卒 業 論 文

時間枠及び乗車時間ペナルティ付き
乗合タクシー問題に対する局所探索法

051600141　　竹田陽

名古屋大学情報文化学部自然情報学科

数理学専攻

2020年1月

時間枠及び乗車時間ペナルティ付き
乗合タクシー問題に対する局所探索法

051600141　　竹田 陽

概 要

乗合タクシー問題 (Diai-a-ride problem,DARP) とは, 異なる移動需要を持つ顧客を同じ車両で同時に輸送する際に, 効率の良い車両の割り当てとスケジューリングを考える問題である. 乗合タクシー問題では, 顧客の乗降点と到着時刻に対する時間枠制約，乗車時間に対する制約が与えられたときに, 車両の運用にかかるコストおよび顧客の不満度合いを最小にすることを目的とする.

乗合タクシー問題では、利用者を出発地から目的地まで輸送するサービスにおいて、リクエスト全ての点を訪問すること、車両はデポから出発しデポに戻ること、リクエストのペアは必ず同じ車両が訪問すること、出発地より後に到着点を訪問することの全てを満たすルートと車両の割り当てを考える問題である。利用者は、出発地と目的地の場所、またその地点に車両が到着してほしい時間枠を指定することができる。また、輸送の時間に対して、最大乗車時間を指定することができる。乗合タクシー問題に対しては，多くの研究がなされている．多くの先行研究では, 利用者の最大乗車時間とリクエストの乗車時刻と降車時刻に対しての時間枠をハード制約として与えている.

時間枠と乗車時間に対する制約を区分線形で凸のペナルティ関数として与えることで、乗合タクシー問題の顧客の不満度合いをより柔軟に表現できるようにした。その問題を「時間枠及び乗車時間ペナルティ付き乗合タクシー問題」と定義する。

本研究では、時間枠及び乗車時間ペナルティ付き乗合タクシー問題に対して、ルート内の近傍操作とルート間の近傍操作を用いた局所探索法を用いたアルゴリズムを提案する。ルート内の近傍操作としては、挿入近傍と交換近傍の 2 種類を提案し実装した。挿入近傍とは、ルートのリクエストを 1 つ選択し、他のルートに挿入する操作のことで、交換近傍とは、ルートのリクエストを 1 つ選択し、他のルートのリクエストと交換する操作のことである。それぞれを計算実験により比較したところ、挿入近傍を用いたほうが交換近傍を用いるより良い解を見つけることができることがわかった。

Local search algorithm for Dial-a-ride problem with convex time window penalty

051400141 Kiyoshi Takeda

Abstract

The Dial-a-Ride Problem (DARP) consists of designing vehicle routes and schedules for multiple users who specify pickup and delivery requests between origins and destination. The aim is to design a set of minimun cost vehicle route while accommodating all requests. Side constraints include vechile capacity, route duration, maximum ride time, etc. DARP arises in share taxi service or transporting people in health care service.

A number of papers about DARP has been published. most of the previous research deal with time windows and maximum ride time as hard constraints.

In this study, we consider with the DARP where time windows and maximum ride time are soft constraints. These soft constraints make the DARP more generic. We propose iterated local search algorithm to find a solution to the problem.

目 次

第1章　はじめに

近年, 新しい移動手段として乗合タクシーサービスの需要が増加してきている．また, 老人や介護が必要な人を自宅などからヘルスケアセンターなどまで輸送するようなサービスも, 高齢化に伴い需要が増加してきている．都心部では、乗合タクシーの実証実験がさかんに行われたりしており、実用化が進んできている．これらのサービスにおいて、利用者の満足度を考慮しつつサービスの実行にかかるコストを最小化する問題は乗合タクシー問題 (Dial-a-Ride problem, DARP) と呼ばれる．乗合タクシー問題の制約としては, 車両の容量資源の最大容量を超えて乗車することはできない (容量制約), 車両が回るルートの大きさ (最大ルート距離)，顧客が車両に乗っている時間の長さ (最大乗車時間) が与えられる．

本研究では, 時間枠及び乗車時間をペナルティ関数で与えてソフト制約とする乗合タクシー問題について考える。ペナルティ関数は区分線形凸関数であるとする。区分線形関数にすることで, 時間枠より少しの遅延は許容できる場合など, 様々なケースを柔軟に考慮することができるようになり, 先行研究より汎用的な問題とすることができる。

3 章では pickup and delivery problem(PDP) を紹介し，乗合タクシー問題 (DARP) について詳しく説明する．4 章では時間枠及び乗車時間ペナルティ付き乗合タクシー問題の定式化を行う。5 章では本問題に対する提案手法を紹介する。6 章では、挿入近傍と交換近傍による解の精度の比較と既存研究の最良解との比較を行った結果を紹介する。また、車両数を減らした際の結果も紹介する。

第2章　先行研究の紹介

乗合タクシー問題に対して、多くの研究がなされている。ここでは、DARP に関する既存研究をいくつか紹介する。

Jaw らは, 各リスエストが時間枠を持ち、複数の車両で容量制約、乗車時間制約、時間枠制約を守りつつ車両の移動コストを最小化する問題に対する近似解法として、連続挿入法を提案した [3]。この解法は、ルートに挿入した時の目的関数値の増加が最小になるようなリクエストを選択してルートに挿入していく解法である。この手法は大きく分けて、リクエストの挿入が実行可能となるルート内の挿入可能箇所を求めることと、求めた挿入箇所の中でコストの増加が最も小さくなるような挿入箇所を選択することの 2 段階に分かれている。

Cordeau らは,DARP に対してタブー探索アルゴリズムを提案した [2]。この手法では、あるルートからリクエストをひとつ取り除き, 別のルートに挿入する、挿入近傍を用いている。解の探索では、実行不可能な解の探索を許しており、目的関数に制約違反ペナルティ関数が用意されている。目的関数に制約違反ペナルティ関数を加えたものを $f(s)$ とする。解の評価関数に、リクエストとそのリクエストが属するルートのペアが解に採用された回数に比例するペナルティ関数を加える。このペナルティ関数を $p(s)$ とすると、解の評価関数は $f(s)$ と $p(s)$ の和で表すことができる。そうすることで、頻繁に訪れられる解に訪問しないようになる.

Braekers らは、DARP に対しての発見的解法として焼きなまし法 (Simulated Annealing,SA) を用いた手法を提案した [1]. この手法では、挿入近傍、交換近傍、2-opt*,r-4-opt、削除近傍の 5 つの近傍操作を使用している。挿入近傍とは、ランダムにルートを選択して、そのルートからリクエストを 1 つ取り除いて別のルートの最適な位置に挿入する操作によって得られる解集合である。交換近傍とは、ランダムにルートを選択して、そのルートからリクエストを 1 つ取り除き、他のルートのリクエストと交換することによって得られる解集合である。2-opt*とは、異なるルート間の辺の交換操作のことである。 DARP では、リクエストのペアは同じ車両が訪問しなければならないので、乗客が乗っていない辺が交換の対象である。r-4-opt は、ルート内のリクエストの訪問順を変更する操作のことである。上記 4 つの近傍操作は、解を改善するために行われる操作であるが、最後の削除近傍は、サービスにおいて使用される車両数を減らすことを目的とした操作である。具体的には、ルートを 1 つランダム選択し、そのルートからリクエストを全て取り除く。そしてその取り除かれたリクエストをランダムな順番で別のルートに挿入していく操作のことである。

適応的巨大近傍探索 (Adaptive large neighborhood search, ALNS) とは, 局所探索ある解の一部を 破壊して再構築する近傍探索のフレームワークである [5]. 破壊の発見的解法 (破壊方法) と再構築 の発見的解法 (構築方法) が複数用意され, 1 つの破壊方法と 1 つの構築方法を組み合わせて一つ の近傍操作となる. 各反復ごとにどの破壊方法・構築方法を選ぶかは, 過去の反復での解の改善履歴を基に計算した値によって選択される. 破壊方法、構築方法は問題に合わせて自由に決めることができる。Pisinger らは、時間枠付きの PDP に対して ALNS を用いる手法を提案した [4]。Pisinger らは、近傍となる破壊方法を 7 つ、構築方法を 2 つ使用している。

可変近傍法 (Variable neighborhood search,VNS) とは、複数の近傍を定義し、暫定解に対してある近傍から 1 つ解を選択し、局所探索を適用するという手順を繰り返す方法である。局所探索

の反復において、暫定会の更新が起きたかによって次の解を選択する近傍を変えるのが特徴である。Paolo らは、DARP に対して可変近傍法 (Variable neighborhood search,VNS) を用いた手法を提案した [6]。この論文では、交換近傍、挿入近傍、実行不可能解に対する挿入近傍、連鎖近傍、削除近傍の 5 つの近傍が使用されている。

第3章　問題定義

この章では，まず pickup and delivery problem(PDP) について説明し．そのあとに乗合タクシー問題 (DARP) について詳しく説明する．

3.1　Pickup and delivery problem

Pick and Delivery problem (PDP) は, 荷物を出発地まで目的地まで輸送する問題である．PDP では, 荷物を受け取る地点 (pickup) と荷物を配送する地点 (delivery) のペアからなる n 個のリクエストと m 台のが与えられたときに以下の制約を満たしつつ与えられたコストを最小化するルートを求める問題である．

1. 与えられた n 個のリクエスト全ての地点を訪問する．
2. 全ての車両はデポから出発してデポに帰る．
3. リクエストでペアになっている出発地と目的地は同じ車両が訪問する．
4. それぞれのリクエストにおいて, 必ず出発地を訪問した後に目的地を訪問する．

ここでのリクエストとは，荷物の輸送要求のことを指す．与えられるコストとしては各頂点間の距離や時間などがある．よく制約として考えられるものは, リクエストの訪問点を決められた時間内に訪れなければならない時間枠制約や車両の容量を制限した容量制約などがある．また, 車両の種類が複数あり，車両によって最大容量が異なる多資源制約や，複数のデポを考慮する問題など，様々な拡張が考えられている．

3.2　乗合タクシー問題 (Dial-a-ride problem)

乗合タクシー問題 (Dial-a-ride problem,DARP) は, PDP を人の輸送に特化した問題である、DARP は PDP と違い人を輸送するため, 車両に乗っている乗車時間が長いと利用者の不満がたまってしまう．そこで乗車時間やリクエストの訪問時間のずれなどで評価される不満度を考慮する必要がある．

顧客の満足度は、利用者の乗車時間と出発地と目的地での待ち時間ではかる。本研究では, 利用者が乗車時刻, 降車時刻, 乗車時間のそれぞれに対して希望を持つことを考える. それぞれの希望はそれぞれ連続区分線形凸関数のペナルティ関数で表される．ペナルティ関数を区分線形関数にすることで、任意の間隔ごとに関数を設定することが可能になり、3 分の遅延なら許容できるが 20 分の遅れは許容できないなどの表現が可能になる. また, 乗車時間に応じてペナルティをかけることができるので, 不満度を柔軟に表現することが可能になる. したがって, DARP をより汎用的に解くことが可能となる.

事前に全てのリクエストがわかっている問題を静的 DARP, リクエストが全てはわかっておらず問題を解く過程でリクエストが次々と与えられる問題を動的 DARP といい, 本研究では静的 DARP を考える．

3.3 乗合タクシー問題の問題複雑度

DARP は, 使用する車両数を 1, リクエストにおける出発地を全てデポとする巡回セールスマン問題 (traveling salesman problem, TSP) ととらえることができる．TSP は NP 困難 [7] であることが知られているため, DARP も NP 困難である．

第4章　定式化

乗合タクシー問題は、$G=(V,E)$ の完全有向グラフ上で定義される．$V=\{\{0,2n+1\},P,D\}$ を頂点集合とし, $A=\{(i,j)\mid i,j\in V, i\neq j\}\}$ を各頂点間の辺集合とする．ここで, デポを 0 と $2n+1$ で表し, V の部分集合 $P=\{1,...,n\}$ を乗車地点の集合, V の部分集合 $D=\{n+1,...,2n\}$ を降車地点の集合とする．$i\in P$, $i+n\in D$ に対して,V 上の 2 点のペア $(i,i+n)$ は乗車地点の頂点 i から目的地に対応する頂点 $i+n$ への乗客輸送リクエスト (以下、単にリクエストと呼ぶ) とする．それぞれの頂点 $v_i\in V$ には, 負荷 q_i とサービス時間 s_i が与えられる.

各リクエストは, m 台の車両 $k\in K=\{1,2,..m\}$ で訪問される．σ をルートの集合とし, σ_k を車両 k の訪問する頂点の順列とすると, $\sigma=\{\sigma_1,...,\sigma_m\}$ である．また, $\sigma_k(h)$ は車両 k が h 番目に訪問する頂点をあらわす．頂点 $i,j\in V$ 間の距離を c_{ij}, 時間を t_{ij} とする．車両の容量を Q とし, 車両 k の $\sigma_{k(h)}$ における容量を $q_{\sigma_k(h)}$ とする．x_{ij} を 0-1 変数とし, 頂点 i,j を結ぶ辺がルートに含まれれば 1，そうでなければ 0 とする．各車両は 1 つのデポから出発しデポに帰る．ここで, n_k は車両 k が訪問するデポ以外の頂点の数である

この問題に対して、目的関数として

1. 車両運用コスト最小化,

2. 顧客満足度最大化

の 2 つを考える。車両運用コストとしては、様々なコストが考えられるが本研究では車両が走行するルートの総距離とする。ルートの総距離を $d(\sigma)$ とすると,

$$d(\sigma)=\sum_{k\in K}\sum_{h=0}^{n_k}c_{\sigma_{k(h)},\sigma_{k(h+1)}}$$

と表せる．各リクエストにおける乗車時刻に対するペナルティ関数を g_i^+, 降車時刻に対するペナルティ関数を g_i^-, 乗車時刻に対するペナルティ関数を g_i とする．頂点 i でのサービス開始時刻を τ_i とし, I をリクエスト集合として, I_σ をルート σ に属するリクエストの集合とする．また, 利用者の不満度を $t(\sigma)$ とすると,

$$t(\sigma)=\sum_{i\in I_\sigma}(g_i^+(\tau_i)+g_i^-(\tau_{i+n})+g_i(\tau_{i+n}-\tau_i))$$

と表せる。各ルートについて車両の割り当てとリクエストの訪問順が決まっている 1 つのルートが与えられた場合に、各頂点でのサービス開始時刻を決定する必要がある。本研究では、目的関数と制約が全て線形の式で表すことが可能なので、線形計画問題 (linear programming problem, LP) として解くことができる。

本研究では、2 つの目的関数の重み付き和の最小化を考える。そうすることで、少し顧客の不満度が上がっても良いのでルートの総距離を短くしたい場合や、顧客の不満度を最優先で考えたい場合など、様々な状況を考慮することができる。

それぞれの係数を定数 α と β としたとき、以下のように定式化できる:

$$\begin{aligned}
&\text{minimize} && \alpha d(\sigma) + \beta t(\sigma), && && (1)\\
&\text{subject to} && \sum_{i \in V} x_{ij} = 1 && (j \in P \cup D), && (2)\\
& && \sum_{j \in V} x_{ij} = 1 && (i \in P \cup D), && (3)\\
& && \sum_{k \in K} n_k = 2n, && && (4)\\
& && \sigma_k(0) = \sigma_k(n_k + 1) = 0 && (k \in K), && (5)\\
& && 0 < q_{\sigma_k(h)} < Q \quad && (h \in \{0, ..., n_k\}, k \in K), && (6)
\end{aligned}$$

$$\begin{aligned}
& && g_i(\tau_{i+n} - \tau_i) > s_i + t_{i,i+n} && (i \in I), && (7)\\
& && \tau_{\sigma_k(h+1)} \geqq \tau_{\sigma_k(h)} + s_{\sigma_k(h)} + t_{\sigma_k(h),\sigma_k(h+1)} && && \\
& && \quad (h \in \{0, ..., n_k\}, k \in K). && && (8)
\end{aligned}$$

式 (2), (3) は訪問点が 1 回ずつしか訪問されないことを, 式 (4) は全てのリクエストが実行されることを表している. 式 (5) は全ての車両がデポから出発してデポに帰ることを表している. 式 (6) は車両の容量制約を表している. 式 (7), (8) はリクエストの先行制約とサービス時刻に関する制約を表している.

第5章　提案手法

本節では, 本研究で提案する手法について説明する. 本研究では, リクエストの割り当てと訪問順を局所探索法を用いて求める. それぞれの反復で得られたルートに対しては, LP ソルバーを使って最適なサービス開始時刻を決定するアルゴリズムを提案する.

5.1　初期解生成

リクエストをランダムに選び, 車両 k にリクエストのペアが連続となるようにルートの最後に挿入する. これを $k = 1$ から m まで繰り返した後, 未割り当てのリクエストがあれば $k = 1$ として同様の操作を続ける. これを未割り当てのリクエストがなくなるまで続ける. このように生成することで、デポから出発してリクエスト全てを訪問し、同じ車両で出発地のあとに目的地を訪問するという制約を必ず守る初期解を生成することができる。

5.2　制限の緩和

本研究では, 局所探索を行う上でより自由に探索を行うために, 車両における容量制約を緩和し, 容量制約を破った時のペナルティを計算するペナルティ関数を定義する. 容量制約のペナルティ関数を目的関数に加えた評価関数を用いて解を評価することにより, 実行不可能解も探索可能になる.

車両の容量制約のペナルティは, 車両の最大容量を超えて乗った人数として, QP と表す. 車両 k に対してルートの i 番目を訪問後に容量を超えて乗っている人数を QP_i^k とすると,

$$QP = \sum_{k \in K} \sum_{i \in n_k} QP_i^k,$$

と表せる. γ を定数とすると、ペナルティを加えた評価関数を $f(\sigma)$ とすると

$$f(\sigma) = \alpha d(\sigma) + \beta t(\sigma) + \gamma QP,$$

で定義する.

5.3　局所探索法

局所探索法 (local search) とは、解を逐次的に改善させていく手法である。解に変化を少し加える操作を近傍操作と呼び、近傍操作によって生成されうる解の集合を近傍と呼ぶ。局所探索法では、適当な初期解からはじめ、現在の解の近傍内に、より良い解が存在すればその解に移動する、という操作を近傍内に改善がなくなるまで反復する方法である。本研究では、近傍操作は大きく分けてルート内の操作とルート間の操作の 2 種類を用いる。

5.3.1 ルート内の近傍操作

1 つのルートの 1 つの頂点を選び、同じルート内の別の箇所に挿入し直す操作である。本研究では 1 つのルートで改善がなくなるまでルート内の近傍操作を行うが、探索する近傍サイズはルート内のデポを除いた頂点数を a とすると $2a^2$ とした。

5.3.2 ルート間の近傍操作

ルート間の近傍操作では、挿入近傍と交換近傍の 2 種類を実装し、計算結果を比較した。2 種類ともに、ルート間の近傍操作を行い、現在の解の評価関数よりも改善した場合、操作後の近傍解に移動する。

挿入近傍

1 つのルートから 1 つのリクエストペアを選び、別のルートに挿入する操作である。挿入する場所は評価関数が最も良くなる場所とする。

交換近傍

1 つのルートから 1 つのリクエストペアを選び、別のルートのリクエストペアと交換する操作である。挿入近傍と同じく、新たに挿入する場所は、最も評価関数がよくなる場所とする。

5.3.3 局所探索アルゴリズム

時間枠及び乗車時間ペナルティ付き乗合タクシー問題に対する局所探索アルゴリズムとして、まずルート内で近傍操作を改善がなくなるまで行い、得られた局所最適解に対してルート間の近傍操作を行い、最良の箇所に挿入する。改善している場合、局所最適解を更新し、現在の解から移動する。

以下に、x_{init} を初期解、x を現在解、x_{init} を暫定解とした際の提案手法の概要を示す。

Algorithm 1 提案手法

1: $x := x_{init}$ とする。
2: 改善がなくなるまでルート内の近傍操作を行う
3: ルート間の近傍操作を行う
4: 変化があったルートに対してルート内の近傍操作を行う
5: **if** 改善した **then**
6: $\quad x_{best} := x$
7: **end if**
8: x_{best} を局所最適解として出力して終了.

第6章　計算実験

6.1　実験環境

実験に用いるプログラムは C++を用いて実装し、計算機はプロセッサ 1.4GHz Intel Core i5, メモリ 16GB 2133 MHz LPDDR3 の macOs を搭載したものを使用した。探索における最適なサービス開始時刻を決定する際の LP ソルバーとしては、Gurobi Optimizer (ver 9.0.0) を使用した。

6.2　問題例の作成方法

DARP では多くの既存研究があるが、本研究では時間枠及び乗車時間に対して区分線形で凸のペナルティ関数で与えている。このような問題設定のインスタンスは存在していないため、ベンチマークとしてよく使用される Cordeau らによって提供されている [2] インスタンスに修正を加えて計算実験を行う。修正方法を以下に示す。

時間枠に関しては、サービス開始可能時刻を e、サービス開始最遅時間を l とすると、0 以上 e 以下に対しては傾き-1、e から l に対しては値が 0、l 以上に対しては傾きが 1 となるような、区分数が 3 のペナルティ関数を作成する。この修正作業を全てのリクエストに対して行う。乗車時間に関しては、乗車時間の閾値を L とすると、0 以上 L 以下に対しては値が 0、L 以上に対しては傾きが 1 となるような区分数 2 のペナルティ関数を作成する。この修正作業を全てのリクエストペアに対して行う。

6.3　インスタンスについて

計算実験に使用するインスタンスは、以下の特徴を持つ。

1. 訪問点におけるサービス時間 d を 10 とする。
2. 乗降人数は 1 人とする。
3. 訪問点 v_i と v_j 間の距離 c_{ij} と時間 t_{ij} は、2 つの頂点のユークリッド距離とする。
4. 乗車時間の閾値 (最大乗車時間) を 90 とする。
5. 車両の最大容量を 6 人とする。
6. それぞれの車両のルートの最大の長さを 480 とする。

以上の特徴を持つリクエスト数 24 から 144 のインスタンスを使用して計算実験を行った。

6.4 実験結果

ここでは、提案手法について行った計算実験の結果を示す。顧客数と車両数がそれぞれ同じであるインスタンスが 2 つずつあるため、添字 a, b で区別する。

表における局所最適解は、ルートの総距離を示す。

6.4.1 挿入近傍と交換近傍の比較

局所探索におけるルート間の近傍操作として挿入近傍と交換近傍の比較を行った。目的関数におけるペナルティ係数 α, β は 1 とする。試行回数を 10^4 回としたときの計算結果を表 6.1 に示す。

ほぼ全てのインスタンスにおいて、挿入近傍による値のほうが交換近傍による値よりも良い性能であることが確認でき、平均で 6.5%良い性能であることが確認できた。

6.4.2 既存研究との比較

本研究での提案手法で、時間枠と乗車時間のペナルティの値を大きくすることで、既存研究との計算結果を比較することができる。6.4.1 で挿入近傍のほうが良い解を得られることが確認できたため、挿入近傍を用いて出力した解と Cordeau らによって示された最良値を比較する。また、目的関数におけるペナルティを $\alpha = 1, \beta = 500$ とする。Cordeau らによって示された最良値は試行回数が 10^5 回の値であるため、同様に 10^5 回の試行を行い、結果を比較した。計算実験の値と最良解との相違を表 6.2 に示す。

複数のインスタンスでペナルティの値が 0 である解を得ることができ、Cordeau らが示した最良解との比較を行うことができた。規模の小さいインスタンス (r1a,r2a,r1b,r2b) に対しては、最良解との相違が 30%程度の解を得ることができた。一方、規模の大きいインスタンスに対しては、あまり良い結果が得られなかった。これは、局所最適解を得るための十分な試行回数を行うことができなかったことが理由と考えられる。また、ペナルティに対する係数を $\beta = 500$ と大きくしているので、ルートの総距離についてはあまり最小化されなかった考えられる。

6.4.3 目的関数の係数の違いによる計算結果

本研究では、目的関数をルートの総距離と顧客の不満度合いの重み付き和として、その最小化を考えている。2 つの係数 α, β の値を変化させた際の計算結果を表 6.3 に示す。

平均を見ると、係数 β が小さいときは出力されたルートの長さは小さく、ペナルティの値が大きいことがわかる。一方、係数 β が大きいときは出力されるルートの長さは大きいものの、ペナルティの値は小さいことがわかる。これらより、目的関数における定数 α,β の値を変えることでルートの長さとペナルティの重要度によって出力するルートを変化させることができることが確認できた。

6.4.4 車両数を減らした計算結果

実社会の乗合サービスにおける車両数を減らすことは、運用コストの削減に大きくつながると予想される。そこで、Cordeau らによって提供されているインスタンスの規模の小さいものに対して、車両数を 1 台少なくして局所最適解を出力したものと、車両台数に変化を加えないもので

表 6.1: 挿入近傍と交換近傍の比較

	size		挿入近傍		交換近傍	
問題例	顧客数	車両数	最良値	計算時間 (分)	最良値	計算時間 (分)
r1a	24	3	216.005	2.14	233.376	1.728
r2a	48	5	514.503	2.93	566.30	2.81
r3a	72	7	960.476	4.917	959.992	4.745
r1b	24	3	195.6	2.417	209.956	2.482
r2b	48	5	437.568	3.340	453.139	3.247
r3b	72	7	921.741	4.658	1031.662	4.719
average			540.982	3.4	575.738	3.288

比較を行った。目的関数におけるペナルティを $\alpha = 1, \beta = 1$ とし、6.4.2 と同様に 10^5 回の試行を行った。計算実験の値を表 6.4 に示す。

実験を行った 6 つのインスタンス全てで、ペナルティが発生しているものの実行可能なルートを出力できていることを確認できた。平均すると解の精度は 4%程度悪り、ペナルティ値は増加しているものの、車両数を削減できているため、サービス自体のコストを削減することが可能であると考えられる。

表 6.2: Best Known Score との比較

	size		本研究			Besk Known Score	
問題例	顧客数	車両数	局所最適解	ペナルティ値	GAP(%)	1000	10000
r1a	24	3	212.49	0.00	11.8	190.79	190.02
r2a	48	3	386.03	0.00	27.7	302.08	302.08
r3a	72	7	776.31	0.56	45.8	532.08	532.08
r4a	96	9	1005.17	0.00	75.4	572.68	572.68
r5a	120	11	1077.68	0.00	69.1	636.97	636.97
r6a	144	13	1506.35	9.56	87.9	801.40	801.40
r1b	24	3	195.6	0.00	18.9	164.72	164.46
r2b	48	5	388.43	0.00	31.0	301.28	296.06
r3b	72	7	777.09	0.00	57.5	498.20	493.30
r4b	96	9	973.73	0.00	81.6	548.89	535.90
r5b	120	11	976.45	0.00	65.5	592.65	589.74
r6b	144	13	1317.47	0.00	77.1	766.55	743.60
average			796.86	0.84	51.8	524.91	515.38

表 6.3: 係数 β を変化させた際の計算結果

	size		$\beta = 1$		$\beta = 10$		$\beta = 50$		$\beta = 500$	
問題例	顧客数	車両数	局所最適解	ペナルティ値	局所最適解	ペナルティ値	局所最適解	ペナルティ値	局所最適解	ペナルティ値
r1a	24	3	202.19	1.07	208.99	0.00	208.99	0.00	212.49	0.00
r2a	48	5	424.79	1.67	391.61	0.02	386.03	0.00	386.03	0.0
r3a	72	7	708.97	17.18	803.64	0.00	763.27	3.50	776.31	0.56
r4a	96	9	911.66	14.99	928.49	0.00	1005.17	0.00	1005.17	0.00
r5a	120	11	1038.52	19.37	1094.43	0.00	1077.49	0.00	1077.68	0.0
r6a	144	13	1350.81	22.38	1428.04	49.88	1494.52	39.64	1506.35	9.56
r1b	24	3	200.74	0.00	197.48	0.00	203.61	0.00	195.6	0.0
r2b	48	5	403.99	1.38	398.94	0.00	386.47	0.00	388.43	0.0
r3b	72	7	686.30	15.71	752.70	0.14	777.09	0.00	777.09	0.0
r4b	96	9	894.65	3.70	919.01	0.00	958.40	0.24	973.73	0.0
r5b	120	11	964.83	12.25	1007.51	0.00	976.45	0.00	976.45	0.0
r6b	144	13	1264.03	22.45	1297.15	0.34	1314.24	0.00	1317.47	0.0
average			754.29	11.01	785.66	4.19	795.97	3.615	796.86	0.84

表 6.4: 車両数を減らした際の比較

		台数に変化なし			1 台削除		
問題例	顧客数	車両数	局所最適解	ペナルティ値	車両数	局所最適解	ペナルティ値
r1a	24	3	202.19	1.04	2	235.23	0.44
r2a	48	5	386.03	0.00	4	390.20	13.08
r3a	72	7	708.97	17.18	6	749.78	45.47
r1b	24	3	200.74	0.00	2	217.61	2.40
r2b	48	5	388.43	0.00	4	377.78	11.46
r3b	72	7	686.30	15.71	6	707.90	27.26
average			428.77	5.65		446.41	16.18

第7章　まとめと今後の研究計画

時間枠及び乗車時間ペナルティ付き乗合タクシー問題に対して、局所探索法による解法を提案した。近傍操作においては、挿入近傍と交換近傍の 2 種類を考え、それぞれを使用した場合の実験結果を比較した。挿入近傍を用いたほうが、交換近傍を用いるより良い精度の解を出力することが確認できた。また、時間枠及び乗車時間のペナルティの値を大きくし、Cordeau らによる既存研究の研究結果と比較した。規模の小さいインスタンスに対しては相違 30%程度の解を出力したが、規模の大きいインスタンスに対してはあまり良い精度の解を出力することができなかった。また、規模の小さいインスタンスに対して、車両数を 1 台減らした際の解を出力し、比較を行った。時間枠及び乗車時間のペナルティの値が発生してしまうものの、実行可能な解を出力できることがわかった。

今後の研究計画としては、局所最適解ではなく大域的な最適解を出力できるようにするため、局所探索ではない手法を提案していきたい。また、規模の大きいインスタンスに対しても精度の良い解を得るために、近傍操作の見直しなどを行っていく。また、本研究の提案手法では近傍操作によって得られたルートの訪問順の全てに対して線形計画問題を LP ソルバーを使って解を出力したが、ある程度解の改善の可能性があるルートの訪問順に対してのみ線形計画問題を解くようにすることで計算時間を大幅に削減できると考える。したがって、解の改善可能性の判断についても手法の提案、実装を今後行っていきたい。

謝辞

本研究の遂行にあたり, 熱心な指導と助言を頂きました柳浦睦憲教授, 胡艶楠助教に深く感謝の意を表します. 提案手法の検討やその有用性において活発に議論を頂きました, 高田氏, ビトル氏に大変お世話になりました. 深くお礼申し上げます. 日々の研究室生活においては柳浦研究室の皆様にお世話になりました. 皆様のおかげで有意義な研究活動に勤しむことができました. 深くお礼申し上げます.

参考文献

[1] Kris Braekers, An Caris, and Gerrit K Janssens. Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots. Transportation Research Part B: Method- ological, Vol. 67, pp. 166-186, 2014

[2] Jean-François Cordeau and Gilbert Laporte. A Tabu search heuristic for the static multi-vehicle dial-a-ride problem. Transportation Research Part B: Methodological. 37. 579-594. 2003.

[3] Jang-Jei Jaw, Amedeo R Odoni, Harilaos N Psaraftis and Nigel H.M.Wilson A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. Transportation Research Part B: Methodological. Vol.20, No.3, pp. 243-245, 1986.

[4] David Pisinger and Stefan Ropke. A general heuristic for vehicle routing problems. Computers operations research, Vol. 34, No. 8, pp. 2403 - 2435, 2007

[5] Stefan Ropke and David Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transportation science, Vol. 40, No. 4, pp. 455 - 472, 2006.

[6] Paolo Detti, Framcesso Papalini, and Garazi Zabalo Manrique de Lara. A multi-depot dial-a-ride problem with heterogeneous vehicles and compatibility constrains in healthcare. omega, Vol. 70, No. 4, pp. 1-14, 2017.

[7] 柳浦睦憲, 茨木俊秀, 組合せ最適化 メタ戦略を中心として, 朝倉書店, 2001