時間枠制約付きチームオリエンテーリング問題に対するタブー探索法
A Tabu Search Algorithm for the Team Orienteering Problem with Time Windows

情報工学専攻　丹治 春人
Haruto TANJI

概要

時間枠制約付きチームオリエンテーリング問題は，オリエンテーリング問題を拡張した問題である．時間枠制約付きチームオリエンテーリング問題の目的は，利益，サービス時間，時間枠を持つ顧客の集合と顧客間の移動時間が与えられたとき，収集する総利益を最大化する訪問経路を求めることである．

本研究では，先行研究で提案された多点スタート反復局所探索法に対し，パス再結合を組み込んだアルゴリズムを提案する．パス再結合は，タブー探索法などのメタ戦略で用いられるアプローチであり，初期解と誘導解から多様な良質解を多数生成する．生成した解を初期解とし，再び反復局所探索を行うことで解を改善する．

時間枠制約付きチームオリエンテーリング問題のベンチマーク問題に対し，数値実験を行った．提案手法は，12 個の問題例において，先行研究で発見された最も優れた解に対する改善解を発見した．

キーワード: オリエンテーリング問題, 時間枠制約, パス再結合.

1　はじめに

オリエンテーリング問題とは，利益とサービス時間を持つ顧客の集合と顧客間の移動時間が与えられたとき，集める利益を最大化する訪問経路を求める問題である．求める経路を複数に拡張した問題をチームオリエンテーリング問題と呼び，さらに，求める経路が複数かつ各顧客に時間枠制約が付いた問題を時間枠制約付きチームオリエンテーリング問題（team orienteering problem with time windows, TOPTW）と呼ぶ．時間枠制約とは，各顧客のサービスを決められた時間枠の中で開始する制約である．各顧客は高々一回しか訪れてはならず，訪れない顧客があってもよい．

TOPTW は NP 困難であり[1]，厳密解法のアルゴリズムは非常に時間がかかると考えられている．時間枠制約付きチームオリエンテーリング問題の先行研究のほとんどはヒューリスティックな解法を提案している．

Amarouche ら[2]は，反復局所探索法を繰り返す多点スタート反復局所探索法を提案した．新たな初期解の生成において，適応メモリ戦略や過去の探索で得た質の高い経路を用いることで，アルゴリズムを強化している．

本研究では，先行研究で提案された多点スタート反復局所探索法に対し，パス再結合を組み込んだアルゴリズムを提案する．

2　問題定義

頂点集合$V = \{0,1,2,\dots,n\}$と辺集合$A = \{(i,j): i \neq j, i,j \in V\}$をもつ完全有向グラフ$G = (V,A)$を考える．頂点0はデポを表し，デポ以外の頂点は顧客を表す．各頂点$i \in V$は非負の利益p_iと非負のサービス時間σ_i，時間枠$[e_i, l_i]$を持つ．デポを除く各顧客には高々一度訪問し，各顧客のサービスは決められた時間枠の中で開始する必要がある．このため，各頂点$i \in V$に時間枠の終了時刻l_iより遅く訪問してはならず，開始時刻e_iより早く到着した場合は，e_iまで待たなければならない．各辺$(i,j) \in A$には三角不等式を満たす非負の移動時間$c_{i,j}$が与えられる．

また，車両集合$M = \{1,2,\dots,m\}$を考える．各車両は時刻0にデポから出発し，時刻L_{max}までにデポに帰還するような経路rを一つ走行する．経路はある車両が訪問する顧客の順序付き部分集合である．ここで，経路rで収集する総利益を$P(r) = \sum_{i=1}^{size(r)} p_{r[i]}$，またその所要時間を

$$C(r) = \sum_{i=1}^{size(r)} (c_{r[i-1],r[i]} + W_{r[i]} + \sigma_{r[i]}) + c_{r[size(r)],0}$$

と表す．$r[i]$は経路r上のi番目の顧客であり，W_uは顧客uでの待ち時間を表す．所要時間はデポへの到着時刻と対応する．経路は所要時間$C(r)$がL_{max}以下かつ各顧客の時間枠の終了時刻以前に到着するときのみ実行可能とする．

TOPTW の実行可能解Sは，各顧客を高々一度のみ訪問する最大m個の実行可能な経路rから構成される．TOPTW の目的は，収集する総利益$\sum_{r \in S} P(r)$を最大化

することである．

3　アルゴリズム

Amarouche ら[2]によって提案された多点スタート反復局所探索法を行った後，本研究ではパス再結合によって新たな解を多数生成し，それらの解を初期解とした反復局所探索を行う．

3.1　多点スタート反復局所探索法

文献[2]の多点スタート反復局所探索法は，二つの解表現による探索空間を移動し，高品質な解で見つかる経路を記憶しておく適応メモリ戦略を用いる．二つの解表現のうち一つは実行可能な経路の集合で顧客の重複のない経路解である．もう一つは実行可能な経路を連結し，連結した経路に含まれない顧客をすべて挿入して得られる巨大ツアー解である．図 1，図 2 に顧客が 9 人のときの解表現の例を示す．適応メモリ戦略とは，タブー探索法などのメタ戦略で用いられる戦略の一つであり，ここでは探索の全期間にわたって高品質な経路を記録する長期メモリを用いる．

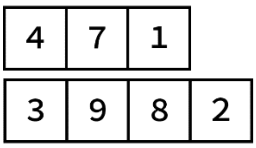

図 1:二つの経路からなる経路解の例

図 2:図 1 の経路解を連結し，
未訪問顧客を挿入して得られる巨大ツアー解の例

多点スタートを開始する前の初期解は貪欲法で作成する．作成した初期解を用い，長期メモリを初期化する．多点スタートでは記憶した高品質な経路から構築する巨大ツアー解を分割して得られる解を初期解として反復局所探索法を繰り返す．反復局所探索法では巨大ツアー解に対しランダム摂動を加えて得られる解を初期解として局所探索法を適用する．局所探索法では七つの近傍を適用する．

3.2　パス再結合

パス再結合は初期解と誘導解と呼ばれる二つの解を用意し，初期解から誘導解へ向かう解の列を生成する手法であり，本研究では 2 つの解の良い特徴を併せ持つ多様な解の生成を狙って適用する．

本研究のアルゴリズムの疑似コードをアルゴリズム 1 に示す．初めに行う多点スタート反復局所探索法にて，パス再結合に用いる初期解と誘導解の候補となる解を長期メモリに記憶する．記憶した解を用いたパス再結合によって多様な良質解を新しく生成する．生成した解を多点スタート反復局所探索法の初期解として再び探索を行う．

アルゴリズム 1

入力: M:長期メモリ
　　S_{init}:多点スタートの初期解
出力: S_{best}:探索全体で最も優れた解

```
1  begin
2      S_best ← multiStart(M, S_init)
3      S_inits ← pathRelinking()
4      for j ← 1 to length(S_inits) do
5          S_best ← multiStart(M, S_inits[j])
6      return S_best
```

解を記憶した長期メモリから初期解S_{init}と誘導解S_{guide}を選び，パス再結合に用いる．具体的には，長期メモリに記憶した10個の解から2個取る組合せを全て試し，パス再結合にかかる反復回数が大きいものから5組のパス再結合で生成される解を採用する．

パス再結合の詳細な手続きは次の通りである．ここで削除よりも挿入を優先することで初期解に誘導解の特徴を取り込み，それまでの探索で訪れていない良質な解の生成を狙う．まず初期解にのみ存在する頂点集合$V_{init} \subset V$と誘導解にのみ存在する頂点集合$V_{guide} \subset V$, 誘導解の各頂点$i \in V$に対し頂点iの一つ前に訪問する頂点$prev_i$と頂点iが誘導解で属する経路g_iを記憶しておく．初期解S_{init}から反復を始め，まず$i \in V_{guide}$である頂点iであって，パス再結合中に削除していない頂点iをS_{init}のいずれかの位置に挿入できるか試す．挿入できなければ，$i \in V_{init}$である頂点iのいずれかをS_{init}から削除する．$V_{init} = \emptyset$であれば，誘導解の頂点$i \in V$であって，誘導解で属す経路g_iとは異なる経路$k \neq g_i$に属す頂点iのいずれかをS_{init}から削除する．誘導解と異なる経路に属す頂点も全て削除していれば，誘導解の頂点$i \in V$であって，頂点iの一つ前にある頂点が記憶した$prev_i$と異なる頂点iをS_{init}から削除する．一度削除した頂点は，挿入を試すときに挿入する頂点iの一つ前の頂点が記憶した$prev_i$と一致す

るときのみ，S_{init} に挿入する．各反復におけるS_{init} を中間解としてS_{inits} に挿入することで記憶する．反復はS_{init} とS_{guide} の二つが一致したら終了する．

4　数値実験

web サイト[3]から入手できる TOPTW のベンチマーク問題に対し数値実験を行った．ベンチマーク問題は文献[4], [5], [6]で提案されたもので，[5]と[6]については車両数（1 から 4）ごとに分類する．また，これらの問題例は Solomon と Cordeau の 2 タイプに分かれており，1 つの分類に 10 から 56 までの問題例が含まれる． さらに，Solomon タイプの問題例については頂点の分布と時間枠に関して，さらに分類がつけられる．頂点の分布はランダム分布（R），クラスター分布（C），ランダム-クラスター分布（RC）の三種類に分かれている．時間枠は小さい問題例が*100 に, 大きい問題例が*200 に分かれている．

数値実験は乱数のシード値を変えて各問題例に対して 10 回行った．計算機は 3.50GHz の Intel® Xeon® CPU E5-2637 v3 を使用し，Rust 言語で実装した．

結果を表 1，表 2，表 3，表 4 に示す．表 1，表 2，表 3 は既知の最良値を 1 としたときの提案手法の目的関数値を表し，1 を超える値は最良値を更新する問題例があったことを意味する. 各問題例に対する 10 回の実験の平均値に対し，全問題例の平均をとったものを平均値の列に表し，各問題例に対する最良値を最良値の列に表す．表 4 は既知の最良値を上回った問題例について，既知の最良値と提案手法の最良値を示す．

提案手法は，12 の最良解を更新した．一つの問題例に対する平均計算時間は 11.9 時間であった．10 回の試行で得た全問題例の最良値と既知の最良値を比較すると，提案手法は 370 の問題例のうち 328 の問題例で既知の最良値以上の目的関数値を取る解を得た．つまり 316 の問題例で既知の最良解と同じ目的関数値の解を得た．

表 1：[4]に対する数値実験の結果

問題例	車両数	平均値	最良値
Solomon:C100	-	1.0000	1.0000
Solomon:R100	-	0.9999	1.0000
Solomon:RC100	-	0.9996	1.0000
Solomon:C200	-	1.0000	1.0000
Solomon:R200	-	1.0000	1.0000
Solomon:RC200	-	1.0000	1.0000
Cordeau	-	0.9936	1.0000

表 2：[5]に対する数値実験の結果

問題例	車両数	平均値	最良値
Solomon:C100	1	1.0000	1.0000
Solomon:R100	1	1.0000	1.0000
Solomon:RC100	1	1.0000	1.0000
Cordeau	1	0.9996	1.0000
Solomon:C100	2	1.0000	1.0000
Solomon:R100	2	1.0000	1.0000
Solomon:RC100	2	0.9999	1.0000
Cordeau	2	0.9984	1.0000
Solomon:C100	3	1.0000	1.0000
Solomon:R100	3	1.0000	1.0000
Solomon:RC100	3	1.0000	1.0000
Cordeau	3	0.9977	1.0000
Solomon:C100	4	1.0006	1.0088
Solomon:R100	4	0.9999	1.0010
Solomon:RC100	4	0.9999	1.0000
Cordeau	4	0.9953	1.0000

表 3：[6]に対する数値実験の結果

問題例	車両数	平均値	最良値
Solomon:C200	1	1.0000	1.0000
Solomon:R200	1	0.9999	1.0000
Solomon:RC200	1	0.9996	1.0000
Cordeau	1	0.9897	1.0000
Solomon:C200	2	0.9998	1.0000
Solomon:R200	2	0.9993	1.0021
Solomon:RC200	2	0.9991	1.0006
Cordeau	2	0.9910	1.0011
Solomon:C200	3	1.0000	1.0000
Solomon:R200	3	0.9993	1.0000
Solomon:RC200	3	0.9997	1.0000
Cordeau	3	0.9943	1.0023
Solomon:C200	4	1.0000	1.0000
Solomon:R200	4	1.0000	1.0000
Solomon:RC200	4	1.0000	1.0000
Cordeau	4	0.9920	1.0007

表 4: 先行研究の最良解を改善した問題例

問題例	車両数	先行研究の最良値	提案手法の最良値
r202	2	1353	1354
r205	2	1402	1403
r209	2	1423	1426
rc202	2	1523	1524
pr13	2	845	846
pr09	3	1276	1278
pr12	3	1002	1004
pr14	3	1375	1376
pr17	3	841	843
pr13	4	1392	1393
c108	4	1130	1140
r111	4	952	953

5 まとめ

本研究では，先行研究で提案された多点スタート反復局所探索法に対し，パス再結合を組み込んだアルゴリズムを提案した．パス再結合は過去の探索で得た良質な解を組み合わせ，組み合わせる解それぞれの特徴を併せ持つ解を生成する．数値実験の結果，提案手法は既存の最良解を 12 の問題例で改善した．この新しい結果は，今後の研究のベンチマークとなり得るものである．

今後の課題として，タブーリストを用いた探索の効率化や，実行不可能解を探索空間に含めることによる解の質の改善が挙げられる．

謝辞

本研究を進めるにあたり，大変多くのご指導，ご助言を頂いた中央大学理工学部情報工学科の今堀慎治教授に深く感謝いたします．また，多大なるご助言，ご協力を頂いたアルゴリズム工学研究室の皆様には大変お世話になりました．心から感謝いたします．

最後に，大学 4 年間に加え，大学院に 2 年間通わせていただいた両親，祖父母に深く感謝いたします．

参考文献

[1] Vansteenwegen P, Gunawan A. (2019). Orienteering Problems, Models and Algorithms for Vehicle Routing Problems with Profits. Springer Cham.

[2]Amarouche Y, Guibadj RN, Chaalal E, Moukrim A. (2020). Effective Neighborhood Search with Optimal Splitting and Adaptive Memory for the Team Orienteering Problem with Time Windows. Computers and Operations Research. Vol.123:105039

[3] The Orienteering Problem: Test Instances - Division CIB (online), available from https://www.mech.kuleuven.be/en/cib/op (accessed 2022-12-26)

[4] Vansteenwegen P, Souffriau W, Berghe GV, Oudheusden DV. (2009) Iterated Local Search for the Team Orienteering Problem with Time Windows. Computers and Operations Research. Vol.36(12):3281-3290

[5] Righini G, Salani M.(2008) New Dynamic Programming Algorithms for the Resource Constrained Elementary Shortest Path. Networks. Vol. 51(3):155-170

[6] Montemanni R, Gambardella LM.(2009) An Ant Colony System for Team Orienteering Problems with Time Windows. Foundations of Computing and Decision Sciences. Vol. 34(4):287-306