メタヒューリスティックス解法の評価
– 巡回セールスマン問題を例として –

M2006MM015 眞野 祐樹

指導教員　伏見 正則

1　はじめに

現在，組合せ最適化問題は，経営，経済，科学，工学などの様々な分野で現れている．種々のスケジューリング問題や配員計画問題など，現実に現れる様々な問題が組合せ最適化問題として定式化できるが，その多くについて，厳密な最適解を求めることがきわめて困難 (NP 困難) であることが，計算の複雑さの理論により明らかにされてきた．

コンピュータの出現により，組合せ最適化問題の解を求めるためのアルゴリズムが多数現れ，様々な問題の解を求めることが可能になった．また，近年のコンピュータの高性能化に伴い，以前では解くことが難しかった比較的大きな問題に対しても，以前よりも短時間で計算することが可能となった．

しかし，現段階では組合せ最適化問題に対する様々な解法が存在するにも拘らず，どのような問題においてどのような解法を用いれば計算効率が良く，より厳密解に近い解を見つけることができるのかということについては，あまり研究がされていない．

そこで本研究では，組合せ最適化問題の一つである巡回セールスマン問題を例として挙げ，様々なメタヒューリスティックス解法について研究し，それらをプログラミングすることで，どの解法が効率的に解を求めることができるかを比較・検討し，どの問題に対しても短時間でより高精度な解を見つけることができるアルゴリズムを作成することを目的とする．さらに，巡回セールスマン問題の応用である時間依存巡回セールスマン問題についても，同様の方法で短時間で高精度な解を見つけることができるアルゴリズムを作成することを目的とする．

2　巡回セールスマン問題とは

2.1　概要

巡回セールスマン問題 (TSP) は，ある一定の範囲の中に複数の都市が配置されているとき，その中のある都市からセールスマンが他の都市に向けて出発し，他のすべての都市を一度ずつ訪れ，出発点に戻る．その総移動距離が最短になるような都市の訪問順序を探す問題を巡回セールスマン問題という．この問題は以下のように定式化される．

記号の定義：

V：頂点集合 $\{1, 2, \cdots, n\}$

E：枝集合 $\{(i,j) |\, i, j \in V,\ i \neq j\}$

d_{ij}：枝 (i,j) の距離

$$x_{ij}\text{: 0-1 変数} = \begin{cases} 1 & \text{枝 } (i,j) \text{ を通過する場合} \\ 0 & \text{枝 } (i,j) \text{ を通過しない場合} \end{cases}$$

S：V の真部分集合

$$\text{目的関数：}\quad \min \sum_{(i,j)\in E} d_{ij}\, x_{ij} \tag{1}$$

$$\text{制約条件：}\quad \sum_{j\in V-\{i\}} x_{ij} = 1 \qquad \forall i \in V \tag{2}$$

$$\sum_{i\in V-\{j\}} x_{ij} = 1 \qquad \forall j \in V \tag{3}$$

$$\sum_{i\in S}\sum_{j\in V-S} x_{ij} \geq 1 \qquad \emptyset \neq \forall S \subset V \tag{4}$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in E \tag{5}$$

2.2　計算時間

巡回セールスマン問題では，訪れるべき都市数 n が少ない場合には，巡回路を総当りで計算・比較し最短経路を選んでも，それほど時間はかからない．しかし，都市数が多くなれば，巡回する組合わせは爆発的に増大するため，全ての巡回路を計算・比較するのに膨大な時間がかかってしまう (表 1)．

表 1　巡回セールスマン問題の計算に要する時間 [3]

都市数	巡回路の数	計算時間
6	60	4.32×10^{-10} 秒
8	2520	3.23×10^{-8} 秒
15	4.36×10^{10}	4.87×1.96 秒
20	6.08×10^{16}	4.87×10^{6} 秒 (約 56 日)
25	3.10×10^{23}	43.88×10^{13} 秒 (約 122 万年)

・n 都市に対して $(n-1)!/2$ 通りの巡回路

・10TFlops のコンピュータ (1 秒間に 10^{13} 回浮動小数点演算できる)

3　メタヒューリスティックス解法とは

上記のように，現実的に要望された時間内で厳密な最短経路を求めることは難しい．この困難を避けるために，厳密解を求めることを諦めて，実用的な時間内にできる限り良い最短経路を求めるのが普通である．この目的を達成するために多くの解法があり，総称してメタヒューリスティックスと呼ばれている．

4　TSP における研究方針

巡回セールスマン問題の厳密解がすでに分かっているベンチマーク [2] を用いて，どのメタヒューリスティックス解法を利用すれば，効率的な時間で高精度な解が得

られるかを C 言語でプログラミングをすることによって比較・検討する．計算に使用した PC の環境は以下のとおりである．

OS :Microsoft Windows XP
CPU : Intel Celeron 2.53GHz
RAM: 1GB

また，本研究で使用した近似解法やメタヒューリスティックス解法を以下に示す．

- 近似解法
  - 最近近傍法
  - ランダム化最近近傍法
  - 2-opt 近傍法
- メタヒューリスティックス解法
  - アニーリング法
  - アント法
  - 多スタート局所探索法

5 近似解法

5.1 概要

近似解法の基本戦略として，欲張り法と局所探索法がある．欲張り法は，頻繁に利用される代表的な近似解法であるとともに，メタヒューリスティック解法においても，初期解や探索解の生成に用いられるなど，重要な役割を持っている．また，局所探索法は，手元にある解 (例えば欲張り法で構成されたもの) の改良を試みる一般的な方法であり，これを一般化したものがメタヒューリスティック解法であると捉えることもできる．よって，近似解法はメタヒューリスティックス解法を理解する上で，きわめて重要なものである．近似解法には，最近近傍法，ランダム化最近近傍法，2-opt 近傍法などがある．

5.2 計算結果

本研究では，最近近傍法，ランダム化最近近傍法，2-opt 近傍法について解の精度を比較した．これより，計算時間においては他の解法よりも多少時間がかかるが，解の精度の面では 2-opt 近傍法を用いるのが良いという結果が得られた．そのため，メタヒューリスティックス解法では 2-opt 近傍法を用いることにする．

6 アニーリング法

6.1 アルゴリズム

アニーリング法 (Simulated Annealing (SA 法)) とは，現在の解 x の近傍 $N(x)$ 内の各解 x' に，解の良さに応じた遷移確率 (良い解ほど移行しやすい) を設定し，それに従って次の解を選ぶ．改悪解であっても遷移する確率を与えることにより，局所最適解からの脱出を図るものである．遷移確率は，物理現象の焼きなましにアイディアを借りて，温度と呼ばれるパラメータ t により調整される．基本的なアルゴリズムを以下に示す．

アニーリング法の基本的なアルゴリズム

$f(x)$ は総移動距離を表す．

Step 1 初期解 x を生成し，初期温度 t を定める．

Step 2 ループの終了条件が満たされるまで，以下のステップ a, b, c を繰り返す．

a $N(x)$ 内の解をランダムに 1 つ選び x' とする．

b $\Delta := f(x') - f(x)$ とする (x' が改悪解ならば $\Delta > 0$).

c $\Delta \leq 0$ ならば確率 1，そうでなければ確率 $e^{-\Delta/t}$ で $x := x'$ (解 x' を受理) とする．

Step 3 反復の終了条件が満たされれば，暫定解を出力して終了する．そうでなければ，温度 t を更新した後，Step 2 に戻る．

6.2 計算結果

本研究では単純な冷却スケジュールを用いて，複数のパラメータを変化させて計算を行った．これにより，ほとんどの都市について誤差が 10% 未満になるという非常に高精度な結果が得られた．

7 TSP における計算結果

4 節で述べた近似解法やメタヒューリスティックス解法の各解法の中には，様々な方法が存在し，複数のパラメータの設定をする必要がある．各解法の中で一番精度の良い方法をそれぞれ選び，それらを一つのグラフにまとめた (図 1)．また，このときの計算時間を図 2 に示す．図 1 より，130 都市まではアニーリング法を用い，225 都市以降はアント法を用いるのが良いという結果が得られた．また図 2 より，計算時間においてはアニーリング法を用いたほうが良いという結果が得られた．

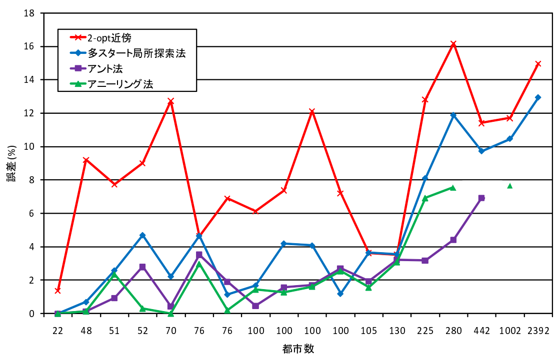

図 1 都市数と最適解との誤差 (解の精度)

8 時間依存巡回セールスマン問題とは

時間依存巡回セールスマン問題 (TDTSP) は，ある一定の範囲の中に複数の都市が配置されているとき，その中のある都市からセールスマンが他の都市に向けて出発し (出発地点をデポと呼ぶ)，他のすべての都市を一度ずつ訪れ，出発地点に戻る．ただし，各都市間の移動時間は都市を訪問する時間帯によって異なる．この総移動時間が最短となるような訪問順序を探す問題を時間依存巡回セールスマン問題という．この問題は以下のように定

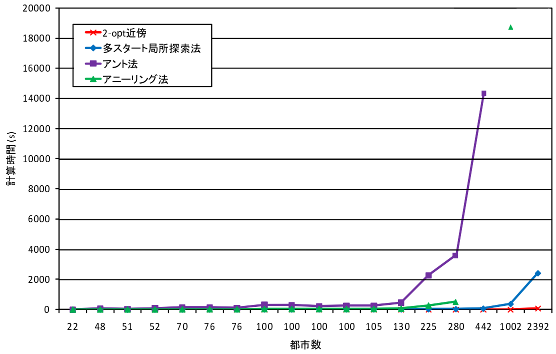

図 2　都市数と計算時間

式化される．ただし，時間帯は枝の出発点で判断する．

記号の定義：

V：頂点集合 $\{1, 2, \cdots, n\}$

E：枝集合 $\{(i,j) | \, i, j \in V, \; i \neq j\}$

M：時間帯集合 $\{1, 2, \cdots, m\}$

t_{ij}^k：時間帯 k における枝 (i, j) 間の移動時間

x_{ij}^k：0-1 変数 $= \begin{cases} 1 & \text{時間帯 } k \text{ における枝 } (i,j) \text{ を通過する場合} \\ 0 & \text{時間帯 } k \text{ における枝 } (i,j) \text{ を通過しない場合} \end{cases}$

S：V の真部分集合

目的関数：
$$\min \sum_{k \in M} \sum_{(i,j) \in E} t_{ij}^k x_{ij}^k \qquad (6)$$

制約条件：
$$\sum_{k \in M} \sum_{j \in V - \{i\}} x_{ij}^k = 1 \qquad \forall i \in V \qquad (7)$$
$$\sum_{k \in M} \sum_{i \in V - \{j\}} x_{ij}^k = 1 \qquad \forall j \in V \qquad (8)$$
$$\sum_{k \in M} \sum_{i \in S} \sum_{j \in V - S} x_{ij}^k \geq 1$$
$$\emptyset \neq \forall S \subset V \qquad (9)$$
$$x_{ij}^k = 1 \implies x_{jl}^{k'} = 0 \qquad (k' < k) \qquad (10)$$
$$x_{ij}^k \in \{0, 1\} \qquad \forall (i,j) \in E \qquad (11)$$

9　TDTSP における研究方針

時間依存巡回セールスマン問題では，巡回セールスマン問題の場合のようにベンチマーク問題は存在しない．そこで本研究では，巡回セールスマン問題で使用したベンチマーク問題 [2] の一部をそれぞれ改良し，それらを用いて計算を行った．データの改良方法を以下に示す．

TDTSP におけるデータの改良方法

各都市間の距離 d_{ij} を入力とする．

Step 1　1 日を m 個の時間帯に分ける．

Step 2　以下のステップ a, b を m 回反復する．

a $m = 1$ のとき，最初の 1 つ目の時間帯における各都市間の移動速度 v_{ij}^1 を $[20, 80]$ の間でランダムにそれぞれ決定する．ただし，$i = j$ のとき $v_{ij}^1 = 0$ とする．

b $m \geq 2$ のとき，以下のステップ (i), (ii) を実行する．

(i) 時間帯が 2 つ目以降の各都市間の移動速度 v_{ij}^m は，1 つ前の各都市間の移動速度 v_{ij}^{m-1} に $[-20, 20]$ の間でランダムに生成した値を加えることで決定する．ただし，$i = j$ のとき $v_{ij}^m = 0$ とする．

(ii) ステップ (i) で得られた v_{ij}^m が $v_{ij}^m \leq 0$ ならば，そのときの都市 i と j のみに対して (i) に戻る．

Step 3　Step 2 で得られた各時間帯における都市間の移動速度 v_{ij}^m で距離 d_{ij} を割ることで，各時間帯における各都市間の移動時間 t_{ij}^m を求める．

上記で作成したデータを用いて，どのメタヒューリスティックス解法を利用すれば，効率的な時間で高精度な解が得られるかを C 言語でプログラミングをすることによって比較・検討する．計算に使用した PC の環境は巡回セールスマン問題の場合と同じである．

また，本研究で使用した近似解法を以下に示す．

① 最近近傍法 (NN1, NN2)

② ランダム化最近近傍法 (NNR)

③ 2-opt 近傍法 (2O)

④ 2-opt 近傍法＋最近近傍法 (2ONN1, 2ONN2, 2ONNR)

⑤ 最近近傍法 +2-opt 近傍法 (NN12O，NN22O, NNR2O)

上記の①, ②の解法は，Malandraki and Daskin によって提案された解法である．彼らは NN1, NN2, NNR の解法の中で，NN2 を用いるのが一番良いということを示した [1]．本研究では，これらの解法を参考にして③〜⑤の解法を新たに提案した．

10　近似解法

ここでは，⑤の解法の NN22O の方法について説明する．この解法は，最近近傍法 (NN2) を行った後，2-opt 近傍法を行うという非常に単純なものである．最近近傍法 (NN2) とは，デポを出発した後，最初に訪れる都市からの都市の訪問方法は従来の最近近傍法と同じであるが，出発地点 (デポ) から最初の都市までについてはすべての通りを調べ，その中で総移動距離が一番短くなるものを解とする方法である．また，TDTSP における 2-opt 近傍法は，TSP の場合とは異なり，これまで用いてきたアルゴリズムをそのまま使用することができない．それは，2-opt 近傍法の操作により，固定されていたデポが他の都市と入れ替わってしまい，総移動時間がまったく違うものになってしまう恐れがあるからである．そこで本研究では，この問題を回避した 2-opt 近傍法を新たに作成した．このアルゴリズムを次に示す．

TDTSP における 2-opt 近傍法のアルゴリズム

$S \subseteq V$ は都市の集合 V の中で未訪問のもの，σ は構成中の巡回路，c は都市数，$i, i+1, j, j+1$ は巡回路の要素番号を表す．

Step 1　$p := 0$ とし，$S = \emptyset$ となる適当な巡回路 σ を一つ作成する．また，この巡回路 σ の総移動時間を求め，これを暫定解とする．

Step 2　$q := p$ とする．

Step 3　$q \geq p + c$ ならば，巡回路 σ を出力して終了．そうでなければ，$r := q + 2$ とする．

Step 4　$r \geq q + c - 1$ ならば，$q := q + 1$ として Step 3 に戻る．そうでなければ, $i := q \bmod c$, $i{+}1 := (q+1) \bmod c$, $j := r \bmod c$, $j := (r+1) \bmod c$ とする．

Step 5　操作後の総移動時間が暫定解よりも小さいならば，巡回路 σ の 2 本の枝 $(i, i+1)$ と $(j, j+1)$ を取り去り，$(i, i+1)$ と $(j, j+1)$ につなぎ替えた後，区間 $[\,i+1, \cdots, j\,]$ を反転させ，操作後の総移動時間を暫定解とした後，$p := (q+1) \bmod c$ として Step 2 に戻る．そうでなければ，$r := r + 1$ として Step 4 に戻る．

これは，区間 $[\,i+1, \cdots, j\,]$ にデポが存在した場合，その枝の組み合わせにおける 2-opt 近傍法をやめ，他の枝の組み合わせについて調べ，枝交換の操作前の総移動距離と操作後の総移動距離を比較して，後者のほうが小さい場合に 2 本の枝 $(i, i+1)$ と $(j, j+1)$ を取り去り，$(i, i+1)$ と $(j, j+1)$ につなぎ替えた後，区間 $[\,i+1, \cdots, j\,]$ を反転させるという方法である．

11　TDTSP における計算結果

9 節で述べた近似解法のうち，①の NN2 と⑤の NN22O の解法を解の精度について一つのグラフにまとめた (図 3)．また，このときの計算時間を図 4 に示す．これらの結果より，都市が多くなるほど NN2 よりも計算時間はかかるが，解の精度の面では，ほとんどの都市数において NN22O のほうが良いという結果が得られた．

12　おわりに

本研究では，TSP においては近似解法とメタヒューリスティックス解法を，TDTSP においては近似解法をいくつか検討した．その結果，TSP ではアニーリング法を用いるのが良く，TDTSP では NN22O を用いるのが良いということが分かった．しかし，TSP においては，本研究で取り上げた解法以外にも様々なメタヒューリスティックス解法が存在するため，それらの解法についても研究をする必要がある．また，TDTSP においては，近似解法の解の比較しか行うことができなかったため，本研究で提案した NN22O を用いたメタヒューリスティックス解法をいくつか試すと，さらに良い結果が得られると期待される．

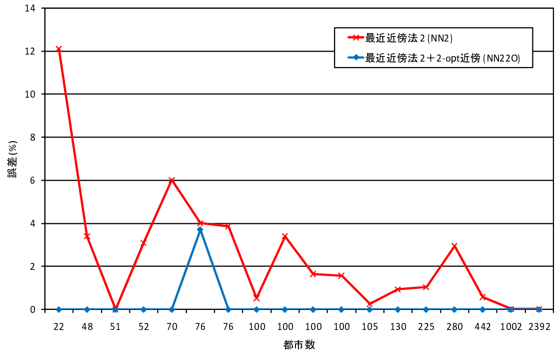

図 3　都市数と最良解との誤差 (解の精度)

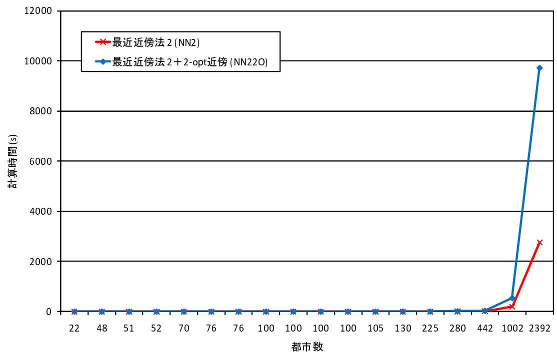

図 4　都市数と計算時間

参考文献

[1] C. Malandraki and M.S. Daskin：Time Dependent Vehicle Routing Problems: Formulations, Properties and Heuristic Algorithms, *Transportation Science*, **26** (3), pp.185-200, August 1992.

[2] TSPLIB：`http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/`

[3] 梅谷俊治：数理科学科特別講義 (OR) 組合せ最適化 ―巡回セールスマン問題を中心として―, 2005, `http://www-sys.ist.osaka-u.ac.jp/~umetani/lecture.html`

[4] 久保幹雄, 田村明久, 松井知己：応用数理計画ハンドブック, 朝倉書店, 2002.

[5] 白石洋一：組合せ最適化アルゴリズムの最新手法 ―基礎から工学応用まで―, 丸善, 2002.

[6] 中道義之, 有田隆也：ACO におけるランダム選択に基づく多様性調節の効果, 情報処理学会論文誌, **43** (9), pp.2939-2947, 2002.

[7] 柳浦睦憲, 茨木俊秀：組合せ最適化 ―メタ戦略を中心として―, 朝倉書店, 2001.

[8] 山本芳嗣, 久保幹雄：巡回セールスマン問題への招待, 朝倉書店, 1997.