

# 互いに独立した複数の巡回路を構築する アントコロニー最適化手法

岩崎 悠 長谷部 浩二

巡回セールスマン問題 (Traveling Salesman Problem: TSP) は、与えられた  $N$  頂点からなる重み付き完全グラフに対し、1 人のセールスマンがすべての頂点を 1 度ずつ通るようなバスの最小コストを求める組合せ最適化問題である。本研究ではこの TSP を拡張し、互いに辺を共有しない  $K$  本の巡回路のコストの和ならびに平均を最小化する 2 つの問題を提案する。この拡張した問題は、元の TSP と比較して計算量の観点から求解がより困難である。そこで本研究では、アントコロニー最適化 (Ant Colony Optimization: ACO) を応用した KI-Average-ACO という近似解法を提案する。具体的には、 $K$  体のアリを複数同時に移動させ貪欲にコストの平均化を行い、2-opt 法を応用してバスの構築成功確率を増加させる。以上で述べた KI-Average-ACO による近似解と二分探索を用いた二次計画法による近似解を比較した結果、bays29 というグラフで  $K = 6$  本の巡回路を構築する場合 KI-Average-ACO は二次計画法で求められたコストの約半分に留まる巡回路を求めることができた。

## 1 はじめに

巡回セールスマン問題 (Traveling Salesman Problem: TSP) は代表的な組合せ最適化問題の 1 つである [4]。この問題は、 $N$  個の都市と都市間を結ぶ重み付き辺からなる完全グラフにおいて、すべての都市を 1 つずつ訪問するような最短の巡回路を求める問題である。TSP はグラフ理論における NP 困難な組合せ最適化問題として有名であり、さまざまな多項式近似アルゴリズムやヒューリスティックアルゴリズムが提案されている。TSP は多くのアルゴリズムのベンチマークとしても有用な問題であるほか、ジョブショップスケジューリングや車両配送問題などの他の最適化

Ant colony optimization for finding multiple mutually-independent circuits.

This is an unrefereed paper. Copyrights belong to the Author(s).

Yu Iwasaki, 筑波大学大学院理工情報生命学術院 システム情報工学研究群 情報理工学位プログラム, Graduate School of Science and Technology Degree Programs in Systems and Information Engineering Master's Program in Computer Science, University of Tsukuba.

Koji Hasebe, 筑波大学システム情報系, Faculty of Engineering, Information and Systems.

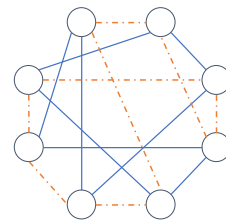


図 1  $N = 8$  の完全グラフにおける  
 $K = 2$  本の互いに独立した巡回路。

問題にも応用されている [4]。

TSP の巡回路を求めるアルゴリズムは物流や旅客輸送の計画などに幅広く応用されている。こうした従来の TSP のアルゴリズムは経路を 1 つ求めるために用いられる。しかしながら、実世界における配送計画やネットワーク経路の最適化などへの応用を考えた際には、複数の独立した巡回路を同時に求められると便利な場合がある。例えば、配送システムを構築する際にあらかじめ配送経路を複数個求めて冗長化することにより、自然災害や交通状態への耐障害性の向上が期待される。

そこで、本研究では完全グラフにおける  $K$  本の互いに独立したコストの合計が最小となるような巡回

路の組を求める問題である K-Independent Traveling Salesman Problem (KI-TSP) を提案する。互いに独立した巡回路とは、図 1 のように、 $K$  個の TSP による経路が 1 本も辺を共有することのないような巡回路の組を意味する。KI-TSP の解を求めることで、コストの合計ができるだけ小さい  $K$  個の巡回路を用意することができる。しかし、 $K$  本のパスの形状について KI-TSP は考慮していないため、形状やコストが互いに大きく異なる巡回路が構成されやすい。

このことから、本研究では KI-TSP の目的関数を変更した K-Independent Average Traveling Salesman Problem (KI-Average-TSP) を併せて提案する。この KI-Average-TSP は、 $K$  個の互いに独立した巡回路のコストの合計と分散の重み付け和を最小化する問題である。KI-Average-TSP では合計に加えて分散もコストとして指標に加えることで、 $K$  本の巡回路のコストを平均化した形状の巡回路を構築できる。

提案する KI-TSP ならびに KI-Average-TSP を解くために、本研究ではアントコロニー最適化 (Ant Colony Optimization: ACO) の基本モデルである Ant System を応用した手法を提案する [1]。ACO は、現実のアリの群れが餌を探すときに最短経路を発見する過程にヒントを得たメタヒューリスティクスによる探索アルゴリズム手法である。ACO は、巡回セールスマン問題などの多くの離散的な組合せ最適化問題に応用されている。

KI-TSP を解くための KI-ACO では、Ant System を  $K$  回だけ同じグラフに対して実行する。 $k$  回目の Ant System によって求められた 1 個のパスに含まれるグラフの辺のフェロモンを 0 とし辺のコストを無限大とする。これによって、 $k+1$  個目のパスの構築では  $k$  個目のパスまでに使用した辺を可能な限り避けることができる。なお、辺のコストを無限大にするアプローチは先行研究において提案されている [3]。

KI-Average-TSP を解く KI-Average-ACO では、1 体ずつアリの巡回路を構築させる Ant System と異なり、 $K$  体のアリの 1 辺ずつ同時に移動させる。これによって、 $K$  体のアリが平等に好ましい辺を貪欲に利用して分散を小さくすることができる。しかし、アリが移動するにつれて同じ辺を 2 回使用する可能

性が高くなり、独立した巡回路の構築失敗率が増加してしまう。そこで、構築の失敗率を抑えるために 2 つのヒューリスティックを用いる。1 つ目は余裕残余地であり、移動先の頂点候補が少ない危険な頂点に優先してアリが移動するための指標である。2 つ目は 2-best-opt であり、2-opt と呼ばれるアルゴリズムを応用した手法である。 $K$  個の巡回路のうち、重複して利用されている辺を巡回路内でうまく交換することによって、重複のない  $K$  個の巡回路に作り変える。これらのヒューリスティックによって、構築失敗率や目的関数の値である重み付け和を小さくできる。

本研究では、提案手法を評価するために 3 つの実験を行った。1 つ目の実験では、KI-Average-ACO で提案した 2 つのヒューリスティックの組み合わせによる性能の違いと特徴を検証した。2 つ目の実験では KI-TSP、3 つ目の実験では KI-Average-TSP を解くアルゴリズムの比較実験を行った。実験の結果、KI-Average-ACO における 2-best-opt はなにもヒューリスティックを用いない場合と比べて最大で 4 倍近くコストを改善することができた。一方、余裕残余地はコストの改善は小さく実行時間は長くなってしまいが、大きいグラフにおいて構築失敗率を減らすことができる有用性が示された。KI-TSP では KI-ACO が、KI-Average-TSP では KI-Average-ACO がよりコストを小さくすることができ、構築する巡回路が多い場合は数値最適化を用いた近似解よりも 2 倍近くコストの小さい巡回路を構築できた。これらのことから、KI-ACO は KI-TSP の近似解を求めるアルゴリズムとして有用であると考えられる。また、KI-Average-ACO は 2 つのヒューリスティックを用いることで構築失敗率を減らしながらコストを小さくでき、グラフや構築する本数が大きくなるほど、二分探索を用いた数値最適化と比べて小さいコストを現実的な時間で求めることができる。

本論文の構成は以下の通りである。第 2 節では、本研究の関連研究を説明する。第 3 節では、KI-TSP 並びに KI-Average-TSP の定義を説明する。第 4 節では、KI-TSP ならびに KI-Average-TSP を解く KI-ACO, KI-Average-ACO について説明する。第 5 節では、提案するアルゴリズムの実験ならびに評価を行

う。最後に、結論とこれからの課題について述べる。

## 2 関連研究

アントコロニー最適化は、現実のアリの群れが餌までの最短経路を発見する過程にヒントを得たメタヒューリスティクスによる探索アルゴリズム手法である [1]。アリは餌を発見すると巣に戻るときにフェロモンと呼ばれる成分を分泌する。このフェロモンによって、アリの群れは巣から餌までの最短経路を発見することができる。ACO はこれらの群れの性質を更新式として定式化することでグラフネットワークに関する問題の近似解を求めることができる。このうち、TSP は ACO に関するベンチマークとして利用されている。

これらのアントコロニー最適化は、トラックなどの配送車でデポ (車の発着地点) から顧客の家まで資源を届ける Vehicle Routing Problem にも適用されている [6]。また、複数のセールスマンがデポから互いに分担して都市を訪問するような部分巡回路を構築する Multiple Traveling Salesman Problem でも良い結果を示している [2]。

一方で、完全グラフにおいて互いに複数の独立した巡回経路を求める研究は少ない。Teng の研究 [7] では、グラフネットワークにおいて複数のハミルトンパスの成立条件などの補題の証明を数学的に行っている。しかし、実際に互いに独立した巡回路を構築するアルゴリズムに関する研究は、著者らの知る限り未だなされていない。

## 3 問題定義

本論文で扱うグラフは  $G = (V, E, d)$  ( $|V| = N, |E| = N(N-1)/2$ ) と表される重み付き無向完全グラフであり、辺  $(i, j)$  の重みは  $d_{ij}$  で表される。

### 3.1 KI-TSP

本論文で提案する KI-TSP は、グラフ  $G = (V, E, d)$  における  $K$  本の互いに辺を共有しない独立した巡回路の組み合わせのうち、 $K$  本の巡回路のコストの合計を最小化する問題である。その形式的な定義は以下の通りである。

**定義 1.** KI-TSP は  $G = (V, E, d)$  において以下の目的関数の解を求める問題である。

$$\begin{aligned} \min. \quad & \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} d_{ij} \cdot x_{ijk} \\ \text{s.t.} \quad & \sum_{k \in K} x_{ijk} \leq 1 & (\forall i, j) \\ & \sum_{j \in V} x_{ijk} = 1 & (\forall i, k) \\ & \sum_{j \in V} x_{jik} = 1 & (\forall i, k) \\ & u_{ik} + 1 - (N-1)(1 - x_{ijk}) \leq u_{jk} & (\forall i, j, k) \\ & x_{ijk} \in \{0, 1\} \\ & 0 \leq u_{ik} \leq N-1 \end{aligned}$$

変数  $x_{ijk}$  はアリ  $k$  による頂点  $i$  から頂点  $j$  へ向かう辺の使用の有無を表す。変数  $u_{ik}$  は巡回セールスマン問題のパスを効率よく探索するための MTZ 制約である。また、これ以降  $K$  個の巡回路のコストの合計 (すなわち、KI-TSP における目的関数の値) を  $\text{cost}_{\text{sum}}$  と表記する。

### 3.2 KI-Average-TSP

KI-Average-TSP は、グラフ  $G = (V, E, d)$  における  $K$  本の互いに辺を共有しない独立した巡回路の組み合わせのうち、 $K$  本の巡回路のコストの平均と、分散の重み付け和を最小化する問題である。その形式的な定義は以下の通りである。

**定義 2.** KI-Average-TSP は  $G = (V, E, d)$  において以下の目的関数の解を求める問題である。

$$\min. \quad \text{cost}_{\text{avg}} + \gamma \cdot \text{cost}_{\text{var}}^{\theta} \quad (1)$$

ただし、

$$\begin{aligned} \text{cost}_{\text{avg}} &= \frac{1}{K} \text{cost}_{\text{sum}}, \\ \text{cost}_{\text{var}} &= \frac{1}{K} \left( \sum_{k \in K} \left( \sum_{i \in V} \sum_{j \in V} d_{ij} \cdot x_{ijk} - \frac{1}{K} \left( \sum_{i \in V} \sum_{j \in V} \sum_{k' \in K} d_{ij} \cdot x_{ijk'} \right)^2 \right) \right). \end{aligned}$$

ここで、 $\gamma, \theta$  は平均と分散の重み付けを行うための

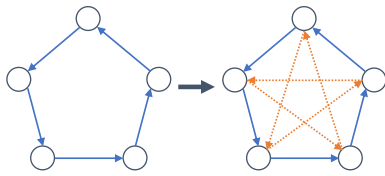


図2 KI-ACOにおける巡回路の構築方法。  
1個ずつ巡回路を構築する。

パラメータであり、制約は KI-TSP の制約と等しい。つまり、目的関数の定義のみが異なる。これ以降  $K$  本の巡回路のコストの平均と標準偏差の重み付け和 (1) を  $\text{cost}_{\text{avva}} = \text{cost}_{\text{avg}} + \gamma \cdot \text{cost}_{\text{var}}^{\theta}$  と表す。

以上の定義から、KI-Average-TSP は KI-TSP を一般化した問題となっている。KI-Average-TSP におけるパラメータ  $\gamma, \theta$  を 0 とすることで  $K$  本の巡回路のコストの平均の最小化の問題となる。KI-TSP では合計の最小化としているが、平均の最小化と合計の最小化の結果は等しくなる。

## 4 アルゴリズム

### 4.1 KI-ACO

KI-ACO は構築したい独立した巡回路の個数である  $K$  の数だけ Ant System を実行する。  $k$  回目の Ant System では、アリとフェロモンを用いて新たに 1 個の巡回路を構築する。巡回路を構築した後はその巡回路に含まれるグラフの辺のフェロモンを 0、コストを無限大とする。これによって、  $k+1$  回目以降の Ant System ではすでに使用された辺に可能な限りアリが移動しなくなり互いに独立したパスを構築できる。ただし、グラフネットワークのフェロモンはすべてのサイクルで共有のものとする。例えば、図 2 で示すように最初の Ant System で青色の巡回路が構築され、次の Ant System でオレンジ色の巡回路が構築される。

この KI-ACO の方法では、はじめは自由に使用できるコストの小さい辺が多く残っているため、構築される巡回路のコストも小さい。しかし、  $k$  のサイクルが進むにつれて使用できる辺の数が減ってコストの大きい辺のみが残り、構築される巡回路のコストも大きくなる。

## 4.2 KI-Average-ACO

### 4.2.1 概要

KI-Average-ACO は  $K$  個の巡回路のコストを平均化するために、通常の ACO と異なり  $K$  体のアリをそれぞれ 1 辺ずつ同時に移動させることでコストの平均化を図る。しかし、  $K$  体のアリを同時に移動させると構築失敗率が増加してしまうため 2 つのヒューリスティクスを併用する。1 つ目は余裕残余地であり、ある頂点から移動できる頂点数が少ない頂点に優先的にアリを移動させるための指標である。これによって、多くの移動できる将来性のある頂点を後回しにすることで構築失敗率を減らすことができる。2 つ目は 2-best-opt であり、  $K$  個の巡回路のうち失敗した巡回路を 2-opt で再構築することで、無駄のない探索を行うことを目指す。これらのアルゴリズムによって、通常の ACO を  $K$  回繰り返すよりも  $K$  個の巡回路の移動コストを平均化させながら、構築失敗率を小さくすることが可能である。KI-Average-ACO のアルゴリズムは、以下のアルゴリズム 1 で示す通りである。

### 4.2.2 アリの同時移動

KI-Average-ACO では、  $K$  体のアリを 1 辺ずつ同時に移動させることですべてのアリができる限り貪欲に好ましい辺を使用できるようにする。このとき毎回の移動後にこれまでの合計移動コストの降順にアリを並べて、その順番で次のアリの移動を行うようにする。これによって、移動コストが大きくなってしまったアリほどより小さいコストやフェロモンの多い辺を優先的に移動できるため、コストの平均化を貪欲に行うことができる。ただし、KI-ACO のアルゴリズムと同様、アリが移動に使用した辺はコストを無限大、フェロモンを逐次に 0 とし可能な限り使われないようにする。この同時移動による構築方法の例を図 3 で示す。ここでは、  $K = 2$  の場合における青とオレンジのアリが巡回路を構築する様子を表している。青とオレンジのアリは 1 辺ずつ同時に移動し、可能な限り自分または他のアリが使用した辺を再び使用しないように次の頂点を決定する。また、対応するコードはアルゴリズム 1 の 15 行目である。

---

**アルゴリズム 1: KI-Average-ACO**


---

```

1 function ki_aco( $G, N, K$ )
2   aco( $G$ )
3   ants = []
4   for  $i \leftarrow 1$  to  $K$  do
5     ants.append(Ant())
6   end
7   for  $t \leftarrow 1$  to  $KT$  do
8     make_tsps( $G, N, K, ants$ )
9     2.best_opt( $G, K, ants$ )
10    pheromone_update( $G, ants$ )
11  end
12 function make_tsps( $G, N, K, ants$ )
13  for  $i \leftarrow 1$  to  $N$  do
14    for  $j \leftarrow 1$  to  $K$  do
15      ants[ $j$ ].move_one_edge( $G$ )
16    end
17    ants.rsorrt(key=ant.costsum)
18  end
19 function 2.best_opt( $G, N, K, ants$ )
20  for  $i \leftarrow 1$  to  $K$  do
21    if ants[ $i$ ].path has duplicated edge
22    then
23      ant = ants[ $i$ ]
24      alts = []
25      for  $j \leftarrow 1$  to  $N$  do
26        for  $k \leftarrow 1$  to  $N$  do
27          at = ant.2_opt(ant.path[ $j$ ],
28            ant.path[ $k$ ])
29          if at.path has no
30            contradiction then
31              alts.append(at)
32            end
33          end
34        end
35      end
36      alts.sort(key=ant.costsum)
37      ants[ $i$ ] = alts[0]
38    end
39  end

```

---

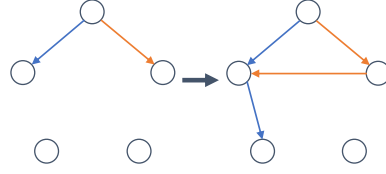


図3 KI-Average-ACOにおける巡回路の構築方法.

#### 4.2.3 移動残余度

KI-TSPの $K$ 体のアリを同時に移動させるという性質から、アリの移動にしたがい、他のアリに既使用された辺を使用せざるを得ない状況が多く発生し構築失敗率が増加してしまう。そのため、 $K$ 体のアリが巡回路を重複なく構築する成功回数を増やすために余裕残余度と呼ばれるヒューリスティクスを導入して確率移動式を定義する。

$K$ 体のアリがそれぞれ $N$ 回の移動のうち既に $t$ 回移動していたとし、アリ $h$ が既に訪問した頂点集合を $U_h$ とする。このとき、アリ $h$ が頂点 $x$ から矛盾なく移動できる頂点集合を返す関数を $R_h^t(x) \subset V$ と定義する。矛盾がないとは頂点 $y \in R_h^t(x)$ が、 $y \notin U_k$ を満たし、 $(x, y)$ の辺が他のアリ $1, 2, \dots, h-1, h+1, \dots, K$ にまだ使用されていないことである。

このとき、アリ $h$ が頂点 $i$ から次の頂点 $j$ を選ぶ移動確率 $P_{ij}^h$ を式(2)のように定義する。ただし、 $\tau_{ij}$ は辺 $(i, j)$ のフェロモン量、 $\eta_{ij}$ はヒューリスティクス値である。また、 $N_h^t(i)$ は $t$ 回目の移動においてアリ $h$ が頂点 $i$ から移動できる頂点集合である。式(2)では、次の頂点 $j$ からさらに移動できる頂点の個数 $|R_h^t(j)|$ で移動確率を割っている。この演算によって、移動できる頂点が少ない余裕のない頂点に優先的に貪欲に移動し、移動できる頂点が多い頂点を後回しにすることができる。これによって、構築失敗率を減らすことができる。時間計算量は乗算で $O(N)$ だけ追加にかかるが、その分構築失敗率が減少し、より効率的に近似解を探索することが可能である。

$$P_{ij}^h = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{u \in N_h^t(i)} [\tau_{iu}]^\alpha [\eta_{iu}]^\beta} \times \frac{1}{|R_h^t(j)|} \quad (2)$$

$$\eta_{ij} = \frac{1}{d_{ij}} \quad (3)$$

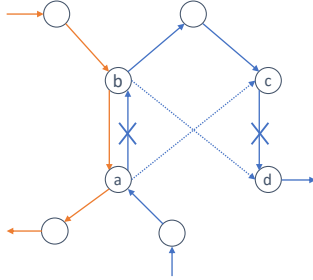


図 4 2-best-opt の例

#### 4.2.4 2-best-opt

$K$  個の巡回路の構築の成功確率をさらに増やし  $\text{cost}_{\text{avva}}$  を最小化するために巡回路の構築後に 2-best-opt というヒューリスティクスを利用してパスを改善する. 2-best-opt はアルゴリズム 1 の 19 行目である.

構築後の  $K$  個のパスにおいて, ある辺  $e = (a, b)$  が複数のアリ  $F (> 1)$  体のパスに重複して利用されるとする. このとき,  $F$  体のアリのパス  $l_1, l_2, \dots, l_F$  に対してそれぞれに 2-best-opt を行う. 2-best-opt の操作は以下の通りである. 図 4 で示す通り, あるパス  $l$  の他のパスにも重複して使用されている辺  $e = (a, b)$  と,  $l$  の他の  $N - 1$  本の辺  $e' = (c, d)$  を 2-opt の操作で交換した場合を考える. このとき, もし交換後に新しくできる 2 本の辺  $e_1 = (a, c), e_2 = (b, d)$  がまだどの  $K$  個のパスにも使用されていないならば辺の利用回数を矛盾なく下げることができるため, 辺  $e'$  を辺の交換候補  $S$  に加える. そして, これらの辺の交換候補のうち, 辺  $e$  と交換することで最もパス  $l$  の移動コストが小さくなるような 1 つの辺  $e' \in S$  を選択して  $e$  と 2-opt で交換する.

この 2-best-opt によって,  $K$  個の巡回路の構築の失敗の要因となっている重複辺を貪欲に修正でき, 構築失敗率ならびに  $\text{cost}_{\text{avva}}$  を小さくすることができる.

#### 4.2.5 フェロモン更新

2-best-opt の実行後,  $K$  個のパスが互いに独立した正しい巡回路である場合は, フェロモン更新式 (4) を実行する. フェロモン更新はアルゴリズム 1 の 10 行目である.

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \sum_{h=1}^K \Delta_{ij}^h, \quad (4)$$

$$\Delta_{ij}^h = \begin{cases} \frac{1}{\text{cost}_{\text{avva}}} & ((i, j) \in l_h) \\ 0 & (\text{otherwise}) \end{cases} \quad (5)$$

$\rho$  はフェロモンの残留率,  $C_h$  はアリ  $h$  の構築した巡回路のコスト,  $l_h$  はアリ  $h$  のパス,  $\text{cost}_{\text{avva}}$  は  $K$  個の巡回路のコストの平均と分散の重み付き和である. ただし,  $K$  個のパスが互いに独立でない場合は, グラフに対するフェロモンの更新式を実行しない. これは, 正しく構築できたときのみ更新を行うことによって探索の集中化を図るためである.

## 5 実験による評価

### 5.1 概要

本節では, 4 節で提案した KI-ACO と KI-Average-ACO のアルゴリズムを評価するために行った 3 つの実験ならびに結果について説明する.

1 つ目の実験は, KI-Average-ACO で提案した 2 つのヒューリスティックの性能検証を行う実験である. この実験によって, KI-Average-ACO におけるヒューリスティックの特徴を明らかとし, 以降の実験で使用するヒューリスティックの組み合わせを定める.

2 つ目の実験は, KI-TSP の問題を解くアルゴリズムの比較を行う実験である. 比較対象は KI-ACO, KI-Average-ACO, 数理最適化の 3 つである.

3 つ目の実験は, KI-Average-TSP の問題を解くアルゴリズムの比較を行う実験である. 比較対象は同様に KI-ACO, KI-Average-ACO, 数理最適化の 3 つである.

これら 3 つの実験における共通のパラメータは表 1 で示す通りである.  $AT$  は KI-ACO における  $k$  本目のパスを構築するときの実行する Ant System のサイクル数を表す. 一方  $KT$  は KI-Average-ACO は  $K$  本のパスを構築するために実行するサイクル数を表す.  $LT$  は, アントコロニー最適化が確率的なシミュレーションであることから, より確からしい結果を得るために行う実験の試行回数である.

表 1 共通パラメータ

parameter	value
フェロモン係数 $\alpha$	1
ヒューリスティック係数 $\beta$	3
フェロモン残留率 $\rho$	0.97
Ant-System のアリの数	$N$
Ant-System のサイクル数 $AT$	200
KI-Average-ACO のサイクル数 $KT$	1000
試行回数 $LT$	10

## 5.2 ヒューリスティックの性能比較

KI-Average-ACO のアルゴリズムに含まれる余裕残余度ならびに 2-best-opt の性能や特徴を計測するための実験を行った。本実験では、余裕残余度と 2-best-opt の利用の有無 4 通りの組み合わせで比較した。ただし、フェロモン更新は必ず行うものとする。

TSPLIB [5] と呼ばれる TSP に用いるグラフデータセットから頂点数の異なる ulysses22, bays29, att48 の 3 つのグラフを用いて組み合わせを比較する。比較の指標は重み付きコスト  $cost_{avva}$ , 実行時間, そして構築失敗率の 3 つである。ここで、1 本でも同じ辺を使用してしまった場合は構築失敗とみなす。また、 $LT = 1000$  回だけ  $K$  本の巡回路を構築しようとするため、例えば構築失敗率が 0.48 の場合は 1000 回のうち 480 回どれか 1 本以上の辺を重複して使用してしまったことを表す。実験の結果は表 2 で示す通りであり、RES は余裕残余度のみを使用する場合、2BO は 2-best-opt のみを使用する場合、RES+2BO はどちらも使用する場合、NONE はどちらも使用しない場合を表す。

表 2 から、3 つのグラフそれぞれで 2-best-opt が大きくコストを削減していることが分かる。特に att48 では  $1.40 \cdot 10^6$  から  $3.49 \cdot 10^5$  に大きくコストを改善できている。一方、余裕残余度は 2-best-opt よりもコストを小さくすることはできず、実行時間が長くなってしまふ。これは、余裕残余度は構築の失敗率を防ぐためにコストを大きくするような辺も使用し、追加計算量が  $O(N)$  だけかかるためである。特徴すべき点として、余裕残余度は 2-best-opt と合わせて使うこ

表 2 KI-Average-ACO における

ヒューリスティックの性能比較

instance	name	cost	time	fail
ulysses22	NONE	inf	12	1.00
	RES	inf	72	1.00
	2BO	$1.57 \cdot 10^5$	14	<b>0.88</b>
	RES+2BO	$1.63 \cdot 10^5$	73	0.90
bays29	NONE	$7.00 \cdot 10^4$	30	0.99
	RES	$2.63 \cdot 10^4$	173	0.96
	2BO	$1.32 \cdot 10^4$	32	0.73
	RES+2BO	$1.17 \cdot 10^4$	178	<b>0.65</b>
att48	NONE	$1.40 \cdot 10^6$	67	0.98
	RES	$6.56 \cdot 10^5$	702	0.91
	2BO	$3.49 \cdot 10^5$	70	0.58
	RES+2BO	$3.92 \cdot 10^5$	696	<b>0.51</b>

とで、bays29 のようにグラフによっては最もコストを小さくすることができるほか、bays29 や att48 のように頂点数の大きいグラフでは構築失敗率を最も小さくすることができる。よって、実行時間に余裕があり大きなグラフで多くの巡回路を構築する場合は余裕残余度と 2-best-opt を併用するのが良いと考えられる。

KI-TSP ならびに KI-Average-TSP の実験では、より実行時間が短い 2-best-opt のみを用いた KI-Average-ACO を使用することとする。

## 5.3 KI-TSP を解くアルゴリズムの比較

KI-TSP の問題を解く 3 つのアルゴリズムの性能に関する比較実験を行った。比較するアルゴリズムは KI-ACO と KI-Average-ACO と数理最適化による求解である。このうち、KI-Average-ACO では合計を最小化すればよいため  $\gamma = 0, \theta = 0$  として探索を行った。また、数理最適化は 3 節で定義した数式を、Gurobipy と呼ばれる Python のライブラリ上で実装し最適解を求めた。KI-TSP は 0-1 整数計画問題に相当する。

実験で使用するグラフは 29 頂点の bays29 というグラフであり、KI-ACO ならびに KI-Average-ACO は表 1 のパラメータを用いた。実験結果は図 5 が示

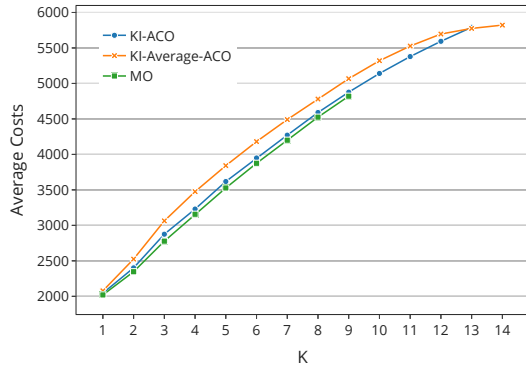


図 5 KI-TSP におけるコストの比較.

すグラフの通りである。MO は数理最適化による求解を表し、縦軸は  $K$  本の巡回路のコストの平均、横軸は構築する巡回路の本数を表す。  $K = 10$  以降において MO のグラフが表示されていないが、これは構築する巡回路の個数に比例して制約式と変数が増加するため計算量が大きくなり、今回の実験環境では厳密解を求めることができなかつたことを表す。

KI-ACO は、KI-Average-ACO と比較してより厳密解に近い巡回路を構築できた。これは 1 個の巡回路を構築するごとにグラフのフェロモンをアリの群れが変化させたことが原因であると考えられる。また、KI-Average-ACO は  $K$  体のアリによる 1000 回のみサイクルであるが、KI-ACO では  $N$  体のアリが  $AT = 200$  のサイクルを  $K$  回繰り返すため探索回数自体が大きいかも寄与していると考えられる。  $K = 14$  個の巡回路を構築できなかった KI-ACO に対して、KI-Average-ACO は  $K = 14$  個の構築に成功しており独立した多くの巡回路を構築することに適しているのではないかと考えられる。

#### 5.4 KI-Average-TSP を解く

##### アルゴリズムの比較

KI-Average-TSP の問題を解く 3 つのアルゴリズムの性能に関する比較実験を行った。比較するアルゴリズムは KI-ACO と KI-Average-ACO と数理最適化による求解である。このとき、KI-Average-TSP の重み付きパラメータは  $\gamma = 1, \theta = 1$  とし、平均と

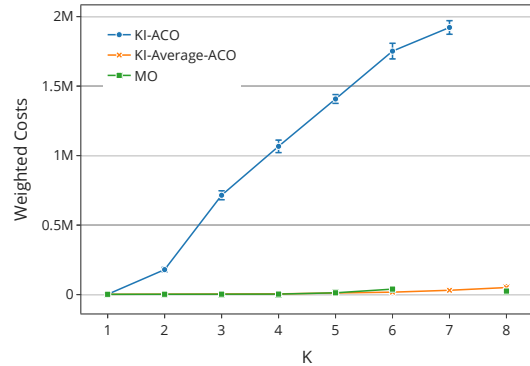


図 6 KI-Average-TSP におけるコストの比較.

分散の重み付け和を最小化することを目的とする。

KI-Average-TSP の最悪時間計算量が  $O(N!^K)$  かつ二次計画問題に相当するため、一般的に数理最適化などにより厳密解を求めることは非常に難しい。実際に著者らの実験環境で厳密解を求めた際には、  $N = 8$  などの比較的小さいグラフや  $K = 3$  などの少ない巡回路の個数の場合しか解を得ることができなかった。そのため、本実験で使用する gr17 という  $N = 17$  頂点のグラフでも同様に厳密解を求めることはできなかった。そこで本研究では、比較対象として数理最適化と二分探索を用いて近似解を求めるアルゴリズムを採用した。これは、分散が  $K$  個の巡回路から計算されることに着目し、最も大きい巡回路のコストと最も小さい巡回路のコストの差  $d$  がどこまで小さくできるかという制約式を加えて予め定めた猶予時間内で計算できるかを二分探索する。これによって、猶予時間で探索できる最も小さいコストの差  $d$  を計算でき、そのときの解は  $K$  個の巡回路のコストの分散が近似解ではあるが小さくなっている。本実験では、二分探索における 1 回の数理最適化の巡回路の構築における猶予時間を 300 秒とした。

実験結果は図 6 ならびに図 7 の示すグラフの通りである。図 6 が示すように、KI-ACO のアルゴリズムを KI-Average-TSP に用いると分散が非常に大きくなる。これは、KI-ACO が現時点で構築できる最小の巡回路を構築し、  $K$  個の巡回路のコストが互いに大きく異なるためである。図 7 で  $K = 7$  の MO の



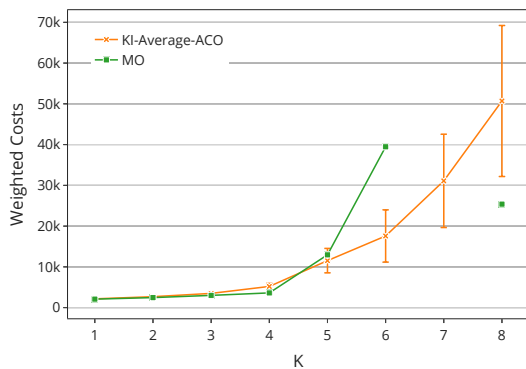


図7 KI-ACOを除いた  
KI-Average-TSPにおけるコストの比較.

グラフが途切れていることは、 $K = 7$ においてMOが猶予時間300秒で7個の巡回路の組を1通りも発見することができなかったことを表す。巡回路の構築の本数が1本から4本の間ではMOの二分探索による数理最適化がより重み付き和を小さくすることに成功している。構築する本数の多い $K = 5, 6, 7$ においてはKI-Average-ACOがよりコストを小さくできている。この結果から、KI-Average-ACOはグラフが大きくなっても許容できる現実的な時間で近似解を探索できるアルゴリズムであると考えられる。しかし、 $K$ が大きくなるにつれてKI-Average-ACOのコストの標準偏差が大きくなってしまっている。これは、KI-Average-ACOが貪欲に探索し局所解に陥っているのが原因である。

## 6 まとめと今後の課題

本研究では、巡回セールスマン問題を拡張したKI-TSPならびにKI-Average-TSPと呼ばれる2つの問題を提案した。また、これら2つの問題を解く手法としてアントコロニー最適化を応用したKI-ACOとKI-Average-ACOを提案した。KI-ACOはAnt Systemを $K$ 回繰り返すことで $K$ 本の巡回路のコストの合計の近似解を求めた。KI-Average-ACOは $K$ 体のアリを同時に移動させることによって分散を小さくし、2-best-optならびに余裕剰余度と呼ばれるヒューリスティックを用いることで構築失敗率を減らすことを目指した。

実験では、KI-Average-ACOにおける2-best-optと余裕剰余度の性能、KI-ACO、KI-Average-ACOのアルゴリズムの性能の検証を行った。ヒューリスティックの性能を調べる実験では2-best-optはコストの削減に大きく寄与し、余裕剰余度は構築失敗率を減らすことでより大きなグラフでの求解を可能とした。また、KI-ACOは合計の最小化に優れ、KI-Average-ACOはKI-Average-TSPの問題で二分探索を用いた数理最適化の解よりも構築する巡回路が大きいときにコストを小さくできた。

本研究の今後の課題は、KI-TSPならびにKI-Average-ACOをより現実的な問題に展開させることである。すべての辺を1度しか使えないという状況は運送やスケジューリングでも少なく、時間をずらすことで2度以上同じ辺を使用できる。そのため、いくらかの辺は複数の巡回路で使用してよいなど、現実の運送やネットワークのトポロジの問題に適用しやすいモデルに変更したアントコロニー最適化のアルゴリズムの開発を行いたいと考えている。

## 参考文献

- [1] Dorigo, M. and Stützle, T.: Ant colony optimization: overview and recent advances, *Handbook of metaheuristics*, Springer, 2019, pp. 311–351.
- [2] Junjie, P. and Wang, D.: An Ant Colony Optimization Algorithm for Multiple Travelling Salesman Problem., 01 2006, pp. 210–213.
- [3] Little, J. D., Murty, K. G., Sweeney, D. W., and Karel, C.: An algorithm for the traveling salesman problem, *Operations research*, Vol. 11, No. 6(1963), pp. 972–989.
- [4] Matai, R., Singh, S. P., and Mittal, M. L.: Traveling salesman problem: an overview of applications, formulations, and solution approaches, *Traveling salesman problem, theory and applications*, Vol. 1(2010).
- [5] Skorobohatyj, G.: MP-TESTDATA - The TSPLIB Symmetric Traveling Salesman Problem Instances, <http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/>.
- [6] Tan, W. F., Lee, L. S., Majid, Z. A., and Seow, H. V.: Ant Colony Optimization for Capacitated Vehicle Routing Problem, *Journal of Computer Science*, Vol. 8, No. 6(2012), pp. 846–852.
- [7] Teng, Y.-H., Tan, J. J., Ho, T.-Y., and Hsu, L.-H.: On mutually independent hamiltonian paths, *Applied Mathematics Letters*, Vol. 19, No. 4(2006), pp. 345 – 350.