寄り道型デマンドバス路線のルート設計に関する手法の検討

坂井　佑理† 　瀧瀬　和樹†† 　浅野　泰仁†† 　吉川　正俊††

† 京都大学情報学科　〒 606–8501 京都府京都市左京区吉田本町

†† 京都大学情報学研究科社会情報学専攻分散情報システム分野　〒 606–8501 京都府京都市左京区吉田本町

E-mail: †{yuuri.sakai,takise}@db.soc.i.kyoto-u.ac.jp, ††{asano,yoshikawa}@i.kyoto-u.ac.jp

あらまし　本研究では寄り道型デマンドバスを活用した高効率な公共交通設計を目標としている．「寄り道型デマンドバス」とはデマンドバスのうち，バス車両が取る運行ルートがあらかじめある程度定められた方式のものを指す．デマンドバスの特徴を活かすことで，より短い走行距離で多くの利用客からのデマンドに応えられる効率の良いバス路線を設計することを目標に設計手法の検討を行った．そのために，現在京都市において実際に固定バス路線が設定されているエリアに着目し，それらと提案手法により設計したデマンドバス路線を比較した．

キーワード　公共交通，デマンドバス，DRT，新交通，域内交通，路線設計

1. はじめに

昨今，地方だけでなく大都市部においてもバス事業者の収益悪化やバス運転手の高齢化・不足が問題となっている (注1)．国勢調査 (注2) によれば中心都市への人口集積は急速に進んでおり，公共交通網維持のためには限られたリソースで多くの利用者を輸送することのできる効率の良いシステムを開発することが必要である．本研究では利用者にとって利便性が高く，事業者にとっても地方都市において優れた収益性を発揮できる可能性がある乗合サービス「デマンドバス」を取り上げ，大型車輌による大量輸送の効率化について論じる．

「デマンドバス」とは利用客からの「いつ（出発時刻）・どこから（乗車希望地）・どこまで（降車希望地）移動したい」というデマンドを予約により受け付け，それに準じてバスの運行ルート・ダイヤをその都度構築する，Demand Responsive Transport (DRT) の一種である．ルート・ダイヤが固定された従来のバスより需要に合わせた柔軟な運行が可能であるとして長年注目され，その方式や運行計画アルゴリズムが数多く研究されており，既に実証実験や実用化の段階に進んでいるサービスも国内外に存在する．例えば東京大学オンデマンド交通プロジェクトが開発したデマンドバス運用システムである「コンビニクル」(注3) は千葉県柏市をはじめとする多くの地域で自治体や診療所などが導入し実運用に至っている．「コンビニクル」は過疎地域・中山間地域をターゲットにしたサービスであり，運用されている車輌の定員や台数も小規模なものとなっている．またイギリスではロンドン交通局によるデマンドバスサービス “Dial-a-Ride” (注4) がサービスを行っている．これは小型バス車輌を用いた無料の会員制乗合バスサービスであり，一般の公共交通機関を利用することが困難な障碍者などを対象にしたものである．

実用例として挙げた 2 つのサービスのように，長年デマンドバスというアイデアに対する評価は，従来の公共交通では採算が取れないような小規模な需要しか見込めない地域・条件への解決策となりうるというものが多かった．しかし近年になって大規模な需要が見込まれる都市部での活用可能性が示され，都市部への導入方法が論じられるようになる．デマンドバスが得意とするデマンドの分布 [1]，鉄道などの大量輸送機関との協調 [2]，1 路線に対し同時に複数台のバス車輌を並行して走らせた場合の運行アルゴリズム [3] など，多くのデマンドを効率的に処理する方法がここ 10 年ほどの間に検討されている．特に野田ら [1] の論文ではマップデータやデマンドデータを人工的に用意した環境を用いて従来の路線バスとデマンドバスをシミュレーションにより比較し，それにより多くのデマンドが発生する都市部において路線バスよりデマンドバスが需要を効率的に処理できる条件が示された．

本研究ではそれらの発展として「寄り道型」と呼ばれるデマンドバスを検討し，実際の環境に基づいて構築した現実的なシミュレーション環境のもと，都市部におけるデマンドバスのさらなる導入可能性を探る．詳しくは後述するが，デマンドバスにおいてバスの走行ルート・時刻表は不定とされることが多い中，寄り道型デマンドバスで曖昧ながら走行ルート・時刻表が事前に設計される．運行事業者にとってあらかじめ運行計画が把握できるということは，出庫・回送や給油といったバス車輌の運用や，各乗務員の行路を決定する上で大きな助けとなる．また利用者から見ても，ダイヤの存在によりいつごろ予約を入れれば待たずにバスに乗れそうか」という予想が立てられる．このような利点にもかかわらず寄り道型デマンドバスの具体的な導入検討はあまりされていない．そこで本研究では事前に設計される曖昧な走行ルート・時刻表の形式として「基本ダイヤ」を独自に考案し，その決定アルゴリズムや基本ダイヤの利用方

(注1)：国土交通省．　バス事業の現状と取り組みについて．http://www.mlit.go.jp/common/001127098.pdf

(注2)：総務省統計局．平成 27 年国勢調査人口等基本集計　結果の概要．http://www.stat.go.jp/data/kokusei/2015/kekka/kihon1/pdf/gaiyou1.pdf

(注3)：東京大学オンデマンド交通プロジェクト．オンデマンド交通・コンビニクル．http://www.nakl.t.u-tokyo.ac.jp/odt/index.html

(注4)：Transport for London. https://tfl.gov.uk/modes/dial-a-ride/

法・評価関数などを検討している．

本稿ではまずデマンドバスの分類を紹介し，その上で提案手法として今回想定するデマンドバスとそれと比較する固定バス路線の設定を説明する．次に行った実験の内容を具体的に述べ，最後に関連研究と今後の方針・課題を挙げる．

2. デマンドバスの分類

デマンドバスの設計方針はルート設計や予約処理手順の違いにより数種類に大別される．提案手法ではルート設計方針を主に「寄り道型」とする．

「寄り道型」とはダイヤ設計の際，サービスを開始する前にバス車輌がとる大まかなルートの予定を立てておく方式を指す．例えばあらかじめ「車輌基地 A を出発し国道 m 号線沿いを走行して車輌基地 B に到着」というようにルートの概要を定めておき，予約を受け付け始めてから実際の運行ルートをその周辺から選択するものである．本研究においては，このあらかじめ決めておく運行計画の概要をを時間的要素も含めて「基本ダイヤ」，予約を受け付けてから決定する実際の詳細な運行計画を時間的要素も含めて「運行ダイヤ」と呼ぶこととする．

寄り道型の主なメリットは実際の運行ルートを決定する際の計算コストをが小さくなることである．後述する予約処理において採用する手順によって事情は多少異なるが，デマンドバスの運行ルート決定に際してはバスの運行スケジュールにルート計算を間に合わせなければならないためリアルタイム性が存在し，計算時間が大きな問題となる．多くの先行研究では交差点などを乗降地点に指定し，それ以外の乗降地点に対する位置的制約は営業エリアのみとする「セミデマンド型」を用いているが，セミデマンド型の場合実際に予約を処理するまでバスの運行ルートはまったく定まらない．対して寄り道型では予約処理前に運行ルートの概要は定まることになる．これにより事前に計算しておくことができる範囲が増えるとともに，車輌や乗務員の行路計画も容易になると考えられる．なお，運行を想定するエリア内において乗降地点の位置的制約をすべて廃したものが「フルデマンド型」と呼ばれる．

3. 提 案 手 法

デマンドバスを用いて固定バス路線に比べ路線網全体でのバスの運行距離を減らすため，複数の固定バス路線をそれより少ないデマンドバス路線で置き換える手法を提案する．この際利用者の不便を招かないよう，利便性は維持ないし向上されなければならないという制約が設けられる．

例えば 2 つの並行する固定バス路線を 1 つの寄り道型デマンドバス路線で置き換えることを考える．「寄り道」をすることで 2 つの固定バス路線の利用者にとって利用可能なバス路線を維持しつつ，Door-to-Door なサービスに近付けることでサービスレベルの向上を目指すことができる．バス路線が「利用可能」であるとはすなわち，利用者の時空間上での座標からバスが走る時空間上での座標までの距離が一定の範囲内であることを指す．空間面では出発地から乗車地までの徒歩移動距離が利用者に不便を感じさせない範囲に収まること，時間面では利用者が

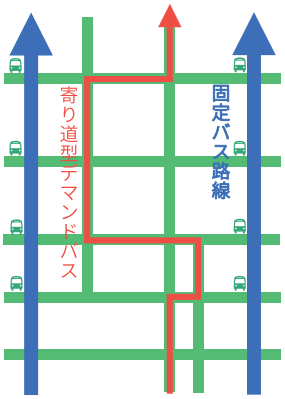

図 1　寄り道型デマンドバスの概略図

乗車地へ到着する時刻とバスが乗車地へ到着する時刻の差，すなわち待ち時間が利用者に不便を感じさせない範囲に収まることを条件として，それらを満たす時空間上の範囲にバスが走行していることをバス路線を「利用可能」であると定義している．さらに元の固定バス路線の運行距離の合計を超えない範囲で増発を行うことで，利用者から見れば高頻度運行化による利便性向上，事業者から見れば減便によるコスト削減を図ることができると考えられる．

このデマンドバスの路線設計に関する処理を行うため，「基本ダイヤ決定アルゴリズム」「運行ダイヤ決定アルゴリズム」と評価関数を提案する．この 2 つは予約受付開始前・後とそれぞれ実行されるタイミングが異なるため，別々のアルゴリズムとした．

3.1 基本ダイヤと運行ダイヤ

「基本ダイヤ」とは「ある程度このルート・時刻表に沿ってデマンドバス路線が設定されている」というデマンドバス路線の大まかなダイヤを表したものであり，過去のデマンドデータを考慮して作成された，未来においても最適と思われる走行エリア・時刻表である．しかし突発的なデマンドの集中などにより過去のデマンドデータと同じように未来においてもデマンドが分布するとは限らないため，基本ダイヤは予約状況に対応してダイヤを変化させるための幅を持たせられている．

「運行ダイヤ」は実際の予約状況を考慮して各バス車輌に対し走行のための詳細なルート・時刻表を定めるものである．すなわち「いつ，どの車輌がどの道を通るか」という情報を表したものとなる．基本ダイヤで表された幅の中から現在の予約を処理するのに最適なダイヤを生成し，これを運行ダイヤとする．

3.2 制　　約

a) デマンド受理の定義

デマンドバス・固定バス路線ともに，利用者が出発地を出発してからバスにピックアップされるまでの所要時間が 20 分以下である場合に「デマンド受理」とする．

b) デマンドバス路線の総延長

本研究では固定路線バスに比べ少ない路線数で利便性を維持できるデマンドバス路線の設計を行うため，デマンドバス路線の総延長は対応するエリアの固定路線バスに比べ短くなければならない．

3.3 評価関数

デマンドバスや固定路線バスの設計・評価を行っている先行研究は数多く存在する．野田ら [1] は設計したデマンドバスと固定路線バスの評価軸として「デマンドの平均達成時間」を用いている．これは利用客がピックアップを希望した時刻から実際にピックアップされるまでの経過時間を示したものである．また，本研究とは異なった形で「寄り道型」の実装方法を定義し寄り道型デマンドバスの並列運行手法を提案した上杉ら [3] の研究では，評価軸として「デマンドの受理率」を用いている．デマンドバスの予約受付においては，利用者が希望したピックアップ時刻までに乗車希望地にバス車輌が到達するようなダイヤが構築できない場合などに予約が拒否され，「不受理」となることがある．そのため上杉らの研究ではその「受理率」の最大化を評価軸に置いてダイヤ決定アルゴリズムを設計した．大量輸送機関である鉄道に協調する形でのデマンドバス運行について論じた上原ら [2] の研究では「平均旅行時間」，すなわち乗換え時間を含めた出発から到着までの所要時間の平均を用いている．平塚ら [4] はデマンド受理率に加え，利用者の平均犠牲量（運賃＋所要時間 (分) × 40 円），バスの台時間あたりの収入の 3 項目についてバス台数との関係性を論じている．また域内交通向けに固定バス路線網の設計を行った Sadrsadat ら [5] は設計されたバス路線を利用可能である人数の都市人口に対する割合を表す「人口カバー率」や公共交通が設定されることによる全体の旅行時間の減少量，バス路線網の総延長や運行コストなどを算出し，設計されたバス路線網により産出可能な総合的な利益を求めている．

評価軸は「利用者から見た利便性」に関するものと，「事業者から見た採算性」に関するものの 2 種類に分類される．本研究では実際にバス事業者による運行への応用が可能であるサービスの設計を視野に入れているため，この 2 種類両方を考慮に入れる必要がある．先行研究で挙げられた評価軸とこの条件を鑑み，本研究では以下の 3 つの評価軸を用いる．

a） 総走行距離あたりのデマンド受理数

「総走行距離」とは路線網全体での 1 日のすべてのバスの走行距離の合計である．本研究においては，デマンドバスの活用による路線網の効率化の軸として全体の便数の削減に注目しているため，少ない総走行距離でより多くのデマンドを受理する路線網が優れていると評価するためにこの項目を設定する．

b） 利用客の平均歩行時間

「平均歩行距離」とは利用者が希望した出発点（交差点でなくても構わない）からピックアップ地点までの歩行時間の平均である．またデマンドバスの利点の一つとして「出発地のドアを出てすぐ車輌に乗車し，到着地のドアの前で降車できる」Door-to-Door 輸送が挙げられる．乗降地点を交差点に限定するセミデマンド型を採用するため完全な Door-to-Door にはならないが，歩行時間が短く Door-to-Door に近い輸送が可能な路線網を高く評価するためにこの項目を設定する．

c） 利用客の平均待ち時間

「平均待ち時間」とは利用者が希望した出発時刻から，ピックアップされるまでの待ち時間の平均を指す．

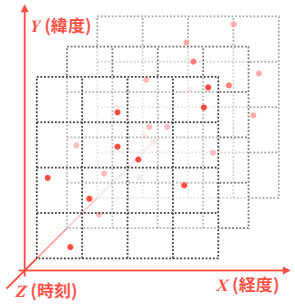

図 2　デマンドが配置された時空間グリッドの概略図

3.4 基本ダイヤ決定アルゴリズム

基本ダイヤは荒い時空間グリッド上に配置されたデマンドを効率的にピックアップできるルート・時刻表を探索することで得られる，通過するグリッドの順序付き集合である．基本ダイヤに含まれる要素のうち，説明のために空間的な要素を「基本ルート」，時間的な要素を「基本時刻表」と呼ぶことがある．「荒い時空間グリッド」とするのは基本ダイヤはあくまでデマンドバス車輌が則るダイヤを大まかにのみ表すものだからである．基本ダイヤの設計においてはバスが通るグリッド内に配置されている利用者を概ねピックアップできるものと考える．ルート設計には遺伝的アルゴリズム (GA) の亜種である人工蜂コロニー (Artificial Bee Colony, ABC) [6] アルゴリズムを用いる．

また，このアルゴリズムにより決定された基本ダイヤの評価を行うために，基本ダイヤ設計時に見積もられたデマンド受理数と実際の運行ダイヤのデマンド受理数の比を算出する．基本ダイヤの想定デマンド受理数は通過するグリッド上のマス内のデマンドをすべて受理できるものと仮定して算出されるため，実際のデマンド受理数よりも高い値が示される．このズレがどれほどであるかを調べるため，この基本ダイヤをもとに後述の運行ダイヤ決定アルゴリズムを用いて運行ダイヤを作成し，実際の運行でのデマンド受理数を算出して，この 2 つの比を基本ダイヤに対するデマンド受理数の補正係数とする．これはデマンドバス路線と固定バス路線を比較する際に，実際の予約・運用なしでも基本ダイヤによるデマンド受理数の見積もりのみでデマンドバス路線の評価値を推定できるようにするためのものである．

3.5 人工蜂コロニーアルゴリズム（ABC 法）

人工蜂コロニー (Artificial Bee Colony, 以下 ABC 法と記載) アルゴリズムとは前述の通り遺伝的アルゴリズム (GA) の亜種であり，群知能の一種である．ABC 法は「収穫・追従・偵察」の 3 つのフェーズからなり，これらを繰り返すことでより良い解を探索する．この節では基本的な ABC 法の流れについて説明する．

a） 初　期　化

初期解のセット $\mathbf{x} = \{x_1, x_2, ..., x_N\}$ を作成し，その中で最も評価値 $f(x_i)$ の高い解を現時点での最適解 x_{best} とする．それぞれの解 x_i には更新試行回数 c_i が紐付けられており，初期値は 0 である．

b） 収穫フェーズ

収穫フェーズで更新を試みる解のセット $\mathbf{x_e} \subset \mathbf{x}$ をランダムに選択し，$\mathbf{x_e}$ の各要素に対して更新を試みる．なお $\mathbf{x_e}$ のサイズは N_e と定める．

$x_i \in \mathbf{x_e}$ である解 x_i に対し $i \neq j$ である $x_j \in \mathbf{x}$ をランダムに選択し，解候補 v_i を次の式に従って作成する．

$$v_{ik} = x_{ik} + \phi(x_{ik} - x_{jk}) \tag{1}$$

この操作は一般的な GA における「交叉」にあたるものであるため，本稿ではこのような場合を「x_i と x_j の交叉」と呼ぶ．k はランダムに決定された次元番号であり，x_{ik} は解 x_i の k 次元目を表す．ϕ は $-1 < \phi < 1$ の乱数である．この解候補 v_i に対し $f(v_i) > f(x_i)$ が成り立つ，すなわち解候補 v_i の評価値が既存の解 x_i よりも高い場合，$x_i = v_i$ とし解 x_i を更新し，更新試行回数 c_i を 0 にリセットする．$f(v_i) \leqq f(x_i)$ である時，すなわち既存の解 x_i の方が解候補 v_i よりも評価値が高い場合は解を更新せず，x_i の更新試行回数 c_i を 1 増やす．

この操作を $\mathbf{x_e}$ の要素すべてに対して行う．

c） 追従フェーズ

追従フェーズでは更新を試みる解のセット $\mathbf{x_o} \subset \mathbf{x}$ を各解の評価値に基づいて決定された確率に従ってルーレット戦略により選択し，$\mathbf{x_o}$ の各要素に対して更新を試みる．なお $\mathbf{x_o}$ のサイズは N_o と定める．解 x_i が選択される確率 p_i は以下の式で求められる．

$$p_i = \frac{f(x_i)}{\sum_{n=1}^{N} f(x_n)} \tag{2}$$

このようにして得られた $\mathbf{x_o}$ の各要素に対して，収穫フェーズと同様に更新を試みる．

d） 偵察フェーズ

偵察フェーズでは更新試行回数が更新試行回数制限 C_l を超過した解を捨て，ランダムに生成した解に入れ替える．すなわち解のセット $\mathbf{x}$ の各要素 x_i について $c_i > C_l$ となったものを入れ替える．

この 3 フェーズを繰り返し行い，より評価値の高い解を求めていく．終了条件は以下の通りである．

e） 終 了 条 件

再帰回数があらかじめ定められた世代数 G に達することが ABC 法の終了条件となる．その時の最適解 x_{best} が ABC 法全体の最適解として出力される．

3.6 今回実装した基本ルート決定アルゴリズム

今回は時間的情報を無視し，出発地座標を元に空間グリッド上に配置されたデマンドの点をより多く受理できるルートを探索するプログラムを ABC 法を用いて作成する．空間的情報のみを扱うため，基本ダイヤのうち基本ルートの一部のみを設計することになる．またルートの起点と終点は固定する．

今回扱うデマンドデータにおいて出発地点が存在する範囲をすべてカバーできる長方形のエリアを対象とする．これを南北方向と東西方向にそれぞれ 12 分割し，図 3 のようなグリッドを設定した．1 マスの南北方向の長さは約 344m，東西方向の長さは約 345m である．

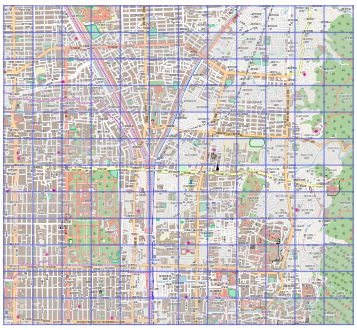

図 3　基本ルートを設定する範囲とグリッド設定

0	0	0	11	29	36	28	28	15	15	1	2
0	1	15	108	355	512	335	229	242	108	22	2
1	7	20	169	542	724	491	412	484	265	76	9
0	7	19	81	292	345	289	390	568	361	94	5
0	3	24	135	251	391	317	381	485	309	83	10
3	21	152	443	704	859	611	560	635	414	100	15
2	36	220	644	845	761	503	383	375	204	43	4
0	7	52	158	264	402	331	373	368	200	61	2
0	0	7	48	173	303	320	318	346	205	48	4
0	0	2	51	213	436	458	291	175	90	24	3
0	0	4	40	176	320	310	127	60	20	2	1
0	0	1	1	12	37	26	10	2	1	0	0

図 4　デマンドデータの 2 次元配列表現

0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	1	1	1	1	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0	0

図 5　基本ルートの 2 次元配列表現
（ルート形状配列）

	0	1	2
0	0	0	1
1	1	1	1
2	1	1	1

図 6　入り組んだルートの例

各マスについて，そのマスに出発点が属するデマンドの数を表示したものが図 4 である．生成された基本ルートの評価値は「通過したマス内に出発点が存在するデマンドの数の合計」とし，これを「受理可能性デマンド数」と呼ぶ．

実装するアルゴリズムにおいては，生成されたルートは図 5 のように表される．すなわち，1 が記されているマスでルートの形状を示す．これを「ルート形状配列」と呼び，記号 route で表す．しかしルート形状配列のみでは入り組んだルートになった際，マスの通過順序が一意に定まらないことがある．

図 6 に示したルート形状配列では例えば，(0, 2) → (1, 2) → (2, 2) → (2, 1) → (1, 1) → (1, 0) → (2, 0) や (0, 2) → (1, 2) → (1, 1) → (1, 0) → (2, 0) → (2, 1) → (2, 1) など複数の通過順序が考えられる．これを一意に定めるため，0/1 の 2 次元配列であるルート形状配列の他に移動方向を表す列を保存しておく．

例えば最初に示した順序であれば [down, down, left, up, left, down] という列になる．これを「移動順序配列」と呼び，記号 direction で表す．

これら 2 つの配列を用いて，アルゴリズムにおける解は $x_i = (route_i, direction_i)$ と表される．ルート形状配列 $route_i$ は以下で説明する ABC 法でのルート最適化で主に用いるものである．一方，移動順序配列 $direction_i$ は生成されたルートを表現するためのものであり，ABC 法ではルート生成の度にルート形状配列とともに出力される以外に用いられることはない．

3.5 節の ABC 法を以下のように基本ルート決定アルゴリズムの実装に適用した．

各解 $x_i \in \mathbf{x}$ は 1 つの基本ルートとした．そして今回の基本ダイヤ最適化に用いる評価関数として，「受理可能性デマンド数」を定義した．これは基本ダイヤが通過するマス内に存在するデマンド数をカウントしたものである．受理可能性デマンド数の最大化が ABC 法における目的関数となる．

初期化においては N 個のランダムなルートを生成し，これを初期解のセットとする．収穫フェーズと追従フェーズにおける解 x_i と解 x_j の選択は標準的な ABC 法と同様に行う．しかし式 (1) が示すように，標準的な ABC 法では解同士を加算・減算した結果もまた解とならなければならない．今回は各解のルート形状配列どうしに交叉操作を行うが，0/1 の 2 値しか値を持たないルート形状配列ではそれが成り立たないため，交叉の方法を以下のように変更した．

解 x_i のルート形状配列 $route_i$ と解 x_j のルート形状配列 $route_j$ を加算し，1 以上の値で囲われている部分にチェックを入れる．このチェックを入れられた範囲内でランダムにルートを生成し，解候補 v_i とする．交叉操作により生成された解候補 v_i で解 x_i を更新するかの判定は，標準的な ABC 法と同様に v_i と x_i の評価値を比較することで行われる．

ABC 法における制御パラメータには以下のものがある．

a） 世　代　数

「収穫・追従・偵察」の 3 フェーズ全体の最大再帰回数を示し，記号 G で表される．再帰回数が G に達することが ABC 法の終了条件である．

b） 解のセットサイズ

解のセット $\mathbf{x} = \{x_1, x_2, ..., x_N\}$ の要素数 N を示す．

c）「収穫蜂」「追従蜂」の個体数

収穫フェーズでランダムに選択される解の個数 N_e と，追従フェーズでルーレット戦略により選択される解の個数 N_o にあたる．一般に $N_e = N_o$ とされる [6]．この呼称は ABC 法が蜂が餌を探索する動きを参考にして設計されたことに由来する．すなわち，収穫フェーズと追従フェーズで解を選択することを蜂が蜜を求めて花を選択する行動になぞらえ，それぞれのフェーズでの「解の選択数」を「探索に従事する蜂の個体数」と表現したものである．

d） 更新試行回数制限

各解 x_i に紐付けられた更新試行回数 c_i の上限を示し，記号 C_l で表される．$c_i > C_l$ となった解 x_i は偵察フェーズで新たな解に入れ替えられる．

3.7　基本ダイヤにおける時間的情報の決定方法

3.6 節で述べたように今回実装したアルゴリズムでは基本ダイヤのうち基本ルート，すなわち空間的情報のみを決定するに留まっている．本来，提案手法で定義した基本ダイヤは図 2 のような時空間グリッド上での移動順序である．これによりバス車輌の速度や時間帯によりルートを変更する路線などの表現が可能になる．今回は実装に至らなかったが，時空間グリッド上での基本ダイヤ決定にも ABC 法が応用できると考えられる．すなわち時空間グリッド状に配置したデマンドをより多くピックアップすることを最適化の指標として，3 次元配列で表現されたルート形状配列を ABC 法にて更新していくことで，完全な基本ダイヤの最適化も可能であると考えている．

3.8　運行ダイヤ決定アルゴリズム

本稿では 3.6 節で述べた基本ダイヤの一部を決定するアルゴリズムのみ実装したが，本来であれば提案手法に挙げた「運行ダイヤ」を決定するアルゴリズムも必要である．しかし運行ダイヤの決定には完全な基本ダイヤが必要なので，今回は運行ダイヤ決定アルゴリズムは実装していない．

4.　実　　験

実際運行されている固定バス路線に対してデマンドバスを設定しその利点を明らかにするため，以下の実験を計画した．今回の実験では，並行する 3 本の大通り（京都市の川端通・東大路通・白川通）に固定バス路線とデマンドバス路線を別個に設定・比較する．固定バス路線はそれぞれの大通りを南下する 3 路線からなるものとする．これらの通りには実際に南北方向に京都市バス・京都バスの路線が運行されている．一方のデマンドバス路線はこれら大通り周辺のエリアを南下する 1 路線と設定する．固定バス路線については 3.3 節で述べた評価軸による評価値，デマンドバス路線については受理可能性デマンド数をそれぞれ求め，固定バス路線とデマンドバス路線の比較・考察を行った．

4.1　使用データ

「基本ダイヤ決定アルゴリズム」「運行ダイヤ決定アルゴリズム」に入力するマップデータと過去のデマンドデータ，そして今回比較対象とする固定バス路線に関するデータを用意した．

マップデータとしてはバス路線を設定する対象エリアの OpenStreetMap データから自動車が通行可能な道路のみを抽出し，無向グラフとした．

デマンドデータとしては，京都市内を対象とした乗換検索サービス「歩くまち京都」の検索ログ（2016 年 7 月 1 日）を加工して用いた．その際，京都市交通局が公表している各バス停留所の 1 日の乗降客数と，検索ログから作成した各停留所の各時間帯におけるデマンド数のヒストグラムからデマンドの母数と分布を推定した．また検索ログに含まれる希望乗車地は鉄道駅・バス停留所の座標であり，利用客の移動開始点（出発地座標と呼ぶ）は含まれていない．そのため，各デマンドについて正規分布を用いて希望乗車地の周囲から出発地座標を選択し，利用者が乗車地まで徒歩で移動する様子を再現している．なお用いるデマンドデータに現れる停留所の配置が南北方向に

表 1　固定バス路線の評価値

走行距離 1km あたりのデマンド受理件数	19.5 件
デマンドを受理できた利用者の平均歩行時間	266 秒
デマンドを受理できた利用者の平均待ち時間	561 秒
デマンド受理数	20517 件
デマンド不受理数	4334 件
デマンド不受理となった利用者の平均歩行時間	359 秒
デマンド不受理となった利用者の平均待ち時間	2309 秒
1 日間での総走行距離	1050km
デマンド受理率	83%

は密，東西方向には疎であるため，南北方向（緯度）は平均 0, 標準偏差 200 の正規分布，東西方向（経度）は平均 0, 標準偏差 400 の正規分布を用い，これを中心である停留所からの座標差とした．これにより各停留所の周囲のデマンドは東西方向に伸びた横長の分布になっている．出発時刻は時間帯別ヒストグラムによる分布から各時間帯の利用客数を推定し，各時間帯においてランダムに生成した．

なお，今回の実装では利用客の行き先は考慮していないため，生成したデータに含まれるのは「出発時刻」「出発地座標」の 2 つである．「出発地座標」は交差点上であるとは限らない．

4.2　固定バス路線

今回対象にする川端通・東大路通・白川通には京都市バス（京都市交通局）・京都バスによりバス路線が設定されており，その停留所とダイヤを利用する．各停留所座標は最寄りの交差点に近似している．京都市バスについては，車輌に搭載された GPS により取得された各車輌の時空間情報から，停留所とみなされた交差点におけるバスの通過時刻を求め，これを時刻表とした．京都バスについては公式時刻表 (注5) から実験対象エリアを南向きに走行する路線の時刻表を抜粋して用いた．

設計した基本ダイヤと比較するため，実験対象エリア内の現在のバス路線に対し 3.3 節で示した各指標の評価値を求めた．デマンドデータは基本ダイヤの設計に用いたものと同じものを使用している．

4.3　基本ダイヤ決定アルゴリズムの実装

今回提案する「基本ダイヤ」の評価を行うため，3.6 節に示した基本ダイヤを簡単にしたものを最適化するプログラムを実装した．

まず，ABC 法における 4 つのパラメータを決定するために，それぞれのパラメータについてそれ以外のパラメータを固定する方法で最適な値を大まかに求めた．以下がその実験結果である．なお，実験はすべて同じ条件で 10 回 ABC 法を実行し，受理可能性デマンド数と実行時間の平均値を取っている．

e）世代数 G

図 7 はセットサイズ $N = 50$，「収穫蜂」「追従蜂」の個体数 $N_e = N_o = 15$，更新試行回数制限 $C_l = 50$，限界ルート長 $L = 25$ の設定の上で ABC 法を世代数 $G = 1000$ まで実行し，各世代における受理できる可能性のあるデマンド数をグラフ化したものである．太線で描画されているものが 10 回の平均値

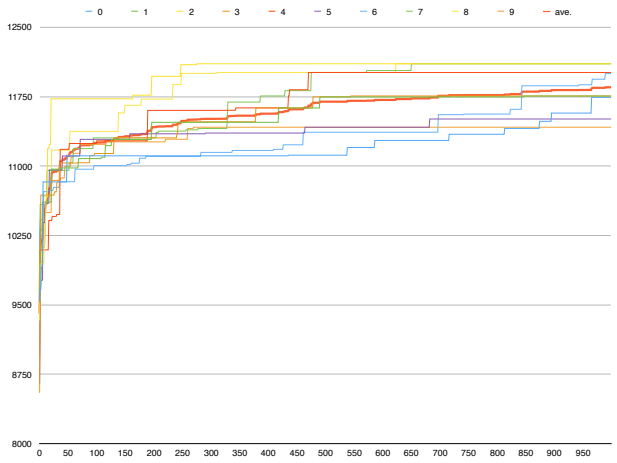

図 7　世代数 G によるデマンド受理数の推移

表 2　セットサイズ N による受理可能デマンド数と実行時間の推移

セットサイズ N	受理可能デマンド数	実行時間 [s.]
50	11604.9	117
100	11609.5	174
150	11436.3	245
200	11467.7	311
250	11400.9	373

である．グラフより 100 世代程度で解の評価値の増加率は鈍化することが分かる．しかし $G = 1000$ の実行時間は平均 395 秒であり，これは十分現実的な実行時間と言える．100 世代以降でも緩やかながら解の最適化は進んでいるため，世代数 G に関しては実行時間とのトレードオフを鑑みた上で大きい値を設定すればよいと考えられる．

以降，世代数 G は十分に大きいと仮定して単位世代あたりの性能と各パラメータとの関係を考察する．「単位世代あたり」とは 1 世代に限らず，ある長さの世代数における最適化性能のことを指すこととする．

f）解のセットサイズ N

世代数 $G = 100$，「収穫蜂」「追従蜂」の個体数 $N_e = N_o = 40$，更新試行回数制限 $C_l = 50$，限界ルート長 $L = 25$ の設定の上で表 2 の通りセットサイズ N を変更し，各セットサイズにおける受理可能デマンド数と実行時間の推移を観察した．セットサイズが大きくなるということは，解 $x_i = \mathbf{x}$ が収穫・追従フェーズにおいて選択される可能性が低くなるということであるため，実行時間や解の配列サイズなどを考慮してセットサイズ N を決定し，それに合わせて単位世代あたりでの更新試行回数に関わる「収穫蜂」「追従蜂」の個体数 N_e, N_o を決定する必要がある．

g）「収穫蜂」「追従蜂」の個体数 N_e, N_o

ABC 法を提案した Karaboga は $N_e = N_o$ と設定しているため [6]，本研究でもそれに倣うこととする．表 3 は世代数 $G = 100$，セットサイズ $N = 50$，更新試行回数制限 $C_l = 50$，限界ルート長 $L = 25$ の設定の上で「収穫蜂」「追従蜂」の個体数 N_e, N_o を変更し，受理可能性デマンド数と実行時間を求めたものである．表 3 から，$N = 50$ の場合は $N_e = N_o = 20$ 程度が適していると考えられ，N_e, N_o は N の 0.4 倍程度に設定

(注5)：バス株式会社．時刻表｜京都バス株式会社．http://www.kyotobus.jp/route/timetable/

表 3　「収穫蜂」「追従蜂」の個体数 N_e, N_o による受理可能デマンド数と実行時間の推移

個体数 N_e, N_o	受理可能性デマンド数	実行時間 [s.]
5	10981.2	70
10	11079.1	77
20	11456.8	84
30	11402.2	122
40	11604.9	117
50	11494.8	112

することとする．

h）　更新試行回数制限 C_l

更新試行回数 C_l は各解が収穫・追従フェーズにおいて選択される回数を考慮して設定される必要がある．極端な例を挙げれば，すべての解が C_l 回以下しか選択されないならば偵察フェーズは意味を失ってしまう．偵察フェーズは局所解からの脱出や探索範囲の拡大に寄与するものであり [7]，偵察フェーズにてある程度解が捨てられなければならない．ある解 x_i が収穫・追従フェーズで選択される単位世代あたりの回数はセットサイズ N と「収穫蜂」「追従蜂」の個体数 N_e, N_o に従うものと考えられる．表 4 は世代数 $G = 100$，セットサイズ $N = 50$，「収穫蜂」「追従蜂」の個体数 $N_e, N_o = 15$，限界ルート長 $L = 25$ の設定の上で更新試行回数制限 C_l を変更し，受理可能性デマンド数と実行時間を求めたものである．この設定では $C_l = 50$ 程度で最も高い評価値が出ている．1 世代で 50 個の回のうちのべ 30 個が選択されるという設定であるので，$\frac{N_e+N_o}{N} : C_l = \frac{30}{50} : 50 = 3 : 250$ を大まかな目安とする．

表 4　更新試行回数制限 C_l による受理可能デマンド数と実行時間の推移

更新試行回数制限	受理可能性デマンド数	実行時間 [s.]
7	11216.6	274
12	11251.3	174
25	11255.2	106
40	11400.7	81
60	11455.4	70
80	11323.6	73
150	11374.6	69

また上の 4 つのパラメータ以外に，基本ダイヤにおけるルート長・起点・終点に関するパラメータが存在する．

i）　限界ルート長 L

限界ルート長 L は ABC 法の直接のパラメータではないが，受理可能性デマンド数に直結するパラメータであるためここで考察する．ルート長は長ければ長いほど多くのマスを回れるため受理可能性デマンド数は大きくなるが，3.2.2 節で示したように問題設定としてルート長は固定路線バスより短くなければならない．今回のグリッドでは南北方向が 12 マスであり，固定バス路線はグリッド上では合計路線長 28 マスである．よって $12 \leqq L < 28$ の範囲で，決定される基本ダイヤに求める効率性（1km あたりのデマンド受理率に相当）を考慮して設定されるべきである．

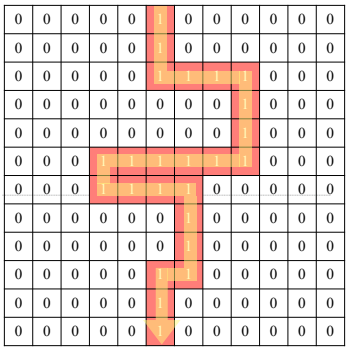

図 8　出力された基本ダイヤ（受理可能性デマンド数 12106）

j）　ルートの起点・終点

今回の実装では ABC 法の解集合 $\mathbf{x}$ に含まれるすべての解について起点と終点は等しいと定めた．その上で南向きに走行する固定バス路線 3 路線をデマンドバス路線 1 路線で置き換えるということと，図 4 に示したデマンドデータの分布を踏まえ，起点を (0, 5)，終点を (11, 15) とした．

以上の考察を踏まえ，パラメータは表 5 のように設定した．

表 5　ABC 法におけるパラメータ設定

世代数	$G = 500$
解のセットサイズ	$N = 50$
「収穫蜂」「追従蜂」の個体数	$N_e = N_o = 20$
更新試行回数制限	$C_l = 70$
限界ルート長	$L = 24$
起点・終点	(0, 5), (11, 15)

表 5 のパラメータを用いて ABC 法を 10 回実行し，最も評価値の高かった出力を解とした．受理可能性デマンド数は 12106 である．図 8 はルート形状配列の上に移動順序配列により表された移動方向を描画したものである．

4.4　考　　察

今回行った実験では基本ダイヤについての実装が不完全であり，4. 2 節で固定バス路線に対し求めたような詳細な評価値が算出できない．そのため「受理可能性デマンド数」という形で，運行ダイヤによっては受理の可能性があるデマンド数をカウントした．この基本ダイヤを元に作成した運行ダイヤでのデマンド受理数はこれより小さくなる．固定バス路線のデマンド受理数は 20517 件であったので，単純に比較しただけでは固定バス路線の方が優れているということになる．

しかし出力された基本ダイヤのルート長はグリッドにして 24 マスであるので，1 マスの辺の長さは約 350m であることを考えると走行距離は 1 回の運行につき 10km 程度になると推測される．固定バス路線の 1 日の総走行距離は 1050km なので，本数との兼ね合いを考慮に入れても路線数を 1 本増やすか，グリッドを粗くして「寄り道」の幅を広げることでデマンドバス路線のカバーエリアを拡げるなどの余地が残されていると考えられる．それに今回は時間的情報を取り扱わなかったため，基本ダイヤは 1 日中同じルートを走り続けるというものになっ

てしまっている．本来であれば時間帯によってデマンドの多いグリッドに路線を切り替えるべきであるので，その点においても改善の余地がある．加えて「受理可能性デマンド数」の定義のために，受理可能なデマンドの出発地座標は 350m 四方のグリッドに縛られている．もしバス車輌がマスの境界近くを走行した場合，隣接するマスに存在する利用者の中には当該のバス車輌の至近にいる者がいる可能性もある．そういった可能性を検討するためには運行ダイヤ決定アルゴリズムの実装まで完了した上で，基本ダイヤ決定時に想定された受理可能なデマンドの分布と，実際の運行で受理可能であったデマンドの分布を比較するなどの手順が必要である．

本研究における貢献点としては，基本ルート決定アルゴリズムへの ABC 法の適用において適切なパラメータ設定に成功したことである．各パラメータについてパラメータの値と評価値，実行時間を比較検討した上でパラメータを決定し基本ルート決定アルゴリズムを実行したところ，それまでで最も高い評価値を持つ基本ルートが得られた．これにより交叉操作を解の形状に合わせて変更し，適切なパラメータ設定を行うことで，ルート設計といった 2 次元配列による解を持つ最適化問題に対しても ABC 法が応用可能であることが示された．同様に 3 次元配列の解に対しても ABC 法の応用可能性があると考えられる．

以上のことを鑑み，今回の実験では具体的な寄り道型デマンドバスの有用性は示すことができなかったものの，実装したアルゴリズムには改善の余地が大幅に残されており，結論を出すためにはさらなる検討が必要であると考察した．またルート設計においては ABC 法の応用が有効であり，提案手法における基本ダイヤの最適化に対しても ABC 法が応用できる可能性が示された．

5. 関 連 研 究

5.1 寄り道型デマンドバスの経路決定

1 路線の 1 回の運行について寄り道型・リアルタイム処理方式のデマンドバスを実装する試みは上杉ら [3] が行っている．K 台の車輌が同時に同方向に並行して走行することでマップ上に分散した利用者を効率的にピックアップし，1 路線によるカバーエリアを拡げていることが特徴的である．なおここでの「寄り道型」は 1 運行における始点と終点を定めその間を走行するのに掛けられる時間に制限をかけることで，2 点間の最短経路からの迂回をある程度までに抑えるというものである．

5.2 GA を用いた固定バス路線網の設計

1 都市に対する固定バス路線網の設計は Sadrsadat ら [5] がイラン第二の都市マシュハドを対象に遺伝的アルゴリズム (GA) を用いて行っている．1 つの路線網を 1 つの染色体，1 つのバス路線を 1 つの遺伝子で表し，遺伝子が持つオン／オフの値によりバス路線網を表現する．ここで染色体とは遺伝子の集合であり，染色体 j の遺伝子 i の値はバス路線網 j においてバス路線 i が採用されているかどうかを示している．目的関数は（利益）−（コスト）の最大化としている．この研究のポイントとして目的関数が簡易であり高速に動作することが挙げられている．

5.3 デマンドバスシミュレーション

野田ら [1] は大都市圏を想定したデマンドバスシミュレーションを行い，デマンドバスが得意とするデマンド分布の形状について考察し，またデマンド頻度が高いほど固定バス路線に比べデマンドバスが有利になることを示した．上原ら [2] はデマンドバスの都市圏での運用という設定で，那覇都市圏の通勤時間帯においてデマンドバスと大量輸送機関である鉄道を連携して運用することを考察している．こちらでも利用者数の増加に伴い旅行時間が改善されていくことが示されており，大規模な需要に対するデマンドバス活用の可能性が示唆されている．

6. 今後の方針・課題

本稿では簡単のために狭い範囲の 1 方向のみで利用者の行き先を考慮しないものと問題を設定し，基本ダイヤの決定においてはデマンドの時刻を考慮せず空間上の情報のみを利用して 2 次元的な設定のみを行った．基本ダイヤを用いての運行ダイヤの決定アルゴリズムの設計，基本ダイヤの時空間上への拡張などが現段階の問題設定においても課題として残されている．段階的に実装の幅を広げていくことで大幅な改善が可能であると考えられるため，そのための具体的な手法を考案しよりデマンドバスの特徴を活かしたダイヤ設計を目指すことが第一の方針である．まずはその点を解決し，将来的には域内交通全体の路線網で固定バス路線とデマンドバス路線の設計・比較検討をし，都市部におけるデマンドバス導入への考察を深めたいと考えている．

謝　　辞

本研究は JSPS 科研費 JP17H06099, JP15K00423 の助成を受けたものです．ここで心より感謝申し上げます．

文　　献

[1] 野田五十樹, 篠田孝祐, 太田正幸, and 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. **情報処理学会論文誌**, 49(1):242–252, 2008.

[2] 上原和樹, 赤嶺有平, 當間愛晃, 根路銘もえ子, and 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**, 56(1):46–56, 2015.

[3] 上杉健太郎, 渡邉豊英, and 向直人. K-means 法を用いた寄り道型デマンドバスの並列運行手法, March 2008.

[4] 平塚順 and 大和裕幸, editors. **シミュレーションに寄るオンデマンドバスの利便性評価**. 東京大学オンデマンド交通プロジェクト 乗り合い型交通システムコンビニクル, 2005.

[5] H Sadrsadat, H Poorzahedi, A Haghani, and Elham Sharifi. Bus network design using genetic algorithm. 1:210–225, 01 2012.

[6] Dervis Karaboga and Bahriye Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39(3):459–471, November 2007.

[7] 飯村伊智郎 and 中山茂. 高次元関数最適化における artificial bee colony アルゴリズムの探索性能評価. **システム制御情報学会論文誌**, 24(4):97–99, 2011.