

オンデマンドバスのための乗り換えを含むリアルタイム ルートスケジューリング

政野 博紀^{1,a)} 柴田 直樹¹ Juntao Gao¹ 南 和宏² 伊藤 実¹

概要：近年、特に過疎地域において、固定路線バスの不採算路線の代替手段としてデマンドバスが注目され、実用化されている。しかし、多くのデマンドバスが予約にオペレータの介在が必要であるなど利便性が高いとは言えず、また採算性も低い。これに対し、自由な経路を持ち、より低遅延で予約の受理率が高いデマンドバスを自動でスケジューリングするシステムの研究が行われている。しかし、これらの研究では主に都市の中心部を対象としており、過疎地域においても運用した場合の採算性が十分に考慮されていない。本稿では過疎地域における運行も想定して、徒歩移動とデマンドバス間または鉄道など既存交通との乗り換えをナビゲートするデマンドバスのシステムを提案する。本システムにおいて、デマンドバスは自由な経路を走行するが、ユーザに対しデマンドバスの迂回が少なくなる乗降地点まで徒歩経路を案内し、乗り換えも案内する。これによりデマンドバスの迂回経路を削減し、運行の効率化を図る。

1. はじめに

地方都市の過疎地域では、バスなどの不採算路線が廃止され、ますます公共交通が衰退していく現状にある。地方自治体が住民の移動手段確保のために運行するコミュニティバスも財政負担を招き、その維持が困難になりつつある [1]。これに対し、より効率的なバスの運行形態としてデマンドバスが注目され、一部の自治体では実際に運行されている。デマンドバスは、定時的な固定ダイヤを持たず、乗客の需要に合わせて動的にダイヤや経路を設計する、より柔軟なバス運行形態である。その中でも固定経路と自由経路の運行形態、バス停固定と自由乗降の運行形態が存在するが、利便性の観点から自由経路・自由乗降でドア・ツー・ドアな方式（いわゆるフルデマンド方式）が望まれる。現在、実際に運行されているデマンドバスは、その多くがオペレータによる予約の受付とバスの経路設計が行われ、採算性も低い。加えて、デマンドバス特有の迂回による遅延の低減、乗り合い率の向上、予約受理率の向上、より自由な運行経路の設計などが課題である。

デマンドバスの経路設計問題は Dial-A-Ride Problem (DARP) として定式化され、タイムウィンドウを用いて経路設計を行うアルゴリズム、Advanced Dial-A-Ride with

Time Windows (ADARTW) が Jaw らにより提案されている [2]。また ADARTW に類似する手法を用いて、経路設計を自動で行うデマンドバスシステムが研究されている [3][4]。これらのデマンドバスシステムは都市の中心部における実証実験において、乗客の希望地点間を相乗り型交通を用いて輸送する手法がシステムとして動作することを確認した。

デマンドバスがより必要とされる公共交通が行き届かない地域で運用する場合、より利用者が少ない地域でも採算性が確保可能か検討する必要がある。デマンドバスの運行コストを削減するには、迂回経路の削減と相乗り率の向上によってバスの運行台数を削減しつつ、多くのユーザを輸送する必要がある。そこで本稿では、バス停非固定の自由乗降かつ自由経路の予約を必要としないフルデマンド式バスを前提とし、徒歩移動とデマンドバス間、または鉄道など既存交通との乗り換えをナビゲートするリアルタイムデマンドバスのシステムを定式化する。提案システムは完全なドア・ツー・ドアではなく、バス路線の最適化に加えて徒歩移動可能な範囲でバスの迂回経路が短くなるようユーザに徒歩移動をナビゲートする。徒歩移動のナビゲートはバス迂回経路の距離的短絡だけではなく、バスが高速で移動できない生活道路へのバスの侵入を抑制することによる迂回経路の時間的短絡も期待できる。徒歩移動のナビゲートにより、中型サイズ以上のバスが法的または物理的に侵入できない細街路に発生するユーザクエリも考慮可能である。また、他のデマンドバス、固定路線バス、鉄道等と

¹ 奈良先端科学技術大学院大学 情報科学研究科
Graduate School of Information Science, Nara Institute of
Science and Technology

² 統計数理研究所
The Institute of Statistical Mathematics

^{a)} masano.hiroki.ma7@is.naist.jp

組み合わせる乗車することにより、デマンドバスの迂回が短縮される場合にはその乗り換えも案内する。特に郊外では、鉄道路線はあるが駅間が長く駅からの目的地が遠い場合などがあることから、既存交通とデマンドバスの組み合わせの乗車のニーズが考えられる。加えて、バスの運行台数を需要により動的に増減し、余剰となったバスは営業所に戻るようにすることで運行コストの削減を図る。

DARP は NP 困難であることが知られており、デマンドバスのスケジューリングの計算コストが問題となる。本研究においては、徒歩経路や乗り換えを案内するための計算も必要となり、より一層の計算コスト増加が見込まれる。一方、スマートフォンを使用するユーザの許容待ち時間はおよそ 8.4 秒であることが分かっている [5]。そこで、ユーザのクエリに対するサーバの応答を 2 種類用意することを考える。第 1 の応答は、レスポンスタイム 10 秒程度を目安とした簡易的な応答（フォアグラウンド応答と呼ぶ）である。フォアグラウンド応答では、短い計算時間内で見つかったスケジュールの中で最も良い結果を仮決定し、ユーザにいったんクエリを受理可能か提示する。第 2 の応答は、バスの迂回を最適化したスケジュールの計算結果をフォアグラウンド応答後に提示する応答（バックグラウンド応答と呼ぶ）である。これらの応答により、ユーザが応答待ちを意識しないようにバスのスケジュールの最適解を計算する。

本稿では、以上のような問題の定式化を行いシステムの構成について検討する。また、システムが実用的な計算時間でスケジュールを算出可能か、実用的なコストで運用可能かを評価する手法について検討する。このとき、渋滞の考慮も行う。

2. 関連研究

本章では、デマンド型交通のスケジューリングについて、それぞれの関連研究の特徴をまとめる。デマンドバスを特徴付けるうえで基本的な運行形態とシミュレーションに関する項目は、表 1 にまとめる。そのうえで、本研究の位置づけについて述べる。

2.1 東京大学デマンド交通プロジェクト「コンビニクル」

東京大学では、バス停が固定されたデマンドバスにおいて、「ゆとり時間」と呼ばれるタイムウィンドウを設定した運行方式を提案している [3]。デマンドバスは、新たなユーザの乗り合い要求によりバスが迂回するので、乗車中のユーザは余計な乗車時間を費やすことになる。そこで、あらかじめ乗客はゆとり時間を設定し、サーバはデマンドバスが経路の迂回によって遅延したとしても、少なくともゆとり時間の範囲内には到着することをユーザに保証することで、通勤通学など時間制約があるユーザも利用可能としている。

タイムウィンドウを用いた方式はベーシックな運行方式ではあるが、千葉県柏市において固定バス停 88 ヶ所を用いた実証実験が行われたほか、いくつかの自治体でコンビニクル方式の導入事例もある。一方で、より利便性の高いシステムとして、非固定のバス停や、予約をせずに乗車できる方式が望まれる。

2.2 公立はこだて未来大学「SAVS」プロジェクト

公立はこだて未来大学の SAVS プロジェクトでは、平常時は通常のタクシーとして運用している車両を、ある一定の時刻のみ乗合型のデマンド型交通として運用を行うことを提案している [4]。デマンド型交通としての運用時におけるスケジューリング手法は、ユーザが持つ締切時刻をオーバーしないという条件下で、新たな乗り合い要求が来た場合に既存のスケジュールに挿入する逐次挿入法を採用しており、先述したコンビニクルと類似している。

SAVS では、固定のバス停を持たないドア・ツー・ドアの運行が可能であるとし、事前に予約するのではなくリアルタイムに配車を行う。函館市中心部における実証実験も行われており、クエリの送信から約 5 分で乗車可能であったことが報告されている。

SAVS は、ドア・ツー・ドアの経路設計においてタクシーに匹敵する経路の自由度を持つものの、課題としては単純な相乗りのスケジューリングにとどまっており、更なる効率化の余地がある。料金設定についてもタクシー以下でバス以上と想定されているものの、対象としている地域が市街地のみで、具体的な運行コストと、より広い地域や過疎地域でも運行した場合の採算性が考慮されていない。

2.3 ライドシェアタクシーの経路最短化によるコスト削減

日本国外においては、タクシーのライドシェアとしてデマンド交通の研究が行われている。Ma らは、タクシーの走行距離の最小化を目的として、タクシーのライドシェアシステムをシミュレーションにより検証している [6]。スケジューリングアルゴリズムはタイムウィンドウと用いた逐次挿入法で、先述したコンビニクルと同様である。

Ma らの論文で特筆すべきは、相乗りによって「ユーザ 1 人にあたりの運賃は削減し、ドライブは通常のタクシー輸送より損しない」という料金モデルを設定したうえで、走行距離と相乗り率から採算性について議論されている点である。タクシーの GPS 軌跡と実際の乗車履歴を用いた北京のシミュレーションにおいて、11%のタクシー走行距離の削減と、7%のユーザの運賃削減が報告されている。

しかし、多くのタクシーを短時間で探索するためにシミュレーション領域をグリッドで分割し、ユーザと同じセル内のタクシーを優先して探索していることから、より効率の良いタクシーのスケジューリングが見逃されている可能性がある。また、タクシー乗車の 1 クエリに対して 1 人

表 1: 関連研究における仮定および評価方法

	バス停	予約	乗り換え	車両	ユーザ数	実証実験
コンビニクル [3]	固定	必要		定員 4 人 × 2, 定員 8 人 × 1, 定員 20 人 × 1	2482 人	○
SAVS[4]	非固定	リアルタイム		定員 4 人 × 18, 定員 8 人 × 2	320 人	○
Ma ら [6]	非固定	リアルタイム		タクシー × 7088		
上原ら [7]	非固定	必要	○	定員 20 人 × 124	2300 人	
DARPT[8]	半固定		○	3~13 台	24~114 人	

と固定の設定になっており現実的ではない。

運行コストについても走行距離のみが考慮されており、走行時間や走行台数について考慮されていない。実際は渋滞等で停止している場合にもコストは発生するので、これらの考慮は必要である。また、タクシー運賃から 7%削減された運賃は日常利用するには依然として高額で、乗り換えや迂回経路の削減によって一層の運行コスト削減の必要がある。

2.4 大型バスとデマンドバスの乗り換え

大型バスとデマンドバスの乗り換えについても研究されている。上原らは、小型のデマンドバスによってユーザを近傍デポまで輸送し、デポで集約したユーザは大型バスによって中距離の輸送を行うクラスターリング手法を提案している [7]。シミュレーションにより ADARTW を用いた通常のデマンドバスと運行コストで比較した結果、乗換え手法はバスの運行台数は増加したものの、トータルのバスの運行距離は削減されたことを報告している。

しかし、軌道交通がほとんど存在しない沖縄県を対象としていることや、デポ間移動の大型バスは独自に設定した路線であることから、既存の交通との乗り換えは考慮されていない。また、デポはユーザの目的地を満たしやすい理想的な位置に設置されていることや、バスの台数は無限であること、都心と郊外間の高需要な通勤利用のみ考慮していること、渋滞を考慮していないことからシナリオは限定的である。

2.5 デマンドバス同士の乗り換え

デマンドバス間の乗り換えを含む問題は、DARP を発展させた Dial-A-Ride Problem with Transfers (DARPT) として定式化され、その解法の評価がシミュレーション実験により行われている [8]。DARPT は ADARTW を適用した DARP と同様にタイムウィンドウを用いて逐次挿入するスケジューリング手法が用いられているが、デマンドバス間で乗り換えができる点が異なる。文献 [8] では Pickup and Delivery Problem with Transfers (PDPT) [9] に用いられる Adaptive Large Neighborhood Search (ALNS) を拡張したうえで DARPT に適用し、焼きなまし法に類似する手法でスケジューリングを行っている。

文献 [8] では、乗り換え地点が固定された運用方式で、デマンドバス間の乗換により運用コストが 8.25%削減されることが報告されているが、自由な乗換地点を設定することでさらにバスの迂回経路が削減されることが期待できる。また、乗降地点は自宅と特定施設間の移動のみとなっており、自由な地点間の移動が考慮されていないことから、フルデマンド方式のデマンドバスを DARPT に適用した評価が求められる。

2.6 本研究の位置づけ

本章でこれまで挙げた研究は、運行コストについて考慮されていない、もしくは限定的な条件下のみの考慮にとどまっている。また、これらの研究は主に都市部やその郊外の輸送を対象としており、過疎地域での運行は考慮されていない。また、デマンドバス間の乗り換えについてもその乗り換え経路の自由度が低く、フルデマンド式をベースとした経路自由度の高い環境下でのシミュレーションを検討する必要がある。

本研究では、より低需要な地域でのデマンドバスの運行を図るため、乗り換え可能なデマンドバスを徒歩移動の案内および既存交通との乗り換えも含めるよう拡張し、迂回経路の削減と余剰バスの営業所への回送を考慮することで運行コスト削減を図る。

3. 問題設定

3.1 提案システム概要

本システムは、ユーザに多少の利便性の低下を許容してもらう代わりに、デマンドバスの運行コストを削減し、より低需要な地域でのデマンドバス運行を目標とする。ユーザがシステムに入力した出発地・目的地に対し、システムはユーザの徒歩移動圏内でバスの迂回が最小となる乗降地点を提示し、ユーザ希望地点と乗降地点間の徒歩経路と乗降時間を地図によりスマートフォン等を通じてナビゲートする。また、鉄道や固定路線バスなど既存の交通、またはデマンドバス間の乗り換えによりデマンドバスの迂回が削減される場合には、乗り換えの案内も同時に行う。スケジューリングにおいて、システムが設定するオーバーヘッド（詳細は 4.2 節で後述）の範囲内で遅延を許容するものとする。

ユーザの乗車要求、バスの位置情報、バスが持つ個々のスケジュール、ユーザの位置情報は、逐一サーバに蓄積される。サーバはユーザのクエリおよびバス情報をもとに、ユーザの要求時間に目的地付近まで移動するバスを探索する。徒歩経路や乗り換えを含む迂回経路最小スケジュールを算出するには多くの計算時間を要することが予想される。そこで本システムでは、サーバの応答をフォアグラウンド応答とバックグラウンドと応答の2種類用意する。フォアグラウンド応答では、10秒程度のレスポンスタイムを目安とし、時間内に見つかった経路で最も良い解を提示することで、ユーザの要求するスケジュールで一旦バスを配車可能かを応答する。バックグラウンド応答では、フォアグラウンド応答の後により良い解がないか、経路を再算出することでバスの迂回を最小化しスケジュールを最適化する。このとき、**余裕度**（定義については4.4節で後述）というスケジュールの余裕さを表す関数を設定し、ユーザの徒歩圏内で余裕度の一番大きいバスに新たな乗車スケジュールを挿入する。バックグラウンド応答では常に経路の最適化を行い、また別の新たなユーザの乗車要求を受領する都度スケジュールを再算出する。スケジュールが更新された場合は既存ユーザに通知する。

3.2 想定環境

3.2.1 ユーザに対する仮定

ユーザはスマートフォンなどの無線通信端末を介して、乗車希望地点、降車希望地点、乗車希望時刻、乗車人数を入力する。ユーザの要求を満たすバスの経路が存在しない場合、代替案を提示しクエリの再入力を促すものとする。

本システムでは、事前予約をするユーザとしないユーザがいることを想定しリアルタイムスケジュールリングを行う。ユーザは乗車したい時刻に端末から乗車要求を出す。また、ユーザは歩行してデマンドバス乗車地点に移動する際、逐一位置情報をサーバに発信する。

3.2.2 デマンドバスに対する仮定

デマンドバスには無線通信機能を持つ車載器が搭載されているものとし、バスの位置情報は逐一サーバに送信されるものとする。この車載器により経路スケジュールや新たな乗客の追加情報などを受信する。また、乗車予定であるユーザの位置情報を受信し、ユーザが予定通りに乗車可能か確認する。近年はタブレット等で、専用の車載器を用いずとも、これらの機能は実現可能であると考えられる。

デマンドバスは車庫（営業所）から出発する。車庫および営業所は、既存のバス会社の設備をそのまま使用すると仮定する。営業所は複数存在し、各営業所に収容可能台数と現在収容されているバス台数が設定されている。バスが営業所から出発するタイミング及び戻るタイミングは、現在のバス予約状況より適切に設定するものとする。

デマンドバスの車両は大きさの異なる複数種類用意する

ことを考える。中型サイズ以上のバスが法的または物理的に侵入できない細街路に徒歩移動不可能なユーザの乗降要求がある場合には、小型車両（タクシーサイズを想定）が優先的に配車される。乗車定員が多い中型以上の車両（マイクロバスや路線バスサイズを想定）は、需要の多い幹線道路を中心に、小型車両から乗り換えたユーザを集約した運用が考えられる。

3.2.3 サーバに対する仮定

サーバはクラウド上に設置するものとし、初期コストを削減する。その時々に必要な処理能力に応じて動的にサーバを確保する。サーバはバスの現在地と乗車人数、スケジュールを車載器から、ユーザの位置情報をユーザ端末から無線ネットワークを介して逐一収集する。また、逐一更新されるバスのスケジュールをもとに、デマンドバス間や既存交通の乗り換え可能状況もあらかじめ計算しておく。入出力やデータについて、サーバにおける想定する流れを図1に示す。詳しいデータの内容については次章で述べる。

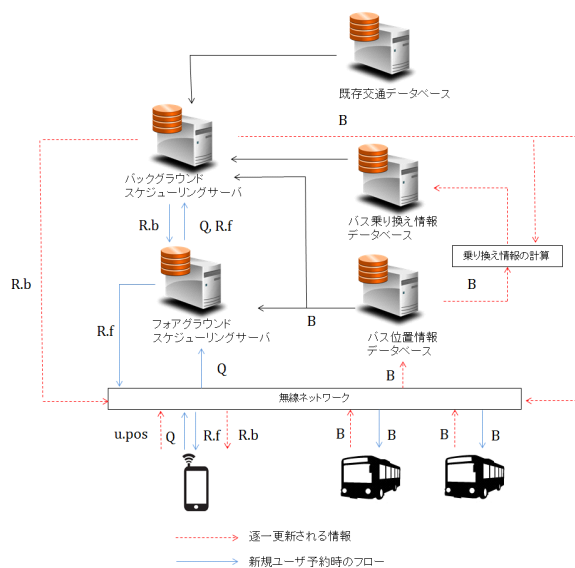


図1: サーバにおける処理のフロー

3.2.4 徒歩移動に関する仮定

ユーザが1回に徒歩移動可能距離は15分程度（1km前後）とする。ただし、長距離の歩行が困難なユーザの存在も考えられる。そこで、一部のユーザは1回の歩行可能距離をさらに制限する（200m前後を想定）。ユーザは徒歩移動中にサーバに位置情報を送信しており、現在地を考慮して歩行途中に歩行経路が変更される場合がある。

3.2.5 乗り換えに関する仮定

鉄道など既存交通は決められたダイヤから遅延しないものとし、乗り換え場所に既存交通の出発時間より前に到着するデマンドバスは乗り換え可能とする。ただし、乗り換

えには数分の時間を要することが考えられるから、数分の固定した余裕時間考慮する。

4. 問題定義

4.1 入力

入力は、あらかじめサーバに入力しておくデータベース入力とユーザからスマートフォンなどの端末を通じてサーバに入力されるユーザクエリ、およびバス車載器からの入力がある。

4.1.1 データベース入力

データベース入力には、地図入力と既存交通に関する静的な運行情報がある。道路トポロジはグラフ $G = (V, E)$ として与えられ、地図入力については表 2 にまとめる。このとき、エリア内にバス営業所が m 箇所存在するとする。

既存交通の鉄道や固定路線バスのスケジュールの集合は、 n 本の運行がある場合に $T = \{transport_1, \dots, transport_n\}$ で与える。例えば $transport_i \in T$ は k 箇所の駅またはバス停に停車するとき、表 3 に示す情報を持つ。既存交通の乗り換え場所（駅やバス停）は、地図情報におけるノードで表現することとする。

表 2: 地図入力

Notation	Definition
$v_i \in V$	交差点ノード
$e_{ij} \in E$	交差点間のエッジ
$speedLimit_{ij}$	ノード間の法定速度
$dist_{ij}$	ノード間の距離
$congestion_{ij}(t)$	時刻 t におけるノード間の混雑度劣化係数
$\{P_1 \dots P_m\} \subset V$	バス営業所を表すノード

表 3: 既存交通運行情報入力

Notation	Definition
$\{station_1, \dots, station_k\} \subset V$	停車駅
$\{arrive_2, \dots, arrive_k\}$	各停車駅における到着時刻
$\{depart_1, \dots, depart_{k-1}\}$	各停車駅における発車時刻
$\{transfer_1, \dots, transfer_k\}$	既存交通同士の乗り換え情報
$\{transfer_i\} \subset T$	$station_i$ で乗り換え可能既存交通

4.1.2 ユーザクエリ

全ユーザの集合を U として、あるユーザ $u_i \in U$ はクエリ Q_i を入力する。クエリ Q は表 4 に示す情報を持つ。また、ユーザクエリとは別に、ユーザの歩行中の位置情報 $u.pos$ も逐一サーバに入力する。

*1 すぐに乗車を希望する場合は入力不要。後述する評価計画においては、すべてのユーザがリアルタイムの乗車を希望するものとする。

表 4: ユーザクエリ

Notation	Definition
$Q.t$	クエリ発生時間
$Q.o.pos \in V$	乗車希望地点
$Q.d.pos \in V$	降車希望地点
$Q.o.t$	乗車希望時間 ^{*1}
$Q.d.t$	降車希望時間
$Q.num$	乗車人数
$Q.walk$	歩行困難か否か

4.1.3 バス車載器からの入力

バス車載器からサーバへの入力 B は、表 5 にまとめる。それぞれの情報は、逐一サーバに送信する必要がある。バススケジュール $B.s$ は、ユーザクエリによる乗降地点とその乗降時間から構成される。

表 5: バス車載器からの入力

Notation	Definition
$B.id$	バス ID
$B.t$	バスのタイムスタンプ
$B.pos$	バスの位置情報
$B.num$	乗車中の人数
$B.c$	バス定員
$B.s$	バススケジュール

4.2 出力

サーバからの出力は、ユーザ端末への出力 R とバス車載器へのスケジュール出力 $B.s$ に大別される。また、ユーザ端末への出力は、フォアグラウンド応答による出力 $R.f$ とバックグラウンド応答による出力 $R.b$ に分けられる。乗客が $n-1$ 回のバス乗り換えにより n 回の乗降を繰り返すとするとき、ユーザ出力を以下に示す。

乗車地点 $\{getOn.pos_1 \dots getOn.pos_n\} \subset V$ 徒歩移動と乗り換えを考慮したバスの乗車地点。

降車地点 $\{getOff.pos_1 \dots getOff.pos_n\} \subset V$ 徒歩移動と乗り換えを考慮したバスの降車地点。

乗車時刻 $\{getOn.t_1 \dots getOn.t_n\}$ 徒歩移動と乗り換えを考慮したバスの乗車時刻。サーバが応答した時点のスケジュールにおける乗車時刻で、新たなユーザの出現による迂回や突発的な渋滞によって遅れる場合がある。

降車時刻 $\{getOff.t_1 \dots getOff.t_n\}$ 徒歩移動と乗り換えを考慮したバスの降車時刻。サーバが応答した時点のスケジュールにおける降車時刻で、新たなユーザの出現による迂回や突発的な渋滞によって遅れる場合がある。

徒歩案内経路 $\{path_1 \dots path_{n+1}\}$ ユーザ希望乗車地点からバス乗車地点、バス降車地点からユーザ目的地、バス乗り換え時の徒歩案内経路。

最遅延乗車時刻 $getOn.t.l$ ユーザに保証する最も遅い乗車時刻。サーバーはユーザの乗車予定のバスが最遅延乗車時刻を超えて迂回するようなスケジューリングは行わない。ユーザの希望乗車時刻と実際の乗車地点までの徒歩移動時間、システムが設定するタイムウィンドウのから以下のように設定される。 $path.t$ は徒歩移動時間、 $TW.o$ はシステムが設定するタイムウィンドウの時間を表す。

$$getOn.t.l = Q.o.t + path.t + TW.o \quad (1)$$

最遅延降車時刻 $getOff.t.l$ ユーザに保証する最も遅い降車時刻。ユーザが乗車中に最遅延降車時刻を超えて迂回するような経路変更は行わない。ユーザの希望地点の $Q.o.pos$ と $Q.d.pos$ の間を乗車希望時刻 $Q.o.t$ から車で最短経路を移動した場合の移動時間 $t.pc$ と、システムが設定する $X\%$ の許容オーバーヘッドをもとに時刻が設定される。

$$getOff.t.l = Q.o.t + t.pc \times X\% \quad (2)$$

目的地到着時刻 $R.o.t$ 最後に乗車するバスの降車時刻に、バス降車地点からユーザの目的地までの徒歩移動時間を足した、ユーザの目的地到着時刻。

フォアグラウンド応答 $R.f$ は1回だけの応答であるのに対しバックグラウンド応答 $R.b$ はユーザのスケジュールが変更される都度出力される。よって $R.b$ では、上記の乗車地点、降車地点、乗車時刻、降車時刻、目的地到着時刻、徒歩案内経路はスケジュールが更新される都度変更される。

4.3 制約

制約条件については、バックグラウンド応答のみに対する制約と、両方の応答に共通する制約を考える。

4.3.1 バックグラウンド応答に対する制約

バックグラウンド応答における制約は、ある時刻におけるバス台数 $busNum(t)$ が、ある時刻におけるユーザ数 $userNum(t)$ と余裕度を考慮して一定台数以上確保されていることである。余裕度 $margin(t, B_i)$ については次項で詳細を述べる。

$$busNum(t) > userNum(t) \times margin(t, B) \quad (3)$$

4.3.2 共通する制約

両応答に共通する制約は、ユーザの要求する時間制約を満たすことである。つまり、最遅延乗車時刻 $getOn.t.l$ および最遅延降車時刻 $getOff.t.l$ よりも遅延するスケジューリングを行わないことである。

$$getOn.t_1 < getOn.t.l \quad (4)$$

$$getOff.t_n < getOff.t.l \quad (5)$$

任意の時刻におけるバス乗車人数 $B.num(t)$ は、バス定員 $B.c$ を超えない。

$$B.num(t) \leq B.c \quad (6)$$

乗り換え元のバスの降車時刻は乗り換え先のバスの乗車時刻より早い。

$$getOff.t_i + path.t_{i+1} < getOn.t_{i+1} \quad (7)$$

ユーザが徒歩により乗車地点に到着する前にバスは発車しない。

$$Q.t + path.t_1 < getOn.t_1 \quad (8)$$

乗り換え回数は C_1 (3回程度を想定) を超えない。

$$n \leq C_1 \quad (9)$$

また本提案システムでは通常ユーザと歩行困難なユーザの2種類の属性に分け、それぞれに最長歩行距離 C_2, C_3 を設ける。

$$path_i.length < C_2(Q.walk = 1) \quad (10)$$

$$path_i.length < C_3(Q.walk = 0) \quad (11)$$

$$C_3 \ll C_2 \quad (12)$$

4.4 目的関数

目的関数についても、フォアグラウンド応答とバックグラウンド応答について考える。フォアグラウンド応答の目的は、短いレスポンスタイムのなかで一旦クエリを受理可能か判別し、無難なスケジュールを応答することにある。よって、目的関数は以下のように到達時間の最小化とする。

$$\text{Minimize } \sum_{u_i \in U} getOff.t_n(u_i) \quad (13)$$

バックグラウンド応答では、迂回を最小化することでデマンドバスの運行を効率化することが目的である。ここで、バスの迂回を距離だけでなく時間も考慮した余裕度 $margin(t, B)$ を導入し、以下に定義を述べる。

余裕度 $margin(t, B)$ ある1台のバス $b_i \in B$ が時刻 t 以降にユーザのスケジュールが許す時間内で迂回できる面積。

例えば、図2に示す場合の余裕度を考える。バス B_1 は時刻 t において3つの乗降スケジュール $B.s_1, B.s_2, B.s_3$ が予定されているものとする。それぞれの乗降の余裕時間 $t.m$ は乗降車時刻 $getOn.t_1, getOff.t_n$ と最遅延乗車時刻 $getOn.t.l$ または最遅延降車時刻 $getOff.t.l$ の差で表現され、今回はそれぞれ3分、15分、10分とする。バスの現在地から $B.s_1$ までの区間余裕度は3分で迂回できる面積である。 $B.s_1$ か

ら $B.s_2$ までの区間では $B.s_2$ の余裕時間が 15 分であるが、さらに後に予定されている $B.s_3$ の余裕時間が 10 分と短いので、 $B.s_1$ から $B.s_2$ までの区間余裕度は 10 分で迂回できる面積である。そして余裕度 $margin(t, B)$ は、図 2 の橙色で示すそれぞれの区間余裕度の面積の総和となる。ただし、 B_2 に示すような他のバスと余裕度の面積が重複している場合、面積は重複して余裕度にカウントしない。

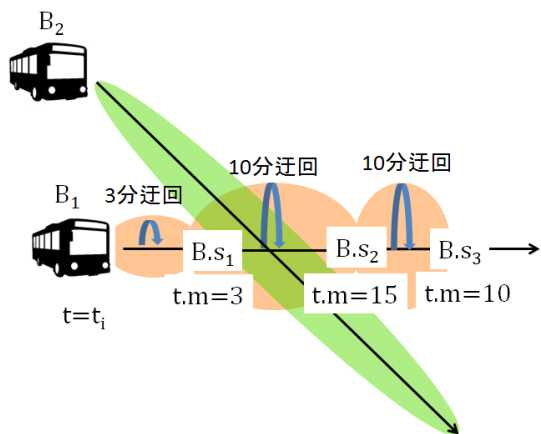


図 2: 余裕度の設定例

バックグラウンド応答における目的関数は、この余裕度 $margin(t, B)$ を最大化する。 t_{start}, t_{end} はそれぞれ評価開始時刻と評価終了時刻を表す。

$$Maximize \sum_{t=t_{start}}^{t_{end}} \sum_{b_i \in B} margin(t, b_i) \quad (14)$$

5. スケジューリング手法

余裕度を用いて徒歩経路および乗り換えを考慮したデマンドバスのスケジューリング方法について述べる。スケジューリングは ADARTW ベースで行うが、これを歩行者や乗り換えに拡張する必要がある。スケジューリングの例を図 3 に示す。それぞれのユーザは徒歩で移動できる範囲を持ち、余裕度により設定されるデマンドバスの迂回可能範囲と重なり合う領域のうち、徒歩移動でユーザが到着する時刻の方がバス到着時刻より早い領域で迂回距離が最小の地点を乗車位置の解とする。

乗り換えに関しては、任意の 2 つのバスの迂回可能範囲に降車地点からの徒歩移動可能範囲を拡張した領域が重なり合い、かつ乗り換え可能な時間的制約を満たす範囲、つまり乗り換え元のバスの方が先に通過する領域を求める。図 3 の B_1 から B_2 へのバスの乗り換えを考えた場合、赤線で囲まれた領域が時間制約をみたすおおよその領域と考えられる。この乗り換え可能領域のうち双方のバスの迂回が最小となる地点が乗り換え地点の解である。既存交通と

の乗り換えも同様の手法で考えることができる。乗り換えによりユーザの目的地付近に到達可能か探索する。

フォアグラウンド応答では、文献 [6] を参考に経路の計算を行うが、ADARTW は NP 困難であるため 10 秒程度を目安に応答できるように計算を途中で打ち切る。バックグラウンド応答では、遺伝的アルゴリズムや焼きなまし法などの発見的手法を用いてバスと歩行者の経路を短縮する。ユーザが徒歩移動中の場合でも、ユーザの現在地を考慮して経路の最適化を行う。スケジュールが変更になった場合、ユーザにその都度通知する。また、フォアグラウンド応答の計算で使用する上記の各領域をバックグラウンドで計算しておくことにより、フォアグラウンド応答の高速化を図る。

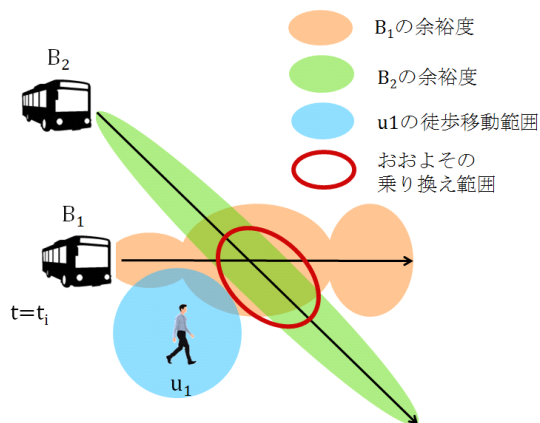


図 3: 徒歩経路および乗り換えの設定例

6. 評価計画

提案手法の有効性を確認するためにシミュレーション実験を行う。実験では、1,000 程度のユーザがデマンドバスを利用する環境下で、実用的な計算時間でスケジュールを算出し、運行コストの削減が可能か評価する。

6.1 実験環境

地図データはフリーの地図データ OpenStreetMap (OSM) [10] を用いる。OSM は空中写真や位置測位機能を持つ携帯端末を用いて測位された地図データであるが、登録ユーザは手動による編集も可能である。OSM では道路の規模に応じてノードにタグ付けがされており、これをもとにデマンドバスが通行可能な道路幅かを考慮可能である。

シミュレータはドイツ航空宇宙センターが開発するフリーの交通流シミュレータ Simulation of Urban MObility (SUMO) を用いる。デマンドバスにおける乗り換え、徒歩移動も対応できるように SUMO をチューニングし、OSM から取得した地図情報を入力する。

6.2 評価項目

提案手法の有効性を評価するための項目を以下に示す。

運行コスト デマンドバスの運行にかかる事業者が負担するコスト。詳しくは後述する。

運賃 運行コストをもとにした単位距離あたりの乗客の運賃を評価する。

ユーザの待ち時間 ユーザが総移動時間と乗車時間の差を評価する。

計算時間 フォアグラウンド応答およびバックグラウンド応答における計算時間を評価する。

デマンドバスの事業者が負担するコストは、事業所などの運用にかかる固定費、燃料費等バスの総走行距離に比例するコスト、ドライバーの人件費等バスの総走行時間に比例するコストの和で表す。デマンドバスの迂回を削減するほど走行距離が削減され、バスを事業所まで回送して動的にバス台数を変化させるほど走行時間が削減されることで運行コストが削減されるモデルである。走行距離のコストにかかる係数を k_1 、走行時間のコストにかかる係数を k_2 として、事業者の負担するコストは以下ようになる。

$$cost = \text{固定費} + k_1 \times \text{走行距離} + k_2 \times \text{走行時間} \quad (15)$$

6.3 比較対象

これらの評価項目について、従来の乗り合いを行わないタクシーと比較する。タクシーはユーザが電話すると最も近くの空車がピックアップし、ユーザの希望地点間を最短経路で走行するものとする。空車が一番近くの駅で待機するものとする。タクシーの料金は走行距離に比例するものとする。タクシーの台数はパラメータとし変化させながら比較を行う。

6.4 ユーザのクエリ発生モデル

文献 [6] で利用されたタクシーの GPS トレースおよび乗客の乗降データ [12] がリリース予定である。このデータが利用できる場合は、1 クエリに対する乗車人数について推定モデルを設定し、データを拡張したうえでデマンドバスおよびタクシーのクエリとして利用する。データが利用できない場合は、道路ネットワークの密度をもとにクエリをランダムで発生させる。

また、ユーザごとに歩行速度を変化させることでユーザが予定時刻に乗車場所に現れない場合を考慮する。ユーザが乗車時刻に遅刻した場合、バスがユーザの到着を待つか、迂回して乗客をピックアップする方法が考えられる。

7. まとめ

本稿では、低需要な地域でのデマンドバスの運用を前提

とし、乗り換えやユーザの徒歩移動を考慮したデマンドバスの運行コスト削減手法の定式化を検討した。デマンドバスの時間的・空間的経路削減手法としてバスの迂回可能面積を利用した余裕度を定義し、そのスケジューリングを提案した。今後は、余裕度や乗り換え可能領域、徒歩移動可能領域の面積の計算方法を検討する必要がある。また、シミュレーションエリアの選定やエリア内の既存交通の運行情報取得方法も検討する必要がある。

参考文献

- [1] 国土交通省 中部運輸局: デマンド型交通の手引き (入手先 (<https://www.tb.mlit.go.jp/chubu/tsukuro/joho/demando/pdf/demando.pdf>)) (2016.06.30 取得).
- [2] Jaw, J. J., Odoni, R. A., Psaraftis, N. H., and Wilson, H.M. N.: 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. 243–257, 1986.
- [3] Tsubouchi, K., Hiekata, K., and Yamato, H.: Scheduling Algorithm for On-demand Bus System, *Information Technology: New Generations, 2009*, pp. 189–194, 2009.
- [4] Nakashima, H., Sano, S., Hirata, K., Shiraishi, Y., Matsubara, H., Kanamori, R., Koshihara, H., and Noda, I.: One Cycle of Smart Access Vehicle Service Development, *In Proceedings of the 2nd International Conference on Serviceology (ICServ2014)*, pp. 152–157, 2014.
- [5] 三好 匠, 江口 真人: ユーザの通信ストレス軽減に向けた QoE 許容限界のモデル化, *電気通信普及財団 研究調査報告書*, no. 28, 2013.
- [6] Ma, S., Zheng, Y., and Wolfson, O.: Real-Time City-Scale Taxi Ridesharing, *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 7, pp. 1782–1795, 2014.
- [7] 上原 和樹, 赤嶺 有平, 當間 愛晃, 根路銘 もえ子, 遠藤 聡志: デマンドバスと大型車両による協調型交通システムの提案, *情報処理学会論文誌*, vol. 56, no. 1, pp. 46–56, 2015.
- [8] Masson, R., Lehuède, F., and Peton, O.: The Dial-A-Ride Problem with Transfers, *Computers & Operations Research*, vol. 41, pp. 12–23, 2014.
- [9] Masson, R., Lehuède, F., and Peton, O.: An adaptive large neighborhood search for the pickup and delivery problem with transfers, *Transportation Science*, vol. 47, no. 3, pp. 344–355, 2012.
- [10] Haklay, M., and Weber, P.: OpenStreetMap: User-Generated Street Maps, *Pervasive Computing, IEEE*, vol. 7, no. 4, pp. 12–18, 2008.
- [11] Behrisch, M., Bieker, L., Erdmann, J., and Krajzewicz, D.: SUMO - Simulation of Urban MOBility, *The Third International Conference on Advances in System Simulation*, 2011.
- [12] Yuan, J., Zheng, Y., Zhang, C., Xie, W., Xie, X., Sun, G., and Huang, Y.: T-drive: Driving directions based on taxi trajectories, *in Proc. 18th SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, pp. 99–108, 2010.

謝辞

本研究の一部は、JSPS 科研費 16K01288, 16K12421, 16K16059 の助成による成果である。