

徒歩移動を考慮した経路探索手法と その鳥取市路線バス経路探索システムへの応用について

楠神 元輝¹ 藤原 好章² 川村 尚生² 菅原 一孔²

二点間の所要時間最短経路を出力するバス経路探索手法を開発し、鳥取市の実際のバス経路探索システムに適用した。システムへの入力は、出発時刻、出発地、目的地の3組で、出発地および目的地はランドマークで指定する。GPS機能付携帯電話からシステムにアクセスした場合は、出発地はGPS機能で与えることもできる。経路探索においてはバス停間の徒歩移動を考慮している。バス停の設置間隔は狭い場合が多く、バス停間を徒歩移動した方が、より短い所要時間で目的地に到達する場合が考えられるからである。また、同じ所要時間の経路においては、なるべく乗換え回数が少なく、徒歩での移動時間が短い経路を出力するようにしている。実験によって提案手法の有効性を確認した。

Path Planning System for Bus Network Including Walking Transfer

Genki Kusugami Yoshiaki Fujiwara Takao Kawamura Kazunori Sugahara

We have developed a path planning system for the bus network in Tottori as Web service. This system finds appropriate paths using location information of the starting point and the destination obtained through GPS or landmark databases and the current time. Our path planning method can find paths including not only bus transfers but walking transfers. Experiments are examined to validate the system.

1 はじめに

いずれの地方も同様の状況であるが、鳥取県でも路線バスの利用者は年々減少の一途をたどっており、平成11年度には70%近い路線が赤字で、路線バス維持のための市町村の負担が大きなものになっている[1]。路線バスの利用者数を増やすためには、様々な視点から路線バスの利便性を高める必要がある。路線バスの不便な点の一つに、経路情報が得にくいということが挙げられる。例えば、鉄道では駅構内の路線図や運賃表等から、また駅員に質問することで経路情報を入手できるが、バス停はほとんどが無人であり、詳しい経路情報を掲示するスペースもないので、入手できる情報は限られている。また、鉄道の駅と異なり、場所を知らせる道路標識等もなく、特に観光客にとっては、どこにバス停があるのかを知ることさえ容易では

ない。さらに、鉄道においては、出発駅と目的駅を指定することで経路情報を出力する経路探索システムが普及しているが[2, 3]、バスを利用する本格的な経路探索システムはまだ存在していない。

そこで我々は、鳥取県のバス会社と協力して、路線バス経路探索システムを開発している。これまでに、バス経路探索を単一始点最短問題ととらえ、経路探索において一般的に使用されるダイクストラのアルゴリズムを適用し、所要時間最短経路を探索する手法を開発した。この手法では、バス停は設置間隔が狭いため、異なる路線に属するバス停間を徒歩で移動することが可能であるという、路線バス経路探索問題特有の性質を考慮し、アルゴリズムやデータ構造を工夫した。そして、この提案手法に基づいた携帯電話から利用できる路線バス経路探索システムを作成した[4]。

この経路探索手法(以後、所要時間最短のみを考慮した探索手法と呼ぶ)では、所要時間が最短となることだけを目標に経路を探索しているため、所要時間最短経路が複数個存在する場合、どの経路

¹鳥取大学大学院工学研究科
The Graduate School of Engineering, Tottori University
²鳥取大学工学部
Faculty of Engineering, Tottori University

が出力されるかは探索順序に依存している．したがって，非実用的，すなわち，乗換え回数が必要以上に多い経路や，徒歩による移動時間が必要以上に長い経路を出力してしまう場合があることが分かった．

この問題を解決するために，所要時間最短のみを考慮した探索手法を行った後，出力された経路を参考としながら再度経路を探索し，乗換え回数と徒歩による移動時間も考慮した経路を出力する手法を開発したので，実験結果とあわせて報告する．本稿の残りは以下のように構成されている．まず，2章で所要時間最短のみを考慮した探索手法を説明した後，3章で新たに提案する，乗換え回数と徒歩による移動時間も考慮した探索手法について述べる．4章で提案探索手法を用いた経路探索システムを説明し，5章で実験結果を通じ，提案探索手法の有効性を示す．最後に6章で結論を述べ，本稿をまとめる．

2 所要時間最短のみを考慮した探索手法

バス経路探索問題は基本的にはネットワークの単一始点最短路問題と考えることができる．

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することでネットワークを構築する．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．たとえば，あるバス路線 α において，8 時台にはバス停 $A \rightarrow B \rightarrow C$ と停車するが，それ以外の時間帯にはバス停 $A \rightarrow C$ と停車する場合，2つの分割路線 $\alpha - 1$ と $\alpha - 2$ 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対してダイクストラのアルゴリズムを適用することで，任意の節から任意の節への所要時間最短経路を求めることができる．ただし，バスを乗り換える場合は，そのバスが到着するまでのバス停での待ち時間を考慮しなければならない．例を図1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 に到着する時刻が10時00分で，次のバスが10時05分に出る場合，待ち

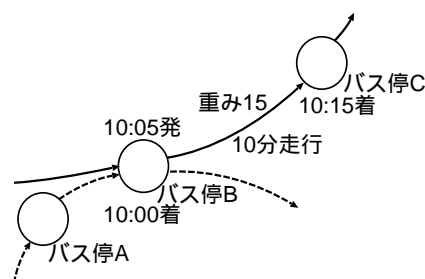


図 1: 待ち時間を含む辺の重み

時間は5分なので，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 $5 + 10 = 15$ 分となる．この待ち時間は到着時刻によって異なるので，辺の重みも一意に決めることができない．

このように，節に到着した時刻に応じて，次の節への辺の重みを動的に計算するアルゴリズムが必要になる．本研究では，ダイクストラのアルゴリズムを，節への到着時刻によって動的に辺の重みを計算するように改良した．ここで，この辺の重みを計算するためのアルゴリズムの計算量は $O(1)$ であり，全体の計算量のオーダーは元のダイクストラのアルゴリズムと変わらない．

しかし，この改良アルゴリズムにより得られる経路は，実際の所要時間最短経路にはならないことがある．なぜなら，バス路線は鉄道とは異なり，2点間を最短距離で結ぶという発想ではなく，なるべく利用客が多い場所を通るように設計されているため，路線が複雑に入り組んでいる上に，バス停の設置間隔は狭い場合が多いので，あるバス停で降りて，他の路線上のバス停まで徒歩で移動して乗り換えることが可能だからである．

図2(a)に示す例において，バス停 A からバス停 H に移動する場合に，徒歩移動を考慮しなければ， $A \rightarrow B \rightarrow C \rightarrow D \rightarrow H$ という経路しか得られないが，徒歩移動を考慮すると $A \rightarrow B \rightarrow F \rightarrow G \rightarrow H$ という経路も探索対象に含めることができる．鉄道の駅は設置間隔が広いので，このような徒歩移動は考慮する必要がなく，バス経路探索問題に特有のものと言える．乗り換えバス停以外の乗り換えに対処するために，ネットワークの各節から他のすべての節に対して徒歩移動による辺を加えるものとする．この辺の重みは，ここで，バス停 A からバス停 B までの徒歩による移動時

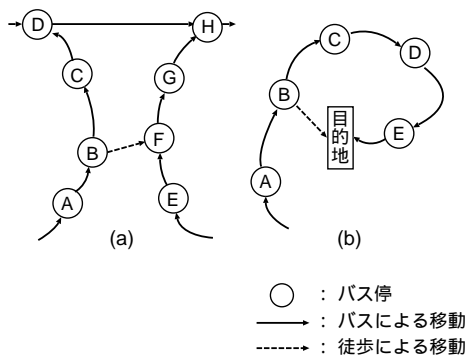


図 2: 徒歩移動が有効となる経路例

間 [分] $T_{A:B}$ であり, 次式で与える .

$$T_{A:B} = [N \times d(A, B)] \quad (1)$$

$d(A, B)$ はバス停 A からバス停 B までの直線距離 [m], N は定数である .

原理的には全バス停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, 適当なしきい値 T_1 を定め, 徒歩による移動時間が T_1 以下のバス停間のみを結ぶこととする .

また, 目的地の最寄りバス停以外のバス停で降り, 目的地まで歩いた方が早いこともあり得る . 図 2 (b) はそのような例で, 目的地の最寄りバス停を E とした場合, 徒歩移動を考慮しなければ, $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow$ 目的地という経路しか得られないが, 徒歩移動を考慮すると $A \rightarrow B \rightarrow$ 目的地という経路も探索対象に含めることができる . 同様に, 出発地に対しても最寄りバス停以外のバス停まで徒歩移動してバスに乗ることを考慮しなければならない . すなわち, 鉄道の経路探索では駅から駅までを経路とするのに対し, バスの経路探索においては, 実際の出発地から目的地までを経路としなければならない . このために, 出発地や目的地も節としてネットワークに加え, 他のすべての節と徒歩移動による辺で結ぶものとする . 原理的には, 出発地から全バス停への辺, 全バス停から目的地への辺が張られるべきであるが, 同様に, 適当な T_2 を定め, 出発地, 目的地とも徒歩による移動時間が T_2 以下のバス停のみと結ぶこととする .

ダイクストラのアルゴリズムでは, ある節に後続する節のうち最小の最短経路推定値を持つものを選ぶ (以後, Extract-Min 操作と呼ぶ) 際の計算量

がアルゴリズム全体の計算量を支配する . このとき用いられるデータ構造のうち, 現在理論的に最も計算時間が短くなるものは, フィボナッチヒープを用いた優先順位付待ち行列であり, Extract-Min 操作 1 回の計算量は節が n 個のとき $O(\log n)$ となる . しかし, 本手法では, フィボナッチヒープではなく, バケット法と呼ばれるデータ構造を用いた [5] . これは, 最短経路推定値が取り得る値毎に待ち行列を用意する方法で, Extract-Min 操作 1 回の計算量は $O(1)$ であるが, メモリ使用量の問題から, 最短経路推定値が取り得る値が限られている場合にしか使用できない . バス経路探索問題においては, 最短経路推定値は分単位で与えられ, 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言える .

3 乗換え回数と徒歩による移動時間も考慮した探索手法

所要時間最短のみを考慮した探索手法は, ダイクストラのアルゴリズムを基本としているため, 所要時間が最短となる経路は出力されるが, 同じ所要時間の経路が複数個存在する場合, その中のどの経路が出力されるかは探索順序に依存している . したがって, 非実用的, すなわち, 乗換え回数が必要以上に多い経路や, 徒歩による移動時間が必要以上に長い経路を出力してしまう場合がある .

そこで, 以下のように最適経路を定義する .

1. 所要時間最短となる経路数が 1 ならば, その経路 .
2. そうでない場合, それらの経路のうち, 乗換え回数最小の経路数が 1 ならば, その経路 .
3. そうでない場合, それらの経路のうち, 徒歩による移動時間が最短の経路数が 1 ならば, その経路 .
4. そうでない場合, 残った経路のうち任意の経路 .

最適経路を, 所要時間最短のみを考慮した探索手法によって得られた経路を条件にして探索範囲を限定しつつ, バス路線ネットワークの全経路を再度探索することによって求める手法を開発した .

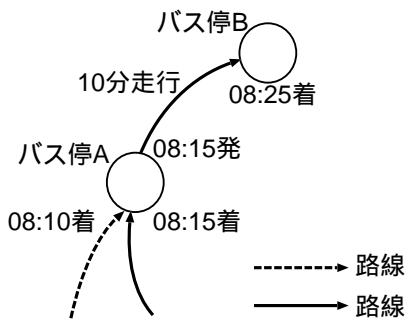


図 3: d で探索範囲を限定してはいけない例

その際、以下の場合にその節以後の探索を行わないことで、探索範囲を限定する。

1. ある節への到着時刻が、出発時刻に出発地から目的地までの最短所要時間を加えた時刻を越えた場合
2. ある節へ到着したときの乗換え回数が所要時間最短のみを考慮した探索手法によって得られた経路の乗換え回数を越えた場合

探索範囲をさらに限定するために、上記の条件 1 を「ある節への到着時刻が、出発時刻にその節の d を加えた時刻を越えた場合」という条件 1' として、より厳しく変更することを考える。ここで、 d とは、所要時間最短のみを考慮した探索手法で経路を得る際に求められている、出発地からその節までの最短所要時間である。

2 章で述べたネットワークでは、このように条件を厳しくすることは不可能である。図 3 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、路線 α 上のバスでバス停 A に到着するのが 08:10 とし、路線 β 上のバスでバス停 A に到着するのが 08:15 としよう。路線 α 上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 上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 08:25 に到着する。バス停 A の d はその節への到着時刻のうちもっとも早いものなので、08:10 である。したがって、厳しくした条件を適用すると、路線 β を通る経路は探索対象とはならなくなってしまうが、この経路の方がより実用的な経路である可能性がある。

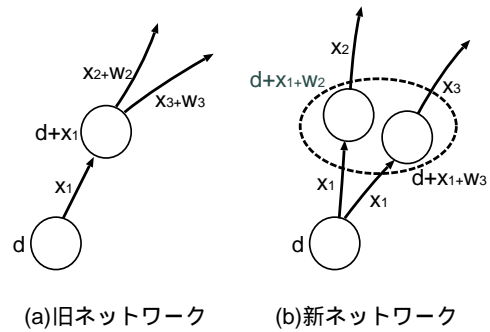


図 4: 旧ネットワークと節を分割した新ネットワーク

この問題に対処するため、 d の定義を、「出発地からその節まで到着した後、その節を出発する時刻のうち最も早いもの」と変更する。図 3 では新しく定義した d は 08:15 となり、路線 β を通る経路も探索対象となる。

ある節から複数の辺が出ている場合、どの辺に向かうかによって、その節の出発時刻が異なる。したがって、上記の条件を適用するためには、節から出る辺の数を 1 にしなければならない。このため、節を分割して新しいネットワークを構築した。図 4 (a) に分割前の旧ネットワークの部分例を、図 4 (b) に分割後の新ネットワークの部分例を、それぞれ示す。図中の $x_i (i = 1, 2, 3)$ は辺の重みを、 $w_i (i = 2, 3)$ は節での待ち時間を示している。図に示されているように、旧ネットワークでは、節での待ち時間は、その節から出る辺の重みとして加えられていたが、新ネットワークでは、節の d に加えられる。

新ネットワークでは条件 1' が適用でき、探索範囲を大幅に限定することができる。

4 路線バス経路探索システム

バス経路探索システムを Web サーバを通じて利用できる CGI プログラムとして作成した。開発言語には Ruby を使用した。

システムの入力画面を図 5 に示す。システムへの入力には、出発時刻、出発地、目的地の 3 組である。出発時刻は初期値として現在時刻が設定されている。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現在鳥取市内の主要な施設 2085 個を MySQL で管理している。また

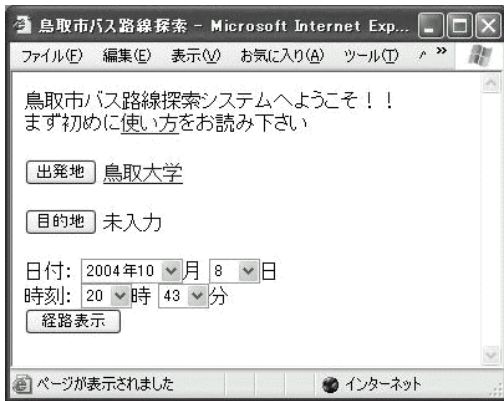


図 5: 入力画面例

ランドマークで指定した出発地，目的地にはその座標を中心とした地図へのリンクが張られている．図 5 の例では，「鳥取大学」をクリックすると，鳥取大学の座標を中心とした地図のページが得られる．なお，GPS 機能付携帯電話からシステムにアクセスした場合は，出発地は GPS 機能で与えることもできる．

システムで扱うデータは，日ノ丸自動車（株）と日本交通（株）によって鳥取県東部で運行されている計 93 路線と 1806 個のバス停を基に，次の 4 種類のデータベースを関係データベースサーバ MySQL によって作成した．

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

(1) は，各レコードがバス停一つに対応しており，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一つに対応しており，分割路線 ID，始点バス停 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ものである．ただし学期間とは，小学校等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上の停車バス停に対応しており，接続 ID，バス停 ID，次のバス停の接続 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

が分割路線の始点バス停の発車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，ネットワークの静的な部分を構築する．そして，探索を行う毎に，出発地と目的地を加えることで経路探索に用いるネットワークを完成させ，経路探索を行う．

5 実験

乗換え回数と徒歩による移動時間も考慮した探索手法の有効性を確認するため，所要時間最短のみを考慮した探索手法との比較実験を行った．なお，2 章の N は $1/50$ ， T_1 は 5， T_2 は 20 とした．ただし， T_2 の値は，バス停が見つからない場合，見つかるまで 10 ずつ増やしていく．

探索結果の比較を図 6 に示す (a) の旧手法は所要時間最短のみを考慮した探索手法で，(b) の新手法は乗換え回数と徒歩による移動時間も考慮した探索手法である．いずれの経路においても，到着時刻は 17:23 で同じであるが，旧手法により出力された経路は乗換えを 3 回行うのに対し，新手法により出力された経路は乗換えを 2 回しか行わないことがわかる．

6 おわりに

節に到着した時刻に応じて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のアルゴリズムと，それによって求めた解を上限値として利用した全解探索を組み合わせることによって，実用的な経路を出力するバス経路探索が可能になった．また，探索アルゴリズムにバス停間等の徒歩移動を組み込むことにより，所要時間がより短く，かつ実用的な経路が出力できることがわかった．開発したアルゴリズムを基に，鳥取県のバス会社と協力することで，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．

本システムは現在試験的に運用中であるが，2005 年度から鳥取県商工会議所により本格運用される予定である．そのために残された課題として，経路の評価基準として，所要時間だけではなく，料金を組み込むことが挙げられる．また，利便性を高めるためには，探索時間をより高速化することも必要である．現在，平均 20 秒程度の検索時間

16:40 出発地『鳥取市立病院』発
 ↓ 徒歩
 16:42 「市立病院」着
 16:45 「市立病院」発
 ↓ 日ノ丸 市立病院線
 16:55 「鳥取駅」着
 17:00 「鳥取駅」発
 ↓ 日ノ丸 鳥大（布勢）線
 17:17 「湖山」着
 17:17 「湖山」発
 ↓ 日ノ丸 湖岸線
 17:19 「鳥商前」着
 ↓ 徒歩
 17:23 目的地『県立鳥取商業高等学校』着

(a) 旧手法の探索結果

16:40 出発地『鳥取市立病院』発
 ↓ 徒歩
 16:42 「市立病院」着
 16:45 「市立病院」発
 ↓ 日ノ丸 市立病院線
 16:58 「鳥取駅」着
 17:00 「鳥取駅」発
 ↓ 日ノ丸 鳥大（布勢）線
 17:19 「鳥商前」着
 ↓ 徒歩
 17:23 目的地『県立鳥取商業高等学校』着

(b) 新手法の探索結果

図 6: 探索結果の比較

を要しているが、システム開発言語を Ruby から C++に変更することで十分な速度が得られるものと考えている。

謝辞

鳥取大学工学部の三好力助教授，鳥取商工会議所の伊藤氏，日本交通株式会社および日ノ丸自動車株式会社の関係各位に感謝します。

参考文献

- [1] 中村理人: 21 世紀の鳥取県の公共交通のあり方に関する研究, 財団法人とっとり政策総合研究センター ニュースレター, No. 17 (2001).
- [2] 駅前探険倶楽部: <http://ekitan.com/>.
- [3] Yahoo!路線情報: <http://transit.yahoo.co.jp/>.

- [4] 楠神元輝, 川村尚生, 菅原一孔: 徒歩移動を考慮した路線バス利用援助システムについて, 電気学会第 15 回情報システム研究会発表資料集, pp. 45-50 (2004).
- [5] Goldberg, A. V. and Silverstein, C.: Implementations of Dijkstra's Algorithm Based on Multi-Level Buckets, Technical Report 95-187, NEC Research Institute, Inc. (1995).