

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

Performance Evaluation of Path Finding Algorithm for Ride-sharing Service

吉塚裕生 *¹

Yuki Yoshizuka

内田英明 *¹

Hideaki Uchida

藤井秀樹 *¹

Hideki Fujii

吉村忍 *¹

Shinobu Yoshimura

*¹ 東京大学大学院工学系研究科

School of Engineering, The University of Tokyo

Ride-sharing service is inexpensive urban mobility which is convenient for the user and can be used in real time. Since this service needs to dispatch the car immediately after reservation by the customer, real-time property is regarded as important. Therefore, we implement this service in the traffic simulator and evaluate the performance of the exhaustive search algorithm and the sequential insertion method for the path finding algorithm for ride-sharing. In the experiment, multiple demand are generated one after another at random positions, and the sum of calculation time and the detour cost required until the cost calculation of all vehicles is completed are measured. Experiments of multiple cases show that the sequential insertion method for computation time can be computed at high speed tolerable to the user and that the accuracy of the solution is not so inferior from the optimal solution.

1. はじめに

1.1 ライドシェアサービス

モータリゼーションの過度の進展は特に都市部において交通渋滞を深刻化させ、旅行時間の増加や大気汚染、地球温暖化などの問題を引き起こしている。一方で、最近ではスマートフォンの大規模な普及と通信費の減少により、Uber や Lyft 社が主導するユーザ中心の MoD (Mobility on Demand) システムが開発されている [1]。MoD システムの一つに、都市型モビリティであるアプリケーションベースのライドシェアサービスがある。このサービスは、一般のタクシーとは異なり相乗りによって複数のユーザの送迎を行う。ユーザは、車両の予約時に乗降地点を入力してライドシェアを希望するかどうか選択する。どちらの場合もユーザに乗車予定時刻と料金の情報を提示するが、一般的にはライドシェアを希望する方がより低い料金が提示される。このようなライドシェアサービスは、交通機関の随意性および信頼性を一定程度維持しつつ、自家用車を所有しない者や運転が困難になった高齢者を含むあらゆるユーザに安価な移動手段を提供する。またユーザにとって利便性が高く、リアルタイムで使用できる、安価なライドシェアサービスを都市部に提供することにより自動車の所有者数が減少するとされており [2]、総交通量の抑制 [3] については渋滞緩和と環境問題の改善効果も期待される。

1.2 ライドシェアサービス提供車両と本研究で想定する利用方法

ライドシェアサービスを提供する車両は Smart Access Vehicle[4]、あるいは将来的に自律型乗合車両に移行することを見越して Shared Autonomous Vehicles[5] などと呼ばれる。いずれも略称は SAV であり、以降本稿でも SAV と表記する。なお本稿の時点では運転手の有無によらず、ライドシェアサービス提供車両として同等に扱う。いずれの場合も SAV の車両サイズはタクシー～中型バス程度が想定されており、このような車両を導入することで、ライドシェアによる交通問題の改善効果が高まる。本研究では、管制センターがユーザの予約を処

理し、各 SAV に問い合わせを行い、その結果に基づいて適切な SAV に予約を割り当てる中央集権的なシステムを想定する。

1.3 本研究の目的

これまで SAV やオンデマンドバスは過疎地での公共交通の代替として考慮に上がることが多かったが、自動運転への期待が高まるにつれ都市部での短距離輸送を担う可能性も指摘されるようになってきた。そこで本研究では、都市部における複数車両での大規模な過密運用を想定した SAV を用いたライドシェアサービスを対象とする。本サービスの流れについて、ある時間に乗車地点 (Origin, 以下 O という) と降車地点 (Destination, 以下 D という) を持つデマンドが発生する。このとき、全ての SAV はデマンドを受け入れた場合の新経路の距離を迂回コストとして算出する。次に、迂回コストが最小となった SAV は、デマンドが割り当てられて経路を更新し、ユーザの送迎を実施する。デマンドが割り当てられなかった SAV は新経路を破棄し、既存の経路を走行する。この経路の更新において、SAV を用いたライドシェアサービスでは逐次挿入法 [6] が用いられる。このライドシェアサービスを実現するうえで重要なのは、リアルタイムにデマンドを割り当てることにかかる計算時間と、ユーザの目的地到着時間に関わる新経路のコストとのバランスである。しかし、逐次挿入法が当該の問題領域においてどの程度の性能・傾向を有するかは評価されていないため、エージェント個別で見れば最適解である全数探索法との比較実験によりそれを明らかにする。

2. 研究手法

2.1 SAV 用経路探索

SAV 用の経路探索は、一般車両の経路探索に加えてユーザのデマンドを満たす制約が必要である。すなわち割り当てられるデマンドの O および D を経由地点として指定した経路探索である。また、ライドシェアサービスはユーザの予約時にリアルタイムでデマンドを SAV に割り当て、ユーザに到着時刻と料金を提示しなければならない。すなわち、一定時間以内に全車両の迂回コストを算出する必要がある。新しいデマンドが割り当てられる際の SAV の経路探索は、経由地点の経由順の列挙と、それぞれの経由地点間の経路探索に分けられる。経由

連絡先: 吉塚裕生, 東京大学大学院工学系研究科,
113-8656 東京都文京区本郷 7-3-1, 03-5841-6994,
y.yoshizuka@save.sys.t.u-tokyo.ac.jp

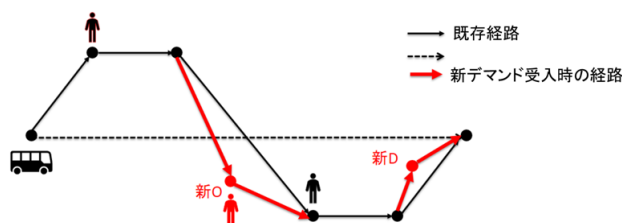


図 1: 逐次挿入法

地点間が 1 本のリンクで結ばれていると仮定すると、この問題は巡回セールスマン問題に帰着する。一般に巡回セールスマン問題を解くアルゴリズムには全数探索法や GA (Genetic Algorithm), SA (Simulated Annealing) などがあるが、これらはデマンドの発生時刻や OD が事前に分かっており、十分な計算時間をかけられるときにのみ有効な手法である。これに対し、ライドシェアサービスに求められるのはリアルタイムにデマンドの割り当てが可能な手法である。計算時間の重要度が高く、リアルタイムなデマンドの割り当てが求められる既存研究 [6] では、逐次挿入法が用いられた。そこで本研究でも、全数探索法と逐次挿入法を交通流シミュレータに組み込み、計算時間と迂回コストについて検証する。あわせて全数探索法で得られる経路の最適解と比較することで、逐次挿入法によって得られる経路の性質について確認する。

2.1.1 全数探索法

任意のデマンド i について、乗車地点 O_i と降車地点 D_i の順序を満たす手法である。つまり、デマンド i について D_i を O_i より先に経由しない（まだ乗車していない客を降車させない）ための最低限の制約である。自分に割り当てられたデマンドの件数を k とおくと計算すべき場合の数は $k!(2k-1)!!$ となる。

2.1.2 逐次挿入法

自分にとって $k-1$ 番目のデマンドが割り当てられた際に決定した経路順序の間に、新たなデマンド i の乗車地点 O_i と降車地点 D_i を挿入する手法である。図 1 にこの例を示す。デマンドの件数を k とおくと計算すべき場合の数は $k(2k-1)$ となる。

2.2 交通流シミュレータへの実装

本研究においては、経路探索アルゴリズムの比較実験のため、知的マルチエージェント交通流シミュレータ ADVENTURE_Mates[7] にライドシェアサービスモデルを実装した。今回、追加した機能とシミュレーションのフローを説明する。

本研究においては、経路探索アルゴリズムの比較実験のため、知的マルチエージェント交通流シミュレータ ADVENTURE_Mates にライドシェアサービスを実現するための管制センターエージェントと SAV エージェントを実装した。図 2 は、従来の ADVENTURE_Mates のシミュレーションフローである。車両エージェントは周辺の車両や道路環境との相互作用によって加速度・速度・位置をステップ毎に更新する。図 2 におけるドライバモデルの処理の前に、本研究では以下のライドシェアサービスモデル処理を新たに組み込んだ（図 3）。

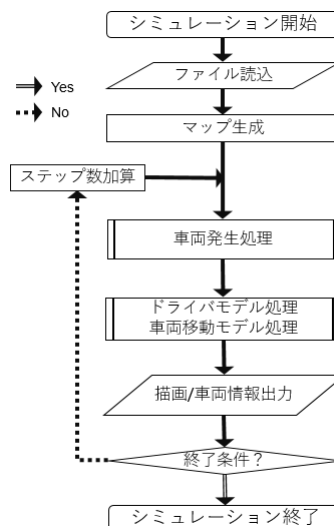


図 2: プログラム構造

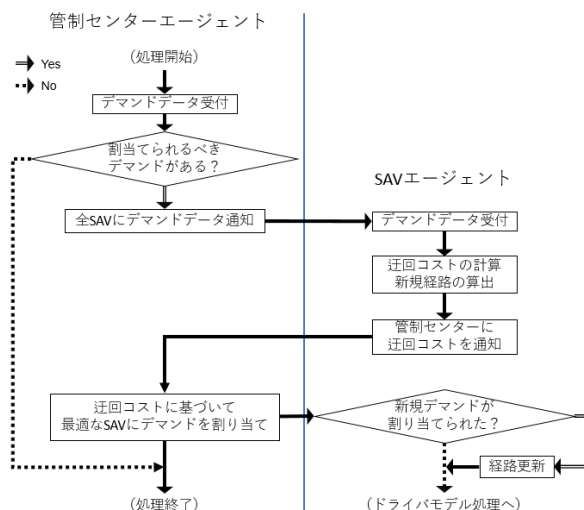


図 3: ライドシェアサービスモデル処理

1. デマンドが発生（O と D、希望乗車時刻の情報を配車センターエージェントに入力）
2. 管制センターエージェントは領域内の全 SAV エージェントにデマンド情報を通知
3. SAV エージェントは新規デマンド情報を基に経路探索を行い、迂回コストを算出
4. SAV エージェントは迂回コストを管制センターエージェントに通知
5. 管制センターエージェントは最小コストとなる SAV を決定、デマンドを割り当て
6. 新規デマンドを割り当てられた車両は予定経路を 3 で求めた経路に更新

表 1: シミュレーション設定

SAV[台]	1	2	4	8
デマンドの発生間隔 [sec]	240	120	60	30
総デマンド件数 [件]	100	100	100	100
総コスト計算数	100	200	400	800
シミュレーション時間 [sec]	24000	12000	6000	3000

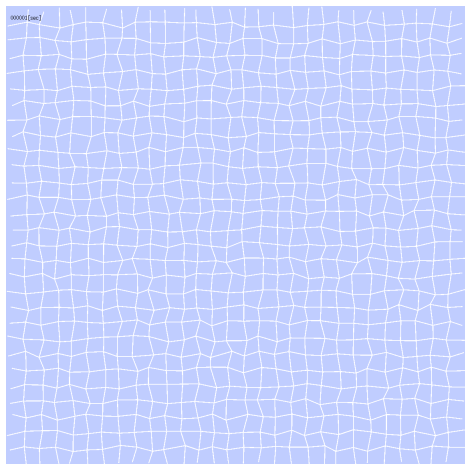


図 4: 歪ませた格子状道路ネットワーク

なお本研究では、3. で求める迂回コストにはデマンドの追加によって新たに生じる経路の距離を用いた。また各経由地点間を結ぶ経路の探索には A*アルゴリズム [8] を用いた。

3. 実験

3.1 実験設定

複数のデマンドの O, D を次々とランダムな位置に発生させ、全 SAV のコスト計算が終了するまでに要した計算時間の総和と迂回コストを計測する。これにより全数探索法と逐次挿入法の性能を評価する。SAV が 1, 2, 4, 8 台の設定を用意し、それぞれのケースについて 100 件のデマンドを与えた。デマンド発生間隔やシミュレーション時間に関する設定を表 1 に示す。これは、1 台の SAV が最大 5 件のデマンドを割り当てられる設定である。

実験に使用する道路環境を図 4 に示す。ノード数 896・リンク数 1624, 2.9km × 2.9km の領域である。また、1 リンクの長さは 100m 程度であり、交差点の座標は x 方向, y 方向に最大 ± 20m 変動させている。厳密な格子状ネットワークであると経路探索時にコストが完全に同一となる候補が多数生成され、計算時間に悪影響を及ぼすためである。なお、シミュレーションには CPU: Intel Xeon E5-2670 v3 @2.30GHz, メモリ: 128GB の計算機を用いた。

3.2 結果

3.2.1 計算時間

出力した計算時間は、新規に 1 件のデマンドが発生ときに全 SAV のコスト算出にかかる時間である。既に SAV がそれまでに持っていたデマンドの件数により計算量が大きく変わり、最大値と最小値の差が大きいため、累積度数で示す。図 5 は全ケースの結果である。

計算時間において明らかな差が得られた。たとえば逐次挿入法は SAV8 台で最大 9.2 秒で計算を終えるのに対して、全

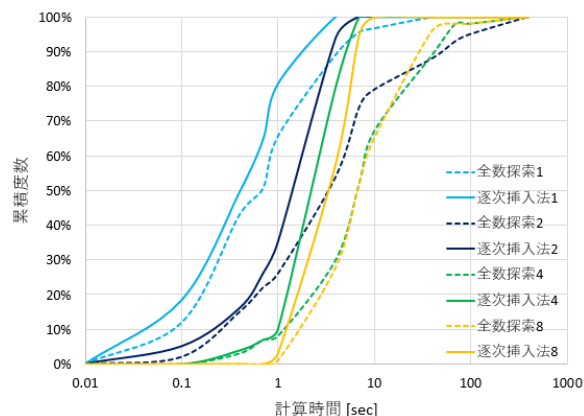


図 5: SAV[1, 2, 4, 8 台]

表 2: 全数探索と逐次挿入法の迂回コスト比

SAV[台]	1	2	4	8
迂回コスト比 100% のケース [%]	97.0	97.0	96.0	100.0
最大コスト比 [%]	104.6	103.7	105.0	100.0

数探索法では 179 秒かかる。計算機の性能にもよるが、全数探索法がサービスの運用に向かないことが改めて確認された。

3.2.2 迂回コスト

全数探索では考え得る経路をすべて調べるため最適解を出すことができる。このため、逐次挿入法でのコストと全数探索との比率である迂回コスト比を調べることで逐次挿入法によって得られる経路の質を評価できる。迂回コスト比は式 (1) に定義する。結果を表 2 に示す。

$$\text{迂回コスト比} = \frac{(\text{逐次挿入法のコスト})}{(\text{全数探索のコスト})} \times 100 \quad (1)$$

表 2 より、どのケースにおいても迂回コスト比は 105% 以内に収まる。逐次挿入法の迂回コストが全数探索に比べて、解の精度がそれほど劣らないことが分かる。また、SAV8 台のとき最大コスト比が 100% となっている点について、領域内に十分な SAV エージェントで満たされる場合、非効率なデマンドの割り当てが発生しないことが示される。計算時間が非常に短いことも考慮し、逐次挿入法が SAV 用経路探索アルゴリズムに採択することは合理的であるといえる。

4. まとめ

本研究ではリアルタイム性が重要視されるライドシェアサービスについて、管制センターエージェントと SAV エージェントをマルチエージェント交通流シミュレータに実装し、SAV の経路探索アルゴリズムについて検証した。複数のケースの実験により、計算時間について逐次挿入法はユーザが許容可能な程度に高速計算可能であり、かつ最適解から解の精度がそこまで劣らないことを示した。今後の課題としては、1 台の車両が受け持つデマンド件数が多いときのコスト比較や、ユーザの待ち時間もコストの変数とすること、1 台の SAV だけでなく全 SAV と全デマンドに対する全体最適化などが挙げられる。

参考文献

- [1] J.Alonso-Mora, S.Samaranayake, A.Wallar, E.Frazzoli, D.Rus: “On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment” Proceedings of the National Academy of Sciences of the United States of America, Vol.114, pp.462-467, 2017.
- [2] R.Hampshire, C.Simek, T.Fabusuyi, X.Di, X.Chen: “Measuring the impact of an Unanticipated Suspension of Ride-Sourcing in Austin, Texas” *ssrn:2977969*, 2017.
- [3] J.Bischoff, M.Maciejewski, K.Nagel: “City-wide Shared Taxis: A Simulation Study in Berlin” IEEE 20th International Conference on Intelligent Transportation Systems (ITSC) Workshop, pp.269-274, 2017.
- [4] 松原仁, 中島秀之, 平田圭二, 佐野渉二: “新しい都市型公共交通サービスのデザイン” サービス学会第一回国内大会, pp.304-307, 2013.
- [5] A.Kornhauser, A.Chang, C.Clark, J.Gao, D.Korac, B.Lebowitz, A.Swoboda: “Uncongested Mobility for All: New Jersey’s Area-wide aTaxi System” Princeton University, 2013.
- [6] 野田五十樹, 篠田孝祐, 太田正幸, 中島秀之: “シミュレーションによるデマンドバス利便性の評価” Information Proceedings Society of Japan, IPSJ, Vol.49, No.1, pp.242-252, 2008.
- [7] 吉村忍, 西川紘史, 守安智: “知的マルチエージェント交通流シミュレータ MATES の開発” シミュレーション, Vol.23, No.3, pp.228-237, 2004.
- [8] P.Hart, N.Nilsson, B.Raphael: “A Formal Bases for the Heuristic Determinatin of Minimum Cost Paths” IEEE Transactions on System Science and Cybernetics Vol.SSC-4, No.2, pp.100-107, 1968.