マルチエージェントで解く巡回セールスマン問題
－TSP Art クイズに挑戦！－

Try to Solve Traveling Salesman Problem by Multi-Agent
- Challenge to TSP Art Quiz!-

林美紗子　大竹裕望子　向直人
Misako Hayashi　Yumiko Otake　Naoto Mukai

1. はじめに

「巡回セールスマン問題」（Traveling Salesman Problem: TSP）は古くから知られる最適化問題の 1 つである。セールスマンがある都市を出発してすべての都市を 1 回訪れるように回り、最終的に最初の都市に戻ってくる最短の巡回路を探す。一見簡単そうに思えるが、都市数が増えるに従って加速度的に解の組み合わせが増加するため、最適解を求めることは極めて難しい。他にも「ナップサック問題」など最適解を求めることが困難な問題はある。しかし、巡回セールスマン問題が別格なのは、世界中にいる一流の応用数学者が何十年調べても、単純に力まかせで調べる以上に大きく改善する方法が知られていないことにある。この問題のあらゆる例を効率良く解くことが出来る方法はいまだに見つかっていない。1962 年には、プロテクター・アンド・ギャンブル（P&G）社の広告キャンペーンとして、TSP が出題されたことがある[1]。この問題は、アメリカの 33 の都市が対象となっていて、イリノイ州シカゴを出発して、またイリノイ州シカゴに戻ってくる距離の合計が最短になるようにルートを計画するものであった。

本稿では、この TSP に焦点を当て、問題を解くためのアルゴリズムをエージェントとして実装し、巡回路が徐々に浮かび上がってくるプロセスを利用してアートに応用することを目的とする。また、アルゴリズムとして、TSP の近似解を導出可能な「最近傍法」「貪欲法」「最遠方挿入法」「最近傍挿入法」を採用する。これらのアルゴリズムは、最適化問題に良く用いられる「遺伝的アルゴリズム」などに比べ実装が容易ではあるものの、理解するにはある程度専門的な知識が必要になる。我々の研究では、アルゴリズムの処理過程を可視化し、題材となる絵を描くことで、アルゴリズムの理解をサポートすることが出来る。さらに、今話題の「ご当地キャラ」等の誰しもが 1 度は見たことのある対象を TSP の題材として利用し、クイズ形式で出題することで、老若男女が楽しめるよう工夫する。

2. アートとしての TSP

TSP をアートに応用した例としてモナリザ版 TSP がある。10 万都市を結ぶモナリザ版 TSP の元になるデータは 2009 年に、Robert Bosch 氏が作成し、「Mona Lisa TSP Challenge」として公開されている[2,3]。図 1 が、現在までに知られている最善のモナリザ巡回路であり、北陸先端科学技術大学院大学の永田裕一氏が遺伝的アルゴリズムを利用して発見した。この巡回路は最適解よりも 0.003%長いとされている。さらに永田氏は他の TSP アートにも挑戦しており、図 2 のゴッホ巡回路をはじめとする 5 作品の最良巡回路の記録を保持している。

他にも、井上氏らは、最短ハミルトン閉路上の連続する 3 点が同一直線上に並ぶとき、閉路の形状を変えずに中央の 1 点が削除可能なことに基づき、TSP アートの都市数を削減する手法を提案した[4]。この手法により、TSP アートのファイルサイズを削減することができる。また、胡氏らは線分の向きが濃淡の等高線に沿うような TSP 線画を導出する手法として、バイラテラル距離に基づく手法を提案した[5]。

我々も TSP アートに焦点を当てると同時に、問題を解くためのアルゴリズムをエージェントとして実装することで、絵が浮かび上がってくるプロセスを利用したクイズを出題する。

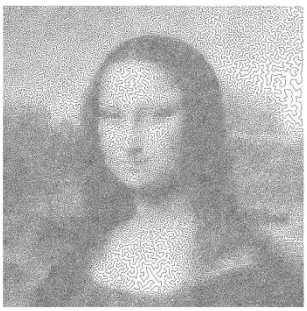

図 1　モナリザ巡回路

図 2　ゴッホ巡回路

3. 代表的なアルゴリズム

本章では TSP を解くための代表的な 4 つのアルゴリズムをエージェントとして実装する方法を紹介する。ここでは、図 3 に示す8×8のキャンバスに配置された$a = (2,7), b = (1,5), c = (6,7), d = (2,1), e = (7,1)$の 5 都市を例に考える。また、表 1 が各都市間のマンハッタン距離である。

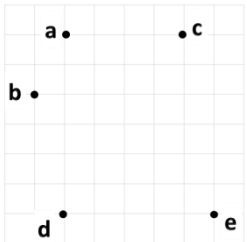

図 3　都市の配置

椙山女学園大学 文化情報学部 文化情報学科
〒464-8662
名古屋市千種区星ヶ丘元町 17 番 3 号

表 1　各都市間のマンハッタン距離

	a	b	c	d	e
a		3	4	6	11
b	3		7	5	10
c	4	7		10	7
d	6	5	10		5
e	11	10	7	5	

3.1 最近傍法エージェント

最近傍法は TSP を解くための最もシンプルなアルゴリズムである。エージェントは、ランダムに始点となる都市を定め、最も近い都市から順に繋いでいく。図 3 を例に考える。まずランダムに始点を決める。ここでは都市dが選択されたとする。ここで、図 4 に示すように、都市dから各都市への距離を比較する。都市dからは、都市bと都市eが距離 5 で最短となる。距離が等しい都市が複数ある場合はランダムに一方の都市を選択する。ここで、都市bが選択されたとすると、図 5 に示すように、都市bからその他の都市への距離を比較する。このプロセスを繰り返すことで巡回路d → b → a → c → e → dが得られる。このアルゴリズムは高速で分かりやすいが、始点の取り方によって生成される巡回路が異なることに加え、最短の巡回路が見つかることはまずない。

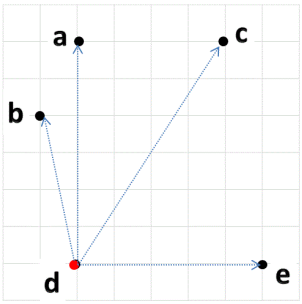

図 4　都市d

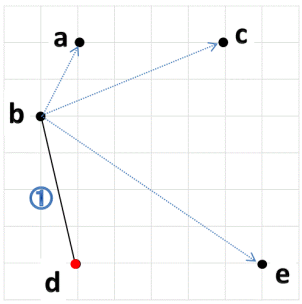

図 5　都市 b

3.2 貪欲法エージェント

貪欲法は名前の通り貪欲に都市間の距離が短い辺から繋いでいく。図 3 には、$\overline{ab}, \overline{ac}, \overline{ad}, \overline{ae}, \overline{bc}, \overline{bd}, \overline{be}, \overline{cd}, \overline{ce}, \overline{de}$の辺があり、それぞれがエージェントとして実装される。最初に、図 6 に示すように、この中で最も距離が短い辺$\overline{ab}$を繋ぐ。次に、残った 9 辺の中で最も距離が短い辺$\overline{ac}$を繋ぐ。この際、それぞれの辺を表す 2 つのエージェントは統合され、1 つのエージェントになる。ただし、辺を繋ぐ際には、生成された巡回路が閉路になるケースを避ける必要がある。例えば、図 7 では、$\overline{bc}$を繋ぐと巡回路a → b → c → aで閉じてしまう。また、都市aは既に都市bと都市cに連結されているため、次に$\overline{ad}$や$\overline{ae}$を繋ぐと都市aから 3 都市が繋がることになってしまい巡回路にならない。上記のプロセスを繰り返すと最終的に$\overline{ab}, \overline{ac}, \overline{bd}, \overline{de}, \overline{ce}$が統合されたエージェントの巡回路が得られる。

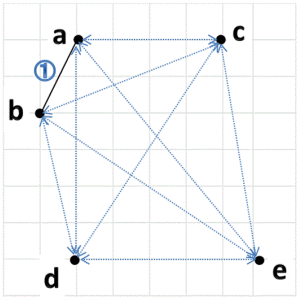

図 6　辺$\overline{ab}$

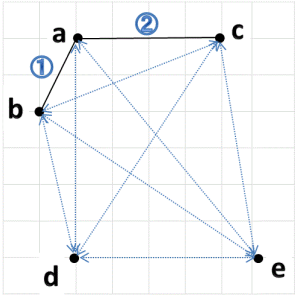

図 7　辺$\overline{ac}$

3.3 最遠方挿入法エージェント

部分的な巡回路に都市を挿入する挿入法と呼ばれるアルゴリズムでは、最遠方、最近傍、最少コスト、無作為といった様々な基準で都市を選択することができる。最遠方挿入法では、部分巡回路がエージェントとして表される。エージェントは、部分巡回路から一番遠い点を選び、経路距離の増分が最小となる位置に挿入する。最近傍法や貪欲法とは異なり、巡回路がステップ毎に更新されるという特徴がある。図 3 を例に考える。図 8 に示すように、まず最も離れた位置関係にある都市aと都市eを結び、部分巡回路a → eを生成する。次に、図 9 に示すように、部分巡回路に含まれる点aと点eから最も遠い都市を調べる。この場合、都市eから都市bまでの距離が最長の 10 となる。そこで、部分巡回路a → eに都市bを加える。都市bを追加する組み合わせは、図 10 に示すように、b → a → e, a → b → e, a → e → bの 3 通りあるが、この中で最短の部分巡回路a → b → eが選択される。この結果、図 6 で生成された部分巡回路a → eは変更され、図 11 の部分巡回路a → b → eに更新されることになる。このアルゴリズムは、早い段階での全体の形がよく、都市の挿入に応じて細部が完成していく。

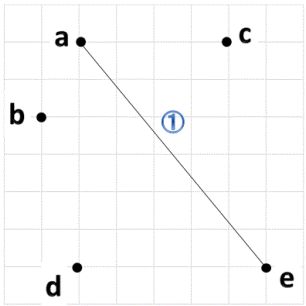

図 8　部分巡回路a → e

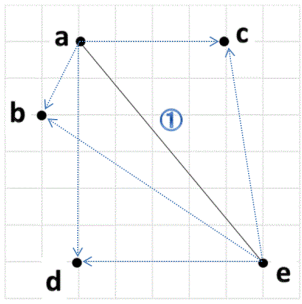

図 9　都市の選択

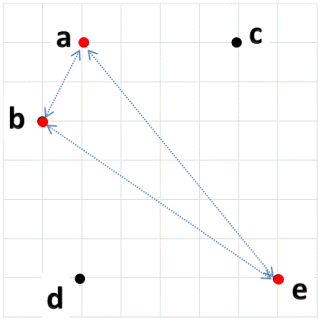

図 10　挿入位置の決定

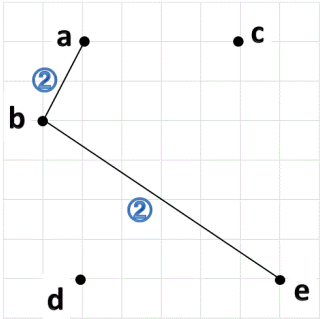

図 11　部分巡回路
a → b → e

3.4 最近傍挿入法エージェント

最近傍挿入法は、最遠方挿入法とは異なり、エージェントは、部分巡回路から一番近い点を選ぶ。図 12 に示すように、最も近い位置関係にある点aと点bを結び、部分巡回路a → bを生成する。次に、図 13 に示すように、部分巡回路に含まれる都市aと都市bから最も近い都市を調べる。この場合、都市aから都市cまでの距離が最短の 4 となる。そこで、部分巡回路a → b に都市cを加える。都市cを追加する組み合わせは、図 14 に示すように、c → a → b, a → c → b, a → b → cの 3 通りあるが、この中で最短の部分巡回路c → a → bが選択される。この結果、図 15 の部分巡回路c → a → bに更新されることになる。

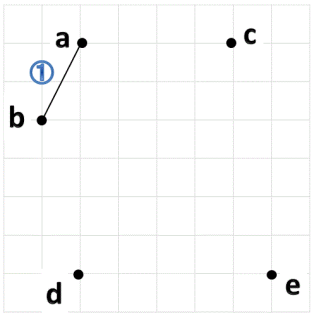

図 12　部分巡回路a → b

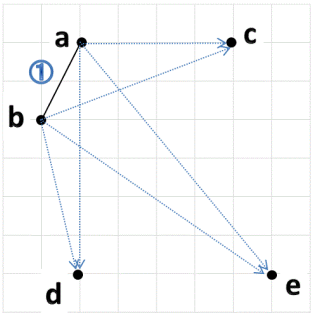

図 13　都市の選択

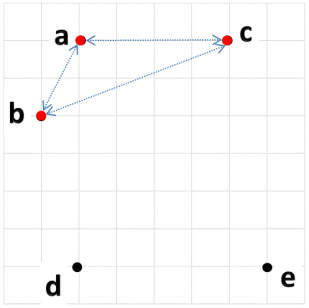

図 14　挿入位置の決定

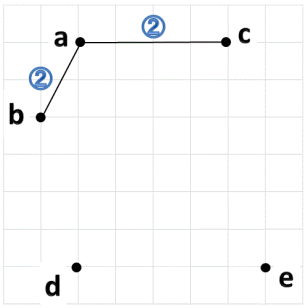

図 15　部分巡回路c → a → b

4. アルゴリズムの評価

本章では前に紹介した 4 つのアルゴリズムを Artisoc で実装し比較する。2000 × 2000のキャンバスに、50、100、150、200、250、300 の都市をランダムな位置に設置し、最近傍法、貪欲法、最遠方挿入法、最近傍挿入法で巡回路を求めたときの所要時間と距離を求めた。各都市数に対して 10 回のシミュレーションを実行し、その平均を結果とする。

図 16 が各アルゴリズムの所要時間の比較である。最近傍法の所要時間が最も短く、都市数の増加に応じて、所要時間が線形に増加していることが分かる。次に、所要時間が短いのが貪欲法である。最近傍法は都市数に比例するのに対して、貪欲法は辺の数に比例することが理由である。一方、最遠方挿入法と最近傍挿入法が最も所要時間が長く、都市数に応じて指数関数的に増加していることが分かる。これは、都市の選択に加え、部分巡回路に挿入する位置を求める計算が必要だからであると考えられる。

図 17 が各アルゴリズムで得られた巡回路の距離の比較である。貪欲法が他のアルゴリズムと比較して最も巡回路の距離が短くなった。一方、最近傍法は、所要時間は短いものの、巡回路の距離は最も長くなる結果となった。

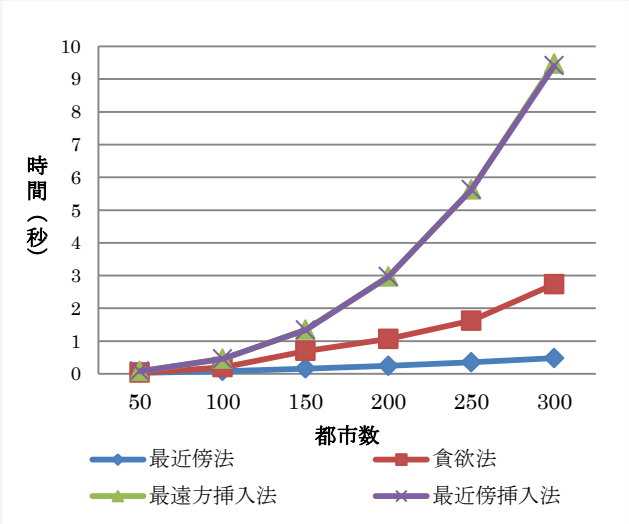

図 16　所要時間

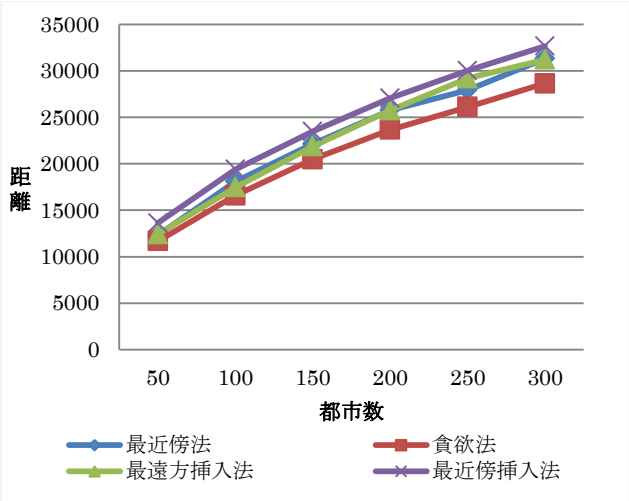

図 17　巡回路の距離

5. 問題作成

本章では、一般的な GIF や PNG などのビットマップ画像から TSP アートの問題を作成する方法について述べる。ここでは、図 18 に示す音声合成システム「VOCALOID」のキャラクターである「初音ミク」を例に挙げて、データ・フォーマットの変換プロセスについて解説する。

図 18　初音ミクの PNG 画像（http://piapro.net/）

最初に、ビットマップ画像から、“線”ではなく“点”の集合で表現する点描画に変換する。変換には、オープンソースの「StippleGen（http://www.evilmadscientist.com/）」を用いる。このソフトウェアは、重み付きボロノイ図の考えに基づき、入力画像から点描画の SVG 画像を出力することができる。また、“White Cutoff”機能を利用することで、密度の低い領域の点を削除することが可能となる。図 19 が StippleGen で生成した初音ミクの SVG 画像である。SVG 画像は XML をベースとした画像形式であり、ブラウザ上で閲覧することができる。例えば、SVG 画像における“点”は下記の circle 要素で表現される。

```
<circle cx="1625.3154" cy="297.3858" r="1.4821429"
style="fill:none;stroke:black;stroke-width:2;"/>
```

この“点”は、X 座標：1625.3154、Y 座標：287.3858、半径：1.4821429 であることが分かる（style 属性で詳細情報の記述が可能）。

図 19　StippleGen で生成した SVG 画像

TSP の問題形式は、「TSPLIB（http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/）」で標準化されている。そこで、本稿においても SVG 画像を標準化された TSP の問題形式に変換して用いる。変換された TSP データを下記に示す。

```
NAME: 初音ミク.svg
COMMENT: MAX_X:341.4876999999999 MAX_Y:530.39757
TYPE: TSP
DIMENSION: 451
EDGE_WEIGHT_TYPE: EUC_2D
NODE_COORD_SECTION
1 196.36279999999988 366.35534999999993
2 199.1452999999999 354.95361
```

“NAME:”などの行はファイルの属性を表している。また、各点は“NODE_COORD_SECTION”以降の行で表され、1 番目の点は X 座標: 196.36279999999988、Y 座標：366.35534999999993 となっている。この TSP データを Artisoc で読み込み「TSP Art クイズ」として出題する。前章で述べたように、TSP を解くためのアルゴリズムにはそれぞれ特徴があり、異なったプロセスで絵を浮かび上がらせていく。初音ミクの TSP データを利用して 300 ステップまで描画した結果が図 20、21、22、23 である。最近傍法は、最初の点から始めて、経路を延長しながら描画していくことが分かる。貪欲法は、長さの短い辺から繋いでいくため、画面全体から浮き上がるように描画される。最近傍挿入法と最遠方挿入法は、点の追加に応じて経路が更新されながら描画していく。この違いが「TSP Art クイズ」の特徴となる。

図 20　最近傍法

図 21　貪欲法

図 22　最近傍挿入法

図 23　最遠方挿入法

6. まとめ

本稿では TSP に焦点を当て、近似解を求めるためのアルゴリズムをエージェントとして実装し、巡回路が徐々に浮かび上がってくるプロセスを利用して TSP アートに挑戦した。アルゴリズムによってアートの描き方が異なるため、学習者の理解をサポートすることが出来る。また、アートの題材として、誰もが知っている対象を選択することで、老若男女問わず楽しめるように工夫した。

今後は、本稿で用いた「最近傍法」「貪欲法」「最遠方挿入法」「最近傍挿入法」といったアルゴリズムだけでなく、最適化問題によく用いられる「遺伝的アルゴリズム」を利用できるようにしたい。モナリザ TSP の最良解は遺伝的アルゴリズムで導出されている。また 3D スキャナーを用いることで平面的な問題だけでなく、立体的な問題に挑戦していきたいと考えている。さらには、老若男女問わずアルゴリズムを理解できるような教育用の教材を開発していきたい。

参考文献

[1] ウィリアム・J・クック, 松浦俊輔（訳）, “驚きの数学 巡回セールスマン問題”, 青土社, (2013).

[2] “The Traveling Salesman Problem”, http://www.math.uwaterloo.ca/tsp/index.html, (2014).

[3] Craig S. Kaplan and Robert Bosch, “TSP art”, Proceedings of Bridges, pp.303-310, (2005)

[4] 井上光平, 浦浜喜一, “タイトル”, 近似誤差最小化による TSP アートの都市数削減, Vol.2009, No.2 (2009).

[5] 胡忠英, 浦浜喜一, “バイラテラル距離に基づく非等方 TSP アート”, 電子情報通信学会論文誌. D, 情報・システム, Vol.94, No.4 (2011).