バスネットワークのための実用的な経路探索システム

川 村 尚 生[†1] 菅 原 一 孔[†1]

我々が先に報告した，徒歩移動を考慮したバス経路探索システムが出力する経路は，必ず所要時間最短ではあるものの，(1) 乗り換えが不必要に多い，(2) 乗車時間が不必要に長い，(3) 好ましくない徒歩移動を含む，等の非実用的なものになる場合があった．そこで，節に到着した時刻に応じて辺の重みを動的に計算する改良ダイクストラ法と，その実行過程で得られるパラメータによって探索範囲を限定した全経路探索を組み合わせることで，このような非実用的な経路の出力を回避する手法を開発した．また，所要時間最短という制約を緩和することで，乗り換え回数，乗り換え時間，徒歩時間の観点から，より実用的と考えられる経路が得られる場合，それらの経路をあわせて探索する手法も開発した．開発した手法を，鳥取市の実際のバスネットワークに適用して，WWW を介して利用できるバス経路探索システムを構築し，公開実験による有効性確認を経て，鳥取県のバス会社，鳥取商工会議所，中国運輸局鳥取運輸支局等の協力により運用を開始した．

Practical Path Planning System for Bus Network

Takao Kawamura[†1] and Kazunori Sugahara[†1]

Our path planning system for bus network that we have previously reported had the possibility of outputting non-practical paths including unnecessary transfers, uselessly long riding time, or non-practical walking transfers. In this paper, we propose a two-stage path planning algorithm to prevent outputting such non-practical paths. As the first stage of the algorithm, we applied a modified Dijkstra's Shortest Path Algorithm to treat dynamically changing costs. Next, as the second stage, a depth-first search with efficient pruning is executed. The proposed algorithm also may find more practical paths in the viewpoint of the number of transfers, riding time, and walking time through relaxing the constraint of minimum required time. We have developed a path planning system based on the proposed algorithm for the bus network in Tottori City as Web service. This system finds appropriate paths using location information of the starting point and the destination obtained through GPS or landmark databases. The developed system is now opened to the public under the cooperation of the bus companies in Tottori prefecture, the Tottori Board of Trade, and the Tottori branch of the Chugoku District Transport Bureau.

1. はじめに

地方における路線バスの利用者数は長期減少傾向にある．たとえば，中国運輸局鳥取運輸支局の調査によると，鳥取県においては昭和 41 年には年間のべ 8 千万人近くあった利用者数が，平成 16 年には 8 百万人弱とおよそ 10 分の 1 になっている．利用者数の減少理由には，モータリゼーションの進展，少子化，過疎化等，やむをえないものも含まれるが，路線バスの利用者数を増やすことは，高齢化社会における生活交通の確保，省エネルギーや環境保全に優れた公共交通機関の促進という点から見て重要な課題といえる．

路線バスの利用者数の増加を図るためには，様々な視点からその利便性を高める努力が必要である．たとえば，バスのリアルタイムの位置情報を利用者に提供するバスロケーションシステムはその試みの 1 つであり，秋田市交通局が 1981 年に試験導入して以来，多数の自治体で運用されている[1),2)]．また，現在，多くのバス会社がインターネット上で時刻表や路線情報の閲覧サービスを提供している．

しかし，利用者にとって最も必要な，経路情報を提供する仕組みについては，これまで十分に取り組まれてきたとはいえない．すなわち，目的地へ行くために，「いつ」「どこで」「どのバスに乗って」「どこで降りるか」等の情報を入手することがきわめて困難である．鉄道においては，出発駅と目的駅を指定すると経路情報を出力する経路探索システム[3),4)] が普及しているが，路線バスには鉄道の経路探索システムをそのまま流用することはできない．複数路線の共通のバス停で

†1 鳥取大学工学部
Faculty of Engineering, Tottori University

の乗り換えだけではなく，任意のバス停から他路線の任意のバス停への徒歩移動による乗り換えを考慮する必要がある．なぜなら，バス路線は 2 地点間を最短経路で結ぶという発想ではなく，利用者が多い場所を通るように設計されているため多数の路線が複雑に入り組んでいるうえに，バス停の設置間隔が狭いので，徒歩移動による乗り換えが可能かつ有効な場合が多いからである．また，出発地や目的地の周辺には複数のバス停が存在していることも多く，行き先によっては距離的に最も近いバス停から乗車することが，出発地から目的地までの所要時間を最短にするとは限らない．また，鉄道駅と異なり，バス停には場所を知らせる道路標識等もなく，どこにバス停があるのかを知ることさえ容易ではない．したがって，鉄道に対して行われているような出発バス停から目的バス停までの経路探索ではなく，出発地や目的地をランドマークで指定する経路検索が必須である．その点，長崎県が提供している県内経路探索システムは，バス停間の徒歩移動を考慮せず，出発地や目的地としてバス停を指定する方式になっており，不十分なものといわざるをえない[5]．

これまでに，バスを含む公共交通機関の経路探索に関して，他の最短路探索手法と組み合わせることを前提に，接続行列を用いて解を効率良く見つけるアルゴリズム[6)–8)] や，乗車時間の不確定性を扱うアルゴリズム[9)] 等が提案されているが，路線バスのための本格的な経路探索システムは報告されていない．

我々は，出発地から目的地までの所要時間最短経路を出力する，徒歩移動を考慮したバス経路探索システム（以後，旧システムと呼ぶ）を開発し，報告した[10)]．旧システムを試験運用した結果，確かに所要時間は最短ではあるものの，非実用的な経路が出力される場合があることが分かった．原因は，バス路線の特性のため，所要時間最短の経路が複数存在するからであった．旧システムのバス経路探索はダイクストラ法[11)] に基づいていたため，所要時間最短経路のうちどれが出力されるかは探索順序に依存しており，非実用的な経路が出力されることがあった．

そこで，我々は，実用的な探索時間で実用的な経路を出力することを目的として，新しいバス経路探索システム（以後，新システムと呼ぶ）を開発した．鳥取県のバス会社，鳥取商工会議所，中国運輸局鳥取運輸支局等の協力により，公開実験による有効性の確認を経て，鳥取市において新システムの本格運用を開始するに至ったので報告する．

本論文は以下のように構成されている．2 章において，旧システムの非実用的な出力経路のパターンを示した後，3 章でそのような経路の出力を回避するための解決方法を述べる．4 章では，所要時間最短という制約を緩和することにより，さらに実用的な経路を出力するための工夫を述べる．5 章において開発した新システムの概要を示した後，6 章でその実験結果を述べ，7 章で本論文をまとめる．

2. 非実用的な経路

この章では，所要時間最短という条件を満たす任意の経路を出力する旧システムにおいて，実際に出力された非実用的な経路を，3 つのパターンに分類して説明する．

2.1 乗り換えが不必要に多い経路

バスの運行ダイヤによっては，直接行けるところを遠回りしても，目的地までの所要時間が変わらない場合があり，その際不必要に多く乗り換える経路が出力されることがある．これは出発時や乗り換え時におけるバス停での待ち時間が長い場合に起こりやすい．例を**図 1** に示す．この例ではバス停 A からバス停 C に行こうとしている．路線 α のバスに乗ってそのままバス停 C まで行けばよいのだが，出発時刻を基準として，路線 β のバスの方がバス停 A を早く出発する場合，図 1 のように，いったんそれに乗ってバス停 B で降車し，路線 α のバスを待って乗り換えるという経路が出力されることがある．ここでいう所要時間とは，ある出発時刻から目的地に到着するまでの時間であり，最初のバスに乗るまでの待ち時間も含む．この例の場合，いずれの経路をとっても所要時間は変わらない．しかし，一般に乗り換え回数は少ないほど望ましく，路線 β を利用する経路は非実用的といえる．

バス路線は複数路線が複雑に入り組んでいるため，このような無駄な遠回りが生じる余地が大きい．また，バスの便数が少ない場合はバス停での待ち時間が長くなり，このような現象はなおさら顕著に見られる．

2.2 乗車時間が不必要に長い経路

乗り換えが必要な場合に，乗車時間が不必要に長い経路が出力されることがある．例を**図 2** に示す．この例ではバス停 A からバス停 C に行こうとしている．路線 α と路線 β は，バス停 B からバス停 D までの区間は同一の経路をたどり，それ以外の部分は違う経

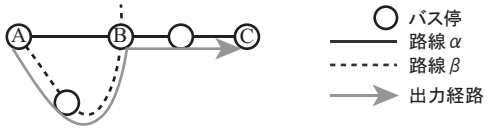


図 1 乗り換えが不必要に多い経路
Fig. 1 A path including unnecessary transfer.

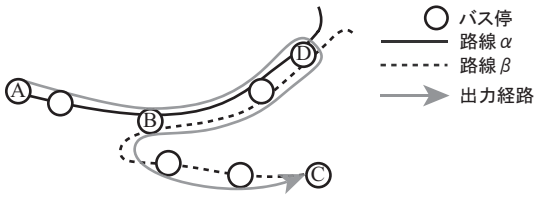


図 2 乗車時間が不必要に長い経路
Fig. 2 A path including uselessly long riding time.

路を走行する．したがって，バス停 B からバス停 D までのどのバス停でも路線 α から路線 β に乗り換えることができる．このとき，バスの運行ダイヤによっては，どのバス停で乗り換えたとしても，乗り換え後に乗るバスは同一となる可能性がある．その場合は，どのバス停で乗り換える経路でも，所要時間も乗り換え回数も等しいので，所要時間を最短にすることのみ考慮する旧システムはもちろん，2.1 節で述べた点を考慮し，乗り換え回数を最少にしようとしても，図 2 に示す $A \to \cdots \to B \to \cdots \to D \to \cdots \to C$ という経路が出力される可能性がある．しかし，乗り換え時間に余裕を持つことや料金面を考慮すると，「乗車時間はできるだけ短くする」ことが望ましく，この例ではバス停 B で乗り換える $A \to \cdots \to B \to \cdots \to C$ という経路が実用的といえる．

バス路線は，特に市街地では複数路線が部分的に重なることが多く，図 2 のような状況は一般的に生じる．

2.3 好ましくない徒歩移動を含む経路

乗り換えが必要な場合に，好ましくない徒歩移動を含む経路が出力されることがある．例を**図 3** に示す．この例ではバス停 A から，路線 α と路線 β の共通のバス停 C を経て，バス停 D に行こうとしている．バス停 C での待ち時間が十分長い場合，バス停 B で降車してバス停 C まで徒歩移動しても，乗り換え後のバスに間に合うことがある．しかし，路線 α のバスに乗り続けていればバス停 C に着くので，利用者がそのような行動をとるとは考えられない．

これは 2.2 節で述べた場合と正反対の状況であり，乗車時間を短くしようとすると好ましくない徒歩移動をする例である．ここで，この問題の解決を，単に「同一路線のバス停まで徒歩移動しない」というルールを導入することで図っても，類似した**図 4** のような状況には対処できない．この例では，路線 α のバスに乗り続けていてもバス停 C に到達しないので，いずれかのバス停から徒歩移動で路線 β に乗り換える必要があるが，バス停 B からではなく，バス停 C により近いバス停 E から歩く経路の方が実用的といえる．

図 3 や図 4 のような状況も，複数路線が複雑に入り組んでいてバス停の設置間隔が短いというバス路線の特性から，市街地では頻繁に生じる．

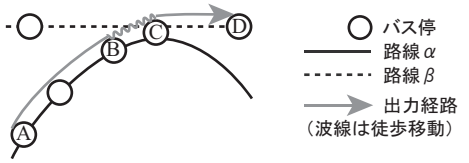


図 3 好ましくない徒歩移動を含む経路 1
Fig. 3 A path including a non-practical walking transfer.

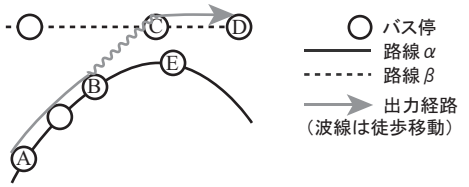


図 4 好ましくない徒歩移動を含む経路 2
Fig. 4 Another path including a non-practical walking transfer.

3. 解 決 方 法

この章では，まず旧システムの経路探索手法を述べ，次いで，2 章で示した非実用的な経路の出力を回避するように改良した新システムの経路探索手法を述べる．

3.1 旧システムの経路探索手法

旧システムでは，バス経路探索をネットワークの単一始点最短路問題ととらえ，以下のように経路を探索していた．

まず，バス停を節，バスが運行する経路を辺，辺の重みをバス停間の所要時間としてネットワークを構築する．このとき，同じ路線を走るバスでも時刻や曜日によっては停まる停留所の数が少ない直行便になる場合があるが，これについては一般便が通る路線とは別の路線が存在するものとして扱う．時間帯によってバス停間の走行時間が多少違う等，同一路線上のバスが通常と異なる状況で走行する場合は他にも存在する．これらもすべて複数の分割路線が存在するものとしてネットワークを構築する．

このネットワークに，経路探索を行うたびに出発地と目的地を節として加え，他の節と徒歩移動による辺で結ぶ．徒歩移動による辺は，バス停を表す節どうしの間にも加えなければならない．原理的には全節間を徒歩移動による辺で結ぶ必要があるが，徒歩による移動時間が一定以上の辺が経路に加わることは考えられないので，徒歩による移動時間が適当な閾値以下の節間のみを結ぶこととする．

そして，完成したネットワークに，ダイクストラ法

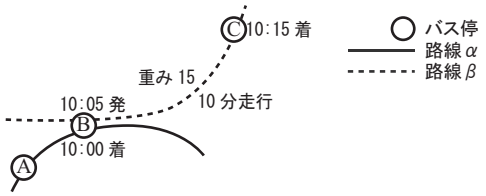


図 5 待ち時間を含む辺の重み
Fig. 5 A weight of an edge includes waiting time.

を適用して経路を探索する．いま，節 A の最短路推定値 d_A，すなわち出発節から節 A までの経路の長さの最小値を用いて節 B の最短路推定値 d_B を更新しようとするとき，d_B は次式で表される．

$$d_B = d_A + l_{A\to B} \tag{1}$$

ここで，$l_{A\to B}$ は節 AB 間の辺の重みで，バス停 AB 間の所要時間を表しているが，本ネットワークではその値は静的には定まらないことに注意を要する．節 A 以前から路線 α 上のバスに乗車している場合は，バス停 AB 間の走行時間 $r_{A\to B}$ が $l_{A\to B}$ に相当するが，節 A から路線 α 上のバスに乗車する場合，次式のように，次に乗るバスが到着するまでの節 A での待ち時間 $w_{A\to B}$ を加えなければならない．

$$l_{A\to B} = r_{A\to B} + w_{A\to B} \tag{2}$$

例を**図 5** に示す．いま，バス停 A から乗り換えバス停 B を経由してバス停 C に移動するものとする．バス停 B への到着時刻が 10:00 で，次のバスが 10:05 に出る場合，バス停 B からバス停 C までの所要時間，すなわち辺の重みは，バス停 BC 間のバスの走行時間 10 分に待ち時間 5 分を加えた 15[分] となる．この待ち時間はバス停 B への到着時刻によって異なるので，辺の重みも静的に定めることはできず，経路探索時に節への到着時刻に応じて動的に計算しなければならない．

旧システムでは，辺の重みを動的に計算するように改良したダイクストラ法を，上記のネットワークに適用して所要時間最短経路を探索していた[10]．ダイクストラ法の一般的な実装方法は，m と n をそれぞれ辺と節の個数としたとき，計算量が $O(m + n\log n)$ である，フィボナッチヒープを用いた優先順位付き待ち行列を用いるものである[12),13)]．しかし本研究では，計算量が $O(m+nC)$ である，1 レベルバケット法[14)] を用いる手法で実装した．ここで，C は辺の重みの最大値である．バケット法は，最短路推定値がとりうる値ごとに待ち行列を用意する手法であり，利用できる場合はフィボナッチヒープを用いるよりも高速になることが多い．バスの経路探索では，最短路推定値が分単位で与えられ，その値もバスで移動する最長の時間で抑えられることから，バケット法に適した問題であるといえる．

3.2 新システムの経路探索手法

3.1 節で述べたネットワークを用い，出発地から目的地までの全経路を，出発地を表す節を根とする木と見なして深さ優先探索し，擬最適経路を見つけるというのが新システムの経路探索アルゴリズムの基本的な考え方である．ただし，実用的な速度で探索を行うために，可能な限り探索範囲を限定する．また，最初に見つかった経路が擬最適経路になるように探索順序に工夫を加えている．ここで，擬最適経路とは以下のような経路である．

(1) 所要時間最短の経路
(2) (1) の経路が複数ある場合は，そのうち乗り換え回数最少の経路
(3) さらに (2) の経路が複数ある場合は，そのうち乗車時間がなるべく短い経路

乗車時間に関する表現が曖昧なものになっているのは，2.3 節で述べた好ましくない徒歩移動を含む経路を排除するためである．すなわち，原則として乗車時間最短の経路を選ぶが，好ましくない徒歩移動を含まないようにするためには乗車時間が増えることもやむをえないものと考える．

全経路探索に先だって，3.1 節で述べた改良ダイクストラ法で所要時間最短の経路を 1 つ見つけておき，その過程で得られるパラメータを探索範囲の限定に用いることを考える．得られるパラメータとしては，まずその経路の所要時間 min_time と乗り換え回数 dijkstra_transfer があげられる．全経路探索で見つける擬最適経路の所要時間は min_time と等しくなければならず，乗り換え回数は dijkstra_transfer を超えてはならない．

これとは別に，3.1 節で述べたネットワークにおいて，辺の重みを，節間の所要時間ではなく，バスを乗り換える場合は 1，そうでない場合は 0 として改良ダイクストラ法を適用することで，所要時間無制限の場合の最少乗り換え回数 min_transfer が得られる．擬最適経路の乗り換え回数が min_transfer 未満になることはありえない．

以上 3 つのパラメータを用いて，全経路探索において，次の条件の少なくとも一方が満たされた場合に，その節より深いレベルの探索を行わないことで，探索範囲を限定する．

(1) ある節に到着したとき，出発地からの乗り換え回数が上限乗り換え回数を超えた
(2) ある節に到着したとき，出発地からの所要時間

が min_time を超えた．

ここで，上限乗り換え回数には最初は min_transfer を与える．探索木をすべて調べ終わっても経路が見つからない場合，上限乗り換え回数を 1 増やして全経路探索をやり直す．これを，経路が 1 つ見つかるか，上限乗り換え回数が dijkstra_transfer を超えるまで繰り返す．このようにすることで，最初に見つかる経路が，所要時間最短の経路の中では乗り換え回数最少であることが保証され，2.1 節で述べたような経路が出力されることはなくなる．

2.2 節で述べたような経路を出力しないようにするためには，可能な限り早めにバスを降りようとする経路が最初に見つかるようにすればよい．このことは，ある節から次の節への探索を行う際，徒歩による移動の辺をバスによる移動の辺より先に調べることで達成される．しかし，2.3 節で述べた，好ましくない徒歩移動を含む経路が見つかる可能性は残される．そもそも，このような徒歩移動を含む経路を出力しないためには，バスを降りて歩かずに乗車し続けることを選ぶ必要があるので，2.2 節の，早めにバスを降りようとする経路を選ぶこととは目標が正反対である．

このような矛盾した両方の目標に対処するため,「バスに乗っていれば接近する方向に向かって，わざわざ降車して徒歩移動しない」というヒューリスティックスを導入する．もちろん，所要時間が短縮できたり，乗り換え回数が減らせたりする場合には，このヒューリスティックスに反する徒歩移動も考慮に入れなければならない．そこで，最初にバスに乗っていても接近しない方向への徒歩移動の辺を調べ，次にバス移動の辺を調べ，最後にバスに乗っていれば接近する方向への徒歩移動の辺を調べることとする．それぞれの方向への徒歩移動による辺が複数ある場合，移動時間の短いものから優先的に調べる．

次に，探索範囲をさらに限定するために，上記の条件 (2)「ある節に到着したとき，出発地からの所要時間が min_time を超えた」を，条件 (2')「ある節に到着したとき，出発地からの所要時間が d を超えた」と，より厳しく変更することを考えてみる．ここで d は，改良ダイクストラ法で経路を得る際に求められている，出発地からその節に到着するまでの最短所要時間である．

3.1 節で述べたネットワークには，条件 (2') を適用することは不可能である．**図 6** でそれを説明する．バス停 A には，そこで行き止まりとなる路線 α とバス停 B まで到達する路線 β が乗り入れているものとする．いま，いずれかの節から 09：50 に出発し，路線

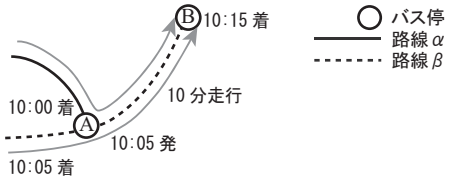


図 6 d で探索範囲を限定してはいけない例
Fig. 6 The searching tree cannot be pruned by d.

α のバスでバス停 A に到着するのが 10：00，路線 β のバスでバス停 A に到着するのが 10：05 としよう．路線 α のバスで到着した場合，バス停 A で 5 分待って路線 β に乗り換えてバス停 B に向かうことになる．一方，路線 β のバスで到着した場合，同じバスに乗ったままバス停 B まで行くことができ，いずれにせよバス停 B には 10：15 に到着する．いま，バス停 A の d_A は，出発時刻 09：50 から最も早い到着時刻である 10：00 までの所要時間なので，10[分] となる．したがって，条件 (2') をそのまま適用すると，出発時刻から 10 分を超えてバス停 A に到着する路線 β を通る経路は探索対象にならなくなる．しかし，図 6 の例では，この経路の方がより実用的な経路である可能性がある．

このように，探索木において，節 A への到着時刻がある時刻 t より遅いという指標で，節 A より深いレベルの探索を打ち切ることはできない．しかし，節 A の出発時刻がある時刻 t より遅いという指標ならば可能である．そこで，d の定義を「出発地からその節を出発するまでの最短所要時間」と変更する．これは式 (2) を次式のように変更することを意味する．

$$l_{A\to B} = r_{A\to B} + w_{B\to x} \tag{3}$$

ここで，$w_{B\to x}$ における x は，節 B の次に向かう節を示す．図 6 の例では，新しく定義した d_A は 15[分] となり，条件 (2') を適用しても，路線 β を通る経路も探索対象となる．

式 (3) から分かるように，節 B から複数の辺が出ている場合，どの節に向かうかに応じて $w_{B\to x}$ の値が異なる．しかし，節 A に到着した段階では，まだ節 B からどの節へ向かうかが分からないので，$w_{B\to x}$ が定まらず，d_A が計算できないことになる．したがって，d の定義を上記のように変更するためには，節から出る辺の数を 1 にしなければならない．ただし，ある節から徒歩移動で他の節に向かう場合は待ち時間が 0 であるから，そうした辺はいくつあってもよい．

このため，節を路線ごとに分割して新しいネットワークを構築する．**図 7**(a) に分割前の旧ネットワークの部分例を，図 7(b) に分割後の新ネットワークの

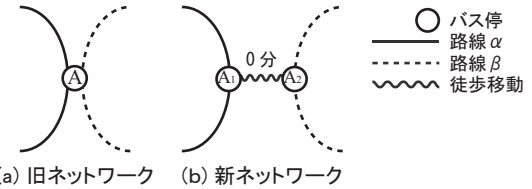


図 7 旧ネットワークと節を分割した新ネットワーク
Fig. 7 New network is obtained through dividing nodes in the old network.

```
search_path()
{
  改良ダイクストラ法により所要時間最短経路 p0 を得る;
  dijkstra_transfer = p0 の乗り換え回数;
  辺の重みを乗り換えの有無にした改良ダイクストラ法に
  より所要時間無制限の経路 p1 を得る;
  min_transfer = 経路 p1 の乗り換え回数;
  for (max_tr = min_transfer;
       max_tr <= dijkstra_transfer; max_tr++)
    traverse({}, 出発地,0,0,max_tr,false);
}

traverse(path,node,t,tr,max_tr,iswalking)
{
  path に node を追加;
  if (node == 目的地)
    path を出力し，以後の探索を打ち切る;
  if (!iswalking) {
    N = {バスに乗っていても接近しない方向にある
         node から徒歩移動可能なバス停};
    walk(N,path,node,t,tr,max_tr);
  }
  next = node の次のバス停;
  if (next が存在する) {
    t_next = t + node から next までの乗車時間;
    d_next = next の最短路推定値;
    if (t_next <= d_next)
      traverse(path,next,t_next,tr,max_tr,false);
  }
  if (!iswalking) {
    N = {バスに乗っていれば接近する方向にある
         node から徒歩移動可能なバス停};
    walk(N,path,node,t,tr,max_tr);
  }
  path から node を削除;
}

walk(N,path,node,t,tr,max_tr)
{
  for (next in N) {
    t_next = next 出発時の出発地からの所要時間;
    d_next = next の最短路推定値;
    if (tr < max_tr && t_next <= d_next)
      traverse(path,next,t_next,tr+1,max_tr,true);
  }
}
```

図 8 新システムの経路探索アルゴリズム
Fig. 8 New path plannning algorithm.

部分例を，それぞれ示す．この例では，路線 α と路線 β の共通のバス停 A を，路線 α の A_1 と路線 β の A_2 に分割している．このとき，A_1 と A_2 間は所要時間 0 の徒歩移動による辺で結ぶ．この辺を移動することで路線 α と路線 β 間の乗り換えが表現される．

新ネットワークには条件 (2') が適用でき，探索範囲を大幅に限定することができる．

以上の経路探索手続き search_path() を**図 8** にまとめる．擬最適経路は探索木を再帰的に探索する手続き traverse(path,node,t,tr,max_tr,iswalking) で得られる．各引数は順に，出発地から node の直前までの経路，次に調べる節，出発地からの所要時間，出発地からの乗り換え回数，上限乗り換え回数，node に徒歩移動で到着したか否かをそれぞれ示す．また，手続き walk(N,path,node,t,tr,max_tr) は，node から徒歩移動可能な節集合 N を調べる手続きで，第 2 引数以降の意味は手続き traverse() と同様である．

本経路探索アルゴリズムは，改良ダイクストラ法によって所要時間最短経路を求めた後，その過程で得られるパラメータにより探索範囲を限定した全経路探索を行うことで，所要時間以外の属性を考慮した経路を出力する．すでに提案されている，所要時間が短い方から上位 k 個の経路を求めるアルゴリズム[15),16)] を用いたうえで，複数の属性に基づく評価指標により各経路の評価を行い，出力経路を選定する手法も考えられる．しかし，本研究で対象としている問題の場合，バス経路は同じ所要時間を持つ経路が多数存在することがあるため，この手法で有効な擬最適経路を出力するためにはパラメータ k を十分大きくとる必要があり，効率が悪い．また，本研究では，乗り換え回数と乗車時間という，単純かつ少数の評価指標のみを考慮しているため，最初に見つかる経路が擬最適経路となるようにアルゴリズム中に評価を組み込むことが可能であり，そうする方が，多数の非実用的な経路を生成してから評価によって絞るよりも効率が良いと考えられる．

4. より実用的な経路を出力するための工夫

3.2 節で述べた方法で，2 章の問題を回避する擬最適経路が出力できる．我々は，擬最適経路が，所要時間最短という制約の下では，利用者にとって最も満足度の高い実用的な経路であると考えている．しかし，たとえば，多少余計に時間がかかっても，乗り換え回数が減らせるならそちらを選ぶ利用者も多いであろう．そこで，所要時間最短という制約を緩和して，以下の観点から改善されている経路もあわせて探索する．

(1) 乗り換え回数

所要時間が最大 T_a 増えても乗り換え回数が減らせる経路を以下のようにして探索する．まず，上限乗り換え回数に min_transfer を与えて全経路探索を行い，min_time+T_a 以内に目的地に到達できる経路が見つかれば，そのうち所要時間が最も短い経路を出力する．経路の発見にかかわらず上限乗り換え回数を 1 増やして全経路探索を再実行する．これを，上限乗り換

え回数が max_transfer になるまで繰り返す．ここで，max_transfer とは擬最適経路の乗り換え回数である．

(2) **乗り換え時間**

バスは道路事情により運行バス停への到着が遅れることが多い．この遅れは乗り換え時に問題となる．すなわち，乗り換え前のバスの遅れによって，乗り換え後のバスに乗れない事態が生じる可能性がある．そこで，得られた経路に T_d 未満の乗り換えが含まれる場合は，すべての乗り換えが T_d 以上余裕のある経路も探索する．この経路は，乗り換え前のバスが T_d 遅れると仮定して図 8 のアルゴリズムを実行することで得られる．バスの遅れは，図 8 中の手続き `walk()` において，next を出発するまでの所要時間を求める部分にのみ影響を及ぼす．

(3) **徒歩時間**

擬最適経路を求める際には，徒歩による移動時間の閾値を，バス停間は T_{w1}，出発地およびバス停から鉄道駅に向かう場合は T_{w2}，出発地と最初の乗車バス停間，最後の降車バス停と目的地間については T_{w3} としている．ここで，T_{w2} と T_{w3} は $T_{w1} < T_{w2}$，$T_{w1} < T_{w3}$ を満たすものとする．擬最適経路に T_{w1} を超える徒歩移動が含まれる場合には，T_{w2} や T_{w3} を T_{w1} と等しくして図 8 のアルゴリズムを実行し，より徒歩時間の少ない経路も探索する．

最初に T_{w2} や T_{w3} を T_{w1} より大きく設定するのは，このような場合は長く歩いてもかまわないとする利用者が多いとの考えに基づく．なお，出発地や目的地から T_{w3} で到達できる範囲にバス停が存在しない場合は，T_{w3} を超えて徒歩移動する経路が探索される．

5. バス経路探索システム

バス経路探索システムを Web サーバを介して利用できる CGI として開発した．開発言語には，経路探索部に C を，ユーザインタフェース部に Ruby をそれぞれ使用した．システムの構成図を**図 9** に示す．

システムで用いるデータとして，日ノ丸自動車株式会社，日本交通株式会社により運行されている計 121 路線と 1,081 バス停を基に，(1) バス停データベース，(2) 分割路線データベース，(3) 接続データベース，(4) 始発データベースを作成し，関係データベースサーバ MySQL によって管理している．3.1 節で述べた分割路線数は 680，3.2 節で述べた分割バス停数は 16,414 であり，全国の鉄道駅数が約 9,000 であることから見ても，本システムが扱うデータの大きさが，鉄道用経路探索システムのそれと同程度であることが分かる．

(1) は，各レコードがバス停 1 つに対応しており，

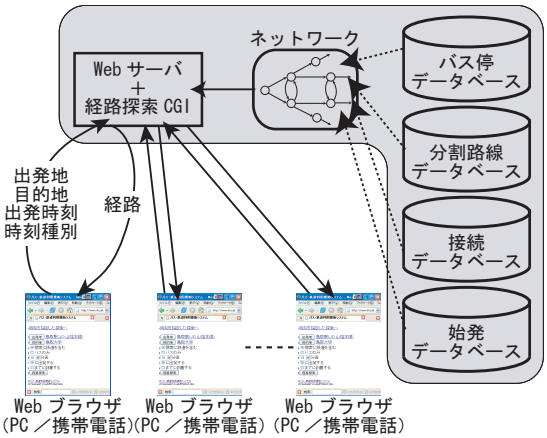


図 9 バス経路探索システム

Fig. 9 Proposed path planning system for bus network.

バス停 ID，バス停名称，北緯，東経の 4 組からなる．(2) は，各レコードが分割路線 1 つに対応しており，分割路線 ID，始発バス停を示す接続情報 ID，有効ビット列の 3 組からなる．有効ビット列は，その分割路線が平日，奇数週土曜日，偶数週土曜日，日曜日，学期間のいつ有効なのかを示す．学期間とは，学校が開校されている期間のことである．(3) は，各レコードが分割路線の停車バス停に対応しており，接続情報 ID，バス停 ID，次の接続情報 ID，次のバス停までのバスの走行時間 [分] の 4 組からなる．(4) は，各レコードが分割路線の始発時刻に対応しており，分割路線 ID と発車時刻の 2 組からなる．

以上 4 種のデータベースから，出発地と目的地を除いた，ネットワークの静的な部分をあらかじめ構築し，約 10MB のファイルとして保存しておく．そして，探索を行うたびに，出発地と目的地の節および徒歩移動による辺を追加してネットワークを完成させ，経路探索を行う．利用者には，擬最適経路のほか，所要時間最短制約を緩和した経路探索により得られた経路をあわせて提示する．なお，4 章で述べた T_a，T_d，T_{w1}，T_{w2}，T_{w3} はそれぞれ，30，5，5，20，20[分] としている．

利用者は，PC や携帯電話等の Web ブラウザから本システムにアクセスする．システムへの入力は，出発地，目的地，出発時刻，時刻種別の 4 組である．出発地および目的地はランドマークで指定する．ランドマークとしては鳥取市内の主要な施設 3,162 カ所を MySQL で管理している．GPS 機能付き携帯電話を用いた場合，出発地として利用者の現在位置を与えることもできる．時刻は特定の日時，あるいは現在時刻より何分後という形で指定し，時刻種別には，指定時

刻に出発するか，指定時刻までに到着したいかのいずれかを指定する．

6. 実　　験

6.1 経路探索速度

本システムの経路探索時間を調べるための実験を，Pentium4 2.8GHz，主記憶 2GB の計算機上で行った．出発地と目的地を 3,162 カ所のランドマークから，出発時刻を 08：00〜20：00 から，それぞれランダムに選び，時刻種別は指定時刻に出発するものとした．出発地，目的地，出発時刻の組に重複がない 100,000 回の試行による経路探索時間の集計結果を**表 1** に示す．なお，経路探索時間には，擬最適経路の探索に要した時間に加えて，4 章で述べた，所要時間最短制約を緩和した経路の探索に要した時間も含まれている．

提案アルゴリズムは探索範囲の主要な限定手段として乗り換え回数を使用するので，乗り換え回数別に集計を行ったが，乗り換え回数は探索時間の最小値以外にはあまり影響を及ぼしていなかった．表 1 の結果から，探索時間のばらつきは大きくなく，条件によらず，実用上まったく問題がない速さであるといえる．

6.2 出力経路改善例

6.2.1 乗り換え回数に関する改善経路例

出発地が『鳥取駅（ＪＲ山陰本線）』，目的地が『鳥取空港』，出発時刻が 08：00 のときの，旧システムの出力経路を**図 10**(a) に，新システムの出力経路を図 10(b) にそれぞれ示す．図 10(c) には，参考のため各地点の位置関係を示す．同図において，バス停 $A \to B \to C \to D$ と移動しているのが旧システムの出力経路で，バス停 $A \to B \to D$ と移動しているのが新システムの出力経路である．この例は図 1 に対応している．すなわち，バス停 B において，直後のバスに乗車することを選択せず，その後のバスを選択することにより，乗り換え回数を減少させた例である．なお，新システムの出力経路においてバス停 B で乗車するバスは，旧システムの出力経路においてバス停 C で乗車するものと同一バスである．当然の結果として，図 10(a)，(b) にも示されているとおり，いずれの出力経路も所要時間は 41 分と等しいが，旧システムの出力経路はバス停 C での不必要な乗り換えを 1 回行うものとなっている．一方，新システムの出力経路では乗り換えなしとなり，改善されている．

表 1　経路探索時間の集計結果
Table 1　Summary statistics for path planning times.

(単位：秒)

乗り換え回数	最小値	最大値	平均値	標準偏差
0	0.03	0.36	0.21	0.08
1	0.05	0.36	0.26	0.07
2	0.08	0.36	0.27	0.06
3	0.12	0.36	0.29	0.05
4	0.18	0.34	0.27	0.07
合計	0.03	0.36	0.25	0.07

6.2.2 乗車時間に関する改善経路例

出発地が『浦富海岸』，目的地が『湖山駅（ＪＲ山陰本線）』，出発時刻が 08：10 のときの，旧システムの出力経路を**図 11**(a) に，新システムの出力経路を図 11(b) に，各地点の位置関係を図 11(c) にそれぞれ示す．同図において，バス停 $A \to B \to D \to E \to F$ と移動しているのが旧システムの出力経路で，バス停 $A \to B \to C \to E \to F$ と移動しているのが新システムの出力経路である．この例は図 2 に対応している．いずれの出力経路も同じ 2 路線を使い，所要時間も乗り換え回数も等しいが，乗車時間が大きく異なる．旧システムでは，バス停 D まで乗車し続けた後に同バス停で乗り換えているため合計 58 分乗車する経路が

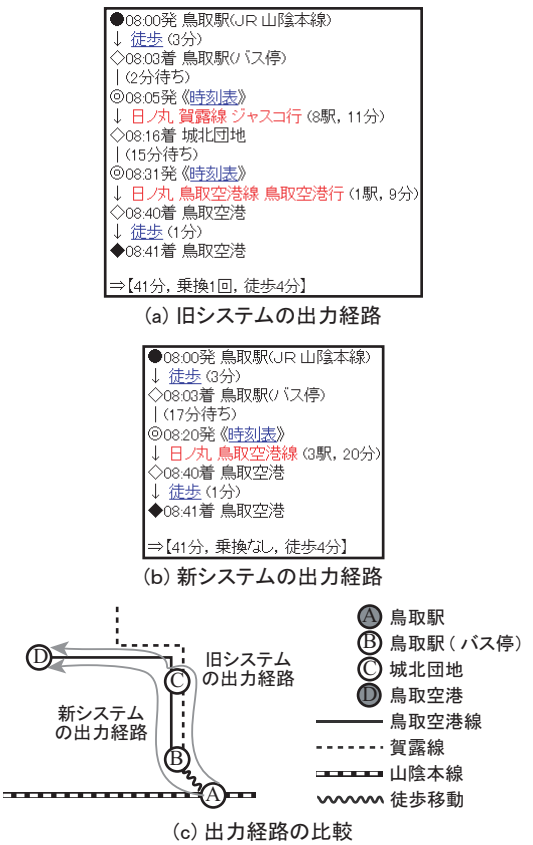
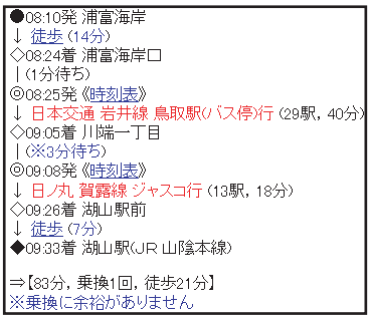
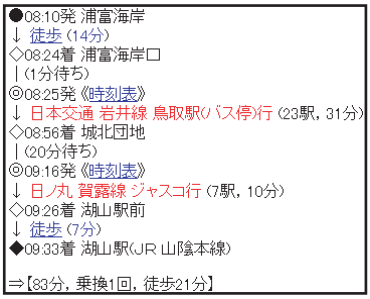


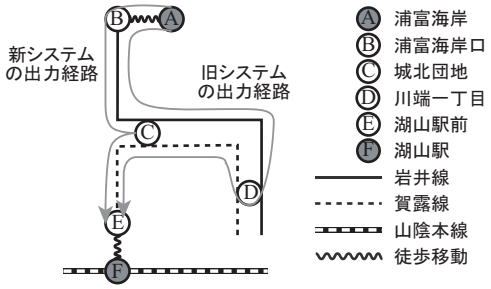
図 10　乗り換え回数に関する改善経路例
Fig. 10　An improved path regarding the number of transfers.



(a) 旧システムの出力経路



(b) 新システムの出力経路



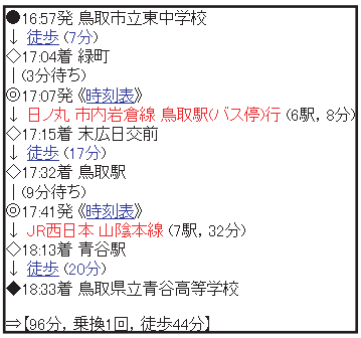
(c) 出力経路の比較

図 11 乗車時間に関する改善経路例
Fig. 11 An improved path regarding riding time.

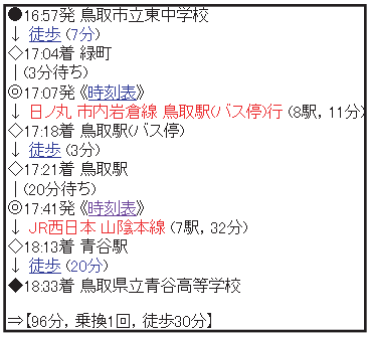
出力されているが，新システムでは，バス停 C で乗り換える経路が出力されている．この経路では合計 41 分しか乗車しないことになり，バス運賃等を考慮すると，より実用的な経路といえる．

6.2.3 好ましくない徒歩移動に関する改善経路例

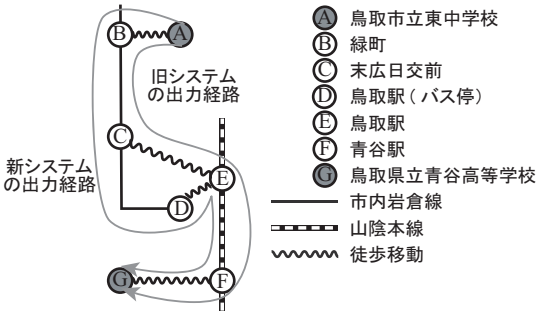
出発地が『鳥取市立東中学校』，目的地が『鳥取県立青谷高等学校』，出発時刻が 16：57 のときの，旧システムの出力経路を**図 12**(a) に，新システムの出力経路を図 12(b) に，各地点の位置関係を図 12(c) にそれぞれ示す．同図において，バス停 $A \to B \to C \to E \to F \to G$ と移動しているのが旧システムの出力経路で，バス停 $A \to B \to C \to D \to E \to F \to G$ と移動しているのが新システムの出力経路である．この例は図 4 と対応している．いずれの出力経路も同じ 2 路線を使い，所要時間も乗り換え回数も等しいが，徒



(a) 旧システムの出力経路



(b) 新システムの出力経路



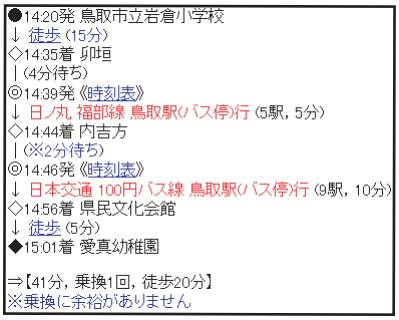
(c) 出力経路の比較

図 12 好ましくない徒歩移動に関する改善経路例
Fig. 12 An improved path regarding the way of walking transfers.

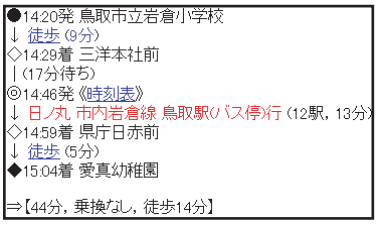
歩時間が大きく異なる．すなわち，旧システムの出力経路では 44 分，新システムの出力経路では 30 分と減少している．旧システムでは，バス停 C からバス停 E まで 17 分徒歩移動する経路が出力されている一方，新システムでは，バス停 E の最寄りバス停 D から 3 分徒歩移動する経路が出力されている．旧システムの経路を利用者が選択するとは考えにくく，新システムの出力経路の方がより実用的といえる．

6.2.4 所要時間最短制約を緩和して得られる提案経路例

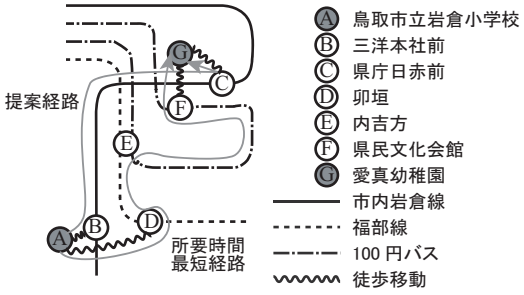
出発地が『鳥取市立岩倉小学校』，目的地が『愛真幼稚園』，出発時刻が 14：20 のときの，所要時間最



(a) 所要時間最短の経路



(b) 提案経路



(c) 出力経路の比較

図 13 所要時間最短制約を緩和して得られる提案経路例
Fig. 13 An advised path obtained through relaxing the constraint of minimum required time.

短の経路を**図 13**(a) に，所要時間最短制約を緩和して得られる提案経路を図 13(b) に，各地点の位置関係を図 13(c) にそれぞれ示す．同図において，バス停 $A \to D \to E \to F \to G$ と移動しているのが所要時間最短経路で，バス停 $A \to B \to C \to G$ と移動しているのが所要時間最短制約を緩和して得られる提案経路である．この例の場合，提案経路は所要時間最短の経路より 3 分遅いだけで，乗り換えの必要がなくなり徒歩時間も 6 分減少する．いずれの経路が実用的であるかは利用者の判断に委ねるべきものであると考え，両経路を併記する．

6.3 公 開 実 験

新システムの有効性を確認するとともに，改善点についての意見を収集するために，公開実験を行い，アンケートおよびモニタ調査を行った[17]．公開実験中のシステム利用は，計 10,156 アクセスで，携帯電話による利用が 68%を占めた．

表 2 アンケートおよびモニタ調査
Table 2 A questionnaire survey on the proposed system.

アンケート	
方法	不特定配布による無記名回答
回答数	176 枚（5,000 枚中）
実施期間	2005-12-22〜2006-01-31
モニタ	
方法	路上依頼によるシステム利用と無記名回答
モニタ数	100 名
実施日	2006-01-17，2006-01-21

表 3 探索結果に対する回答の集計結果
Table 3 The aggregated result of the found paths of the proposed system.

	アンケート	モニタ
満足・十分	49%	75%
情報不足	25%	12%
精度を改善すべき	10%	2%
その他・未回答	16%	11%

表 2 にアンケートおよびモニタ調査の方法，回答者数，実施日を示す． アンケート内容は多岐にわたるが，ここでは，探索結果に対する回答の集計結果のみ**表 3** に示す．選択肢の中で，探索経路の質に関する否定的な項目は「精度を改善すべき」というものだけだが，その割合はアンケートにおいて 10%，モニタにおいて 2%と，それほど高くない．また，精度改善について具体的に寄せられた意見は，低床バス，広告バス等のバスの種類を指定した探索を望むもの等であり，経路が実用的でないという意見は皆無であった．したがって，本システムが実用的な経路を出力できている点については，十分に確認できたといえる．

なお，探索結果が「情報不足」と答えた回答者によって，出力経路に追加すべき情報としてあげられたのは，料金情報，周辺地情報，観光情報の順に多かった．

7. お わ り に

バスネットワークを対象として，節に到着した時刻に応じて辺の重みを動的に計算する改良ダイクストラ法と，その実行過程で得られるパラメータによって探索範囲を限定し効率的に全経路探索を行う 2 段階探索によって，実用的な速度で実用的な経路を探索する手法を開発した．この手法では，所要時間最短で，その中では乗り換え回数最少，かつ，好ましくない徒歩移動を含まない限りにおいて，その中では乗車時間がなるべく短い経路が出力される．また，所要時間最短という制約を緩和することで，乗り換え回数，乗り換え時間，徒歩時間の観点から，より実用的と考えられる

経路が得られる場合，それらの経路をあわせて探索する手法も開発した．

開発したアルゴリズムを基に，実際に鳥取市で利用可能なバス経路探索システムを構築した．本システムは，公開実験による有効性の確認を経て，現在，日本トリップ有限責任事業組合によって市民や観光客に供されている[18]．PC や携帯電話等の，WWW にアクセスできる環境さえあれば，誰でも，どこからでも，本システムを無料で利用できる．

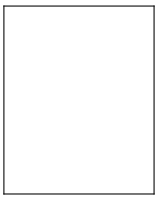
今後の課題として，バス会社やバスの種類を指定した経路探索と，出力経路上に料金，周辺地，観光地情報等を表示することを検討している．また，バスロケーションシステムとの連携の可能性も探りたい．

謝辞　鳥取商工会議所，日ノ丸自動車株式会社，日本交通株式会社，アイコンヤマト株式会社，ソンズ株式会社，中央印刷株式会社，中国運輸局鳥取運輸支局の関係各位に感謝します．

参　考　文　献

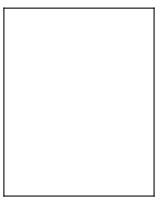
1) 京都市交通局：ポケロケ，http://www.city.kyoto.jp/kotsu/bls/index.shtm.
2) 国土交通省松江国道事務所，松江市交通局，一畑バス（株）：ぐるっとバスなび，http://www.matsukoku-mlit.go.jp/busloc/.
3) 株式会社ぐるなび：えきから時刻表，http://ekikara.jp/.
4) 駅前探険倶楽部：駅探：乗り換え案内 時刻表 路線検索サービス，http://ekitan.com/.
5) 長崎県：長崎県ホームページ：GOOD SITE 経路検索＆地図情報検索＆生活情報サービス，http://goodsite.pref.nagasaki.jp/.
6) Koncz, N., Greenfeld, J. and Mouskos, K.: A Strategy for Solving Static Multiple-Optimal-Path Transit Network Problems, *Journal of Transportation Engineering*, Vol.122, No.3, pp. 218–225 (1996).
7) Liu, C.-L., Pai, T.-W., Chang, C.-T. and Hsieh, C.-M.: Path-Planning Algorithms for Public Transportation Systems, *Proc. 4 Intl. IEEE Conf. on Intelligent Transportation Systems*, pp.1061–1066 (2001).
8) Liu, C.-L.: Best-Path Planning for Public Transportation Systems, *Proc. of 5 Intl. IEEE Conf. on Intelligent Transportation Systems*, pp.834–839 (2002).
9) Wellman, M.P., Ford, M. and Larson, K.: Path Planning under Time-Dependent Uncertainty, *Proc. of the 11th Conf. on Uncertainty in Artificial Intelligence*, pp.532–539 (1995).
10) 川村尚生，楠神元輝，菅原一孔：徒歩移動を考慮するバス経路探索システム，情報処理学会論文誌，Vol.46, No.5, pp.1207–1210 (2005).
11) Dijkstra, E.W.: A Note on Two Problems in Connexion with Graphs, *Numerische Mathematik*, Vol.1, No.296–271 (1959).
12) Fredman, M.L. and Tarjan, R.E.: Fibonacci Heaps and Their Uses in Improved Network Optimization Algorithms, *ACM*, Vol.34, No.3, pp.596–615 (1987).
13) Sneyers, J., Schrijvers, T. and Demoen, B.: Dijkstra's Algorithm with Fibonacci Heaps: An Executable Description in CHR, *Proc. 20th Workshop on Logic Programming*, Vol.1843-06-02, pp.192–199 (2006).
14) Goldberg, A.V. and Silverstein, C.: Implementations of Dijkstra's Algorithm Based on Multi-Level Buckets, Technical Report95–187, NEC Research Institute, Inc. (1995).
15) Eppstein, D.: Finding the k shortest paths, *SIAM Journal on Computing*, Vol.28, No.2, pp. 652–673 (1998).
16) Martins, E. d. Q.V., Pascoal, M. M.B. and Santos, J. L. E.D.: The K Shortest Paths Problem, CISUC Research Report (1998).
17) 中国運輸局鳥取運輸支局：徒歩移動を考慮するバス経路探索システムに係る調査報告書 (2006).
18) 日本トリップ有限責任事業組合：バス・鉄道利用援助システム，http://www.ikisaki.jp/.

(平成 18 年 5 月 15 日受付)
(平成 18 年 11 月 2 日採録)



川村　尚生（正会員）

昭和 40 年生．平成 6 年神戸大学大学院自然科学研究科博士課程単位取得退学．同年鳥取大学工学部知能情報工学科助手，現在，同科助教授．エージェントシステム，バスの利用促進に関する研究に従事．博士（工学）．電子情報通信学会，ソフトウェア科学会，人工知能学会各会員．



菅原　一孔（正会員）

昭和 31 年生．昭和 56 年東京工業大学大学院理工学研究科電子物理工学専攻修士課程修了．同年神戸市立工業高等専門学校電気工学科講師．同校助教授を経て平成 6 年鳥取大学工学部電気電子工学科助教授，現在同学部知能情報工学科教授．計算機工学に関する研究に従事．工学博士．IEEE，電子情報通信学会各会員．