

相乗りのための将来需要を考慮した経路最適化

大社 綾乃^{1,a)} 大滝 啓介¹ 小出 智士¹ 西 智樹¹

受付日 2019年1月24日, 採録日 2019年7月3日

概要: 都市化にともなう渋滞や環境汚染などの問題を解決するため, ライドシェアや相乗りによる移動の効率化が注目されている. 配車サービスにおける相乗りを実現するためには, 膨大な計算量をとまなう顧客と車両の割当ておよび経路生成が必要である. 従来研究により顧客と車両の割当てを効率良く行うアルゴリズムが提案されている. また, 将来需要の予測モデルをもとに生成した架空の顧客を同時に扱うことで将来需要を考慮することもできるが, 車両の総移動距離が大幅に増えるという問題がある. そこで本稿では, 架空の顧客を扱うのではなく, 実在する顧客のみを車両に割り当てた後, 将来需要を考慮した経路を生成する手法を提案する. タクシーの実データを用いた数値実験の結果, 提案手法は車両の移動距離をほとんど増加させずに, 乗車できない顧客数を減らすことができた.

キーワード: 相乗り, ライドシェア, 経路最適化, 割当て最適化, 将来需要

Path Optimization with Future Demand for Ride-sharing

AYANO OKOSO^{1,a)} KEISUKE OTAKI¹ SATOSHI KOIDE¹ TOMOKI NISHI¹

Received: January 24, 2019, Accepted: July 3, 2019

Abstract: Ridesharing services are promising to realize convenient transportation. To find optimal vehicle assignments and their routes is intractable in real city-wide services. Previous work has been developed the algorithm that finds them using possible combinations between customer requests and service vehicles. The approach can also consider future demand using dummy requests sampled with a prediction model in addition to current requests. However, vehicles using the method need to travel a longer distance due to the dummy requests. In this paper, we develop the method to find the detour routes with high-future demand, which satisfy the current requests, instead of using the dummy requests. We demonstrated that our method could reduce the rejection rate of the requests with little additional travels using a real-world taxi trip dataset.

Keywords: ride-sharing, optimal path planning, optimal assignment problem, future demand

1. はじめに

交通ネットワークや情報通信技術, 高精度な位置情報技術の普及にともない, Uber^{*1}や Lyft^{*2}に代表されるような Mobility-on-Demand (MoD) システムがめざましい発展を遂げている. MoD システムでは, 顧客は必要ときに最適なモビリティやサービスを利用することができる. たとえば Uber や Lyft では, アプリを用いて好きなタイミングで配車を依頼することができるだけでなく, 食べ物な

どの荷物の配送が可能なサービスもある. また自動運転技術が普及すれば, これまで運転することができなかった人も気軽に移動できるようになるだけでなく, 車内での過ごし方や保険制度のあり方が変わり, さらに新しいサービスが生まれると期待される [1]. 一方で, 自動運転車の普及にともない車両数が増加するという予測や, 都市化にともない渋滞や環境汚染が悪化するという予測もある [2]. そのようななか, 相乗りやライドシェアは, 複数人で同じ車両を利用することで個人で移動するよりも車両数を減らせるため, 渋滞や環境汚染の緩和への貢献が期待されている [3].

¹ 株式会社豊田中央研究所
Toyota Central R&D Labs., Inc., Bunkyo, Tokyo 112-0004, Japan

^{a)} okoso@mosk.tytlabs.co.jp

^{*1} <https://www.uber.com> (参照 2018-12-10)

^{*2} <https://www.lyft.com> (参照 2018-12-10)

本稿では、タクシーに代表される配車サービスの相乗りに焦点を当てる。

タクシーによる相乗りを実現するためには、顧客の出発地や目的地、希望到着時間などから、相乗りする顧客と車両を割り当て、経路を生成することが必要である。相乗りにより発生する遅延時間を抑えつつ、乗車できる顧客数を最大化することを目的とした相乗り最適化について Alonso-Mora らの研究がある [4]。Alonso-Mora らは、相乗り可能な顧客の組と車両をグラフ構造として構築し、最適に割り当てた後、最短経路で移動する手法を提案している。また、学習した将来需要の予測モデルをもとに生成した架空の顧客を車両に割り当てることで、将来需要も考慮した [5]。この手法は、架空の顧客へ車両を割り当てることで将来需要を考慮した経路を生成するが、経路生成時には架空の顧客と実在する顧客を区別しないため、生成された架空の顧客の位置や精度によっては、その顧客を迎えに行く経路を生成しても、その顧客が実際には存在しない可能性がある。その場合、その移動が損失となり車両の総移動距離が大幅に増加するという問題がある。

そこで我々は、将来需要を考慮し、乗車できない顧客数を減らしつつ、無駄な移動を減らすことを目的とした相乗りシステムを目指す。具体的には、実在する顧客のみで車両との割当て最適化を行った後、顧客の待ち時間や遅延時間の制約を満たす範囲で、期待される将来需要量が最大となるような経路で移動する。提案手法では、将来需要量が高いと期待される経路を生成して移動するため、最短経路で移動する場合より移動距離が増える可能性はあるが、移動中に他の顧客と相乗りできる可能性が高まる。また、実在する顧客のみで割当て最適化を行うため、将来需要量に基づく架空の顧客を用いる従来手法 [5] と比較し、余計な移動距離を減らすことができると期待される。

提案手法の有効性を検証するため、ニューヨーク市マンハッタン島のタクシーの実データを用いて、様々な需要比率と車両台数の環境下で数値実験を行う。数値実験では、提案手法を用いることで、将来需要を考慮しない従来手法 [4] より乗車できる顧客数を増やすことができることを確認する。また、将来需要を考慮した従来手法 [5] より車両の総移動距離が短くなることを確認する。

本稿の構成は、まず 2 章で関連研究について述べ、3 章で本稿で扱う相乗りシステムについて説明する。4 章で提案手法について述べた後、提案手法の有効性を検証するためにに行った数値実験について 5 章で議論し、最後に、全体のまとめと今後の課題を 6 章で述べる。

2. 関連研究

複数の顧客のリクエスト情報から全員の要求を満たす車両の移動経路を生成する問題として、Dial-A-Ride-Problem (DARP) がある [6]。DARP では顧客のリクエスト情報が

既知であることが前提だが、タクシーなどの相乗りの場合、顧客は必要ときに配車を依頼するため、リアルタイムに更新する顧客リクエストや車両情報をもとに割当ておよび経路生成を行う必要がある [7]。Lalos ら [8] は、タクシーによる相乗りシステムを実社会で実現するために必要な要件をあげている。タクシーの相乗りのための顧客や車両に関する制約条件は多岐にわたり、顧客の出発や到着に関する時間制約を扱う研究 [4], [9], [10] や、顧客のコミュニティや快適性を考慮した研究 [11], [12] もある。我々は従来研究 [4], [10] と同様の顧客の時間制約を扱う。

相乗りを最適化する際、乗降車する顧客の順序を決定する必要がある。最適解を得るためには全通りの組合せについて調べる必要があるが、その計算コストは膨大になり現実的な時間で解くことが難しい。そこで、新たな顧客を現在の経路に挿入する方法 [13] や、探索範囲を限定する方法 [14] などがあり、本稿で扱う相乗り最適化では、相乗りする顧客数が少ない場合には全通りの組合せを調べ、多い場合に前者の方法を用いる。

将来需要を考慮した経路生成に関して、Alonso-Mora らの研究 [5] では、予測モデルに基づく架空の顧客を車両に割り当てることで、間接的に将来需要を考慮した経路を生成している。Lin ら [15] は、ある地点からある地点に向かう顧客が発生する確率をもとに経路を生成しているが、運転者やタクシー会社の収入が最大になることを目的としているため、我々の相乗り最適化の問題設定とは異なる。

3. 相乗りシステム

本章では、我々が扱う相乗りシステムについて説明する。

3.1 概要

対象とする相乗りシステムの概要を図 1 に示す。システムは、一定時間ごとに顧客の配車リクエストと車両の位置および乗客の情報を受け取り、車両の割当ておよび経路

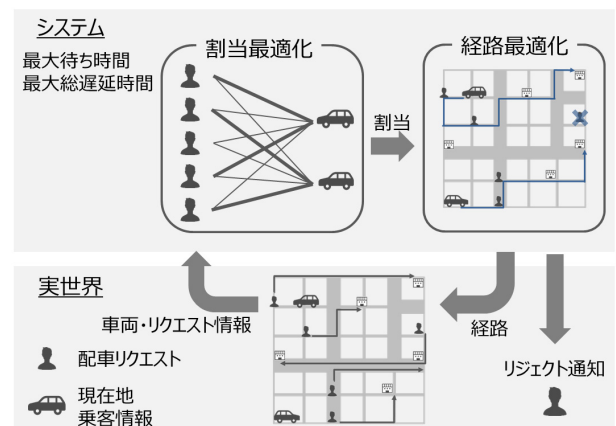


図 1 相乗りシステム

Fig. 1 Schematic overview of our ride-sharing system.

最適化を行った後、生成された経路を車両へ送る．ここでは、あるリクエストを持つ顧客のことを単にリクエストと呼び、乗車して移動している顧客を特に乗客と呼ぶ．具体的には、下記の3つの手順により経路を生成する．1) まず、顧客から配車リクエストを、車両から現在の位置と乗客の情報を受け取る．2) 受け取った情報に基づき、乗客とリクエストの総遅延時間および、どの車両にも割り当てられないリクエスト数を、可能な限り小さく抑えるような車両への顧客の割当てと、乗降車順序を決定する．3) 割当てと乗降車順序に基づき、車両の経路を生成する．2) を割当て最適化、3) を経路最適化と呼び、これらを合わせて相乗り最適化と呼ぶ．システム内では、最大待ち時間および最大総遅延時間が設定されており、それらの制約を満たす必要がある．待ち時間とは、顧客からリクエストを受け取ってから、その顧客が乗車するまでの時間を表す．総遅延時間とは、顧客が相乗りをせずに最短で移動した場合と比較し、相乗りにより生じた移動の遅延時間と待ち時間を合計したものである．なお、最大待ち時間を超えたリクエストについては、システムはその顧客に乗車できないことを通知する．これをリジェクトと呼ぶ．また、車両が空間的に偏ったことによるリジェクトの増加を抑えるために、空車両は空間的に平準化された位置に再配置される．空車両の平準化については、本稿では割愛する．

提案手法は、車両の顧客への割当てと乗降車順序を変更しない範囲で、将来リクエストが発生しやすい経路を生成する手法であり、割当て最適化については従来手法 [4] を用いる．詳細な問題設定は、4.1 節で述べる．

3.2 システムの処理の具体例

相乗りシステム内の処理の詳細について、図 2 を用いて具体的に説明する．人のアイコンは顧客（黒：リクエスト、青：乗客）を、車のアイコンは車両を表す．顧客から伸びる矢印（黒）は顧客の乗降車位置を意味し、車両から伸びる破線（橙）および実線（橙）は、それぞれシステムが生成した経路と実際に車両が移動した経路を表す．具体的な処理の流れを示すために、最大乗車可能人数が2名の車両2台 v_0 , v_1 を配車することができるシステムについて、

顧客から、時刻 $t = 0$ でリクエスト r_0 , r_1 , r_2 を、時刻 $t = 1$ で r_3 を受け取る場合を考える．ただし、顧客の最大待ち時間を2ステップとする．

時刻 $t = 0$: システムは、顧客からのリクエスト r_0 , r_1 , r_2 と車両の位置および乗客の情報を受け取る． v_0 は空車状態、 v_1 は乗客2名を乗せているとする．システムは、割当て最適化により決定された車両 v_0 のリクエスト r_1 , r_2 への割当てと各リクエストの乗降車順序に基づき、経路最適化により車両の移動経路を生成し、車両 v_0 に送信する．車両 v_1 は目的地到着直前の乗客2名を乗せた満車状態であるため、どのリクエストも割り当てられない．リクエスト r_0 は、最大待ち時間2ステップまでに迎えに行くことのできる車両が存在しないため、どの車両も割り当てられない．

時刻 $t = 1$: 顧客からの新たなリクエスト r_3 と車両の位置および乗客の情報を受け取る．車両 v_1 は乗客2名を降ろしたため、空車状態である．システムは、割当て最適化により車両 v_1 にリクエスト r_2 , r_3 を割り当て、乗降車順序を決定し、経路最適化により車両の移動経路を生成し、車両 v_1 に送信する．リクエスト r_2 については、車両 v_1 を配車するよりも短い待ち時間で乗車させることができるため、車両 v_1 に割当てが変更される．車両 v_0 は乗客として r_1 を乗せているが、 r_2 へ配車する必要がなくなったため経路が変更される．なお、乗客は車両の割当てが変更されない．

時刻 $t = 2$: 顧客からのリクエストと車両の位置および乗客の情報を受け取る．ここでは顧客からのリクエストはないが、最大待ち時間2ステップまでに r_0 へ配車することができなかったため、 r_0 にリジェクトを通知する．車両 v_0 は乗客 r_1 を車両 v_1 は乗客 r_2 , r_3 を乗せている．

以上のように、システムは、一定時間ごとに収集した顧客のリクエストと車両情報および、一時刻前までに収集したまだ乗車していない顧客のリクエストに基づき、相乗り最適化により経路を生成し、その情報を車両に送信する．また、最大待ち時間までに乗車させることができなかった場合には、顧客に乗車できないことを通知する．割当て最適化および経路最適化の詳細については、4 章で説明する．

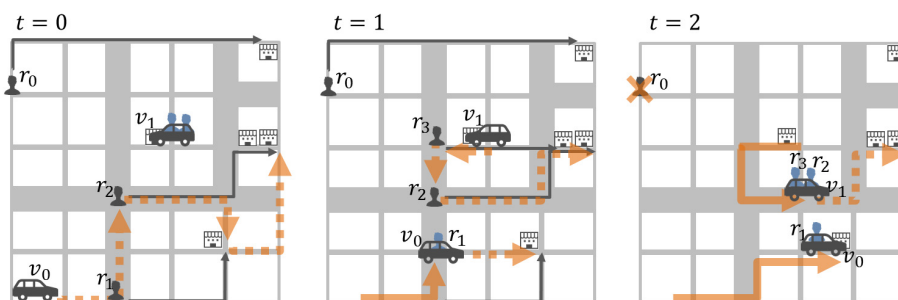


図 2 相乗りシステムの挙動例

Fig. 2 Schematic example of our ride-sharing system behavior.

4. 提案手法

相乗り最適化は、顧客と車両の割当て最適化と車両の経路最適化で構成され、割当て最適化については従来手法 [4] を用い、経路最適化に関して将来需要を考慮した経路の生成手法を提案する。後節では、相乗り最適化の問題設定および割当て最適化について説明する。その後、本稿で提案する将来需要を考慮した経路生成手法について述べ、従来手法との違いについて説明する。

4.1 問題設定

最大乗車可能人数が ν の m 台の車両 $\mathcal{V} = \{v_1, \dots, v_m\}$ は、重み付き有向グラフ $G = (\mathbb{V}, \mathbb{E}, w)$ によってモデル化された道路ネットワーク上を移動する。ここで、 $\mathbb{V}$ はノードの集合、 $\mathbb{E} \subseteq \mathbb{V} \times \mathbb{V}$ はエッジの集合、 w はエッジの重みを表す。相乗りシステムは、一定時間ごとに受け取る車両の情報と顧客の配車リクエストおよび、一時刻前までのまだ乗車していないリクエストに基づいて配車する。システムは、車両 v の現在地 q_v と現在の乗車人数を n_v^{pass} としたときの乗客 $\mathcal{P}_v = \{p_1, \dots, p_{n_v^{pass}}\}$ の情報を受け取る。システムが処理すべきリクエスト集合を $\mathcal{R} = \{r_1, r_2, \dots, r_n\}$ と表し、リクエストが割り当てられた車両に乗車すると、 $\mathcal{R}$ からそのリクエストを削除する。

各顧客はシステムに対し、配車リクエスト $\hat{r}$ として乗降車位置 $(o_{\hat{r}}, d_{\hat{r}}) \in \mathbb{V} \times \mathbb{V}$ を送信する。システムは、受け取った $\hat{r}$ に、顧客が配車リクエストを送信した時刻 $t_{\hat{r}}^*$ および最短可能到着時刻 $t_{\hat{r}}^*$ を付加したものをリクエスト r として扱う。最短可能到着時刻 t_r^* とは、 o_r から d_r への最短経路での移動時間を $\tau(o_r, d_r)$ とすると、 $t_r^* = t_{\hat{r}}^* + \tau(o_r, d_r)$ で表すことができる。

システムは、割当て最適化により車両へのリクエストの割当てと各リクエストの乗降車順序を最適化し、経路最適化により割当てと乗降車順序を変えない範囲で経路を最適化する。割り当てられたリクエストを乗降車順に最短経路で移動した際の乗車時刻を t_r^p 、降車時刻を t_r^d としたとき、 t_r^p 、 t_r^d および車両 v は下記の制約を満たす必要がある。

$$t_r^p \leq t_r^* + \Omega \quad (\forall r \in \mathcal{R}), \quad (1)$$

$$t_r^d \leq t_r^* + \Delta \quad (\forall r \in \mathcal{R}), \quad (2)$$

$$n_v^{pass} \leq \nu \quad (\forall v \in \mathcal{V}). \quad (3)$$

Ω は最大待ち時間、 Δ は最大総遅延時間を表し、システムが設定する全リクエスト共通の値である*³。待ち時間は $t_r^p - t_r^*$ に相当し、総遅延時間は、最短で移動した場合と比較して相乗りにより発生した遅延時間と待ち時間の合計を意味し、 $t_r^d - t_r^*$ に相当する。式 (1) は最大待ち時間までに

乗車すること*⁴、式 (2) は最大総遅延時間までに到着すること、式 (3) は各車両の乗客人数が最大乗車可能人数以下であることを表す。これらを制約条件 $\mathcal{Z}$ とする。

4.2 割当て最適化

本節では、提案手法で用いた Alonso-Mora らの研究 [4] の割当て最適化について述べる。割当て最適化は、車両情報とリクエスト情報を入力とし、各車両に対して割り当てられたリクエストの集合および、車両 v がリクエストを乗降車させる順序 $Order_v$ を出力とする。乗降車順序 $Order_v$ は、現在地と乗客および割り当てられたリクエストの乗降車地点のリストからなる。たとえば図 2 の $t = 1$ を例にあげると、車両 v_0 に割り当てられたリクエストの集合は $\{r_1\}$ 、 $Order_{v_0} = \langle q_{v_0}, d_{r_1} \rangle$ となり、車両 v_1 に割り当てられたリクエストの集合は $\{r_2, r_3\}$ 、 $Order_{v_1} = \langle q_{v_1}, o_{r_3}, o_{r_2}, d_{r_3}, d_{r_2} \rangle$ となる。

割当て最適化における目的は、顧客の総遅延時間の和および、どの車両にも割り当てられないリクエスト数を、可能な限り小さく抑えるように車両と顧客を割り当てることである。これを達成するため、割当てのコスト関数 $\mathcal{C}$ として、

$$\mathcal{C} = \sum_{v \in \mathcal{V}} \sum_{r \in \mathcal{P}_v} \delta_r + \sum_{r \in \mathcal{R}_{ok}} \delta_r + \sum_{r \in \mathcal{R}_{ko}} c_{ko}, \quad (4)$$

を定義する。 $\delta_r = t_r^d - t_r^*$ であり、割り当てられたリクエスト集合を $\mathcal{R}_{ok}$ 、割り当てられなかったリクエスト集合を $\mathcal{R}_{ko}$ 、割り当てられないことに対するペナルティを定数 c_{ko} とする。式 (4) の第 1 項は乗客の総遅延時間の和、第 2 項は割り当てられたリクエストの総遅延時間の和、第 3 項は車両に割り当てられないことに対するペナルティを表し、定数 c_{ko} は十分に大きな値とする。このとき、リクエスト r や車両 v について制約条件 $\mathcal{Z}$ を満たす必要がある。この割当て最適化は整数線形計画問題として解くことができる。詳細は従来研究 [4] に記述されている。

4.3 将来需要を考慮した経路最適化

経路最適化では、割当て最適化により出力された $Order_v \quad \forall v \in \mathcal{V}$ を入力とし、各車両の経路を出力とする。将来需要を考慮する際、車両は、重み $w = (\beta, \rho)$ が関連付いた重み付き有向グラフ $G = (\mathbb{V}, \mathbb{E}, w)$ によってモデル化された道路ネットワーク上を移動すると仮定する。 $\beta, \rho: \mathbb{E} \rightarrow \mathbb{R}$ はエッジの移動時間と将来需要量を表す。将来需要量とは、現時刻からある時刻までの予測リクエスト発生量を意味する。将来需要量の予測モデルについては問

*³ 提案手法は、 Ω および Δ の値をリクエストごとに設定しても対応可能であるが、本稿では簡易化のため共通の値とした。

*⁴ 実際には、前時刻での予想乗車時刻より乗車時刻が遅くなると予想される場合は、そのリクエストは別の車両に割当てを変更することができないとしている。現時刻での予想到着時刻を $t_r^p(t)$ 、前時刻での予想到着時刻を $t_r^p(t-1)$ とすると、 $t_r^p(t) \leq t_r^p(t-1)$ を満たす必要がある。詳細は文献 [4] を参照。

わないため、本稿では説明を割愛する．2 頂点 $s, g \in \mathbb{V}$ 間の経路 $Path_{s,g}$ を頂点の列 $Path_{s,g} = \langle v_0, \dots, v_k \rangle$ ($v_0 = s$, $v_k = g$, $(v_i, v_{i+1}) \in \mathbb{E}$, $0 \leq i \leq k-1$) で表す．また経路 $Path$ の将来需要量を、経路を構成するエッジの将来需要量の和で定義する．顧客の待ち時間および総遅延時間に関する制約条件 (式 (1), (2)) を満たしつつ、将来需要量の和が最大となる経路を最適な経路 $Path^{opt}$ と呼ぶ．

我々の手法は、車両 v について、 v の現在地 q_v から $Order_v$ に従って、最適な経路 $Path^{opt}$ の近似解となる経路 $Path^*$ を生成する．乗降車地点を $Order_v = \langle l_0, l_1, \dots, l_i \rangle$ ($l_0 = q_v$) とすると、各地点間の経路 $Path_{l_k, l_{k+1}}^*$ ($0 \leq k \leq i-1$) を生成し、結合した $Path_{l_0, l_i}^*$ を出力する．

経路 $Path_{l_k, l_{k+1}}^*$ を求めるため、定数 $\lambda_v \geq 0$ を用いた重み関数 $\mathcal{C}_{\lambda_v} : \mathbb{E} \rightarrow \mathbb{R}$ を $\mathcal{C}_{\lambda_v}(e) := \beta(e) - \lambda_v \rho(e)$ で定義し、グラフ $G_{\lambda_v} := G(\mathbb{V}, \mathbb{E}, \mathcal{C}_{\lambda_v})$ を新しい重み付きグラフと定義する． $Path_{l_k, l_{k+1}}^{\lambda_v} := \operatorname{argmin}_{e \in Path_{l_k, l_{k+1}}} \mathcal{C}_{\lambda_v}(e)$ としたとき、 λ_v に関してコスト最小となる経路を近似最適経路、すなわち $Path_{l_k, l_{k+1}}^* := \operatorname{argmin}_{\lambda_v \geq 0} \mathcal{C}_{\lambda_v}(Path_{l_k, l_{k+1}}^{\lambda_v})$ と定義する．ただし求める λ_v の値は、制約条件 (式 (1), (2)) を満たすものとする．なお、 $\lambda_v = 0$ のとき、 $\mathcal{C}_{\lambda_v}(e) = \beta(e)$ となり、 $Path_{l_k, l_{k+1}}^*$ は最短経路と一致し、 λ_v の値が大きいくほど将来需要量が考慮される．

制約条件 (式 (1), (2)) を満たすか判定するためには、車両 v に割り当てられた全顧客の乗降車および全乗客の降車時刻について調べる必要がある．図 2 の時刻 $t = 1$ における車両 v_1 について、 $Order_{v_1} = \langle q_{v_1}, o_{r_3}, o_{r_2}, d_{r_3}, d_{r_2} \rangle$ の経路生成を例に説明する．(q_{v_1}, o_{r_3}) 間の生成された経路 $Path_{q_{v_1}, o_{r_3}}$ が、顧客の制約条件 (式 (1), (2)) を満たすかを判定する際には、図 3 で示すように、 o_{r_3} 以降の乗降車が最短経路で移動すると仮定する．具体的には、現時刻を t_0 とすると、以下のすべての制約条件を満たす場合に、顧客の制約条件を見たす経路であるといえる．

$$\begin{aligned} t_0 + \beta(Path_{q_{v_1}, o_{r_3}}) &\leq t_{r_3}^r + \Omega, \\ t_0 + \beta(Path_{q_{v_1}, o_{r_3}}) + \tau(o_{r_3}, o_{r_2}) &\leq t_{r_2}^r + \Omega, \\ t_0 + \beta(Path_{q_{v_1}, o_{r_3}}) + \tau(o_{r_3}, o_{r_2}) + \tau(o_{r_2}, d_{r_3}) &\leq t_{r_3}^* + \Delta, \\ t_0 + \beta(Path_{q_{v_1}, o_{r_3}}) + \tau(o_{r_3}, o_{r_2}) + \tau(o_{r_2}, d_{r_3}) \\ &\quad + \tau(d_{r_3}, d_{r_2}) \leq t_{r_2}^* + \Delta. \end{aligned} \quad (5)$$

上記の式 (5) を除く各不等式において、右辺と左辺の差分がその区間での迂回可能な時間といえる．提案手法では、

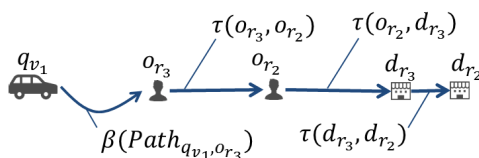


図 3 提案手法における経路生成の例

Fig. 3 Overview of our routing method.

経路 $Path_{l_k, l_{k+1}}^*$ を $k = 0$ から順番に生成し、そのつど車両に割り当てられた全顧客の制約条件を確認する必要があるが、実際には、ある経路 $Path_{l_k, l_{k+1}}^*$ で迂回可能な時間は、 $Path_{l_j, l_{j+1}}^*$ ($k \leq j \leq i-1$) での迂回可能な時間の最小値となる．つまり、ある経路 $Path_{l_k, l_{k+1}}^*$ で迂回可能な時間を $dt_{l_k, l_{k+1}}$ とすると、 $dt_{l_k, l_{k+1}} = \min\{dt_{l_j, l_{j+1}} | k \leq j \leq i-1\}$ となる． $Path_{l_k, l_{k+1}}^*$ が顧客の制約条件を満たすかは、 $\beta(Path_{l_k, l_{k+1}}) \leq \tau(l_k, l_{k+1}) + dt_{l_k, l_{k+1}}$ が成り立つかで判定することができる．

ある車両 v における経路 $Path_{l_0, l_i}^*$ を求めるアルゴリズムを Algorithm 1 に示す．適切な探索範囲 $0 \leq \lambda_v \leq \lambda_{max}$ について、 $\mathcal{C}_{\lambda_v}(Path_{l_k, l_{k+1}})$ を最小化する λ_v の値を二分探索法により求め、 $Path_{l_k, l_{k+1}}^*$ を決定し (3–13 行目)、経路を結合する (14–15 行目) ことを、各地点間で行う．二分探索法を用いるのは、以下 2 点の性質を持つためである．
1) $\mathcal{C}_{\lambda_v}(Path_{l_k, l_{k+1}}^{\lambda_v})$ は、 λ_v の値の増加に従って減少する．
2) λ_v の値が一定以上大きいと $\mathcal{C}_{\lambda_v}$ が負の値を持ち、解を得られない場合がある． λ_v の値を小さくしていくとどこか 1 点 λ_v^* で解を得ることができ、かつ $\lambda_v \leq \lambda_v^*$ の範囲で必ず解を得ることができる．

なお、 $\text{ShortestPath}(G(\mathbb{V}, \mathbb{E}, \mathcal{C}_{\lambda_v}), l_k, l_{k+1})$ は、エッジコスト $\mathcal{C}_{\lambda_v}$ を持つグラフ G 上のコスト最小経路を返す関数である． $\mathcal{C}_{\lambda_v}(e)$ の値により、エッジコストが負の場合でも解が得られる Bellman Ford 法 [16] とエッジコストが非負に限られるが計算コストが小さい Dijkstra 法 [17] を使い分ける．

4.4 従来手法との違い

提案手法および従来手法により生成される経路の様子を図 4 に示す．道路ネットワークの道の太さは、将来の需要量を表す．破線 (青) は最短経路 (従来手法 [4])、破線 (橙) は従来手法 [5] による将来需要を考慮した経路、実線

Algorithm 1 Path Generation with Future Demand

Input: $Order_v \leftarrow \langle l_0, \dots, l_i \rangle$

Output: $Path_{l_0, l_i}^*$

```

1:  $Path_{l_0, l_0} \leftarrow \emptyset$ 
2: for  $k \leftarrow 0$  to  $i-1$  do
3:   [二分探索法による  $\lambda_v$  の探索]
4:    $low \leftarrow 0, high \leftarrow \lambda_{max}$ 
5:   while  $low \leq high$  do
6:      $mid \leftarrow \lfloor (low + high)/2 \rfloor$ 
7:      $\lambda_v \leftarrow mid$ 
8:      $Path_{l_k, l_{k+1}} \leftarrow \text{ShortestPath}(G(\mathbb{V}, \mathbb{E}, \mathcal{C}_{\lambda_v}), l_k, l_{k+1})$ 
9:     if  $\beta(Path_{l_k, l_{k+1}}) \leq \tau(l_k, l_{k+1}) + dt_{l_k, l_{k+1}}$  then
10:       $low \leftarrow mid + 1$ 
11:     else
12:       $high \leftarrow mid - 1$ 
13:    $Path_{l_k, l_{k+1}}^* \leftarrow Path_{l_k, l_{k+1}}$ 
14:   [経路の結合]
15:  $Path_{l_0, l_i}^* \leftarrow Path_{l_0, l_k}^* + Path_{l_k, l_i}^*$ 

```

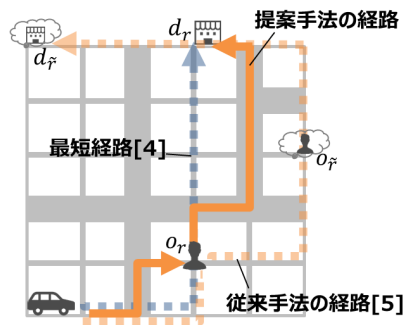


図 4 提案手法および従来手法による経路の例

Fig. 4 Example paths computed by our method and the previous methods [4], [5].

(橙)は提案手法による将来需要を考慮した経路を示している。

従来手法 [4] は、割当て最適化によって割り当てられた車両の現在地と顧客の乗降車位置を最短でつないだ経路を生成する。この手法は、現在の顧客に対してのみ最適化しているため、将来需要を考慮することができない。従来手法 [5] は、従来手法 [4] と割当て最適化の入力が異なり、現在の顧客の配車リクエストに加え、将来需要の予測モデルに基づきサンプリングした架空の顧客 (図 4 中の $\tilde{r}$) の配車リクエストに対して割当て最適化を行い、割り当てられた車両と顧客の乗降車位置を最短でつなぐ経路を生成する。

ここで、実在する顧客リクエスト集合および架空の顧客リクエスト集合を $\mathcal{R}^{now}$, $\mathcal{R}^{future}$ とし、それぞれのリクエストが割り当てられないことに対するペナルティをそれぞれ c_{ko}^{now} , c_{ko}^{future} とすると、割当て最適化の目的関数である式 (4) は、

$$C = \sum_{v \in V} \sum_{r \in \mathcal{P}_v} \delta_r + \sum_{r \in \mathcal{R}_{ok}} \delta_r + \sum_{r \in \mathcal{R}_{ko}^{now}} c_{ko}^{now} + \sum_{r \in \mathcal{R}_{ko}^{future}} c_{ko}^{future},$$

と書き換えることができる。ただし $\mathcal{R}^{now} \cup \mathcal{R}^{future} = \mathcal{R}$ であり、実在する顧客を優先的に割り当てるため、 $c_{ko}^{now} \gg c_{ko}^{future}$ とする。従来手法 [5] では、架空の顧客への配車により間接的に将来需要を考慮した経路を生成していることになるが、経路最適化時には架空の顧客と実在する顧客を区別しないため、サンプリングされた架空の顧客の位置によっては、大幅に遠回りする可能性がある。

そこで提案手法では、割当て最適化は実在する顧客に対してを行い、経路最適化時に将来顧客が現れそうな経路を優先的に通るような迂回路を生成する。具体的には、割当て最適化で計算された割当てと乗降車順序を変更せず、顧客の待ち時間および総遅延時間の制約を満たす範囲で、将来需要量の和が最大となる経路を生成する。

図 4 では、提案手法による経路は、将来の需要量が多い (太い) 道を優先的に移動し、最短経路より移動量が増

えているが、移動中に他の顧客と相乗りできる可能性が高まると期待される。また、実在する顧客のみで割当て最適化するため、従来手法 [5] による経路と比較し、移動距離を減らすことができると考えられる。その結果、提案手法は、従来手法 [5] より少ない移動距離で、従来手法 [4] で移動した場合よりも多くの顧客へ配車できると期待される。

5. 数値実験

本章では、提案手法の有効性を検証するため、タクシーの実データを用いて従来手法 [4], [5] と性能を比較した数値実験について述べる。

5.1 実験設定

5.1.1 リクエストデータ

本実験では、オープンデータであるニューヨーク市マンハッタン島のタクシーのデータ^{*5}をリクエストデータとして用いた。リクエストデータには、乗降車の位置と時刻、乗客数、移動距離、料金などが含まれる。本実験では、乗降車位置を配車リクエスト $\hat{r}$ として用い、乗車時刻をリクエスト時刻 t_r 、各リクエストで乗車する人数を 1 名と仮定した。乗車位置および降車位置はタクシーの GPS により得られる緯度、経度からなるため、後述する地図から抽出したグラフ G 上で最近傍のノードに割り当て、ノード上で乗降車すると仮定した。また、乗降車地点が同じリクエスト、経路が存在しないリクエスト、移動時間が 1 分未満のリクエストは除いた。本実験では、2013 年 5 月 3 日 (金) 08:30~09:00 の 30 分間のリクエストデータから、タクシー会社が VTC とラベル付されたデータのみを抽出し、6,107 件のリクエストを作成した。

5.1.2 地図情報

オープンデータである OpenStreetMap^{*6}を用いた。地図情報は、地点を表すノードと、ノード間をつなぐエッジからなる有向グラフで表現されている。OpenStreetMap では、エッジに様々な種類の道路や小道などを識別するためのキーやその距離の値が付加されている。本実験では、車両が通ることのできる道を抽出するため、“primary”, “secondary”, “tertiary”, “residential”, “unclassified”, “road”, “living street” のキーを持つエッジのみを用いた。そのため、本実験で用いたマンハッタン島のグラフは、ノード数 4,247, エッジ数 9,103 であった。

5.1.3 実験条件

1 分ごとのリクエストデータを集め、リクエストと車両の割当ておよび経路生成を行い、車両位置を更新することを繰り返した。実験条件を表 1 に示す。全車両の速度は

^{*5} Brian, D. and Dan, W.: New York City Taxi Trip Data (2010-2013), <https://databank.illinois.edu/datasets/IDB-9610843> (参照 2018-12-10)

^{*6} <https://www.openstreetmap.org> (参照 2018-12-10)

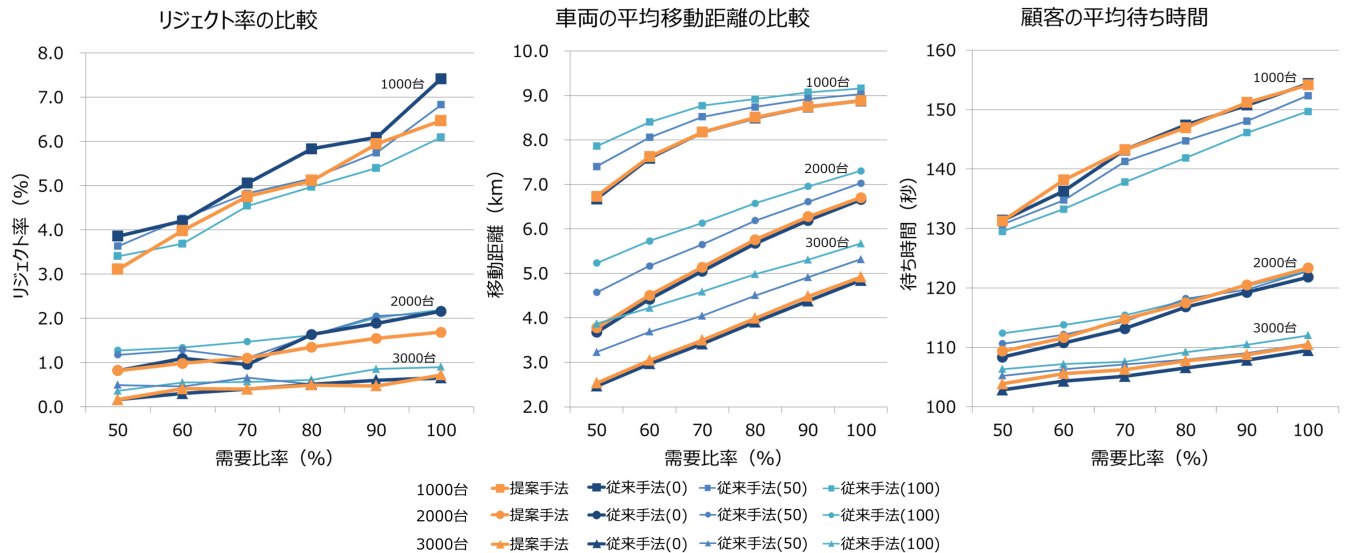


図 5 数値実験による結果 (左: リジェクト率, 中央: 平均移動距離, 右: 平均待ち時間)

Fig. 5 Comparison of several performance metrics. Left: the rejection rate. Center: the average travel distance. Right: the average waiting time.

表 1 実験条件

Table 1 Experiment setting.

パラメータ	値
最大遅延時間 Δ [min.]	8.0
最大待ち時間 Ω [min.]	5.0
車両の速度 [m/s]	5.5
車両台数 [台]	1,000, 2,000, 3,000
最大乗車可能人数 ν [人]	4
需要比率 [%]	50, 60, 70, 80, 90, 100

5.5 [m/s] とし^{*7}, 最大待ち時間 Ω および最大遅延時間 Δ は, 従来研究 [4] を参考にし, それぞれ 5.0 分, 8.0 分とした^{*8}. なお, 様々な需要量の環境における性能を比較するため, 1 分ごとのリクエストデータについて 50%~100% の割合でランダムにサンプリングした. この割合を本稿では需要比率と呼び, 需要比率 100% のときは 6,107 件すべてのリクエストデータを用いた. 車両の初期位置は, 需要比率が 100% の場合のリクエストデータの時系列順の先頭から, 車両台数分のリクエストの乗車位置を用いた.

従来手法 [5] および提案手法では, 10 分単位のリクエストデータを既知の将来需要として用いた. まず, 本実験で用いるリクエストデータを 10 分ごとに区切り, 従来手法 [5] では, 現在時刻を含む 10 分間のリクエストデータから, 0, 50, 100 リクエストをランダムにサンプリングし,

架空の顧客リクエストとして用いた. なお, サンプリング数が 0 の場合は従来手法 [4] に相当する. この架空リクエストは, 時刻が更新されるたびに新たにサンプリングした. なお, 実在する顧客および架空の顧客が割り当てられないことに対するペナルティの値 c_{ko}^{now} , c_{ko}^{future} を, それぞれ 10^6 , 10^3 とした. 提案手法では, 10 分ごとに区切られたデータから各ノードのリクエスト数をカウントし, n_1 から n_2 へ向かうエッジ e の将来需要量 $\rho(e)$ として, n_2 のリクエスト数を用いた.

5.1.4 評価項目

提案手法と従来手法 [4], [5] の性能を比較するため, リジェクト率, 車両の平均移動距離, 顧客の平均待ち時間を評価する. リジェクト率とは, 全顧客に対しリジェクトされた顧客数の割合を意味する. 車両の平均移動距離は, 全車両の移動距離を車両台数で割ることで算出する. 顧客の平均待ち時間は, リクエスト時刻から実際に乗車するまでの平均時間, つまり $\frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} (t_r^p - t_r^e)$ を意味する. なお, リジェクトされた顧客の待ち時間は Ω とした.

5.1.5 実験結果

様々な需要比率環境下で車両台数を変化させたときの提案手法および従来手法の性能を比較したグラフを図 5 に示す. 図 (左) がリジェクト率, 図 (中央) が各車両の総移動距離, 図 (右) が平均待ち時間の比較を表す. 横軸はすべて需要比率を, 従来手法の後の数字はランダムにサンプリングした架空の顧客数を表し, 0 の場合は従来手法 [4], それ以外は従来手法 [5] を意味する.

まずリジェクト率を比較する. 車両台数が 1,000 台の場合, 従来手法に着目すると, サンプリングする架空の顧客数の増加にともないリジェクト率が低下し, 適切に将来需要が考慮できていることが分かる. 一方提案手法は, 従来

*7 ラッシュ時 (08:00~13:00) のタクシーの平均速度 [10]

*8 2018 Factbook, <https://www1.nyc.gov/assets/tlc/downloads/pdf/2018-tlc-factbook.pdf> (参照 2019-04-10) によると, ニューヨーク市の yellow cab を利用する顧客の移動距離の多くは 3.0 マイル以下であり, その平均距離は 3.7 マイル (約 6.0 km) である. 車速を 5.5 [m/s] とすると, 平均乗車時間は 18.2 分となる. 本実験で使用した最大待ち時間 Ω は, 平均乗車時間の約 27% に相当する. Ω および Δ の妥当な値については今後検討の余地がある.

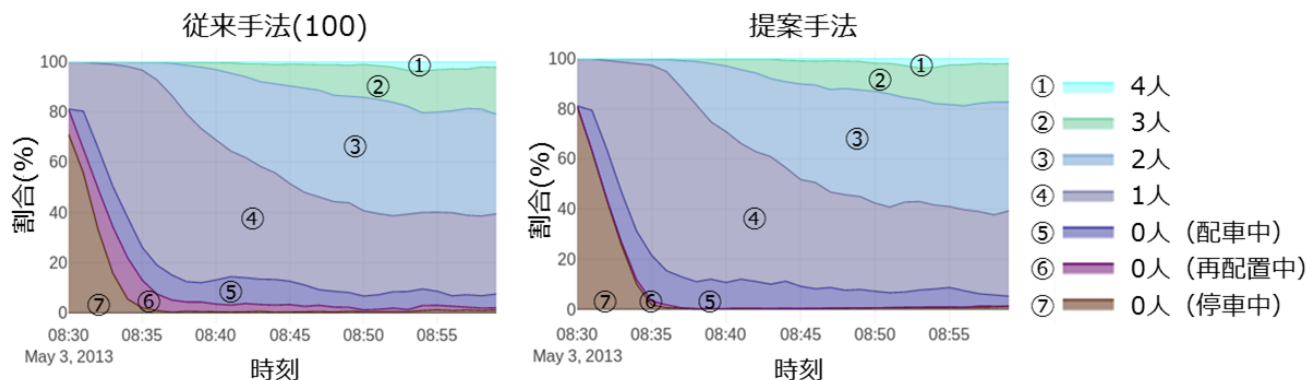


図 6 各時刻における乗車人数の割合 (左：従来手法 (100)，右：提案手法)

Fig. 6 The ratio of the number of passengers. Left: previous method [5]. Right: our method.

手法 (100) には劣る結果となったが、将来需要をまったく考慮しない従来手法 (0) と比べると低いリジェクト率を実現できている。車両台数が 2,000 台の場合は、需要比率が 70% の場合を除いて、提案手法が最も低いリジェクト率を実現できている。車両台数が 3,000 台の場合は、どの手法も大差なくリジェクト率が 1.0% 以下であり、車両台数が十分であったことが分かる。

次に各車両の平均移動距離を比較する。どの車両台数および需要比率であっても、従来手法ではサンプリングする架空の顧客数の増加に従い、移動距離が増加している一方、提案手法は従来手法 (0) と同等である。以上より、需要量に対して車両台数が十分でない場合、提案手法は従来手法 (0) と同等の移動距離で、より低いリジェクト率を実現できることが確認できた。

最後に顧客の平均待ち時間を比較する。すべての車両台数および需要比率において待ち時間の最大値と最小値の差は最大 6 秒以下であり、待ち時間にあまり影響を及ぼしていないといえる。しかし車両台数 1,000 台の場合、サンプリングする架空の顧客数の増加に従い待ち時間が減少しているため、サンプリング数をより増やすことでより短い待ち時間が期待される。提案手法では、従来手法 (0) と大差ない結果となった。提案手法における経路最適化によって生成される経路は、以下の 2 種に分類することができる。

- (1) 各顧客の乗降車地点を最短距離経路でつないだ経路であり、かつ将来需要量が高い経路。これは、最短経路が複数ある場合に起こりうるが、従来手法 (0) では、必ずしも将来需要量が高い経路が選ばれるとは限らない。
- (2) 各顧客の乗降車地点をつなぎ、全顧客の制約条件を満たす範囲で、将来需要量が高くなるよう迂回する経路。待ち時間と移動距離の両方において、提案手法は、従来手法 (0) と同等の結果になったことから、上述した経路の (1) の場合が多くあったと考察できる。また、提案手法の方が待ち時間および移動距離が増加していることから、上

述した経路の (2) の場合も発生していたことが推測できる。提案手法は、従来手法 (0) とほぼ同等の待ち時間および移動距離で、より低いリジェクト率を実現できていることから、将来需要の高い経路を通ることで、効率良く顧客を割り当てることができているといえる。

各時刻での車両の乗車人数を比較するため、従来手法 (100) および提案手法における、乗車人数の割合を示したグラフの一例を図 6 に示す。このグラフは、車両台数が 1,000 台、需要比率が 100% のときの結果を表し、横軸は時刻、縦軸は乗車人数の割合を表している。なお、乗車人数が 0 名の場合、車両の状態は、(1) 割り当てられた顧客へ配車するための移動中、(2) 需要の偏りを平準化するための移動中、(3) 停車中、の 3 種類に分けることができ、図 6 の凡例の 5～7 にそれぞれ対応する。図 6 より、どちらの手法も乗車人数の比率は同様であったが、乗車人数 0 名の場合において、従来手法 (100) は、需要の偏りを平準化するための移動が提案手法より多くあることが分かった。これは、架空の顧客リクエストが割り当てられ、その架空の顧客を迎えに行くような経路が生成されるためである。この理由により、従来手法 [5] において、サンプリング数が増加するほど車両の平均移動距離が増加していると考えられる。

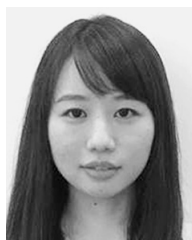
6. おわりに

本稿では、将来需要を考慮した経路最適化に基づく相乗り最適化手法を提案し、ニューヨーク市のタクシーの実データを用いた数値実験により、その有効性を検証した。提案手法では、割当て最適化による顧客の割当てと乗降車順序を変更せず、顧客の待ち時間および総遅延時間の制約を満たす範囲で、将来需要量の和が最大となるような経路を生成する。数値実験の結果、需要に対して車両台数が十分でない環境で、将来需要を考慮しない従来手法 [4] と同等の移動距離および待ち時間で、リジェクト数を減らすことが可能であると確認できた。

提案手法では、全顧客で共通の最大待ち時間および最大遅延時間を設定していた。しかしながら、各顧客の目的地までの距離や、急いでいるかどうかなどの事情により、不公平感や不満の原因になると予想される。そのため、今後は各顧客の希望を考慮することができる方法を検討する予定である。また、提案手法は一定時間ごとに車両の割当ておよび経路最適化を行うため、顧客が乗車をキャンセルすることに対しては対応できるが、乗車後の目的地変更には対応できない。提案手法を実システムで運用する際には、目的地変更可能であるなどのさらなる拡張が必要となる。

参考文献

- [1] Litman, T.: *Autonomous vehicle implementation predictions*, Victoria Transport Policy Institute Victoria, Canada (2017).
- [2] Gehrke, S. and Gehrke, S.R.: A Survey of Ride-Hailing Passengers Friday Transportation Seminar, Technical Report (2018).
- [3] Clewlow, R.R. and Mishra, G.S.: Disruptive transportation: The adoption, utilization, and impacts of ride-hailing in the United States, Research Report UCD-ITS-RR-17-07 (2017).
- [4] Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E. and Rus, D.: On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment, *PNAS*, Vol.114, No.3, pp.462–467 (2017).
- [5] Alonso-Mora, J., Wallar, A. and Rus, D.: Predictive routing for autonomous mobility-on-demand systems with ride-sharing, *Proc. IROS*, pp.3583–3590, IEEE (2017).
- [6] Cordeau, J.-F. and Laporte, G.: The dial-a-ride problem: Models and algorithms, *Annals of Operations Research*, Vol.153, No.1, pp.29–46 (2007).
- [7] Tong, Y., She, J., Ding, B., Chen, L., Wo, T. and Xu, K.: Online minimum matching in real-time spatial data: Experiments and analysis, *Proc. VLDB Endowment*, Vol.9, No.12, pp.1053–1064 (2016).
- [8] Lalos, P., Korres, A., Datsikas, C.K., Tombras, G.S. and Peppas, K.: A framework for dynamic car and taxi pools with the use of positioning systems, *Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, 2009*, pp.385–391, IEEE (2009).
- [9] Ma, S., Zheng, Y. and Wolfson, O.: T-share: A large-scale dynamic taxi ridesharing service, *Proc. ICDE*, pp.410–421, IEEE (2013).
- [10] Santi, P., Resta, G., Szell, M., Sobolevsky, S., Strogatz, S.H. and Ratti, C.: Quantifying the benefits of vehicle pooling with shareability networks, *PNAS*, Vol.111, No.37, pp.13290–13294 (2014).
- [11] Santos, D.O. and Xavier, E.C.: Dynamic Taxi and Ridesharing: A Framework and Heuristics for the Optimization Problem, *Proc. IJCAI*, Vol.13, pp.2885–2891 (2013).
- [12] d'Orey, P.M., Fernandes, R. and Ferreira, M.: Empirical evaluation of a dynamic and distributed taxi-sharing system, *Proc. ITSC*, pp.140–146, IEEE (2012).
- [13] 吉塚裕生, 内田英明, 藤井秀樹, 吉村 忍: ライドシェアサービス向け経路探索アルゴリズムの性能評価, 人工知能学会全国大会論文集, 人工知能学会 (2018).
- [14] Zhu, M., Liu, X.-Y. and Wang, X.: An Online Ride-Sharing Path-Planning Strategy for Public Vehicle Systems, *T-ITS* (2018).
- [15] Lin, Q., Deng, L., Sun, J. and Chen, M.: Optimal demand-aware ride-sharing routing, *Proc. INFOCOM*, pp.2699–2707, IEEE (2018).
- [16] Bellman, R.: On a routing problem, *Quarterly of Applied Mathematics*, Vol.16, No.1, pp.87–90 (1958).
- [17] Dijkstra, E.W.: A note on two problems in connexion with graphs, *Numerische Mathematik*, Vol.1, No.1, pp.269–271 (1959).



大社 綾乃

1993 年生。2015 年大阪府立大学工学部知能情報工学科卒業。2017 年同大学大学院修士課程修了。同年 (株) 豊田中央研究所入社。意思決定や経路の最適化に関する研究に従事。



大滝 啓介 (正会員)

2016 年京都大学大学院情報学研究科知能情報学専攻修了。京都大学博士 (情報学)。同年 (株) 豊田中央研究所入社。意思決定に関するデータ解析と最適化に関する研究に従事。



小出 智士

1981 年生。2004 年大阪大学理学部数学科卒業。2006 年同大学大学院情報科学研究科修士課程修了。同年 (株) 豊田中央研究所入社。時空間データベースと機械学習の研究に従事。



西 智樹

1982 年生。2005 年大阪大学応用理工学部機械工学科卒業。2007 年同大学大学院知能機能創成工学専攻修士課程修了。同年 (株) 豊田中央研究所入社。意思決定の最適化と自動運転および MaaS への応用に関する研究に従事。