自己組織化マップ法による巡回セールスマン問題の解法Ⅱ

西　見　康　平（電子情報工学科）
加　藤　友　彦（電子情報工学科）

Semi-Optimum Solution of Traveling Salesman Problem by Self-Organizing Maps Ⅱ

Kouhei NISHIMI (Department of Information Electronics)
Tomohiko KATO (Department of Information Electronics)

Abstract

The traveling salesman problem is one of the most difficult problems in optimization problems. In this study we improve the method of B. Angeniol et al. based on the self-organizing maps (SOM) by T. Kohonen in several points. We apply the present method systematically to 100-, 500-, 1000-, 5000-, 11849-city problem. The result shows that the computing time is proportional to approximately the square of the number of cities. That is, the present method gives a polynomial algorithm, though within the limits of semi-optimum solutions, for the traveling salesman problem that is one of the representative problems of the NP complete problem.

Keywords: *traveling salesman problem, NP-complete problem, self-organizing map*

１．序

巡回セールスマン問題（Traveling Salesman Problem：以下 TSP と略記）とは，提示された全都市を一度だけ訪問し，最初の地点へ戻ってきたときの経路が最短になるような経路を見出す問題である。全都市を一度だけ巡る経路の数は，都市数を n としたとき $(n-1)/2$ で与えられる。n が小さいときは簡単に最適解が得られるが，n が多くなるにつれて経路の数は飛躍的に増大し，わずか100都市の場合でも 10^{156} のオーダーとなり，スーパーコンピュータを用いても，現実的な時間内ですべての経路は計算し尽くせない。このような性質から，TSP は NP（Non-deterministic Polynomial）完全問題に分類される。都市数が膨大な TSP に対し厳密解を求めることは困難なため，実際的な計算時間の中で比較的良い精度の準最適解を与えてくれる手法が必要となる。

我々の研究室では，平成12年度から卒業研究，平成15年度からは修士論文の一つのテーマとして，TSP の準最適解の研究を続けている。この間，ニューラルネット方式によるモンテカルロシミュレーション（熱浴法，マルチカノニカル法），遺伝アルゴリズム，自己組織化マップ法（以下 SOM と略称）によりプログラムを作成し，結果を比較検討してきた。このうち，天野のマルチカノニカル法の結果[1)]と，小林による SOM の結果[2)]をこの研究論集に発表している。この間の経験から，少なくともニューラルネット方式，遺伝アルゴリズムに比較して，自己組織化マップ法が圧倒的に優れた結果をもたらすことが明らかになった。

本研究では，小林の SOM のアルゴリズムをさらに３つの点で改良し，多数の都市の問題について現実的な時間内で準最適解を求める手法を与えることを試みた。なお，本研究で使用したパソコンは Pentium4, 3.40GHz である。

２．TSP に対する自己組織化マップ法の適用

自己組織化マップ法（Self-Organization　Maps：以下 SOM と略記）とは，コホネン[3)]（Teuvo Kohonen）によって提案された，ニューラルネットワークの一つである[4)]。ニューラルネットワークとは脳機能をモデル化し，計算機上のシミュレーションによって表現することを目指した数学モデルである。ニューラルネットワークには，あらかじめ正解（＝教師信号）が提示され，その正解の方向へ最適化されていく「教師あり学習」と，それを必要としない「教師なし学習」に」分けられる。SOM 法は「教師なし学習」

平成21年５月28日受付

に分類される，入力層と出力層のみを持つ２階層型のモデルであり，入力ベクトルの中に存在するある傾向や相関関係などの情報を見つけ出しやすい長所がある。

このSOM法を初めてTSPに用いたのがアンジェニオール[5)]（Angeniol）らであり，本研究においても，アンジェニオールのアルゴリムを改良したものを用いることとする。

アンジェニオールのアルゴリズムでは，まず二次元平面上に都市と初期ノードを配置する。初期ノードは，通常１つである。

次に，提示された都市に最も近いノードを勝利ノードとする。もし最近傍のノードが既に他都市の勝利ノードであった場合には,同座標にノードを複製し,それを勝利ノードとする。

勝利ノードが決定したならば，勝利ノードを専属する都市の方向へ動かす。このとき他の全てのノードも同時に移動するが，その移動する距離は，勝利ノードから遠い順番にあるものほど短くなっていく。このとき，移動する割合は，あらかじめ与えられるパラメータGによって左右される。Gは更新する近傍範囲を調節するパラメータで，Gが大きいほど多くのノードが更新されるが，小さければ更新されるノードの数と，移動距離は短くなっていく。

以上の動作を全ての都市に対して行い，その終わりにGを更新する。更新係数 α は $0<\alpha<1$ までの範囲で与えられ，通常１に近い数値を与えられる。

また，全都市調査を三回繰り返して，その間一度も選ばれなかったノードは消去される。

これらを繰り返し，全ての都市がノードを獲得し，ノード数に変化がなくなったら終了する。

図１にアンジェニオールのアルゴリズムのフローチャートを示す。

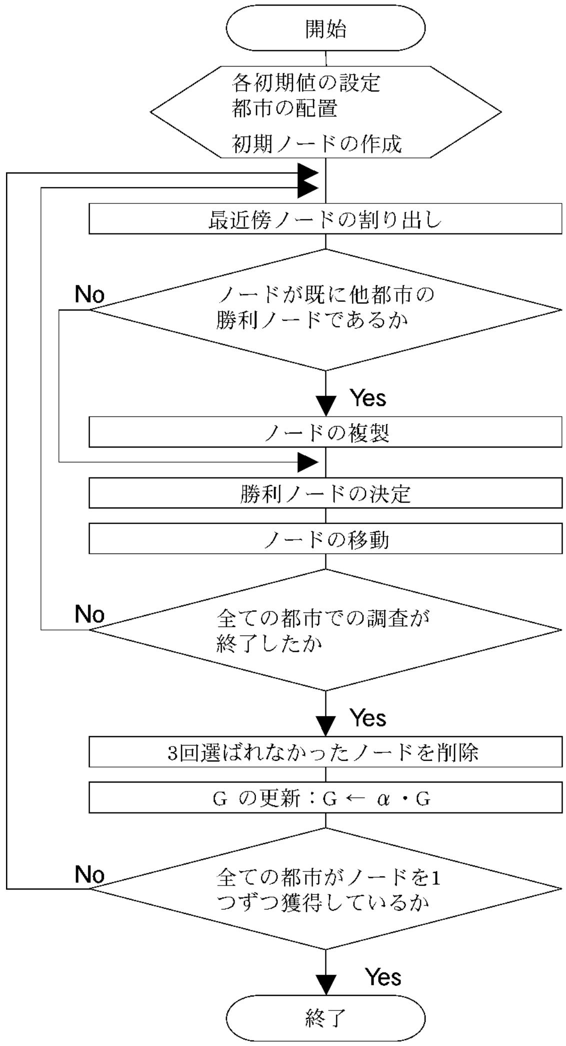


図１　アンジェニオールのアルゴリズムのフローチャート

３．プログラムの改良

自己組織化マップ法は，現実的な時間で準最適解を導き出すのに有用な手法であるが，都市数が多くなれば，計算時間は増大していく。そのため，精度をあまり落とさず計算時間を短縮する方法が求められる。

そこで，前年度までの成果を参考にしながら，プログラムを改良していくことにする。改良点として，

⑴　プログラムにおける構造体の使用
⑵　初期ノード配置の見直し
⑶　更新係数 α の検討

の三つを取り上げ，検討した。

3.1　プログラムにおける構造体の使用

SOM法によるTSPの解決において，都市とノードというそれぞれ複数の要素を持ったデータが存在する。ここではそれを構造体の形にまとめることで，プログラム自体の可読性の向上と，ノードの複製・削除の簡略化，データのやり取りのしやすさの向上を図る。

構造体は以下のように構成する。

```
構造体 City
　{
　　・都市番号
　　・x座標
　　・y座標
　　・勝利ノード
　}
```

```
構造体 Node
　{
　　・x座標
　　・y座標
　　・専属都市
　}
```

情報を構造体ごとにまとめることで，プログラムを視覚的に理解しやすくなる。

構造体を用いることで，ノードの複製・削除の際に，各要素を一度に複製することが可能となる。この際，memcpy関数を用い，構造体のアドレスを指定して作業を行う。

3.2　初期ノード配置の見直し

初期ノードは通常１つだけ与えられるが，ここではその

数と初期ノードの配置について検討し，より良い初期ノードの設定がないかを考察する。

実験に当たって，100都市と1000都市の場合で，それぞれ初期ノードを，

＜１＞座標（０，０）に１つだけ配置した場合

＜２＞都市と同数のノードを円状に配置した場合

＜３＞都市と同数のノードを乱数によってランダムに配置した場合

の三つの配置によって結果にどのような違いが出るかを考察する。

以下に結果を記す。なお，都市の座標は乱数によって決定しており，都市配置を変えるごとに３つのパターンすべてを試行し，それを５回ずつ行った平均を結果として記している。

100都市	配置＜１＞	配置＜２＞	配置＜３＞
距離	8.60	8.46	8.31

1000都市	配置＜１＞	配置＜２＞	配置＜３＞
距離	26.36	26.84	26.23

若干ではあるが，ランダムな配置の場合，良い結果が出ている。ただしその差は少なく，顕著な変化が出るまでには至らなかった。

また，都市と同数の初期ノードを配置した場合，ノード数が１つの場合と比べて計算時間が多くなった。これは初期ノード数が多くなった分，ループの回数が多くなり，そのために時間がかかったものと思われる。

以上のことから，初期ノード配置はあまり結果に変化をもたらさないことが分かった。しかしノードの数や配置の方法などによってはまた違った変化が見られる可能性もあり，まだ検討の余地はあるものと思われる。

3.3　更新係数 α の検討

都市 i が勝利ノード（j_C とする）を決定したとき，勝利ノードとその近傍のノードは都市の方向へ移動することになる。更新しようとしているノード j が，勝利ノードからノード同士のリングに沿ってどのくらいの位置にあるのかを以下の式によって求める。

$$n=\min\{(j-j_C)\bmod N,\ (j_C-j)\bmod N\}$$

n は距離尺度を意味し，この n とパラメータ G から，次式によって更新率 $f(G,\ n)$ を決定する。

$$f(G,\ n)=\frac{1}{\sqrt{2}}\exp\left(-\frac{n^2}{G^2}\right)$$

この更新率を，都市の座標とノードの座標との差に乗算することで更新率が決定する。すなわち，

$$c_k^j \leftarrow c_k^j+f(G,\ n)\cdot(x_k^i-c_k^j)$$

である。ここで c_k^j はノードの座標を，x_k^i は都市の座標を表している。

以上の式に基づいて各ノードの移動を全都市において終了させた後，更新係数 α によってGを更新する。

$$G \leftarrow \alpha\cdot G$$

解を収束させるために，Gは減少されなければならない。そのため α は $0<\alpha<1$ の範囲内の数値を与えられ，通常は１に近い値を与える。添付したソースプログラムでは $\alpha=0.95$ としている。

更新係数 α は定数であるが，本項ではこれを全都市調査 t 回の関数として $\alpha(t)$ と置き換えることにした。ノード数の増減に応じて，以下のように α を更新していく。

$$\alpha(t+1)=\alpha(t)+\frac{\gamma}{M}\cdot\{C(t)-C(t-1)\}$$

M は全都市数を，$C(t)$ は全都市調査 t 回目におけるノードの数を表している。γ は定数であり，通常は負の定数として設定する。これによって，

・ノード数が前回に比べて増えた場合

　$\alpha(t)<\alpha(t-1)$ となり，Gの減少量が多くなる。

　ノードの移動半径が狭まりやすくなる。

・ノード数が前回に比べて減った場合

　$\alpha(t)>\alpha(t-1)$ となり，Gの減少量が小さくなる。

　ノードの移動範囲が狭まりにくくなる。

となることが分かる。

すなわち，全都市調査序盤はノードの数が大きく増えるため α の値も小さくなり，G も大きく減少するが，ノードが減少し始めると G はある程度減少率を保ったまま更新されていくことになる。

これによって計算時間や経路長にもたらされる変化と，γ の値によって結果がどのように左右されるかを，検討した。まず，定数 γ の値を様々に変えながら，それによる結果を検討する。

γ の値を0.0，−0.2，−0.4，−0.6，−0.8，−1.0として，５パターンのランダムな都市配置においてプログラムを実行した。その計算時間と経路長の平均値を下表に示す。初期ノー

γ	計算時間 [s]	経路長
0	7.84	26.05
−0.2	7.87	26.54
−0.4	8.89	26.03
−0.6	13.02	26.74
−0.8	22.51	28.65
−1.0	88.02	29.01

ドは都市と同数を乱数によって配置するものとし，その他の初期値は以下の通りである。

・都市数：1000
・G：40.0
・α：0.95

この結果より，γ＝0 である場合，つまり α が定数である場合に比べ,劇的に計算時間が短くなっていることが分かる。経路長は減少した後，γ＝-0.4 を過ぎたあたりから徐々に長くなっていっているが，計算時間は指数関数的に下がり続けている。

また都市数に応じてどのように計算時間に差が出たかを，次の表に示す。なお，条件は上と同じである。

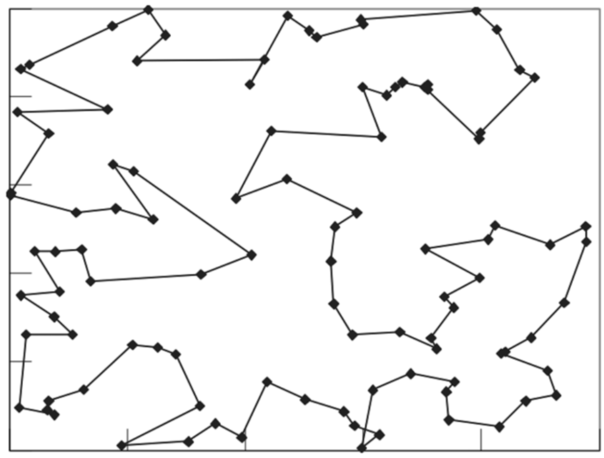
		α＝0.95（定数）	$\alpha(0)$＝0.9 γ＝-0.4
100都市	所要時間	0.64s	0.16s
	経路長	8.18s	8.34s
500都市	所要時間	18.18s	2.79s
	経路長	18.48s	18.36s
1000都市	所要時間	1m28.02s	13.02s
	経路長	26.05s	26.03s

このことから，パラメータ γ には都市数に関係なく計算時間を減少させる効果があることが分かった。しかし γ の値によっては精度が下がる場合もあり，γ を設定する場合には，適切な値を与える必要がある。

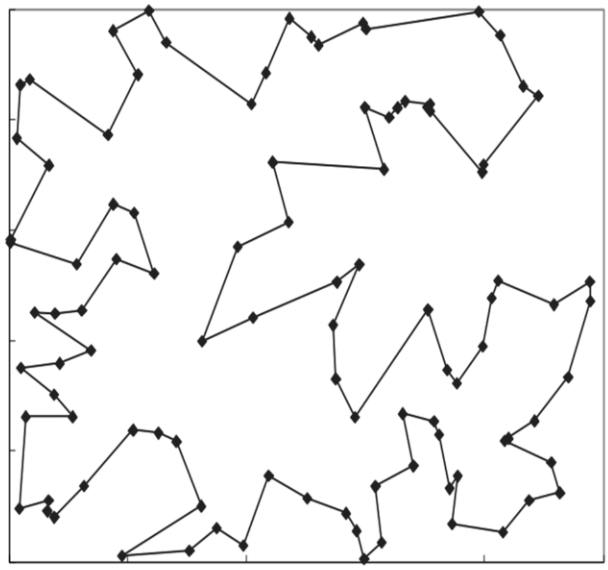
4．自己組織化マップ法の適用結果

上記の方法を，ランダムに配置した100都市，500都市，1000都市，5000都市の問題に適用した。その結果を以下に示す，100都市の問題については，比較のため γ＝0 の場合と γ＝-0.4 の場合の結果を示す。

<100都市>　G＝10.0　$\alpha(0)$＝0.95　γ＝0.0
計算時間：0.36s　経路長：8.24

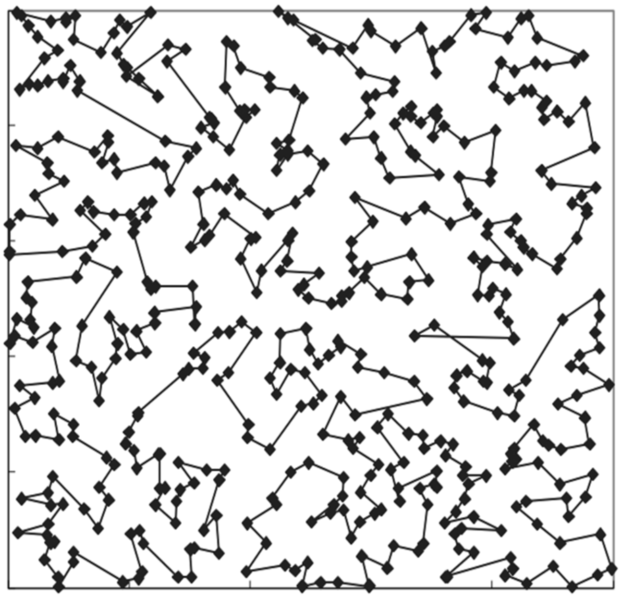


<100都市>　G＝10.0　$\alpha(0)$＝0.95　γ＝-0.4
計算時間：0.08s　経路長：8.06



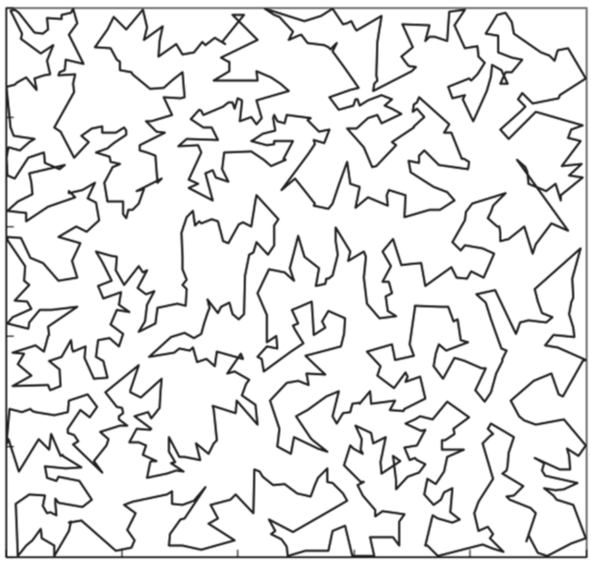
γ＝-0.4 の場合は計算時間が 4 分の 1 以下になり，経路も短い結果となっている。このことから，500都市以上の問題については，γ＝-0.4 の場合の結果のみを示す。

＜500都市＞ G＝20.0　$\alpha(0)$＝0.95　γ＝-0.4
計算時間：2.46s　経路長：18.78



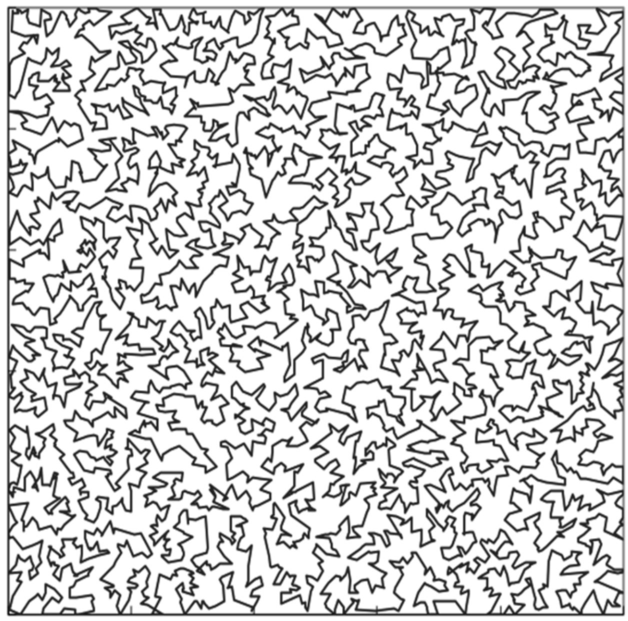
なお，γ＝0 の結果は，計算時間 15.23s，経路長 18.47 である。

＜1000都市＞ G＝40.0　$\alpha(0)$＝0.95　γ＝-0.4
計算時間：13.28s　経路長：25.75



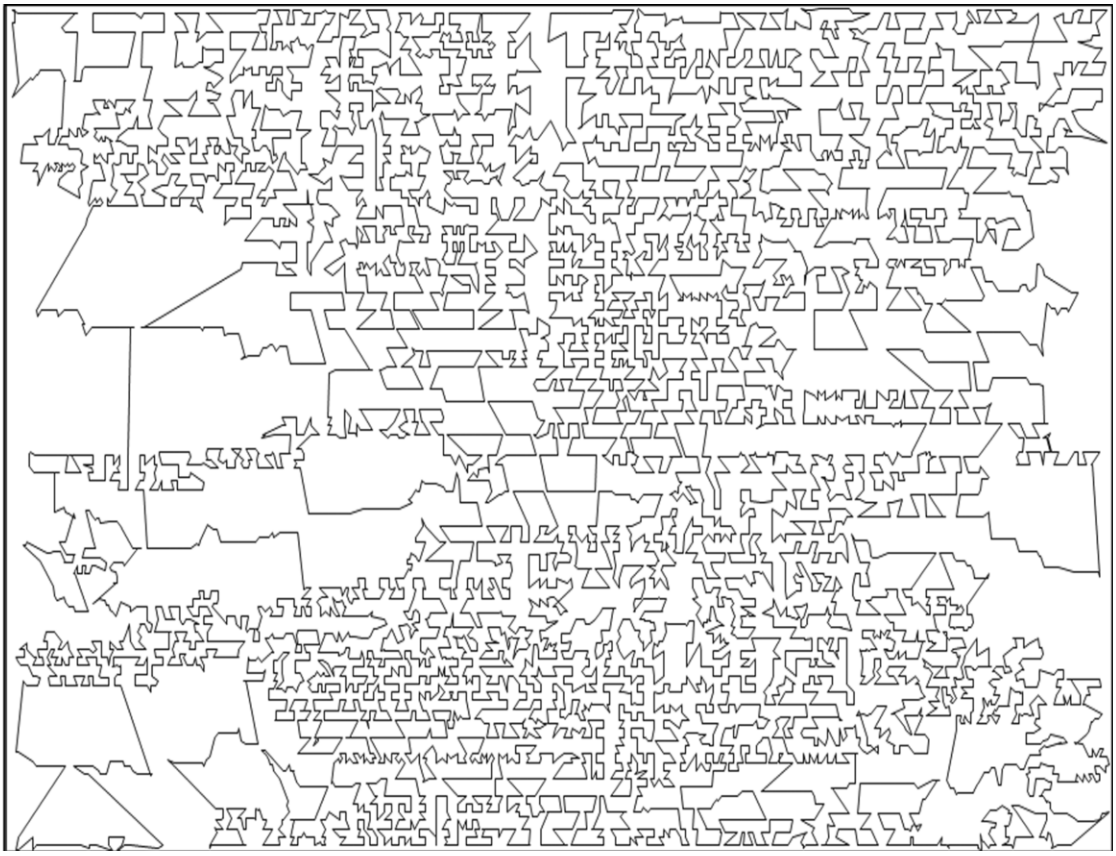
なお，γ＝0 の結果は，計算時間 87.49s，経路長 25.86 である。

＜5000都市＞ G＝40.0　$\alpha(0)$＝0.95　γ＝-0.4
計算時間：361s　経路長：57.59



ここまでの問題は，都市の配置をランダムに設定したものであるが，最後に TSP に関する世界的なデータベースである TSPLIB に登録されている標準問題の内の11849都市問題に適用した結果を示す。

標準問題＜11849都市＞G＝40.0　$\alpha(0)=0.99$　$\gamma=-0.2$
計算時間：2468.86s　経路長：1079022.296



この結果を見ると，厳密解が知られていないのでどの程度の制度化か評価できないが，経路が交差しているところはほとんどないことから見て，かなり良い精度の準最適解と思われる。

5．考察

4.で示した５つの問題に対する結果に関して，都市数 n と所要時間 t の関係を見るために，図２に n と t の log-log プロットを示す。結果はほとんど直線に乗っており，その勾配は約2.2となっている。すなわち，都市数 n と所要時間 t の関係は

$$t=6.2\times10^{-6}n^{21}\,[\mathrm{s}]$$

という簡単な式で表現される。TSP は NP 完全問題の代表的問題であるが，ここで報告する SOM を用いた方式では，低い次数の多項式で処理できることが分かった。準最適解ではあるが，この結果は有益なものであると考える。TSPLIB に登録されている最大の都市数は，85900であるが，上式を外挿すれば，約40時間で計算できることになる。今後の課題としたい。

本研究は，ここ数年において加藤研究室で取り組まれてきた巡回セールスマン問題の研究を引き継ぐものである。３つの改良点に関して，構造体の使用については，プログラムの構造を簡明にできた利点があるが，計算時間の短縮につながったかどうかは明らかでない。初期ノードの配置の検討については，期待したほどの成果が得られなかったが，さらに他の配置を考えれば効果が出るかも知れない。更新係数 α の検討については有効な改良が達成できたと判断する。

しかし当初の目標であった精度の向上と計算時間の短縮のうち，後者は成果があったが，前者についてはあまり成

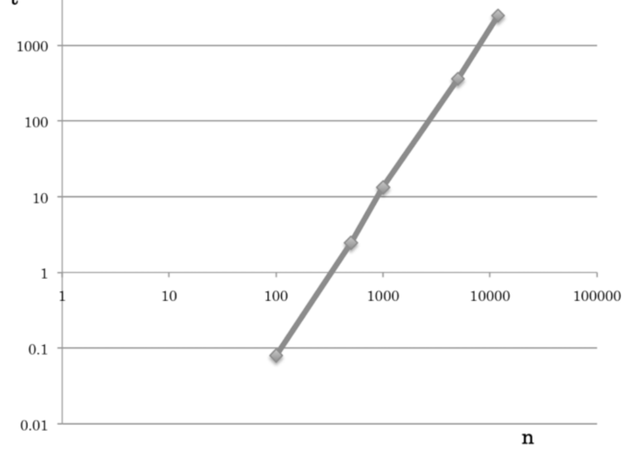


図２　都市数 n と所要時間 t の log-log プロット

果を上げられなかった。精度と計算時間は，あらかじめ設定しておくGの値に多少左右されるが，よほどGを大きく変えない限り，大幅には変化しない。この点では小林の行った徐冷法と組み合わせることが有効かもしれない。

参考文献

［1］ 天野恵美子，加藤友彦：巡回セールスマン問題に対するマルチカノニカル法の適用，福岡工業大学研究論集37巻第1号11頁，2004.

［2］ 加藤友彦，小林　徹：自己組織化マップ法による巡回セールスマン問題の解法

［3］ T. Kohonen；徳高平蔵・岸田　悟・藤村喜久朗訳；自己組織化マップ，シュプリンガー・フェアラーク東京株式会社，1996.

［4］ 徳高平蔵・岸田　悟・藤村喜久朗；自己組織化マップの応用，海文堂出版，1999.

［5］ BERNARD ANGENIOL, GAELDE LA CROIX VAUBOIS AND JEAN-YVES LE TEXIER, Self-Organizing Feature Maps and the Traveling Salesman Problem, Neural Network, 1 (1988) 289.