3J2-OS10-6

# オンデマンド交通運行計画生成アルゴリズムの地域適合化

## Area Adapted Scheduling Algorithm for the On-demand Bus System

坪内 孝太[*1]
Kota Tsubouchi

大和 裕幸[*1]
Hiroyuki Yamato

柳澤 龍[*1]
Ryu Yanagisawa

[*1] 東京大学大学院 新領域創成科学研究科
Graduate School of Frontier Sciences, The University of Tokyo

The problem of the developed On-demand Bus service with cloud computing technology is not to be able to react the complicated request from local government. This is because the same calculation algorithm is introduced in all projects. In order to answer the complicated requirement to the On-demand Bus scheduling algorithm from each area, the area adapted scheduling algorithm is developed and evaluated by the computer simulation experiment. According to the result of computer simulation, the proposed scheduling algorithm performed as designed.

## 1. 序論

オンデマンド交通とは，利用者が事前に行った予約にあわせて運行計画を変えながら効率的に移動する新しい乗り合い交通サービスである．

既存のオンデマンド交通サービスは市町村がサーバを購入し運用する「サーバ購入タイプ」によって提供され，システムにかかる費用が普及の課題となっていた．大和ら[大和 2008]および坪内ら[Tsubouchi 2011]はサーバをデータセンタで運用しインターネットを通じてサービスを提供する「サービス購入タイプ」を開発し，低コストで運用できる新しいオンデマンド交通サービスを開発した．また，開発したシステムは，①乗客からの予約が入る度に運行計画を作成するリアルタイム処理方式を採用しており，②時刻・経路非固定型のデマンド運行に対応し，③ゆとり時間と呼称する特殊な Time Window[Jaw 1986]を用い，スケジューリングによる遅延というオンデマンドバススケジューリング特有にみられる遅延が起こらないといった 3 点の特徴を持った独自の運行計画生成アルゴリズムを中心としたシステムであり，延べ 30 自治体において実証運行を行ってきた(2011 年 3 月末時点)．

本システムの課題は，画一的な運行計画生成アルゴリズムを基本としているために，地域の細やかな要求に応えきれないことである．たとえば，三重県玉城町では，利用者が何時に到着したいという希望を多少ずらしてでも乗り合いを多く生じるようなアルゴリズムが良いという希望があるのに対し，千葉県柏市のようなサラリーマンや学生を対象としたサービスの場合は，運行の効率性よりも利用者の希望になるべく近い運行計画を生成するアルゴリズムが良いという希望がある．利用者の希望に近い運行計画を作ることと乗り合いを多く生じることは相反する要求であり，トレードオフの問題となる．

そこで，本稿ではパラメータを変えることで，地域に適合した運行を提供する事ができる運行計画生成アルゴリズムの開発および評価を目的とする．

## 2. 開発したオンデマンド交通システム

### 2.1 開発したシステムの概要

開発したオンデマンドバスの概要について図 1 に示す．まず，乗客は Web や電話からデマンド情報(何時に・どこから・どこまで移動したいか？という情報)を計算システムに伝える．このとき，デマンド情報は予約システムを介して伝達される．

乗客のデマンド情報が計算システムに伝わると，計算システムはこの乗客よりも前に予約した他の乗客のデマンド情報と，新しい乗客のデマンド情報を合わせて運行計画を更新する．更新が成功すると，再度予約システムを介して乗客にスケジュールが伝えられる．また，この時更新された新しい運行計画はオンデマンドバスに搭載されている車載システムに伝達され，運転手は常に最新の運行スケジュールを把握することが出来る．乗客はシステムから通知された時刻に指定された待ち合わせ場所で待機する．これによりサービスが成立する．

バスの位置は GPS を用いて常に取得する．乗客の乗降状況や運行状況については，運転手が車載システムを操作して，1 度の発着イベント毎にデータベースに蓄積される．その際に，バスが実際の移動でかかった時間を保存し，データベースに特殊な形式で保存される．これをマイニングすることでデータベースは実移動にかかる時間を導出でき，そのことが移動時間の正確な見積もりに貢献する．また，蓄積された発着イベントや GPS により取得した現在位置を地図上に表示することで，乗客がインターネットを介し，運行状況を確認することができる．

本稿の主眼である運行計画生成アルゴリズムの地域適合化は，次節で述べる．

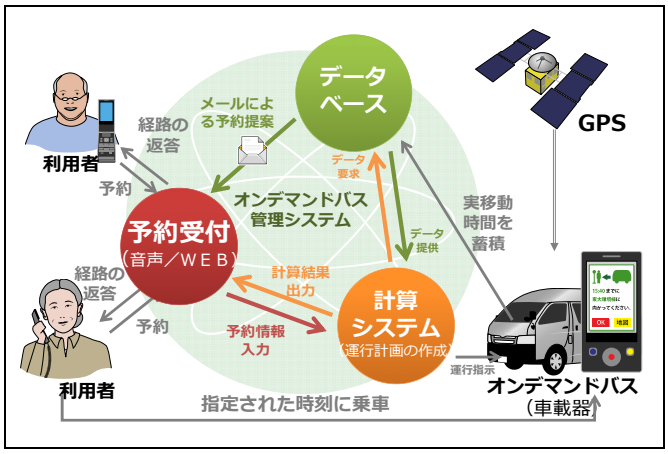


**図1.** 開発したオンデマンド交通システムの概要

連絡先：坪内孝太，東京大学大学院新領域創成科学科人間環境学専攻，〒277-8563 千葉県柏市柏の葉5－1－5 環境棟 283 号室，tsubouchi@is.k.u-tokyo.ac.jp

## 2.2 運行計画生成アルゴリズムの概要

はじめに問題の定式化を行った後，解く問題について示し，アルゴリズムの概要について述べる．

### (1) 問題の定式化

まず，表 1 に問題の定式化必要な変数を示す．

**表 1** 問題を定義する変数

| 変数 | 説明 |
|---|---|
| $N$ | 全乗客数を示す． |
| **R** | 全乗客からなる運行計画情報を示す． |
| $n$ | 対象としている乗客 $n$ を示す． |
| $R_n$ | 乗客 $n$ の運行計画情報を示す． |
| $DPT_n$ | 乗客 $n$ の希望乗車時刻(the Desired pick-up time)を示す． |
| $EPT_n$ | 乗客 $n$ が許容できる乗車時刻の最早時間(the Earliest pick-up time)を示す． |
| $LPT_n$ | 乗客 $n$ が許容できる乗車時刻の最遅時間(the Latest pick-up time)を示す． |
| $APT_n$ | 乗客 $n$ の実際の乗車時刻(the Actual pick-up time)を示す． |
| $DDT_n$ | 乗客 $n$ の希望降車時刻(the Desired delivery time)を示す． |
| $EDT_n$ | 乗客 $n$ が許容できる降車時刻の最早時間(the Earliest delivery time)を示す． |
| $LDT_n$ | 乗客 $n$ が許容できる降車時刻の最遅時間(the Latest delivery time)を示す． |
| $ADT_n$ | 乗客 $n$ の実際の降車時刻(the Actual delivery time)を示す． |
| $+n$ , $-n$ | 乗客 $n$ の乗車(あるいは降車)のイベントを表す． |
| $TT(x,y)$ | 地点 $x$ から地点 $y$ までの移動時間(Travel time)を示す． |
| $p(x)$ | イベント $x$ が生じる地点を示す． 例: $p(+n)$ |
| $Bus_n$ | 乗客 $n$ が降車するバスの番号を示す． |
| $DRT_n$ | 乗客 $n$ が乗り合いなしで移動し，出発地から目的地まで直行したときの直行乗車時間(the Direct ride time)を示す． |
| $MRT_n$ | 乗客 $n$ が許容する最大の乗車許容時間(the Maximum ride time)である． |
| $ST_n$ | 乗客 $n$ が設定するゆとり時間(Slack Time)を示す． |
| $SA_n$ | 乗客 $n$ が設定する希望とのずれ時間(解の探索範囲: Search Area)を示す． |
| $IPT_n$ | 乗客 $n$ に約束した乗車時刻(the Informed pick-up time)を示す． |
| $IDT_n$ | 乗客 $n$ に約束した乗車時刻(the Informed delivery time)を示す． |
| $V$ | 全車両数を示す． |

### (2) 解く問題について

オンデマンド交通の問題は $N$ 人の乗客を $V$ 台の車両に乗せる **R** を導出することを考えるスケジューリングアルゴリズムである．R は以下の式で定義することができる．

$$R = \{R_1, R_2, \cdots\cdots, R_N\} \quad (1)$$

また，各乗客の運行計画を示した $R_n$ は以下のように定義することができる．以下より，運行計画情報は，バスが運行するのに必要な情報であり，乗客 n が乗車する車両番号，乗客 n との待ち合わせ場所に到着する時刻，乗客 n を降ろす時刻の3つの情報から成るものである．

$$R_n = \{ Bus_n,\ APT_n,\ ADT_n \} \quad (2)$$

オンデマンド交通を利用する際に，乗客 $n$ は乗車バス停 $p(+n)$と降車バス停 $p(-n)$を指定する必要がある．さらに，希望乗車時刻 $DPT_n$ か希望降車時刻 $DDT_n$ のどちらかを指定する．スケジューリングの結果，乗客 $n$ に通知乗車時刻 $IPT_n$ と通知降車時刻 $IDT_n$ を伝える．しかし，$IPT_n$ と $IDT_n$ の時刻通りにバスが発着するとは限らない．実際には乗客 $n$ は実際の乗車時刻 $APT_n$ にバスに乗車し，実際の降車時刻 $ADT_n$ にバスから降りる．ただし，$IPT_n$ と $IDT_n$ および $APT_n$ と $ADT_n$ は以下の条件を満たしている．

$$EPT_n \le APT_n (= IPT_n) \le LPT_n \quad (3)$$

$$EDT_n \le ADT_n \le IDT_n \le LDT_n \quad (4)$$

数式(3)は実際の乗車時刻 $APT_n$ と通知した乗車時刻 $IPT_n$ の関係を示している．$APT_n$ と $IPT_n$ に差があってはいけず，$EPT_n$ と $LPT_n$ の間に乗客は乗車する．

数式(4)は実際の降車時刻 $ADT_n$ と通知した降車時刻 $IDT_n$ の関係を示している．とともに $EDT_n$ と $LDT_n$ の間に乗客は降車しなければならず，通知した時刻 $IDT_n$ よりも実際に到着する時刻 $ADT_n$ が遅くなってはいけない．

#### ①初期のタイムウィンドウの設定方法

本アルゴリズムは各乗客に設定されるタイムウィンドウを設定することで解の探索範囲を絞り込むことにある．まず，乗客の予約情報に設定されたタイムウィンドウは以下の通りになる．乗客は希望到着時刻を設定し，それによって以下の式のようにタイムウィンドウが設定される．

$$LDT_n = DDT_n \quad (5)$$

$$EDT_n = LDT_n - ST_n \quad (6)$$

$$DRT_n = TT(p(+n), p(-n), hour(DDT_n)) \quad (7)$$

$$EPT_n = EDT_n - DRT_n \quad (8)$$

$$LPT_n = EPT_n + ST_n \quad (9)$$

#### ②予約完了後に設定されるタイムウィンドウ

予約を完了し通知乗車時刻 $IPT_n$ と通知降車時刻 $IDT_n$ を通知した場合，タイムウィンドウが変化し，さらに制約条件を強めることができる．乗客に一度 $IPT_n$ を通知すると，バスは遅くとも $IPT_n$ に乗客 $n$ を乗せる必要がある．そのため，$EPT_n$ と $LPT_n$ および $APT_n$ すべて $IPT_n$ と同時刻となる．

一方，降車時刻については $IDT_n$ を通知すると $LDT_n$ の値が $IDT_n$ となる．$ADT_n$ は今後の予約状況によって変わるが，$LDT_n$ と $LDT_n$ との間に来ることになる．

$$IPT_n = APT_n = EPT_n = LPT_n \quad (10)$$

$$LDT_n = IDT_n \quad (11)$$

$$EDT_n \le ADT_n \le LDT_n (= IDT_n) \quad (12)$$

また，乗客 $n$ がバスに乗車している時間は $MRT_n$ を超えないことおよび出発バス停が到着バス停よりも後に来ないことから，以下の式が成り立つ．

$$0 \le ADT_n - APT_n \le MRT_n \quad (13)$$

### (3) 運行計画生成アルゴリズム

運行計画生成アルゴリズムは全バスに対して予約を挿入できるかを試みるヒューリスティックなアルゴリズムである．まえならえアルゴリズムの導入にあたって次の 2 つの変数を導入する．

**表 2** 運行計画生成アルゴリズムに用いる変数

| 変数 | 説明 |
|---|---|
| $S(e)$ | イベントのフィージブル関数．あるイベントの状態がフィージブルかを示す．フィージブル関数については詳細を後述する． |
| *TabuList* | 過去の計算リストを指す．一度の予約処理の計算過程でフィージブルでないと判断された **R** が蓄積されている． |

運行計画生成アルゴリズムの解法をフローチャートで示す（図 2）. バスには既に *n-1* 人の予約が成立し，経路が生成されているとする. ここで *n* 番目の乗客が予約をし，乗客 *n* の予約を挿入することを試みる. 初期状態として *n-1* 人までで構成されている経路があり，±1 から±(n-1)までの各イベントにおいて前節で説明したタイムウィンドウが設定されている. まえならえアルゴリズムによって乗客 *n* の実乗車時刻（通知乗車時刻）$APT_n$ $(IPT_n)$と実降車時刻（通知降車時刻）$ADT_n$ $(IDT_n)$とを決定する.

まず $APT_n$ を仮定し，挿入を試みる. 挿入を試みたあと，フィージブル性をチェックし，可能であれば経路を決定とする. 一方，フィージブル性がなければ調整を行い，フィージブルでないイベント *e* に対して別の $APT_e$ を設定する. フィージブルでない場合は，過去の計算リスト *TabuList* を確認し，インフィージブルな運行計画 **R** を確認する. 過去に計算した運行計画 R と同じものが導出された場合，その後の計算も同じ計算が循環することを示しており，これ以上の運行計画計算は無駄となる.

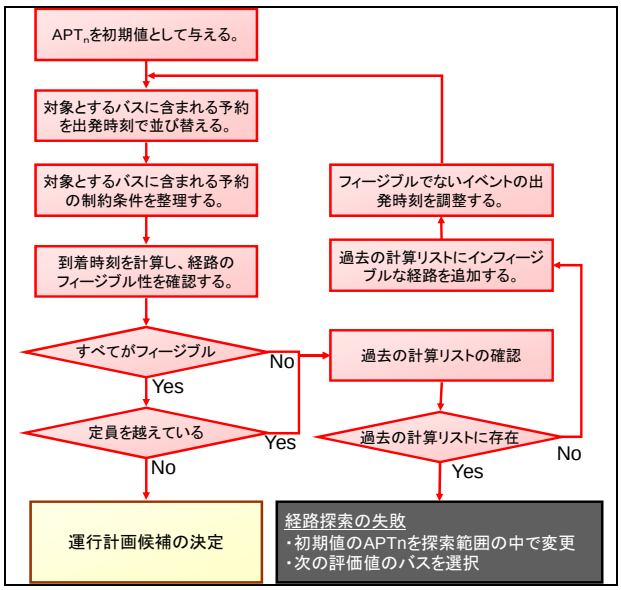


**図 2** 運行計画生成アルゴリズムのフローチャート

もし，過去に同じ計算を行っていなければ，今回導出されたインフィージブルな運行計画を *TabuList* に保存し，$APT_n$ を調整後，運行計画計算を行う.

なお，初期値として与える $APT_n$ は，$SA_n$ に従った探索範囲の中で，1 分ずつずらしながら探索を行う. すなわち $APT_n$ は以下の探索範囲に含まれる.

$$DPT_n - SA_n \le APT_n \le DPT_n + SA_n \qquad (14)$$

フィージブル関数 *S(e)*はイベントの関数がフィージブルかどうかを確認する関数で次の 6 種類の値を取り得る. *S(e)*の詳細は表 3 に示す通りである.

**表 3** フィージブル関数

| *S(e)* | 説明 |
|---|---|
| 0 | イベント *e* において状態はフィージブルである. |
| -1 | イベント *e* において状態はフィージブルでない. +e が-e よりも後のイベントなっている. $(APT_e>ADT_e)$ |
| -2 | イベント *e* において状態はフィージブルでない. 実際の到着時刻 $ADT_e$が最遅到着時刻 $LDT_e$よりも遅い. $(ADT_e>LDT_e)$ |
| -3 | イベント *e* において状態はフィージブルでない. 実際の到着時刻 $ADT_e$が最早到着時刻 $EDT_e$よりも早い. $(ADT_e<EDT_e)$ |
| -4 | イベント *e* において状態はフィージブルでない. 実際の出発時刻 $APT_e$が最遅出発時刻 $LPT_e$よりも遅い. $(APT_e>LPT_e)$ |
| -5 | イベント *e* において状態はフィージブルでない. 実際の出発時刻 $APT_e$が最早出発時刻 $EPT_e$よりも早い. $(APT_e<EPT_e)$ |

フィージブル関数が負の場合には調整を行う. 調整の結果すべてのイベントがフィージブルとなればスケジューリングは終了する. また，検索限界の条件内に解が見つからない場合にも同様にスケジューリングは不成立ということで返す. フィージブル関数によって表 4 のように操作を行う.

**表 4** フィージブル関数が負の場合の調整

| *S(e)* | 調整方法 |
|---|---|
| 0 | 調整不要 |
| -1 | $APT_e^* = LPT_e$ |
| -2 | $ADT_{e-1}^* = ADT_{e-1} - (ADT_e - LDT_e)$ |
| -3 | $ADT_{e-1}^* = ADT_{e-1} + (EDT_e - ADT_e)$ |
| -4 | $APT_e^* = LPT_e$ |
| -5 | $APT_e^* = EPT_e$ |

計算が終了する条件は，最大乗車定員をすべてのイベントで超えていないかつ，すべてのイベントがフィージブルという条件になる. 下記の通りとなる.

$$\sum_e^N S(e) = 0 \quad \cap \quad \sum_e^N C(e) = 0 \qquad (15)$$

ただし，C(e)は，イベント e において定員を超過していないかを確認する関数であり，以下の様に定義される.

$$\begin{cases} C(e) = 0 & in \quad CAP(e)_{Bus_n} \le CAP_{Bus_n} \\ C(e) = -1 & in \quad CAP(e)_{Bus_n} > CAP_{Bus_n} \end{cases} \qquad (16)$$

## 2.3 運行計画生成アルゴリズムの評価（地域適合）

運行計画生成アルゴリズムでは，複数の初期値を与えるため，多くの場合，複数のフィージブルな経路が導出される. したがって，得られたフィージブルな経路の中で最適な運行計画を導出する.

評価には以下の式(16)により求まる評価点 $P_s$ を用いる. ただし *a* は，$0 \le a \le 1$ を満たす実数とし，*a* の値を変えることで地域に適合した運行計画を提示できることを考えた. *s* はフィージブルと判断されたスケジュールとする. *RT* は実乗車時間を示し，*AS* は乗合いの件数を示す. f はそれぞれの値を得点 P に正規変換する関数であり，個々では乗合最大人数と実乗車時刻の最大時間が同じスケールに変わる線形関数を設定した.

$$P_s = a \times f_{AS}(AS_s) + (1-a) \times f_{RT}(RT_s) \qquad (17)$$

*AS* と *RT* は以下の式で定義する. $CAP(e)_v$は，車両 v のイベント e 時の乗車人数を示す.

$$AS_s = \frac{\sum_{e=+N\sim(-N)-1} D(p(e), p(e+1)) \times CAP(p(e))_{Bus_N}}{ADT_N - APT_N} \qquad (18)$$

$$RT_s = ADT_n - APT_n \qquad (18)$$

すなわち，$AS_s$の値が大きければ大きいほど乗り合いが多く発生していることを示し，$RT_s$が大きければ大きいほど，利用者が遠回りをして長く乗車していなければならないことを示している．$a$ の値を調整する事で，“乗り合いの生じやすさ”と“利用者の要求の遵守度”といった相反する要求の妥協点を設定することができる．

## 3. シミュレーション実験による評価

### 3.1 シミュレーションの概要

坪内ら[坪内 2010] が開発したオンデマンド交通導入設計シミュレータを用いてシミュレーション実験を行う．

シミュレーションを始める前にシミュレータは，xml 形式の発生乗客データおよび地域データを読み込み，コンピュータ上に実験環境を生成する．発生乗客データは①出発地，②目的地，③希望到着時刻，④予約時刻といった情報からなり，①出発地，②目的地，③希望到着時刻の分布は地方自治体の担当者が入力した発生パターンと一致している．④予約時刻は，希望到着時刻の 30 分前から 3 時間前の幅でランダムに決定する．

シミュレーションが始まるとコンピュータ内の時間が進行する．ここで予約時刻になると同コンピュータ内の予約システムに予約を行う．その結果，希望到着時刻に最も近い予約（到着予定時刻，車両番号）が返されるが，希望到着時刻と通知される到着予定時刻が±20 分以上の誤差があった場合，予約は不成立となる．本稿では，成立した予約のみによって作られた運行計画を評価し，乗り合いの数や各乗客の平均遠回り時間について調べる．

シミュレータ内には車両エージェントがオンデマンドバス運行を行っており，動的に変更する予約情報に応じて乗客の乗降を行う．この際，単位時間毎に車両の乗車人数等を記録する．

### 3.2 シミュレーションの結果

式(17)の $a$ の値を変えることで，乗合率と顧客の乗車時間という相反する指標の重みづけを変えることができ，それにより地域の要求に適合した運行計画生成アルゴリズムが実現できる．提案手法の妥当性をシミュレーション実験によって示す．

評価には，埼玉県北本市において得られた 1 ヶ月間の実証実験のログデータから一定の需要パターンを作成したものを，インプットデータとして用いた．$a$ の値を変えることによって，違う挙動の示すアルゴリズムになることを確認する．

**表 5** 北本市実証運行の概要

| | |
|---|---|
| 期間 | 平成 21 年 10 月 1 日～12 月 31 日 |
| 運休 | なし |
| 利用時間 | 8:30−17:30 |
| エリア | 北本市内全域 |
| 乗降場所 | 共通乗降場（公共施設，病院，商業施設，金融機関，駅）<br>自宅前（車両が入れない場合はその付近） |
| 車両 | 12 人乗りワゴン車 2 名 |

シミュレーションの結果を表 6 に示す．表中の平均遠回り時間は，乗客一人あたりの実乗車時間から直行移動時間を引いたものの平均値である．表中の乗合のケース数はたとえば $a$=0.75 の時，5 人乗合のケースが 2 件，4 人乗合が 3 件，2 人乗合が 5 件，乗合でないケースが 7 件あることを示す．表を見ると，同じ運行計画生成アルゴリズムを用いているにもかかわらず，$a$ の値を変える事で，期待する運行計画が作られていることが分かる．

**表 6** シミュレーションの結果

| $a$ | 平均遠回り時間 | N 人乗合のケース数 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 0 | 5.8 min. | - | - | - | - | 3 | 1 | 5 | 15 |
| 0.25 | 6.4 min. | - | - | - | - | 2 | 1 | 7 | 14 |
| 0.5 | 6.7 min. | - | - | - | - | 3 | 4 | 3 | 11 |
| 0.75 | 7.3 min. | - | - | - | 2 | 3 | - | 5 | 7 |
| 1 | 8.4 min. | 1 | - | - | 3 | - | - | 2 | 10 |

具体的には，$a$ の値を大きくすれば平均遠回り時間が長くなる反面，大人数での乗り合いのケースが増えている．したがって，地域に最適な a の値を発見することができれば，地域に適合したオンデマンド交通運行計画生成アルゴリズムとなる．

## 4. 結論

オンデマンド交通システムにおける地域の複雑な要求に対応するために，地域適合型の運行計画生成アルゴリズムを開発し，シミュレーション実験によって提案アルゴリズムの有効性を検証した．

開発したアルゴリズムは，作成された複数のフィージブルな運行計画の中から一つを選択する際の評価関数の重み付けをパラメータ $a$ によって変化させることで，地域の複雑な要求に対応するものである．シミュレーション実験では，実際に $a$ を変える事で，生成される運行計画が期待する特徴を有したものとなることを確認し，本手法の有効性を示した．

本稿では運行管理者が重視する乗合いおよび乗客が重視する実車時間という相反する指標を評価関数に採用したが，この評価関数の種類を増やすことでより複雑な地域の要求に対応した運行計画生成アルゴリズムとなることが期待できる．

### 謝辞

本研究を行うにあたり，JST-CREST および JST-ASTEP の資金をいただきました．実証実験をするにあたり，北本市役所様のご助力を賜りました．ここに記して謝意を表します．

### 参考文献

[大和 2008] 大和裕幸，坪内孝太，稗方和夫: オンデマンドバスのためのリアルタイムスケジューリングアルゴリズムとシミュレーションによるその評価，運輸政策研究，Vol.10 No.4, pp002-010，2008.

[Tsubouchi 2011] Kota Tsubouchi, Hiroyuki Yamato, Kazuo Hiekata: Innovative on-demand bus system in Japan, IET Journal Intelligent Transportation Systems, Vol.4, No.4, pp270 – 279, 2011

[Jaw 1986] Jaw, J., Odoni, A., Psaraftis, H., and Wilson, N: Heuristic Algorithm for the Multi-Vehicle Advance Request Dial-A-Ride Problem With Time Windows, Transpn. Res.-B Vol.20B. No.3, pp.243-257, 1986.

[坪内 2010] 坪内孝太，大和裕幸，稗方和夫:オンデマンドバスの導入設計シミュレータの開発と評価，人工知能学会論文誌，Vo.25, No.3, pp.400-403, 2010