Dial-a-Ride 問題

竹田 陽

2019 年 11 月 28 日

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, however, 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.

1 はじめに

近年, 乗合タクシーサービスは新しい移動手段として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．これらのサービスの特徴としては, 利用者はリクエストとして出発地と到着地, それぞれに対して場所と時間枠を指定することができる．本研究では, これらのような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ことを考える．このような問題は Dial-a-Ride 問題 (Dial-a-Ride problem, DARP) と呼ばれ, 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間と降車時間に対しての時間枠をハード制約として与えている．本研究では, 最大乗車時間と時間枠をペナルティ関数で与えてソフト制約とする．こうすることで, 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．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．

静的 DARP に関して, 先行研究において様々な手法が提案されている．Jaw らは, この問題に対して近似解法として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連続挿入法を提案した [2]．Cordeau らは, 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際にタブーサーチ探索を用いる手法を提案した [1].

2 問題説明

まず, Pickup and Delivery problem (PDP) について説明する．

2.1 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.

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.

2.2 Dial-a-Ride problem

Dial-a-Ride 問題 (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．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 [3] であることが知られているため, DARP も NP 困難である.

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．このように制約を与えることで, 少しの遅延を許容する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.

2.3 記号の定義と定式化

本問題は, $G = (V, E)$ 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. 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.

この時, 以下のように定式化できる:

$$\text{minimize} \quad \alpha d(\sigma) + \beta t(\sigma), \tag{1}$$

$$\text{subject to} \quad \sum_{i \in V} x_{ij} = 1 \qquad (j \in P \cup D), \tag{2}$$

$$\sum_{j \in V} x_{ij} = 1 \qquad (i \in P \cup D), \tag{3}$$

$$\sum_{k \in K} n_k = 2n, \tag{4}$$

$$\sigma_k(0) = \sigma_k(n_k + 1) = 0 \qquad (k \in K), \tag{5}$$

$$0 < q_{\sigma_k(h)} < Q \quad (h \in \{0, ..., n_k\}, k \in K), \tag{6}$$

$$g_i(\tau_{i+n} - \tau_i) > s_i + t_{i,i+n} \qquad (i \in I), \qquad (7)$$

$$\tau_{\sigma_k(h+1)} \geqq \tau_{\sigma_k(h)} + s_{\sigma_k(h)} + t_{\sigma_k(h),\sigma_k(h+1)} \quad (h \in \{0, ..., n_k\}, k \in K). \qquad (8)$$

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.

3 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反復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, 乗降時間と乗車時間に関してペナルティ関数の値が閾値以下であるかを判定し, 閾値以下であれば最適なサービス開始時刻を決定して評価する. サービス時刻を決定する問題は, 制約と目的関数が全て線形の式で表すことが可能であるため, 線形計画問題 (linear programming problem, LP) として定式化し, 解くことが可能である.

3.1 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.

3.2 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.

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.

3.3 サービス開始時刻の決定の有無

局所探索によって得られる全てのルートに対してサービス開始時刻を LP で計算するのは非効率である．そのため，現在の最良解よりも改善する可能性があるルートのみに対してサービス開始時刻を決定する．改善の可能性の判定は，以下の手順で行う．

Algorithm 1 ルートの改善可能性の判定

Input: $G = (V, E)$, 各頂点 i におけるペナルティ関数と閾値, 各リクエスト j の乗車時刻のペナルティ関数と閾値，ただし $V = \sigma_k \cup s$, $E = \sum_{h=0}^{n_k} t_{\sigma_{k(h)},\sigma_{k(h+1)}}$ である.

Output: s からの最長路の計算可能性

1: 各頂点 i に対して，ペナルティが閾値以下になるような時刻 l_i, u_i を計算
2: **for** $e = 1$ to $|n_k|$ **do**
3: 　s から $\sigma_k(e)$ に対して容量 l_i の辺を追加
4: 　$\sigma_k(e)$ から s 対して容量 $-u_i$ の辺を追加
5: **end for**
6: s から $\sigma_k(0)$,$\sigma_k(n_k + 1)$ に対して容量 0 の辺を追加
7: $\sigma_k(0)$, $\sigma_k(n_k + 1)$ から s に車両の最大使用時間の容量の辺を追加
8: 各リクエスト $j \in I_\sigma$ において，容量が乗車時間の閾値の負の値の辺をリクエストと逆向きに追加
9: 動的計画法で s からの最長路を計算
10: 最長路 s を計算可能なら 1, 不可能なら 0 を出力.

以上の手順を行うことで, グラフ G において正の閉路の存在を判断することができる. 動的計画法 (DP)

で計算を行っていき，最長路を計算することができるのならばグラフ内に正の閉路が存在しないことを示しており，乗降時間と乗車時間のペナルティが閾値以下であるような時刻の組み合わせが存在することを示している．反対に, 正の閉路が存在することは, サービス開始時刻もしくは乗車時間において閾値を超える部分が存在するということを示している.

動的計画法では, 変数として $opt(i,v)$ を用いる. $opt(i,v)$ は, 始点 s で頂点数 n のグラフ G における, たかだか i 本の辺を用いた $s-v$ パスの最大コストをあらわす. $n-1$ 本の辺を用いて s から全ての点への最大コストを計算した後, n 本の辺を用いて s から全ての点への最大コストを計算した際に, 新たに更新があった場合には正の閉路があると言えるので, 計算を終了する. 全ての頂点に関して $i=n-1$ と $i=n$ で更新がない場合のみ, 正の閉路は存在しない.

$opt(i,v)$ を計算するにあたって, 複数の状態を考える必要がある。以下の

1. $i-1$ 本以下の辺を用いる場合,
2. i 本の辺を用い，最初の辺が (v,w) である場合

の 2 種類が存在し, 1 の場合には

$$opt(i,v) = opt(i-1,v)$$

で表現でき，2 の場合は

$$opt(i,v) = t_{vw} + opt(i-1,w)$$

と表現することができる.

よって，以下のように定式化できる,

$i=0$ の場合,

$$opt(i,v) = \begin{cases} 0 & (v=s), \\ -\infty & (v \neq s), \end{cases}$$

$i>0$ の場合,

$$opt(i,v) = \max\{opt(i-1,v), \max_{w \in V}(opt(i-1,w)+)+t_{vw}\}$$

3.4 反復局所探索法

DARP に対する反復局所探索法に基づく解法を提案する. 概要を以下に示す.

初期解生成後，ルート内とルート間での局所探索を行い，ルートの改善可能性を判断し，改善可能性があればその後ルートを改善する．この局所探索を改善がなくなるまで行い，得られた解に対して kick という操作を行い，局所探索では探索されないような解を初期解として再び探索を行う.

今後，具体的な近傍操作や kick の方法について考察していく.

4 まとめと今後の課題

本論文では, ソフト時間制約付き DARP について説明した. 今後は, プログラムを実装し提案手法の有効性を確認する.

参考文献

[1] 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.

[2] 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. 243-2435, 1986.

[3] 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