

CA-002

# グループ巡回セールスマン問題と多重訪問巡回セールスマン問題 に対する発見的解法

平野 愛翔<sup>†</sup>  
静岡大学<sup>†</sup>

長谷川 和樹<sup>‡</sup>  
静岡大学<sup>‡</sup>

植田 佳佑<sup>§</sup>  
静岡大学<sup>§</sup>

安藤 和敏<sup>¶</sup>  
静岡大学<sup>¶</sup>

呉 偉<sup>||</sup>  
静岡大学<sup>||</sup>

## 1 はじめに

巡回セールスマン問題 (traveling salesman problem, TSP) は与えられた全ての都市をちょうど1度通るような最小の巡回路を探す問題であり、そのシンプルな問題構造から、基盤配線、配送計画、スケジューリングなど幅広い実問題への応用例が挙げられた。しかし実応用の場面では、他の制約を追加することがあり、TSP に対する多くの拡張問題も研究されてきた。

本研究では、全ての都市が複数のグループに分割されている場合に、各グループから1つずつ都市を訪問し、全てのグループをちょうど1度訪れる最小のグループ巡回路を求めるグループ巡回セールスマン問題 (group TSP, GTSP) と、各都市を複数回通ることを考える多重訪問巡回セールスマン問題 (many-visits TSP, MVTSP) を考える。2つの拡張 GTSP と MVTSP のそれぞれに対して発見的解法を提案し、また GTSP に対しては2つの数理モデルを設計する。計算実験を行い、数理モデルおよび、提案手法の評価を行う。

## 2 問題説明

### 2.1 巡回セールスマン問題

頂点 (都市) の集合を  $V = \{1, 2, \dots, n\}$  とした有向完全グラフ  $G = (V, E)$  と、2 頂点間  $(i, j) \in E$  の移動にかかるコスト  $c_{ij}$  が与えられるとき、巡回セールスマン問題 (traveling salesman problem, TSP) は全ての頂点をちょうど1度通る巡回路のうち、コストが最小となるものを見つける問題である。本研究で扱うコスト行列  $c$  はメトリックで

あると仮定する。すなわち、三角不等式

$$c_{ij} + c_{jk} \geq c_{ik} \quad \forall i, j, k \in V \quad (1)$$

を満たす。TSP は NP 困難であることが知られており、 $P \neq NP$  の仮定のもと、厳密に解く多項式時間アルゴリズムは存在しない。

TSP を表現する多くの数理モデルがこれまでに提案されており、その中 DFG モデル [1], MTZ モデル [2], GG モデル [3] などが有名である。本研究では、現状最も効率よく部分巡回路を排除できる DFG モデルを扱う。変数  $x_{ij}$  を辺  $(i, j) \in E$  を通るなら1、通らない場合0を取る0-1決定変数とすると、DFG モデルは、

$$\min \sum_{(i,j) \in E} c_{ij} x_{ij} \quad (2)$$

$$\text{s.t.} \quad \sum_{j \in V} x_{ij} = \sum_{j \in V} x_{ji} \quad \forall i \in V \quad (3)$$

$$\sum_{j \in V} x_{ij} = 1 \quad \forall i \in V \quad (4)$$

$$\sum_{i \in S, j \notin S} x_{ij} \geq 1 \quad \forall S \subset V, S \neq \emptyset \quad (5)$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in V, \forall j \in V \quad (6)$$

のように書くことができる。目的関数 (2) は巡回路のコストを最小化する。制約 (3) は流量保存則を示し、制約 (4) は各頂点  $i$  を訪問することを保証する。制約 (5) は部分巡回路を排除する制約である。しかし、制約 (5) の本数が  $O(2^n)$  となり、全ての制約をあらかじめ追加すると非常に効率の悪いモデルとなるため、制約 (5) を緩和した問題から解き始め、必要に応じて対応する部分巡回路排除制約を遅延制約として追加する枠組みの方が有効であることが知られている。

### 2.2 グループ巡回セールスマン問題

有向完全グラフ  $G = (V, E)$  とコスト行列  $c$  に加え、頂点集合  $V$  の分割 (グループの集合)  $\mathfrak{G} =$

Heuristic algorithms for the many-visits TSP and group TSP

<sup>†</sup>Aito HIRANO, Shizuoka University

<sup>‡</sup>Kazuki HASEGAWA, Shizuoka University

<sup>§</sup>Keisuke UEDA, Shizuoka University

<sup>¶</sup>Kazutoshi ANDO, Shizuoka University

<sup>||</sup>Wei WU, Shizuoka University

$\{V_1, V_2, \dots, V_m\}$  が新たに与えられるとする。ここで、 $V$  の分割  $\mathfrak{V}$  は以下を満たす集合族と定義する。

$$V_k \cap V_l = \emptyset \quad \forall V_k, V_l \in \mathfrak{V} \quad (7)$$

$$\bigcup_{V_k \in \mathfrak{V}} V_k = V \quad (8)$$

$$V_k \neq \emptyset \quad \forall V_k \in \mathfrak{V}. \quad (9)$$

以上の入力に対して、グループ巡回セールスマン問題 (group TSP, GTSP) は全てのグループの任意の 1 頂点をちょうど 1 度通るグループ巡回回路のうち、コストが最小となるものを探す問題である。既存研究では、条件 (7) を緩和した問題定義も見られるが、グループ間の共通頂点が存在する場合、それらの頂点に対してダミー頂点を加えることで、本研究の問題に等価変換ができる。GTSP は TSP を (各グループに点が 1 つしかない) 特殊ケースとして含むため、NP 困難である。

本研究では GTSP に対して 2 種類の数理モデル、頂点モデルとグループモデル、を提案する。表記の簡略化のため、集合族  $\mathfrak{V}$  のラベル集合を  $\Lambda = \{1, 2, \dots, m\}$  と定義し、 $E_{kl}$  を 2 つのグループのラベル  $k, l \in \Lambda$  に対して、

$$E_{kl} = \{(i, j) \mid \forall i \in V_k, \forall j \in V_l\}$$

で与えられる辺の部分集合と定義する。

頂点モデルでは、TSP のモデル (2)–(6) に使用した 0-1 決定変数  $x_{ij}$  を利用し、以下のように定式化できる。

$$\min \sum_{(i,j) \in E} c_{ij} x_{ij} \quad (10)$$

$$\text{s.t.} \sum_{i \in V_k} \sum_{j \in V \setminus V_k} x_{ji} = \sum_{i \in V_k} \sum_{j \in V \setminus V_k} x_{ij} \quad \forall k \in \Lambda \quad (11)$$

$$\sum_{i \in V_k} \sum_{j \in V \setminus V_k} x_{ji} = 1 \quad \forall k \in \Lambda \quad (12)$$

$$\sum_{k \in S, l \notin S} \sum_{(i,j) \in E_{kl}} x_{ij} \geq 1 \quad \forall S \subset \Lambda, S \neq \emptyset \quad (13)$$

$$\sum_{j \in V} x_{ji} = \sum_{j \in V} x_{ij} \quad \forall i \in V \quad (14)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in E. \quad (15)$$

目的関数 (10) は巡回回路のコストを表す。制約 (11) はグループに対する流量保存則を示し、制約 (12) にて丁度 1 点の訪問を保証する。制約 (13) は部分

巡回回路排除制約であり、TSP のモデル (2)–(6) と同様に遅延制約として必要に応じて加える。以上の制約によって全てのグループ間を連結性を保証しながら訪問することを可能にするが、始点と終点が異なる経路が構築されてしまう可能性がある。したがって、制約 (14) を加え、各頂点に対しても流量保存則を満たすことで実行可能なグループ巡回回路を構築する。

2 つ目のモデルとして、グループモデルを提案する。変数  $z_{kl}$  をグループ  $k, l \in \Lambda$  間に直後の移動関係があるとき 1、ない場合 0 を取る 0-1 決定変数、 $\alpha_i$  を頂点  $i \in V$  を訪問するとき 1、しないとき 0 を取る 0-1 決定変数とする。また  $c'_{kl}$  を 2 つのグループ  $k, l \in \Lambda$  に対して直後の移動関係を持つときはその移動にかかるコストを示し、持たない場合は 0 を取る、すなわち、

$$c'_{kl} = \begin{cases} c_{ij} & \text{if } z_{kl} = 1, \alpha_i = 1, \alpha_j = 1 \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

を満たす連続補助変数とする。以上の変数  $z, \alpha, c'$  を用いて、GTSP のグループモデルを以下のように定式化できる。

$$\min \sum_{k,l \in \Lambda} c'_{kl} \quad (17)$$

$$\text{s.t.} \sum_{i \in V_k} \alpha_i = 1 \quad \forall k \in \Lambda \quad (18)$$

$$c'_{kl} \geq c_{ij}(z_{kl} + \alpha_i + \alpha_j - 2) \quad (19)$$

$$\forall (i, j) \in E_{kl}, \forall k, l \in \Lambda$$

$$\sum_{l \in \Lambda} z_{lk} = \sum_{l \in \Lambda} z_{kl} \quad \forall k \in \Lambda \quad (20)$$

$$\sum_{l \in \Lambda} z_{lk} = 1 \quad \forall k \in \Lambda \quad (21)$$

$$\sum_{k \in S, l \notin S} z_{kl} \geq 1 \quad \forall S \subset \Lambda, S \neq \emptyset \quad (22)$$

$$z_{kl} \in \{0, 1\} \quad \forall k, l \in \Lambda \quad (23)$$

$$\alpha_i \in \{0, 1\} \quad \forall i \in V \quad (24)$$

$$c'_{k,l} \geq 0 \quad \forall k, l \in \Lambda. \quad (25)$$

目的関数 (17) は  $c'_{kl}$  を用いて巡回回路のコストの最小化を表す。制約 (18) は同じグループに属する点間の移動を制限する。また制約 (19) によって  $c'_{kl}$  が式 (16) にしたがうことを保証する。ここで、グループモデルでは、グループを一つの頂点と見ることが

できるため、TSP の制約 (3)–(5) を制約 (20)–(22) のように書き換えることができる。

2 つのモデルを比較すると、頂点モデルは制約を  $O(2^{|Q|} + |Q| + |V|)$  本含み、グループモデルは制約を  $O(2^{|Q|} + |Q|^2|V|^2)$  本含むため、頂点モデルはグループモデルと比べ、部分巡回路排除制約以外の制約の本数が少ない。しかし、グループモデルは 0-1 変数を  $O(|Q|^2 + |V|)$  個含む一方で、頂点モデルは 0-1 変数を  $O(|V|^2)$  個含むため、変数の数ではグループモデルの方が少ない。

### 2.3 多重訪問巡回セールスマン問題

自己ループを含む有向完全グラフ  $G^\circ = (V, E^\circ)$ 、コスト行列  $c$ 、および各頂点  $i \in V$  における需要  $d_i$  ( $\in \mathbb{Z}_{\geq 0}$ ) が与えられるとき、多重訪問巡回セールスマン問題 (many-visits TSP, MVTSP) は全ての頂点  $i$  をちょうど  $d_i$  ( $\geq 1$ ) 回通る連結閉路のうち、コストが最小のものを探さ問題である。ここで、頂点のもつ需要を満たすために、解閉路に (自己ループも含む) 同じ辺を複数回通過することができる。またメトリックの仮定のもと、頂点  $i$  の自己ループのコスト  $c_{ii}$  は、 $c_{ii} \leq \min_{j \in V \setminus \{i\}} \{c_{ij} + c_{ji}\}$  を満たすものとする。与えられる全ての需要  $d_i$  が 1 であるときの MVTSP は TSP となるため、MVTSP も NP 困難である。MVTSP を各頂点  $i$  を  $d_i$  個の独立した頂点とみなす場合は、TSP として解くこともできるが、変換後のグラフの頂点が  $n$  個から  $\sum_{i \in V} d_i$  個になるため、 $d_i$  の合計値が大きい場合、TSP として解くことが困難である。

MVTSP に対しては Sarin ら [4] によって、複数の数理モデルが提案されている。そのうち、代表的な一つは、 $x_{ij}$  を辺  $(i, j) \in E^\circ$  の通る回数を表す整数変数に拡張し、

$$\min \sum_{(i,j) \in E^\circ} c_{ij} x_{ij} \quad (26)$$

$$\text{s.t.} \sum_{j \in V} x_{ij} = \sum_{j \in V} x_{ji} \quad \forall i \in V \quad (27)$$

$$\sum_{j \in V} x_{ij} = d_i \quad \forall i \in V \quad (28)$$

$$\sum_{i \in S, j \notin S} x_{ij} \geq 1 \quad \forall S \subset V, S \neq \emptyset \quad (29)$$

$$x_{ij} \in \mathbb{Z}_{\geq 0} \quad \forall (i, j) \in E^\circ \quad (30)$$

のように書ける。TSP の DFJ モデル (2)–(6) と異

なる点は、訪問回数を表す制約 (28) である。また部分巡回路排除制約 (29) が存在するため、TSP の DFJ モデルと同様に遅延制約として追加するアプローチが有効である。

MVTSP から制約 (28) の代わりに、

$$\sum_{j \in V} x_{ij} \geq d_i \quad \forall i \in V. \quad (31)$$

を追加した緩和問題 MVTSP<sup>( $\geq$ )</sup> を考える。このとき、以下の定理が成立する。

**定理 1.** MVTSP と MVTSP<sup>( $\geq$ )</sup> の最適値は同値である。

*Proof.* MVTSP<sup>( $\geq$ )</sup> は MVTSP の緩和問題であるから、MVTSP<sup>( $\geq$ )</sup> の最適値は MVTSP の最適値以下になる。

一方、MVTSP の最適値は MVTSP<sup>( $\geq$ )</sup> の最適値以下であることを証明するために、解  $\mathbf{x}^{(\geq)}$  を制約 (28) を満たさない MVTSP<sup>( $\geq$ )</sup> の任意の最適解とする。すなわち、 $\mathbf{x}^{(\geq)}$  における頂点  $i \in V$  の訪問回数  $d(\mathbf{x}^{(\geq)}, i)$  を

$$d(\mathbf{x}^{(\geq)}, i) = \sum_{j \in V} x_{ij}^{(\geq)}$$

とすると、 $d(\mathbf{x}^{(\geq)}, i') > d_{i'} \geq 1$  となる頂点  $i'$  が存在する。ここで、 $\delta^+(\mathbf{x}^{(\geq)}, i')$  と  $\delta^-(\mathbf{x}^{(\geq)}, i')$  をそれぞれ以下のように定義する：

$$\delta^+(\mathbf{x}^{(\geq)}, i') = \{j \mid x_{ij'}^{(\geq)} \geq 1\}$$

$$\delta^-(\mathbf{x}^{(\geq)}, i') = \{j \mid x_{ji'}^{(\geq)} \geq 1\}.$$

解  $\mathbf{x}^{(\geq)}$  が最適解であるため、 $\mathbf{x}^{(\geq)}$  には  $i'$  における自己ループが存在せず、 $i'$  が  $\delta^+(\mathbf{x}^{(\geq)}, i')$  と  $\delta^-(\mathbf{x}^{(\geq)}, i')$  の両方に属さない。ここから、コストを増やさずに

$$\begin{cases} d(\bar{\mathbf{x}}^{(\geq)}, i') = d(\mathbf{x}^{(\geq)}, i') - 1 \\ d(\bar{\mathbf{x}}, i) = d(\mathbf{x}^{(\geq)}, i) & \text{if } i \neq i' \end{cases} \quad (32)$$

となるような実行可能解 (最適解)  $\bar{\mathbf{x}}^{(\geq)}$  を作成できることを示す。グラフ  $G(\bar{\mathbf{x}}^{(\geq)})$  を  $\bar{\mathbf{x}}^{(\geq)}$  による  $G^\circ$  の (連結) 辺誘導部分グラフとする。  $G(\bar{\mathbf{x}}^{(\geq)})$  から頂点  $i'$  を削除するときの  $G(\bar{\mathbf{x}}^{(\geq)})$  の頂点誘導部分グラフ  $G(\bar{\mathbf{x}}^{(\geq)}, i')$  の連結性により、2 つの頂点  $k$  と  $l$  を選ぶ：

- $G(\mathbf{x}^{(\geq)}, i')$  が連結であるとき,  $\delta^-(\mathbf{x}^{(\geq)}, i')$  と  $\delta^+(\mathbf{x}^{(\geq)}, i')$  からそれぞれ任意の 1 つの頂点を選択して  $k$  と  $l$  とする ( $k$  と  $l$  が同じ頂点になってもよい).
- $G(\mathbf{x}^{(\geq)}, i')$  が非連結であるとき,  $k$  と  $l$  を, それぞれ  $\delta^-(\mathbf{x}^{(\geq)}, i')$  と  $\delta^+(\mathbf{x}^{(\geq)}, i')$  から, かつ異なる連結成分から選ぶ. ( $G(\mathbf{x}^{(\geq)})$  が連結であることより, そのような  $k$  と  $l$  が必ず存在する.)

選んだ  $k$  と  $l$  の下で, 解  $\bar{\mathbf{x}}^{(\geq)}$  を

$$\begin{cases} \bar{x}_{ki'}^{(\geq)} = x_{ki'}^{(\geq)} - 1 \\ \bar{x}_{i'l}^{(\geq)} = x_{i'l}^{(\geq)} - 1 \\ \bar{x}_{kl}^{(\geq)} = x_{kl}^{(\geq)} + 1 \\ \bar{x}_{ij}^{(\geq)} = x_{ij}^{(\geq)} \text{ if } (i, j) \neq (k, i'), (i', l), (k, l) \end{cases} \quad (33)$$

のように構築する. 解  $\bar{\mathbf{x}}^{(\geq)}$  が (32) と (33) をみたすため, 流量保存則と被覆制約が満たされる. また,  $k$  と  $l$  の選び方により, 解  $\bar{\mathbf{x}}^{(\geq)}$  の連結性も保証される. すなわち, 解  $\bar{\mathbf{x}}^{(\geq)}$  が MVTSP $^{(\geq)}$  の実行可能解である. 三角不等式 (1) により解  $\bar{\mathbf{x}}^{(\geq)}$  の評価値は解  $\mathbf{x}^{(\geq)}$  の評価値以下である.

したがって各点  $i'$  に対して  $d(\bar{\mathbf{x}}^{(\geq)}, i') = d_i$  となるまで上記の解変更を繰り返すことで, コストを増やさず,  $\bar{\mathbf{x}}^{(\geq)}$  を MVTSP の実行可能解に変換できる. よって, MVTSP の最適値は MVTSP $^{(\geq)}$  の最適値以下である.  $\square$

### 3 グループ巡回セールスマン問題に対する発見的解法

GTSP の任意の実行可能解はグループの訪問順序 (グループ巡回路)  $s$  と各グループで訪問する頂点の集合  $P$  で表現することができる.

#### 3.1 グループ訪問巡回路に対する最適な訪問頂点の決定

全体的なアルゴリズムを提案する前に, グループ巡回路  $s = (s_1, s_2, \dots, s_m)$  が与えられるときに最適な訪問頂点集合  $P$  を求める多項式時間アルゴリズムを提案する.

図 1 は  $s = (s_1 = 2, s_2 = 4, s_3 = 1, s_4 = 3)$  に基づく最適な訪問頂点集合  $P$  を構築する手順を

示している.  $s$  が巡回路であるため, 先頭グループを全てのグループの中でサイズが最小となるグループとしても, 一般性を失わない (図 1a では  $V_2$  である). 先頭グループから任意の 1 つの頂点を  $i'$  とし,  $s$  のもとで  $m+1$  層の多層グラフ (multi-stage graph)  $G_s(i')$  を構築する.  $G_s(i')$  の第 1 層と第  $m+1$  層は頂点  $i'$  のみを含み, その他の層  $k$  に対しては  $V_{s_k}$  の全ての頂点から構成される (図 1b, 1c). 多層グラフでは第  $k$  層と第  $k+1$  層の全ての頂点間にのみ有向辺が存在する. ここで,  $G_s(i')$  の第 1 層から第  $m+1$  層までの任意の経路で経由した頂点の集合が 1 つの訪問頂点集合  $P$  に対応し, 経路のコストは  $s$  の順番に従うときの GTSP の評価値に対応する. したがって,  $G_s(i')$  上の最短路は,  $i'$  を含む最適な  $P$  に変換できる. 全ての  $i'$  に対応する  $G_s(i')$  を考慮することにより, グループ巡回路が  $s$  のとき, 最適な訪問頂点集合  $P$  を求めることができる (図 1d). 以上のアルゴリズムの疑似コードを Algorithm 1 に示す.

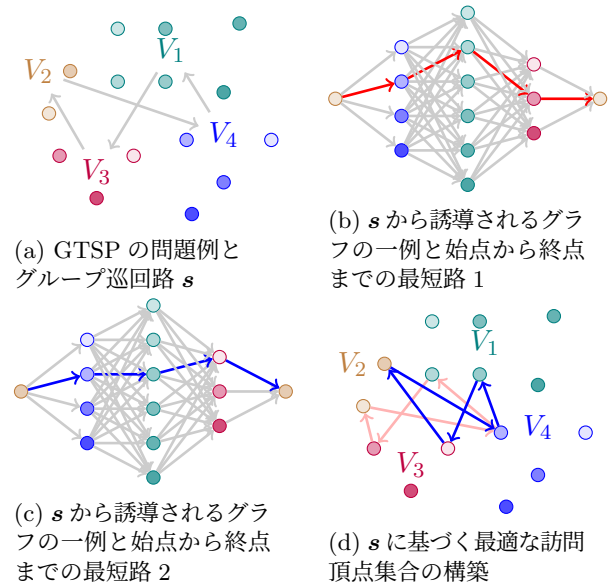


図 1:  $s$  に基づく最適な訪問頂点集合の構築

多層グラフは有向非巡回グラフの特殊ケースであるため, 1 つの  $G_s(i)$  に対する第 1 層から第  $m+1$  層までの最短経路を求めるための計算量は,  $G_s(i)$  の辺の数に依存し,  $O(n^2)$  である. 先頭グループのサイズは  $\frac{n}{m}$  以下であることから Algorithm 1 の計算量は  $O(\frac{n^3}{m})$  である.

**Algorithm 1**  $s$  に基づく最適な訪問頂点集合の決定.

```

1:  $P$  を初期化する:  $P \leftarrow \emptyset$ .
2:  $v^*$  を初期化する:  $v^* \leftarrow \infty$ .
3:  $k'$  を  $\operatorname{argmin} \{|V_k| \mid k \in \Lambda\}$  の任意の要素とする.
4: for all  $i \in V_{k'}$  do
5:    $G_s(i)$  の第 1 層から第  $m+1$  層までの最短経路問題を解き,  $P'$  とその評価値  $v_{P'}$  を得る.
6:   if  $v_{P'} < v^*$  then
7:      $P \leftarrow P'$ ,  $v^* \leftarrow v_{P'}$ .
8:   end if
9: end for
10: return  $P$ .
```

### 3.2 反復局所探索法

Algorithm 1 をサブルーチンとして, GTSP の反復局所探索法を提案する. 反復局所探索法の擬似コードを Algorithm 2 に示す. はじめに, 初期のグループ巡回路  $s$  を求めるために, 以下のように重み付き有向完全グラフ  $G^{(\Lambda)}$  を構築する.  $G^{(\Lambda)}$  の頂点集合はグループのラベル集合  $\Lambda$  とし,  $G^{(\Lambda)}$  上の  $k, l \in \Lambda$  の移動にかかるコストを, 対応するグループ  $V_k$  とグループ  $V_l$  の重心間の距離と設定する.  $G^{(\Lambda)}$  に対して TSP を解き, 得られるグループ巡回路を  $s$  とする. 次に  $s$  を初期解として, 局所探索法を行う. 近傍解の評価関数として Algorithm 1 を用いることで, TSP の近傍を利用できる. 本研究では, そのうち 2opt と  $k$ OR-opt ( $k = 1, 2$ ) を使用する. 局所探索法を終え, 局所最適解  $s'$  を得る. また GTSP 専用の近傍操作として, sequence-location (SL) 操作を提案する. SL 操作は, Algorithm 1 より  $s'$  に対する訪問頂点集合  $P$  を求め,  $P$  から誘導される  $G$  の部分グラフに対して TSP を解くことで新たな  $s'$  を構築する操作である. SL 操作を評価値の改善がなくなるまで繰り返す. 最後に  $s'$  の評価値  $v'$  と暫定値  $v^*$  を比較し, 改善できた場合, 暫定解  $s^*$  を  $s'$  で更新する. その後局所探索法の新たな初期解  $s$  を生成するため,  $s^*$  に対して double bridge 操作を適用する. 以上の一連の操作を制限時間に達するまで繰り返す.

## 4 多重訪問巡回セールスマン問題に対する発見的解法

定理 1 から, MVTSP と MVTSP<sup>( $\geq$ )</sup> は等価なため, 本研究では MVTSP<sup>( $\geq$ )</sup> に対して発見的解法を

**Algorithm 2** GTSP に対する反復局所探索法.

```

1:  $v^*$  を初期化する:  $v^* \leftarrow \infty$ .
2: グラフ  $G^{(\Lambda)}$  に対する TSP を解き, 求めた解を  $s$  とする.
3:  $s^* \leftarrow s$ .
4: repeat
5:    $s$  を初期解とする局所探索を行い, 近傍解の評価に Algorithm 1 を利用する. 得られた局所最適解とその評価値をそれぞれ  $s'$  と  $v'$  とする.
6:    $s'$  を初期解にし, SL 操作に基づく反復改善を行い,  $s'$  と  $v'$  を更新する.
7:   if  $v' < v^*$  then
8:      $v^* \leftarrow v'$ ,  $s^* \leftarrow s'$ .
9:   end if
10:   $s^*$  を double bridge 操作で更新し, 得られた解を  $s$  とする.
11: until 制限時間に到達する.
12: Algorithm 1 に  $s^*$  を与え,  $P^*$  を求める.
13: return  $s^*$ ,  $P^*$ .
```

構築し, その後定理 1 で示したアルゴリズムに則り MVTSP の解へ変換する.

MVTSP<sup>( $\geq$ )</sup> ((26)–(27), (31), (29)–(30)) の任意の実行可能解を  $x$  とする.  $x$  は流量保存則 (27) を満たすため, 解  $x$  から誘導されるグラフ  $G(x)$  は複数の単純閉路を用いて被覆することができる. この性質を用いて, 集合被覆問題 (set covering problem, SCP) の数理モデルをもとに MVTSP<sup>( $\geq$ )</sup> の SCP モデルを提案する. グラフ  $G^\circ$  に含まれる全ての単純閉路の集合を  $J_{\text{all}}$ , 単純閉路  $j \in J_{\text{all}}$  のコストを  $\tau_j$  とする. 単純閉路  $j$  に点  $i \in V$  が含まれるときと含まれないときの状況をそれぞれ  $a_{ij} = 1$  と  $a_{ij} = 0$  で表す. 以上の記号と, 変数  $y_j$  を単純閉路  $j$  の選択回数を示す整数変数とし, MVTSP<sup>( $\geq$ )</sup> の SCP モデルを以下のように表現できる.

$$\min \sum_{j \in J_{\text{all}}} \tau_j y_j \quad (34)$$

$$\text{s.t.} \sum_{j \in J_{\text{all}}} a_{ij} y_j \geq d_i \quad \forall i \in V \quad (35)$$

$$y_j \in \mathbb{Z}_{\geq 0} \quad \forall j \in J_{\text{all}}. \quad (36)$$

ここで, 目的関数 (34) は選択した単純閉路のコストの総和を表す. 制約 (35) は被覆制約を表す. 現状の SCP モデル (34)–(36) は選択した単純閉路間の連結性を保証しない. また,  $J_{\text{all}}$  のサイズは  $n$  に対する指数オーダーになるため, SCP モデルの構築, 求解は困難である. したがって,  $J_{\text{all}}$  の代わり

に  $J_{\text{all}}$  の部分集合  $J$  を用いるモデル  $\text{SCP}(J)$  を考える. 提案手法では  $\text{SCP}(J_{\text{all}})$  の最適解に選ばれやすい単純閉路のみを  $J$  に加え,  $J$  のサイズを制限する手法を提案する. また連結の保証がある解を得るための方法を提案する.

本研究では  $J$  の構築をするために,  $\text{MVTSP}^{(\geq)}$  の数理モデルから制約 (29) を除き, 線形緩和した線形計画モデル (LB モデル) を利用する. ここで, LB モデルは完全単模性をもつため, 線形緩和をしても整数最適解が存在する. 提案手法では, LB モデルの最適解  $\mathbf{x}^*$  の単純閉路分解を行う. 解  $\mathbf{x}^*$  から得られた単純閉路集合を  $Q_{\mathbf{x}^*}$  とするとき, 解  $\mathbf{x}^*$  が  $\text{MVTSP}^{(\geq)}$  の緩和問題の最適解であるため, 単純閉路  $C \in Q_{\mathbf{x}^*}$  が  $\text{MVTSP}^{(\geq)}$  の最適解に含まれる可能性が高い. したがって,  $Q_{\mathbf{x}^*}$  の全ての閉路を  $J$  に追加する. しかし, LB モデルは部分巡回路排除制約 (29) が緩和されたため, 構築された集合  $Q_{\mathbf{x}^*}$  に含まれる単純閉路を全て用いても,  $\text{MVTSP}^{(\geq)}$  の実行可能解を構築できない可能性がある. そこで, 本研究では集合  $Q_{\mathbf{x}^*}$  を連結させる連結補正閉路  $\bar{C}$  を以下のように構築し,  $\bar{C}$  も同時に  $J$  に加える. 例として, 図 2 のような 6 個の単純閉路  $C_1, C_2, \dots, C_6$  を含む  $Q_{\mathbf{x}^*}$  を考える. 閉路  $C_1$  と  $C_2$ ,  $C_3$  と  $C_4$  が連結であるため,  $Q_{\mathbf{x}^*}$  に 4 つの連結成分を持つことがわかる. ゆえに, 連結補正閉路  $\bar{C}$  を構築するには, 各連結成分をそれぞれグループとみなした GTSP を解くことで求められる. 以上の流れで, LB モデルの最適解  $\mathbf{x}^*$  に対

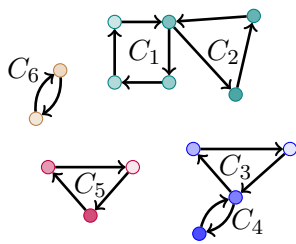


図 2: 集合  $Q_{\mathbf{x}^*}$  に含まれる単純閉路

して構築される  $Q_{\mathbf{x}^*}$  と  $\bar{C}$  を  $J$  に加える. ここまでを 1 反復とし, 2 反復目から次の制約を追加した LB モデル ( $\text{LB}(X)$ ) を考える.

$$\sum_{(i,j) \in C} x_{ij} \leq L(C, \mathbf{x}^*) \quad \forall \mathbf{x}^* \in X, \forall C \in Q_{\mathbf{x}^*}. \quad (37)$$

ここで,  $X$  は今までに得られた解からなる集合であり,  $V(C)$  を単純閉路  $C$  に含まれる点の集合と

し,  $L(C, \mathbf{x}^*)$  は以下のように定義する.

$$L(C, \mathbf{x}^*) = \begin{cases} 0 & C \text{ が自己ループ} \\ \sum_{(i,j) \in C} x_{ij}^* - |V(C)| & \text{その他の場合.} \end{cases}$$

制約 (37) を課すことで, 毎回の反復で異なる単純閉路の追加が保証できる. 提案手法では,  $J$  のサイズが十分な数 (パラメータ  $\alpha$ ) になるまで反復を続ける.  $J$  を構築するためのアルゴリズムを Algorithm 3 に示す.

### Algorithm 3 集合 $J$ の構築手法.

- 1: 単純閉路集合  $J$  を空集合で初期化する:  $J \leftarrow \emptyset$ .
- 2: 解集合  $X$  を空集合で初期化する:  $X \leftarrow \emptyset$ .
- 3: **while**  $|J| \leq \alpha$  **do**
- 4:   緩和問題  $\text{LB}(X)$  を解き, 解  $\mathbf{x}^*$  を得る.
- 5:   解  $\mathbf{x}^*$  をもとに, 単純閉路集合  $Q_{\mathbf{x}^*}$  を構築する.
- 6:   単純閉路集合  $Q_{\mathbf{x}^*}$  の連結補正閉路  $\bar{C}$  を構築する.
- 7:   集合  $J$  を更新する:  $J \leftarrow J \cup Q_{\mathbf{x}^*} \cup \{\bar{C}\}$ .
- 8:   集合  $X$  を更新する:  $X \leftarrow X \cup \{\mathbf{x}^*\}$ .
- 9: **end while**

$J$  の構築後,  $\text{SCP}(J)$  を解いて連結の保証がある解を得るための手法を提案する.  $\text{SCP}(J)$  の実行可能領域の部分集合  $Y$  を用いて,  $\text{SCP}(J)$  に以下の制約を加えたモデルを  $\text{SCP}(J, Y)$  とする:

$$\sum_{j \in \Phi(\mathbf{y}^*)} y_j \geq 1 \quad \forall \mathbf{y}^* \in Y. \quad (38)$$

ここで,  $\Phi(\mathbf{y}^*)$  は  $\mathbf{y}^*$  に含まれた連結成分のうち 2 つ以上の連結成分間を連結させることができる閉路からなる  $J$  の部分集合である. 制約 (38) を追加することで, 連結保証のない  $\mathbf{y}^*$  を排除することができる. 提案手法では,  $\text{SCP}(J, Y)$  の最適解  $\mathbf{y}^*$  を求め,  $\mathbf{y}^*$  に対応する閉路群が連結でないとき,  $\mathbf{y}^*$  を  $Y$  に追加し, 再び  $\text{SCP}(J, Y)$  を解く. このような反復を連結保証のある解  $\mathbf{y}^*$  を得るまで続ける.

## 5 計算実験

GTSP と MVTSP に対して数理モデルと発見的解法との比較実験を行った. 計算環境は CPU 4.00 GHz Xeon E-2286G, メモリ 64 GB である. プログラミング言語は Python 3.11.9 であり, 数理モデルの計算と, MVTSP の発見的解法に関わる線形モデルと整数モデルは数理最適化ソルバーの Gurobi 9.5.1 を使用した. TSP を解く際には LHK

法 [5] を用いた。各手法または各モデルにおいて、1つの問題例に対する制限時間は 600 秒とした。

GTSP の問題例は TSPLIB にある 4 問 (berlin52, kroA100, u159, kroA200) をベースに頂点の座標を設定し、 $k$ -means 法を用いて  $m$  グループの分割を行った。各  $n$  と  $m$  の組み合わせに対し 5 問生成し、その平均値で各手法を評価する。表 1 は GTSP の各手法で得られる解の %gap の平均値を示す。ここで、解の質を表す「%gap」は評価値を  $UB$ 、頂点モデルの下界値を  $LB_1$ 、グループモデルの下界値を  $LB_2$  とし、

$$\%gap = \frac{UB - \max\{LB_1, LB_2\}}{\max\{LB_1, LB_2\}} \times 100$$

で定義する。また「inf」は 5 問中に実行可能解が得られない問題例が存在することを表す。 $n$  が小さいときは、3つの手法とも同程度の質の解を求められた。しかし、 $n$  が大きくなるに連れて、数理モデルで得られる解よりも、発見的解法で得られる解の質が高くなる傾向がある。

表 1: GTSP に対する解の精度 (%gap)

問題例		頂点	グループ	発見的解法
$n$	$m$	モデル	モデル	
52	5	0.0	0.0	0.0
52	10	0.0	0.0	0.1
52	26	0.0	0.0	0.0
100	10	2.5	2.5	2.9
100	20	2.4	2.7	2.4
100	50	6.6	inf	6.5
159	15	23.4	23.4	23.3
159	31	32.4	32.9	30.9
159	79	inf	inf	25.1
200	20	50.5	48.7	48.3
200	40	74.5	59.6	38.6
200	100	inf	inf	25.0

MVTSP の問題例は、GTSP と同じく TSPLIB の 4 問を用いた。各頂点  $i \in V$  の需要  $d_i$  は一様乱数  $U(1, 5)$ ,  $U(6, 10)$ ,  $U(15, 20)$  に従い設定し、それぞれタイプ I-III と呼ぶ。また同様に各設定に対して乱数問題例を 5 問生成した。発見的解法の中で GTSP を解く際は、反復局所探索法 (Algorithm 2) を 10 秒間用いた。またパラメータ  $\alpha = 100$  と設定した。表 2 は MVTSP の各手法で得られる解の %gap の平均値を示す。数理モデル (26)–(30) は  $n = 52$  の問題例でのみ最適解を求めたが、その他の  $n$  では実行可能解すら求めることができなかった。一方で発見的解法は全ての問題例に対して、実

行可能解を得ることができた。しかし、小規模の問題例に対しても最適解は得られなかった。

表 2: MVTSP に対する解の精度 (%gap)

問題例		数理モデル	発見的解法
$n$	タイプ		
52	I	0.0	3.7
52	II	0.0	1.0
52	III	0.0	0.4
100	I	inf	9.7
100	II	inf	4.3
100	III	inf	1.8
159	I	inf	8.1
159	II	inf	3.5
159	III	inf	1.5
200	I	inf	10.6
200	II	inf	4.2
200	III	inf	1.8

## 6 今後の課題

これまで自己ループを含む有向完全グラフ  $G^\circ$  に対する MVTSP について議論した。以降は自己ループを含まない有向完全グラフ  $G$  に対する MVTSP を考える。このとき、以下の定理が成立する。

**定理 2.** 自己ループを含まない有向完全グラフ  $G$  に対する MVTSP に実行可能解が存在する必要十分条件は、

$$\sum_{j \in V \setminus \{i\}} d_j \geq d_i \quad \forall i \in V \quad (39)$$

である。

*Proof.* (十分性の証明)  $G$  に対する MVTSP の任意の実行可能解  $\mathbf{x}$  において式 (39) を満たすことを示せばよい。ここで実行可能解  $\mathbf{x}$  は制約 (27) と制約 (28) を満たすため、

$$d_j = \sum_{k \in V \setminus \{j\}} x_{kj} = \sum_{k \in V \setminus \{j\}} x_{jk} \quad \forall j \in V \quad (40)$$

が成立する。したがって、点  $i$  を除く全ての点  $j \in V$  に対する式 (40) の和を取ると、

$$\begin{aligned} \sum_{j \in V \setminus \{i\}} d_j &= \sum_{j \in V \setminus \{i\}} \sum_{k \in V \setminus \{j\}} x_{jk} \\ &= \sum_{j \in V \setminus \{i\}} \sum_{k \in V \setminus \{j\}} x_{kj} \geq \sum_{j \in V \setminus \{i\}} x_{ij} = d_i \end{aligned}$$

が成立する. ゆえに式 (39) を満たす.

(必要性の証明) 必要性を証明するために, MVTSP の実行可能解を構築できる min-max decrement 法 (Algorithm 4) を提案する. Step 12 で得られる  $R$  は連結性の保証がある実行可能解である. ここで, Algorithm 4 を用いて実行可能解

---

**Algorithm 4** Min-max decrement 法.
 

---

- 1: 全ての頂点  $i \in V$  において,  $\bar{d}_i \leftarrow d_i$  として初期化する.
  - 2:  $R$  を空のリストとして初期化する.
  - 3: **repeat**
  - 4:  $j \leftarrow \operatorname{argmin}_{i \in V} \{\bar{d}_i \mid \bar{d}_i \geq 1\}$ .
  - 5:  $\bar{d}_j \leftarrow \bar{d}_j - 1$ .
  - 6:  $R$  の最後尾に頂点  $j$  を追加する.
  - 7:  $j \leftarrow \operatorname{argmax}_{i \in V} \{\bar{d}_i\}$ .
  - 8:  $\bar{d}_j \leftarrow \bar{d}_j - 1$ .
  - 9:  $R$  の最後尾に頂点  $j$  を追加する.
  - 10: **until**  $|\{i \in V \mid \bar{d}_i\}| \geq 1$ .
  - 11: **if**  $|\{i \in V \mid \bar{d}_i\}| = 0$  **then**
  - 12: 実行可能解  $R$  を返す.
  - 13: **else**
  - 14: 実行可能解を生成せずに終了する.
  - 15: **end if**
- 

を構築できず, 終了した場合 (Step 14) を考える. このときの  $\bar{d}$  には,  $\bar{d}_{i'}$  が正の値である頂点  $i'$  がただ 1 つ存在し, その他の頂点  $j \in V \setminus \{i'\}$  では  $\bar{d}_j = 0$  である. したがって, 終了状態において

$$\bar{d}_{i'} > \sum_{j \in V \setminus \{i'\}} \bar{d}_j \quad (41)$$

が成立する. ここで, 一つ前の反復を考えると, 式 (41) の両辺がそれぞれ 1 ずつ加えられるため, 式 (41) は一つ前の反復でも成立する. したがって, Algorithm 4 を用いて実行可能解を構築できないならば, 以下の式を満たす.

$$d_{i'} > \sum_{j \in V \setminus \{i'\}} d_j. \quad (42)$$

上記の対偶を考えると, 式 (39) を満たすならば Algorithm 4 によって MVTSP の実行可能解を構築できる.  $\square$

4 節で議論した自己ループを含む有向完全グラフ  $G^\circ$  に対する MVTSP の発見的解法は, 定理 1 のもと, MVTSP<sup>( $\geq$ )</sup> の解を変換して MVTSP の実行可能解を得た. しかし, 自己ループを含まない

有向完全グラフ  $G$  に対する MVTSP は, 定理 1 が成立しないケースが存在する. したがって  $G$  に対する MVTSP には, 4 節で提案した発見的解法と異なるものを考える必要がある.

## 7 おわりに

本研究では, 巡回セールスマン問題 (TSP) の拡張問題である多重訪問巡回セールスマン問題 (MVTSP) とグループ巡回セールスマン問題 (GTSP) を扱い, MVTSP に対しては発見的解法を提案し, GTSP に対しては発見的解法と 2 種類の数理モデルを構築した. 計算実験では, GTSP と MVTSP のそれぞれに対して, 提案手法の有効性を確認できた.

## 参考文献

- [1] George Dantzig, Ray Fulkerson, and Selmer Johnson. Solution of a large-scale traveling-salesman problem. *Journal of the Operations Research Society of America*, Vol. 2, No. 4, pp. 393–410, 1954.
- [2] Clair E Miller, Albert W Tucker, and Richard A Zemlin. Integer programming formulation of traveling salesman problems. *Journal of the ACM*, Vol. 7, No. 4, pp. 326–329, 1960.
- [3] Bezalel Gavish and Stephen C Graves. The travelling salesman problem and related problems. Working paper OR 078-78, Massachusetts Institute of Technology, Operations Research Center.
- [4] Subhash C Sarin, Hanif D Sherali, and Liming Yao. New formulation for the high multiplicity asymmetric traveling salesman problem with application to the chesapeake problem. *Optimization Letters*, Vol. 5, pp. 259–272, 2011.
- [5] Keld Helsgaun. General  $k$ -opt submoves for the Lin–Kernighan TSP heuristic. *Mathematical Programming Computation*, Vol. 1, pp. 119–163, 2009.