愛媛大学教育学部紀要 第70巻 145-150 2023

# 高等学校教科「情報」におけるプログラミング教育としての TSP アート

（数学教育講座） 河村 泰之

# TSP Arts as Programming Education in the High School Subject "Information"

KAWAMURA Yasuyuki

（ 2023 年 9 月 1 日受付、2023 年 11 月 28 日受理 ）

［要約］ 本稿では、プログラミング教育における題材選びの難しさと多岐にわたる教育目標に対処するための一案として、TSP アート（Traveling Salesman Problem Art）を取り上げる。TSP アートは、巡回セールスマン問題を基にしたアート作品であり、計算論的な考え方とクリエイティビティが組み合わさる点で教育に適している。巡回セールスマン問題は、与えられた頂点を一度ずつ訪れて元の地点に戻る最短経路を求める問題である。TSP アートでは、この問題を解く過程で生成される経路が絵となる。教育的な観点からは、TSP アートはアルゴリズム理論とプログラミングスキルの両方を鍛えることができる。特に、現実的な時間内で解を求めるためのアルゴリズムの工夫が必要とされる。これは、生徒にとってはアルゴリズムの効率性や計算時間の重要性を理解する良い機会となる。さらに、TSP アートは画像処理の基礎も学べるため、多角的な知識とスキルの習得が期待できる。発展として、より高度なアルゴリズムの比較や、問題設定の拡張（例：複数のセールスマン）も考えられる。

［キーワード］巡回セールスマン問題，TSP アート

## Ⅰ．背景

2020 年より、小学校におけるプログラミング教育が必修化された。この変化は高等学校においても影響を及ぼし、より高度な内容の教育が求められている。しかし、適切な教材やテーマの設定は容易ではない。各種アプリで子どもたちがイメージするのは、自分が操作できる部分や画面に表示されて見える部分であり、その他の重要性がわかりにくい。例えば、教科書で定番の”並べ替え”では、並べ替えた結果はわかりやすいが、逆に結果がわかりやすいため、並べ替えがどれだけ難しい問題なのか関心を持てない者も多い。また、並べ替えのアルゴリズムの重要性を知るには、少し深い理解が必要であり、直感的ではないため、興味を持たせるのは簡単ではない。テーマ設定を難しくする他の要因としては、プログラミング教育の応用が多岐にわたることが挙げられる。産業界からは、あるときはデータサイエンスが、またあるときは人工知能が扱えるように、と様々な要望があがる。メーカーでは、ハードウェアや OS の基礎知識を踏まえた組み込み系のプログラミングがまだまだ求められている。このように多様な要求があり、特定の分野を扱うこともできないためテーマ設定を難しくしている。

生徒の興味・関心と社会の多様な要求が交錯する中で、どのように教育目標を設定し、具体的な教材を選定するかは重要な問題である。本稿では、これらの課題に対する一つの解決策として、TSP アートを教育の場に導入する方法について考察する。教科書[4]で基本文法や配列・関数を終えた後に”並べ替え”が採用され

ていることからわかるように、入門の後は計算効率を数学的に考えることが重要となる。TSP アートは、高校生に興味を持たせつつ、計算効率を考えさせることのできるテーマとして有力である。

## Ⅱ. TSP アートの概要

まず、巡回セールスマン問題について紹介する。最適化の分野で広く研究されている問題であり、この問題を基にして作成されるアートが TSP アートである。生徒の創造的なモチベーションを利用して計算論的な要素を体験でき、プログラミング教育の教材として魅力あるテーマである。

巡回セールスマン問題は古くから知られている問題であるがオリジナルについてははっきりしない。少なくとも、1932 年には Karl Menger がこの問題を定義している[1]。ここでは、グラフを用いた定義を紹介する。

**巡回セールスマン問題（グラフ理論での定義）**

**$n$ 個の頂点集合 V と、それらのいくつかの 2 点対を結んだ辺の集合 E からなるグラフを G=(V,E) とする。辺の重みが d:E→R で与えられたとき、V の中のすべての頂点をちょうど 1 回ずつ経由する巡回路(Hamilton 閉路)で、辺の重みの総和を最小にするものを求めよ。**

日常の言葉では次のように、セールスマンでたとえられる。

**巡回セールスマン問題（平易な言葉での説明）**

**あるセールスマンは、会社を出発して $n$ か所の顧客を訪問した後、会社に戻ってこなければならない。このとき、移動距離を最小にするにはどのようなルートをたどるか。**

現実世界で考えると移動手段や評価基準が煩雑になるので、平面上の頂点として考える。辺の重みを 2 点間のユークリッド距離とすると、もっとも短い一筆がき（最短経路）を探すことになる。

この問題は巡回セールスマン問題(Traveling Salesman Problem)として広く知られており TSP と略される。TSP では、はじめに頂点集合が決まれば、その時点でもちろん最短経路が決定する。（1 通りとは限らない。）そこで、うまく頂点集合を与えることで、最短経路が絵を表すように考えられたのものが TSP アートである。有名な例としてモナリザがある。Waterloo 大学が公開しているインスタンス[2]を図１に、その顔の部分だけを拡大したものを図２に示す。図１では絵に見えるが、解像度を上げると頂点を結んだ経路だということがわかる。このインスタンスは 10 万頂点で生成されている。

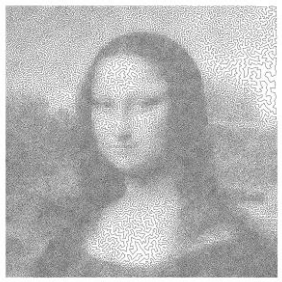


図１ Monalisa TSP art instance

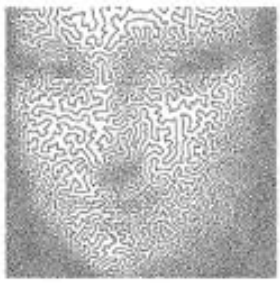


図２ 顔部分の拡大図

## Ⅲ. 手順

目標となる絵があれば、TSP アートを作ることは難しくない。しかし、頂点数が多くなるととても計算が終わらないので、基本的な手順を示しながら、注意すべき点を説明する。

手順としては次の 4 ステップを考える。

**TSP アートを作成する手順**

**Step 1: 画像を選ぶ**
**Step 2: 頂点集合を作る**
**Step 3: TSP を解く**
**Step 4: できあがる図の確認**
**もし満足のいく絵になっていなければ、**
**頂点の追加・削除して調整し Step 2 へ**

Step 1：画像を選ぶ

モナリザのような芸術を目指すと完成が遠くなる。基本的には濃淡のない２値画像（モノクロ）で作ると良い。写真よりイラストの方が作りやすいかもしれない。ここでは、愛媛大学のマスコットキャラクター、えみかをモノクロ処理したものを例にとる（図３）。この画像は 150×150 ピクセルに約６千頂点で構成される。

図３　モノクロ処理したえみか (約 6,000 点)

Step 2：頂点集合を作る

プログラミングを想定するとテキスト形式の画像ファイルだと中が見えて良い。具体的には pbm 形式などがわかりやすいが、表示するソフトウェアを準備する手間を惜しむなら png や bmp でも基本的には同じことができる。CSV で黒の頂点座標だけ保存すると効率が少し上がる。

Step 3: TSP を解く

ここが要となるステップである。アートを作るだけなら既存のツールを使うのが簡単である。ツールは日進月歩なので、その時々で最も良いものを探すのが良い。例えば、先ほどのモナリザのページに紹介されている Concorde TSP Solver[3] は古くからよく利用されている。近年は整数計画法のソルバーの性能が急速に上がっているので、最速を求めるなら検討すると良い。

またプログラムを自作するときは、厳密なものはあきらめて適度に近似するのがよい（理由は後述）。

Step 4: できあがる図の確認

TSP を解いてみるとわかるが、Step 2 で作る頂点集合をかなり修正する必要がある。図４に簡単に求めた経路(最短ではない)を示す。これを見ると、例えば、閉路なので離れたものを表すのは苦手であることに気づく。他には、思いもよらないところがつながってしまう点などが気になってくる。そこで、Step 2 に戻り、不要な点を削除したり、新たな点を加えたりすることで経路を意図的に変えようという意図が働く。

修正したいところは多く見つかるがアートはこだわると終わりがない。適度なところで妥協することも重要である。満足のいくアートを作ることは教科「情報」の目的ではないので、絵の修正に授業時間を多くの費やすことのないよう注意しなければならない。

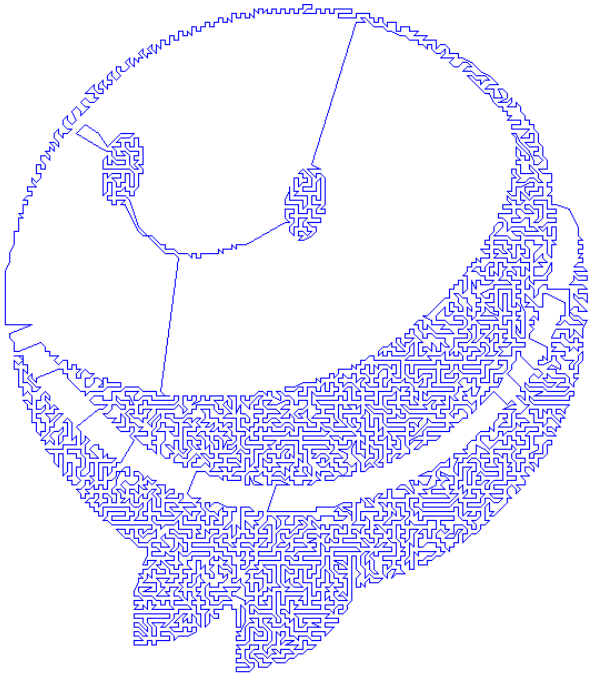


図４　近似アルゴリズムでできあがった TSP アートえみか

## Ⅳ. TSP の解法

TSP は計算論で NP 困難と呼ばれるクラスの問題であり、また、NP 完全であることが知られている。つまり、頂点数が大きくなると現実的な時間で解けるアルゴリズムは見つかりそうにない。そこで、頂点数を少なくするか、精度をあきらめることになる。経路を考えるとき、すべての経路を計算すると、頂点数が $n$ の場合、組合せは $n!$ 通りあり、この方法を試そうとすると現在のコンピュータでも計算できるのは 15 頂点くらいまでなので工夫が必要であるが、実は、単純な工夫でかなりよくなる方法が知られている。それは、2-opt と呼ばれる方法である。まずどんな方法でも良いから閉路を作り、2 つの辺を入れ替えることで距離が短くなるならば、その 2 つを入れ替えるということを繰り返す。図５で概念を説明する。

図５(a)のようにまず閉路を作る。図は曲線であるが実際は線分の列になっているので経路長の計算は容易である。(b) では、交差している部分をつなぎ直すことで全体の長さが短くなっていることがわかる。(c) 同じ操作を繰り返す。交差がわかりやすいが、交差していなくても全体が短くなるなら入れ替える。(d) このような 2 辺がなくなれば完成とする。

この方法は局所解に陥ることも多く、必ずしも最短にならないことは注意しておく。

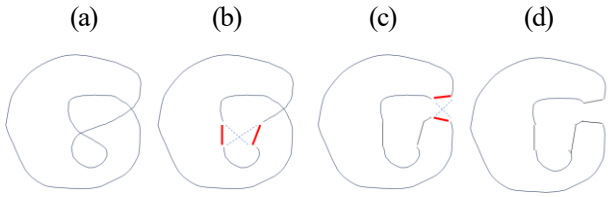



図５　2-opt で改善する様子のイメージ

## Ⅴ．生徒への課題の与え方

この方法でアートを考えるとき、本質的には TSP の効率を考えることになる。”並べ替え”という定番トピックでは実感できなかった計算効率の重要性が、自分の活動の中で現れることで、関心につながる。

実際に Step 3 で近似解を求めるとき、頂点数が多くなるととても時間がかかる。図４の近似解は、日常使いのノートパソコン（2.5GHz の Core i5-7200U、メモリ 8GB）で計算に 1 時間ほど要した。頂点を追加・削除して絵を確認するたびに同じくらい時間がかかる。Step 2～4 を繰り返す（何回も TSP を解く）とわかるが、アルゴリズムを考えてプログラムする時間より、待ち時間の合計の方が長くなるので、はやく終わるプログラムがいかに重要かわかる。待ちきれなくなるので、自然と早いアルゴリズムに興味を持つ。

そこで、他の解き方についてアイディアを求めると、例えば、次のような案が出る：ある頂点からはじめて、もっとも近い頂点を探してつなげていく。この最近傍法は簡単な発想で、実装するのも難しくない。現実でたくさんまわるとき、序盤はこのような方法をとるセールスマンやトラックドライバーは多いと言われている。（単純に近いところばかり目指すと、最後に戻ってくるのが大変なことも多いので、途中でアルゴリズムを変える。）

この方法は単純に実装すると、自分以外のすべての頂点との間の距離を計算することになり、総回数は

$$(n-1)+(n-2)+\cdots+1=n(n-1)/2$$

となる。計算時間は 2-opt より少しはやくなることも多いが、求まる経路が比較にならないくらい悪い。

他にも、多様なアイディアが出るかもしれないが、その場で思いつく方法はたいていがグリーディアルゴリズム（貪欲法）に分類される方法で、どんな局所解に陥るか見極めれば欠点が見えてくることが多い。生徒から案がでたとき、すぐに計算量が見積もることはプログラミングを教える教員にとって重要なスキルである。

高校生にとっては自分のアイディアを言語化できることも大切だし、それを実装できるようになることも重要である。また、実装するスキルがつくと、自分のアイディアの表現方法も豊かになる。

入門者に対して気をつけることは、アイディアを出しても、プログラムできないこともある点である。例えば、著者が実際に授業を行ったときに高校生から出た案として「まず、外側をぐるっと回るように戻って・・・」という表現があった。"外側"、"ぐるっと"、"回るように"、"戻ってくる"、どれもプログラムでそのまま記述するのが難しい表現だけで説明されたので記憶に残っている。本質的には問題設定を説明しているだけで、解法について説明できていない。表現力がないだけなのか、それとも解法を考えていないのかによって次に与える課題が変わる。

## Ⅵ．発展課題

TSP アートを作った後の発展内容について触れる。より高度なアルゴリズムを考えることもでき、派生問題を考えることもでき、また、画像処理や並列計算など話題を多角的に広げることもできる。

### 1．アルゴリズム論として発展

TSP はとても有名で古くから知られているので、書籍にも web サイトにも情報が豊富にある。できあがったプログラムをダウンロードすることも容易である。

定番の発展は、シミュレーションしてアルゴリズムを比較することである。つまり、方法 A と方法 B で様々な点集合に対して最短経路を計算する。時間で比較することが多いが、解の近似率も重要である。

一般化した問題を考えることも有効である。たとえば、セールスマンが $k$ 人いて、$n$ か所の顧客を訪ねる問題を考えると、$k$=1 が TSP なのでこの問題の一般化と言える。セールスマンではなくトラックドライバーだとしたとき、載せる荷物に制限があるなど現実にありそうな設定を追加すると派生問題は無数に作ることができる。派生問題を作るときは、単純なアイディアだけでなく、数学的な意味を検討することは重要である。

もともとの TSP でも厳密解は求められないのだから一般化した問題や派生問題は厳密解が見つからない前提となることが多い。

2. 画像処理としての課題

高校生にとって TSP アートはアルゴリズム的に考察を深めるだけでなく、画像の仕組みを考える練習にもなる。テキストファイルで画像を扱えるだけで画像処理のイメージが格段に変わる。白黒画像の次は、濃淡のあるグレースケール画像などを扱うこともできるが、TSP アートではカラー画像から格段に手間がかかるので濃淡画像でとどめておくのが良い。

3. 動画としての課題

2-opt で解が改善する様子を動画にするのも楽しい。静止画を連続で出力し、それをつなぎ合わせるだけで動画ができあがる。2-opt を繰り返すだけで、TSP アートができあがっていく様子を動画で作ったものを公開した[5]。この例ではランダムな閉路から始めると 56100 回の更新があり、1 秒で 600 枚の画像を表示しても約 1 分半かかる。その一部を付録に掲載する。

## Ⅶ. まとめ

最適化の分野で研究されてきた巡回セールスマン問題（TSP）から発展した TSP を高等学校教科「情報」で扱った経験をもとに、TSP アートの扱いについて整理した。

## 参考

[1] K. Menger, "Das botenproblem", in Ergebnisse eines Mathematischen Kolloquiums 2 (K. Menger, editor), Teubner, Leipzig, pp. 11-12, 1932.

[2] R. Bosch, "Mona Lisa TSP Challenge", https://www.math.uwaterloo.ca/tsp/data/ml/monalisa.html 2009.

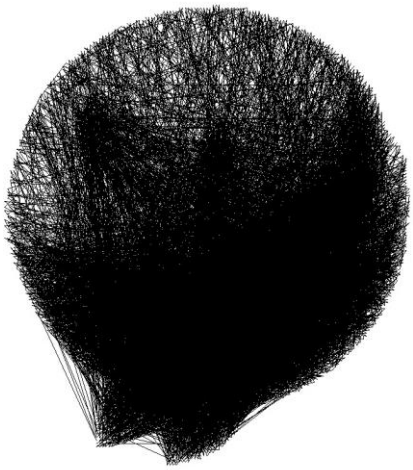
[3] W Cook , "Concorde TSP Solver", Waterloo university, https://www.math.uwaterloo.ca/tsp/concorde/ (Last update March 2015)

[4] 萩谷昌己ほか，「高校情報Ⅰ Python」，実教出版, 2022.

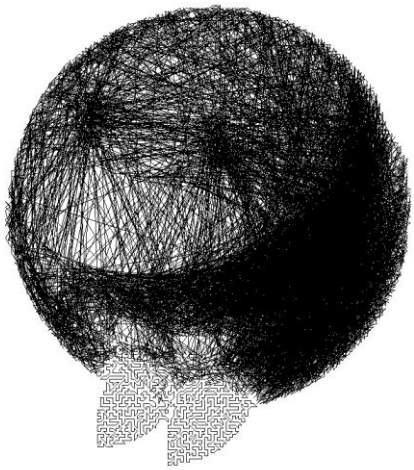
[5] https://youtu.be/3gPZZoHHG1Y (Upload September 2023)

付録

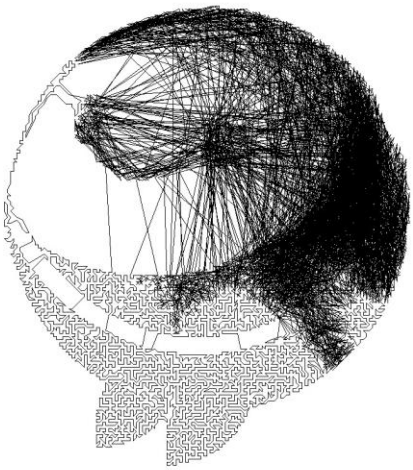
モノクロのえみかでランダムな閉路から 2-opt を繰り返す経過
すべての様子は [5] https://youtu.be/3gPZZoHHG1Y (約 1 分半の動画)



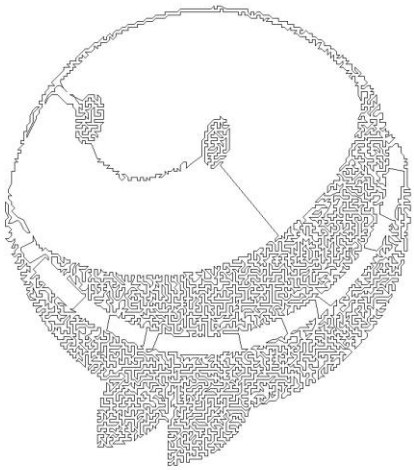
初期経路（ランダムにすべての頂点を一筆がき）



進捗度 33% (更新 18,903 回)



進捗度 66% (更新 37,807 回)



プログラム終了時 (更新 56,710 回)