日本オペレーションズ・リサーチ学会
2023 年春季研究発表会

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パス再結合

中央大学　*丹治　春人　TANJI Haruto
01015123　中央大学　今堀　慎治　IMAHORI Shinji

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．

Amarouche ら[1]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，文献[1]の手法にパス再結合を組み込んだアルゴリズムを提案し，TOPTW のベンチマーク問題に対する数値実験により性能評価を行う．

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$\sigma_i$，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[1]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．

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(M)
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[2]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[3]，[4]，[5]で提案されたもので，[4]と[5]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．

数値実験は乱数のシード値を変えて各問題例に対して 5 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 に示す．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 5 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．

提案手法は，11 個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 15.1 時間であった．追加の実験結果は当日報告する．

表 1：数値実験の結果

インスタンス	車両数	平均値	最良値
[3](Solomon)	-	0.9936	1.0000
[3](Cordeau)	-	0.9999	1.0000
[4](Solomon)	1	1.0000	1.0000
[4](Solomon)	2	0.9999	1.0000
[4](Solomon)	3	1.0000	1.0000
[4](Solomon)	4	1.0001	1.0088
[4](Cordeau)	1	0.9996	1.0000
[4](Cordeau)	2	0.9986	1.0000
[4](Cordeau)	3	0.9975	1.0007
[4](Cordeau)	4	0.9954	1.0000
[5](Solomon)	1	0.9999	1.0000
[5](Solomon)	2	0.9995	1.0021
[5](Solomon)	3	0.9996	1.0000
[5](Solomon)	4	1.0000	1.0000
[5](Cordeau)	1	0.9898	1.0000
[5](Cordeau)	2	0.9906	1.0011
[5](Cordeau)	3	0.9938	1.0023
[5](Cordeau)	4	0.9917	1.0007

参考文献

[1] 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.

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

[3] 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) pp.3281-3290

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

[5] 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) pp.287-306