令和 5 年度　学士論文

時間制約付き carrier vehicle routing 問題に対する二次錐計画問題を用いた解法

東京工業大学　情報理工学院　数理・計算科学系
学籍番号　19B31048
脇田康平
指導教員 山下 真 教授

2023 年 2 月 27 日

目次

1　はじめに

本論文の構成は以下である。第 2 節では、二次錐計画問題 (SOCP) に関する前提知識として二次錐計画問題と内点法アルゴリズム、中心パスなどについて説明をする。第**??**節では、緩和法とペナルティ法に基づく変形について説明する。第 5 節では、数値実験を行った結果を示し、各手法の評価を行う。第 6 節で、まとめを行う。

1.1　記法

この論文で扱う記法を述べる。

- $\|\cdot\|$ はユークリッドノルムとする。
- 行列 $\boldsymbol{A} \in \mathbb{R}^{k\times m}, \boldsymbol{B} \in \mathbb{R}^{k\times n}$ を横方向につなげてできる新たな行列 $\boldsymbol{C} \in \mathbb{R}^{k\times(m+n)}$ を $\boldsymbol{C} = (\boldsymbol{A}, \boldsymbol{B})$ と表記する。
- 行列 $\boldsymbol{A} \in \mathbb{R}^{m\times k}, \boldsymbol{B} \in \mathbb{R}^{n\times k}$ を縦方向につなげてできる新たな行列 $\boldsymbol{C} \in \mathbb{R}^{(m+n)\times k}$ を $\boldsymbol{C} = (\boldsymbol{A}; \boldsymbol{B})$ と表記する。
- 上付き添え字 T は行列やベクトルの転置を示す。
- $\boldsymbol{x}^T\boldsymbol{y}$ はベクトル $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ の内積 $\sum_{i=1}^n x_i y_i$ である。
- 2 つのベクトル $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ に対して、$\boldsymbol{x} \circ \boldsymbol{y}$ を次のように定義する。

$$\boldsymbol{x} \circ \boldsymbol{y} = \begin{pmatrix} \boldsymbol{x}^T\boldsymbol{y} \\ x_0 y_1 + y_0 x_1 \\ \vdots \\ x_0 y_n + y_0 x_n \end{pmatrix}$$

- 2 つの行列 $\boldsymbol{A}, \boldsymbol{B}$ に対して $\boldsymbol{A} \oplus \boldsymbol{B} = \begin{pmatrix} \boldsymbol{A} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{B} \end{pmatrix}$
- n 次元のベクトル $\boldsymbol{e} \in \mathbb{R}^n$ かつ、1 番目の要素のみが 1 で、その他の要素は全て 0 であるベクトルを $\boldsymbol{e}_n = (1; \boldsymbol{0})$ と表す。
- 対角成分以外は 0 で、対角成分は 1 行目が 1, それ以外が -1 である行列を $\boldsymbol{R}_n \in \mathbb{R}^{n\times n}$ とする。$\boldsymbol{R}_n = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -1 \end{pmatrix}$

2　先行研究

2.1　An integrated integer programming model with a simulated annealing heuristic for the carrier vehicle traveling salesman problem

時間制約なし焼きなまし解法

2.2　Exact Solutions for the Carrier Vehicle Traveling Salesman Problem

Gambella らの手法を概説する。本研究では時間制約がない場合の CVTSP を対象としている。本論文で着目すべきは CVTSP を MISOCP として定式化をしている点であり、これは時間制約付きに容易に拡張できる点で非常に理想的といえる。

3　問題設定

3.1　対象とする問題について

本研究が対象とする CVTSPWT(Carrier Vehicle Travering Salesman Problem with Time Window) についての説明を与える。
この問題では、速度 V_c である carrier と 速度 V_v である vehicle の二つの輸送機器を用いる。ただし, $V_c < V_v$
CVTSPWT とは以下の条件を満たす carrier の経路のうち、スタート地点 (q_s) からゴール地点 (q_f) まで最も少ない時間で航行できる経路とその時間を求める問題である。

1. どの目的地にも少なくとも一回訪れる。その際、訪れるのは vehicle でも carrier でも良い

2. ある目的地 i には (u_{i1}, u_{i2}) の間の時間に訪れる。

3. vehicle には一度に航行できる距離 a があり、これを超えてはいけない。航行可能距離は、carrier と同じ座標に来たら再び a に戻る。

4. vehicle は carrier から離陸して着陸するまでの間に一箇所しか訪れられない。

本問題に対してドローンの航続可能距離を 0 とすると TSP と同じ設定となることから、TSP より少なくとも難しい問題である。

3.2 混合整数二次錐計画問題

二次錐 (K) これを用いて二次錐計画問題 (SOCP) の一般的な形は (数式) と表せる。
SOCP をノルムを用いて変形すると以下の数式のように表せる。

$$\begin{aligned} \min \quad &: \quad \boldsymbol{f}^T \boldsymbol{x} \\ \text{subject to} \quad &: \quad ||\boldsymbol{A}_i x + \boldsymbol{b}_i||_2 \leq \boldsymbol{c}_i^T \boldsymbol{x} + \boldsymbol{d}_i, i = 1, \ldots, m \\ &\quad \boldsymbol{F}\boldsymbol{x} = \boldsymbol{g} \end{aligned}$$

二次錐計画問題はノルムを扱えるため汎用性が高いだけでなく、ソルバーを用いて高速で求解をすることができるという特徴がある。
次に混合整数二次錐計画問題 (MISOCP) の定義について述べる。

$$\begin{aligned} \min \quad &: \quad \boldsymbol{f}^T \boldsymbol{x} \\ \text{subject to} \quad &: \quad ||\boldsymbol{A}_i x + \boldsymbol{b}_i||_2 \leq \boldsymbol{c}_i^T \boldsymbol{x} + \boldsymbol{d}_i, i = 1, \ldots, m \\ &\quad \boldsymbol{F}\boldsymbol{x} = \boldsymbol{g} \\ &\quad \boldsymbol{x}_i \in \mathbb{Z}, (i = 1, \ldots, n) \end{aligned}$$

ここで、$f \in \mathbb{R}^n, \boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d}$ が定数で $x \in \mathbb{R}^n$ が目的変数である。混合整数二次錐計画問題は決定変数が離散であるため組合せ最適化のアルゴリズムを用いて最適値を求める必要がある。

3.3 MISOCP として定式化

ここでは、対象問題を解くためのモデルを MISOCP を用いて定式化する。これは、Gambella らの先行研究の拡張である。
まずは定数を定義する。

n	number of target points
q_i	set of target points coodinates
q_{min}	vector of the minimum of the q_i
q_{max}	vector of the maximum of the q_i
V_v	vehicle speed
V_c	carrier speed
a	vehicle operational range
p_o	coordinates of the starting point of the trajectory
p_f	coordinates of the ending point of the trajectory

次に決定変数を定義する。

Q_i	coodinates in the i th target point to be visited

w_{ij} binary variables taking 1 if target point j is visited in position j

$p_{to,i}$ coordinates of the takeoff point for the visit of Q_i

$p_{l,i}$ coorinates of the landing point after the visit of Q_i

$t_{i,1}$ time taken by the vehicle from $p_{to,i}$ to reach Q_i

$t_{i,2}$ time taken by the vehicle from$Q_i$to reach $p_{l,i}$

t_i time taken by the carrier from $p_{to,i}$ to reach $p_{l,i}$

T_1 time take by the carrier from $p_o$to reach $p_{to,1}$

T_i time take by the carrier from $p_{l,i-1}$to reach $p_{to,i}$

T_{n+1} time take by the carrier from $p_{l,n}$to reach p_f

MISOCP モデルを定義する。

$$\begin{array}{rcl} \min & : & \sum_{i=1}^{n} t_i + \sum_{i=1}^{n+1} T_i \\ \text{subject to} & : & \|\boldsymbol{A}_i x + \boldsymbol{b}_i\|_2 \leq \boldsymbol{c}_i^T \boldsymbol{x} + \boldsymbol{d}_i, i = 1, \ldots, m \\ & & \boldsymbol{F}\boldsymbol{x} = \boldsymbol{g} \\ & & \boldsymbol{x}_i \in \mathbb{Z}, (i = 1, \ldots, n) \end{array}$$

4 提案手法

4.1 焼きなまし法の説明

焼きなまし法 (Simulated Annealing) とは、大域的最適解を求めるためのメタヒューリスティクスの一種である。 局所的最適解を避けるために、目的関数値が一時的に悪くなるような状態へも遷移するという特徴がある。演算を繰り返すごとに値の変化量が小さくなっていくため、最終的に一つの値に収束する。

最小化問題に対する焼きなまし法の疑似アルゴリズムを示す。

初期状態を一つとり、最良状態とする。その時のスコアを最良スコアとする。

イテレーション数 $k = 0$ をとる。最大イテレーション数 k_{max} をとる

while $k < k_{max}$ **do**

現在の温度を求める。

新しい状態を近傍関数から得る。

新しい状態のスコアを得る。

現在のスコアと温度をもとに遷移確率を求める。

if new score ¡ best score **then**

最良状態、最良スコアにそれぞれ現在の状態とスコアを代入する。

end if

if new score¿ best score and random¡prob **then**

state=next state, score = next score

end if
end while

4.2 提案手法

対象とする問題は、目的地を辿る順番を決める問題と順番が与えられた際にドローンの発着場所を決める問題の二つに分割することができる。前者は組合せ最適化問題であり、ソルバーで解くのが困難であるため焼きなまし法を用いる。後者は SOCP として定式化することができ、ソルバーで求解する。
以下に提案手法のアルゴリズムの全体を示す。

初期状態を一つとり、最良状態とする。その時のスコアを最良スコアとする。
イテレーション数 $k = 0$ をとる。最大イテレーション数 k_{max} をとる
while $k < k_{max}$ **do**
現在の温度を求める。
新しい状態を近傍関数から得る。
新しい状態のスコアを得る。
現在のスコアと温度をもとに遷移確率を求める。
if new score ¡ best score **then**
最良状態、最良スコアにそれぞれ現在の状態とスコアを代入する。
end if
if new score¿ best score and random¡prob **then**
state=next state, score = next score
end if
end while

5 数値実験

5.1 定数の設定

5.2 初期点とパラメータの設定

5.3 テスト問題の生成方法

5.4 実験結果

5.4.1 計算時間と精度の評価

6 結論

謝辞

参考文献