、おおよそ 10 %程度収束までにかかる時間が短縮された。

制御	最適解	準最適解	有意味解	無意味解
なし	48 %	15 %	22 %	15 %
あり	51 %	15 %	28 %	6 %

表 2: エネルギー関数の結合係数の制御による最終状態の改善

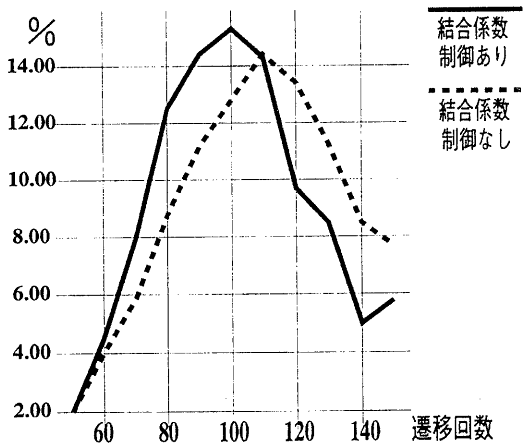

図 8: エネルギー関数の結合係数の制御による収束速度改善

4.3 全体評価

2TSP という問題を扱った研究が他に見あたらないので、全体的な評価を次のような形で

行った。

都市数の変化 全都市数が6,8,10の3通りの場合の解の状態をシミュレーションにより評価

修正による変化 8都市2TSPについて一部の解に修正を加えることによりどの程度解の状態が改善されるかの評価

都市数	最適解	準最適解	有意味解	無意味解
6	56 %	8 %	27 %	9 %
8	51 %	15 %	28 %	6 %
10	45 %	10 %	35 %	10 %

表 3: 都市数による2TSPの全解最終状態

6,8,10都市2TSPをニューラルネットワークを用いて解いた結果を示す。実験数は各々100である。また都市配置は各実験毎にランダムにとってある。表3は解を第4.1節のように分類し、それぞれへの収束率を表したものである。これらを見ると都市数の増加ともに解の状態が悪化していることが分かる。しかし有意味解の中には図9のような（準）最適解に近いものが多く含まれた。図中Aの解は一部のループが独立した部分巡路となってしまった場合で、これを「一部はずれ解」と呼ぶことにする。Bの解は巡路の一部が他のセールスマンの分担になってしまった場合で、これを「他ループ所属解」と呼ぶことにする。図9Bでは実線と点線の部分がそれぞれのセールスマンの巡路という解になってしまった場合である。点線の一部は実際には実線となるはずである。8都市2TSPの解では表4のように不備のあった解の多くがこのどちらかに属すことが分かる。「一部はずれ解」の場合もっとも近い巡路を持つセールスマンの巡路の一部として取り込むことにより（準）最適解になることが多い。またこのような修正は簡単に行える。「他ループ所属解」の場合、1つ1つのループを1人のセールスマンの巡路とみなすことで簡単に（準）最適解へ修正できる。つまりこの2種類の不備のある解は容易に修正できるので、これを修正した結果を示しておく。また表5は表3と同じ形式で8都市2TSPの解の修正前と修正後をかいたものである。修正の結果最適解または準最適解

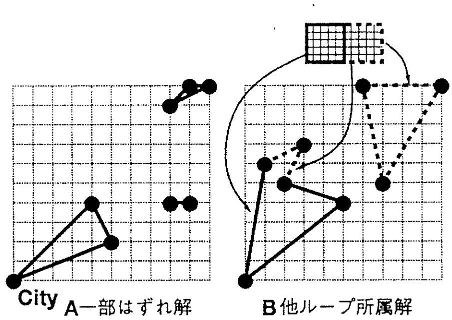

図 9: 修正により (準) 最適解となる解

一部はずれ	他ループ所属	TSPの解	その他
50 %	25 %	20 %	5 %

表 4: 8都市2TSでの不備のある解の内容

都市数	最適解	準最適解	有意味解	無意味解
8（修正前）	51 %	15 %	28 %	6 %
8（修正後）	65 %	22 %	7 %	6 %

表 5: 修正前後の8都市2TSPの全解最終状態

への収束率は80 %を越え、TSPをニューラルネットワークで解いた場合に近い値を得ることができた。この修正は都市数や分業人数の増加によって修正数が増加すると思われるが、修正を自動化することも容易で、簡単な修正で解の状態が改良されると考えられる。

5 おわりに

nTSPという問題は応用範囲が様々考えられるにも関わらず、これまでこの種の問題に対する研究というのは、確認した範囲では見あたらない。その理由として

1. 非常に解空間が広く[1]アルゴリズム的に解くことが困難である。

2. 問題設定が計算時間・メモリ等の点で非常に大きくなる。

という2点が考えられる。そこでこれらの困難な状況に対して、ニューラルネットワークを用いることにより実用時間内に準最適解を求める手法を適用し、nTSP問題の簡単な場合として2TSPによるシミュレーションを行った。その際、この種の手法の未解決部分である初期値の問題を取り上げ、nTSPにおいては分業状態を示唆するような初期状態を持たせることがよいことが実験的にいえた。また一般的な問題おいてもエネルギー関数の設定において弱い制約条件というものを新たに取り入れ、これによって複雑な問題もある程度ニューラルネットワークを用いて解くことができることを示し、エネルギー関数の結合係数を制御することで性能が上がった。そして局所的最小値を避けられるのではないかという可能性もある。このような工夫により2TSPにおいては実用時間内[2]に準最適化解がかなりよい確率で得られることがいえた。しかしながら大規模な問題に対する研究はまだ行えておらず、様々なスケジューリング問題等にnTSPを基礎技術として応用が可能であると思われるが、これらについて具体的に構成することは行えていない。これらの点は今後の課題である。

参考文献

[1] 甘利俊一. ニューロコンピュータの現状と将来. 共立出版, 1990.

[2] 甘利俊一. ニューラルネットの新展開–研究の最前線を探る–. サイエンス社, 1993.

[3] 上坂吉則. ニューロコンピューティングの数学的基礎. 近代科学社, 1993.

[4] 村島定行, 萬膳義久. 巡路中の都市の隣接性に基づいた巡回セールスマン問題のニューラルネットへの埋め込み. NC90 48, 電子情報通信学会, 3 1991.

[5] 中野馨. Cでつくる続脳の情報システム. 啓学出版, 1992.

[6] 麻生秀樹. ニューラルネットワーク情報処理. 産業図書, 1988.

[7] 林豊, 清水利幸, 松本直樹. 接点接続行列を用いたＴＳＰの解法. NC92 139, 電子情報通信学会, 3 1993.

[1] TSPよりもさらに広い

[2] ワークステーション(SS10)で数秒〜30秒程度