

多数経路の比較容易性を向上させた経路編集型乗換案内インタフェース

内山 雄貴[†] 小林 亜樹^{††}

[†] 工学院大学 工学部情報通信工学科 〒163-8677 東京都新宿区西新宿 1-24-2

^{††} 工学院大学 情報学部情報通信工学科 〒163-8677 東京都新宿区西新宿 1-24-2

E-mail: [†]c514016@ns.kogakuin.ac.jp, ^{††}aki@cc.kogakuin.ac.jp

あらまし 一般的な乗換案内サービスよりも利用者の意図に従った柔軟な経路選択を行える乗換案内インタフェースとして、経路編集方式[1]を提案している。しかし、既存研究では多数経路の表示と選択に問題があった。そこで本稿では、経路編集時に利用者が選択できる経路数を増加させつつ、迅速な経路間の比較判断を行えるインタフェースについて提案する。具体的には、経路編集時に新たな候補集合を得るにあたって、複数の順位付け条件から選択できるようにするとともに、追加候補経路の既存表示経路との共通部分経路を省略表示し、さらに、他の候補経路の表示を適当に簡約化することによって、利用者の注目箇所がわかりやすいようにした。経路を迅速に発見することができるについて、既存のインタフェースと比較を行うことで検証を行う。

キーワード 乗換案内, 経路探索, ユーザインタフェース, エッジ集約

1. はじめに

公共交通の円滑な利用を目的として乗換案内^(注1)サービスが利用されている。このような乗換案内サービスの利用経路を検索する操作では、利用者は経済合理的な消費行動に基づいた指標だけでなく、自身の知識や経験、日常的な外的要因を組み合わせ、経路を閲覧することにより所望経路を発見し、行動の意思決定をしている。[2]

現在普及している乗換案内のインタフェースでは、表示された検索結果において複数の候補経路を閲覧、比較を行うことで所望経路を得ている。しかし、乗換経路探索 Web サービスの利用者調査[3]によると、約8割の利用者が上位経路を選択しており、複数経路の比較を行わずに経路を決定している。その原因の一つは、従来の乗換案内サービスでは多数経路の比較が迅速に行うことができないからであると考えられる。利用者調査[4]によると、検索結果に従って即交通行動に移る利用者が大半であり、所望経路発見までの迅速性は重要であると筆者らは考える。

JR 東日本が提供しているスマートフォン向けアプリ「Go! by Train」[5]では、結果表示画面において、多数の候補経路の表示がされ、経路間の比較が行える。しかしながら、比較の迅速性は考慮されていない。

本研究では、知識条件という経路分割手法の意義を説明する利用者モデルを導入し、拡張した経路編集方式下での多数経路の比較容易性を向上させる省略表示を提案する。本稿ではまず、乗換案内、経路探索のモデル化を行う。その後、本提案を用いた試作システムの実装および評価実験を行い、所望経路を発見できるまでの探索所要時間を測定し、提案手法が比較容易性の向上に有用であるかを検証する。

(注1)：本来、乗換案内という単語はジョルダン株式会社の提供する公共交通利用経路案内システムを指すが、本稿では一般的な呼称に習い経路案内サービス全般を指す言葉として用いる。

2. 乗換案内のモデル化

2.1 システムのモデル化

本研究において筆者らは、乗換案内システムのモデル化を行っている。[1]

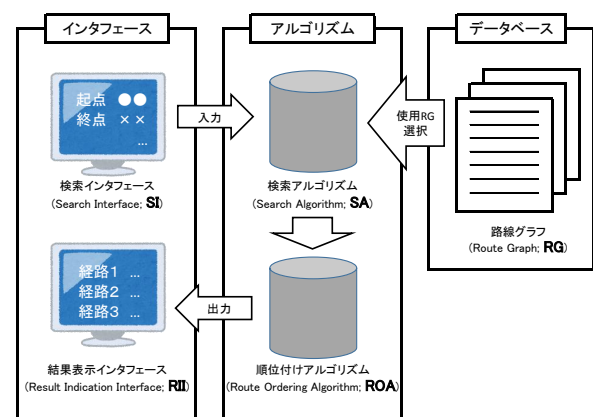


図1 乗換案内のシステムモデル

乗換案内システムは、図1に示すように、利用者がシステムに条件を入力するのに用いる検索インタフェース (Search Interface; SI)、入力された条件に従い経路の探索を行う検索アルゴリズム (Search Algorithm; SA)、探索された経路に対して順位付けを行う順位付けアルゴリズム (Route Ordering Algorithm; ROA)、利用者が検索結果を閲覧し比較するのに用いる結果表示インタフェース (Result Indication Interface; RII) により構成される。

検索インタフェース (Search Interface; SI) では、利用者が乗換案内で検索を行うためのクエリを入力するインタフェースである。一般的な乗換案内サービスでは、利用者は乗車駅、下車駅、経由駅で表される検索対象区間の情報と、時刻、優等列車や飛行機などの利用条件、表示する経路の順位付け条件を入力する。

検索アルゴリズム (Search Algorithm; SA) は、SI で入力された条件に基づき、条件に合致する経路を探索するアルゴリズムである。SA ではまず、入力された時刻と利用条件により使用する路線グラフ (Route Graph; RG) を決定し、利用可能な経路を探索する。

順位付けアルゴリズム (Route Ordering Algorithm; ROA) は、SA で探索された経路を SI で指定された順位付け条件に基づき並び替えるアルゴリズムである。あらかじめ SI で指定された条件に従って検索結果を並び替えることで、利用者の所望経路が結果表示画面上ですぐに発見できるようになる。

結果表示インタフェース (Result Indication Interface; ROA) は、ROA によって並び替えられた検索結果のうち上位数件を表示するインタフェースである。経路情報同士を比較したり、再検索操作を行ったり、実際に行動する際の情報を取得するために用いられる。

2.2 経路探索のモデル化

本稿では、経路探索を乗車駅と下車駅を入力とし、システムが提示した候補経路中から任意の一経路を最終的に選択することとモデル化する。このとき、最終的に選択された経路を所望経路と呼ぶ。所望経路は検索前に既に利用者の意識上で明確であることもあり、また検索操作過程において潜在的な所望経路が変化することも考えられるが、本稿では結果的に選択される経路に対して所望経路と呼ぶこととする。なお、簡単のため以下では所望経路が全候補経路集合を閲覧しても存在しない場合については無視して議論を進める。

一般的な乗換案内サービスにおける ROA では、時間、運賃、乗換回数の 3 種類の指標をもとに候補経路集合を並び替えている。この 3 種類の指標を順位付け条件と呼ぶとする。しかし、順位付け条件だけでは利用者の所望経路が得られない場合がある。例えば、利用者が

- 空いている A 線を経由して目的地へ向かいたい。
- 普段利用している B 線で目的地へ向かいたい。

などの欲求がある場合、所望経路が得られない場合が多い。これは、一般的な乗換案内サービスにおいてはこのような指標が存在せず、欲求に対する候補経路集合を得られないからである。しかしながら、上記のような交通網に関する知識に加え、一般的な乗換案内サービスの利用経験があり、「ある順位付け条件を選択し、示された候補経路集合から所望経路を呼び出すことができる」という、候補経路集合の出現傾向による知識も持ち合わせているときには、順位付け条件を利用者が選択するということは所望経路を得るにあたって有用である。この 2 種類の知識を知識条件と呼ぶこととする。

乗換案内サービスを使用して経路を探索する際に、知識条件が成立している区間においては、順位付け条件を純粹にその順位付け条件のために用いてはいない。すなわち、順位付け条件は所望経路が含まれた候補経路集合を得るための近似として用いている。このとき、利用者は「A 線を利用する」といった正解経路を知っている場合も含んでいるが、ほとんどの利用者は正解経路が存在している場合でも乗換案内サービスを使うことが多い。これは、経路には具体的な時刻や運賃などの詳細な属

性情報も含まれ、これらは非常に多くの組み合わせが存在するため、それらを正確に知識として記憶しており、かつその記憶に確信が持てる利用者というものは非常に想定しにくいと考えられる。そのため、正解経路を知っている場合でもその区間を含んだ経路を探索している。

このような知識条件を満たす利用者の割合などについての直接の調査を筆者らは知らない。しかし、利用者調査 [3] によると、前記 3 指標が経路の選択理由であるとしたのは全体の約 45% に過ぎず、それでも乗換案内サービスを利用しているという事実は、利用者の相当数において、何らかの知識条件を満たす状態にあることを強く示唆するものである。

3. 提案手法

3.1 概要

本研究では、利用者が経路候補中の任意の駅で経路を分割し、改めて検索部より経路候補集合を得る方式を前提に、再検索時の順位付け条件変更時に経路候補数が多くなり、画面表示上煩雑になる点を解消することを目指す。具体的には、経路候補中の共通している区間の省略表示を行うことで経路の差分を明確に示し、また他の経路候補の簡約化表示を行うことで、利用者の注目箇所を分かりやすく示す手法を提案する。

3.2 経路分割のモデル化

2.1 節のようにモデル化すると、単一の経路上の任意の駅で経路検索を分割して行っても、SA と ROA の組み合わせにより出力される候補経路集合は同一であることになる。しかし、実際には ROA によって経路数を限定する過程で候補からふるい落とされる経路が存在するために、一般に短い区間における検索のほうに候補経路の多様性が増す結果となる。以前の手法 [1] では、ここで増加した候補経路を活用するためのインターフェイスとすることが主眼であった。

しかし、ROA に与える順位付け条件を異なるものとした場合には、仮に同一の候補経路数までに限定したとしても出力される候補経路の多様性がさらに増すこととなり、多くの経路選択候補を得ることでより柔軟な経路選択を実現できる可能性がある。2.2 節で述べたような知識条件を持つ利用者の場合にはこのような利用を行う可能性が想定でき、本稿ではこのような場合に得られる多数の候補経路集合をどのように提示するかという観点を検討している。

3.3 経路分割による所望経路発見の手順

経路分割にあたって、まず入力された順位付け条件 $O_c((1) \text{ 式})$ をもとにした全区間 $[S_0, S_d]$ の候補経路集合 $R_1((2) \text{ 式})$ を得る。ここで入力する O_c は、分割駅 S_y までの O_c となる。

$$O_c = \{t_o, f_o, c_o\} \quad (1)$$

$$R_1 = \{r_{11}, r_{12}, \dots, r_{1y}, \dots, r_{1n}\} \quad (2)$$

O_c のうち、 t_o は時間順、 f_o は運賃順、 c_o は乗換回数順を示している。

次に、任意の一経路 $r_{1y} \in R_1$ 内にある経路途中の駅 S_y を選択する。このときの S_y とは知識条件成立不成立の境界駅であ

り、また任意の一経路 r_{1y} とは、区間 $[S_0, S_y]$ における所望経路である。

S_y を選択した後、 $[S_y, S_d]$ での部分区間において、選択された O_c をもとに再検索を行うことで、追加候補経路集合 R_2 ((3) 式) を得る。

$$R_2 = \{r_{21}, r_{22}, \dots, r_{2y}, \dots, r_{2n}\} \quad (3)$$

再検索を行い、得られた R_2 は部分区間 $[S_y, S_d]$ の経路しかないため、 $[S_0, S_y]$ においては r_{1y} における同区間と結合を行う。最後に R_2 から区間 $[S_y, S_d]$ における所望経路 r_{2y} を選択することで、全区間 $[S_0, S_d]$ における所望経路を得る。

3.4 経路数問題の解決手法

経路分割にあたって、追加候補経路集合を得ることで多数の候補経路が存在することとなる。このうち、 R_2 中の経路はどれも $[S_0, S_y]$ では r_{1y} と共通と見做せる。既存の経路編集方式では、これら同内容の共通部分経路をそれぞれ表示しているため、煩雑となり、経路間の差分を認識する妨げとなってしまう。

そこで提案手法では、共通部分経路を表示せず、代わりとして木構造的に集約することで、共通であることと、差異があることを可視化して提示する。これにより、より候補経路間の差分を容易に判断が行える。

また、 r_{2y} の発見の際に、利用者の注目は r_{1y} とその派生経路集合にあると考えられるが、 R_1 上での経路についても比較したい場合はある。そこで、提案手法では R_1 を駅名、路線名を省略し、 R_2 よりも簡約表示化を行い、注目領域の隅に残す。

既存の経路編集方式では、経路を分割し r_{1y} を選択すると、分割後の候補経路画面では R_1 のうち r_{1y} 以外の経路は非表示となっている。これは、 R_1 をすべて表示すると迅速な経路比較が行えないからである。

このような描画を行うことで多数経路が比較できるとともに、分割後の経路候補である R_2 をより注目箇所を明確化する効果もある。その一方で、 R_3 では簡約化しつつも、順位付け条件でもある所要時間、乗換回数といった指標に関しては比較できるよう描画している。このように、 R_2 とあわせて R_3 を簡約描画することで、利用者の注目箇所を分かりやすく示しつつ、多数経路の比較を行うことのできるインタフェースとなる。

4. 試作システム

4.1 システム構成

本システムでは、利用者が指定したそれぞれの部分経路に順位付け条件を入力し、所望経路を発見するインタフェースについて、Web ブラウザ上で実行できるシステムとして実装を行った。なお、本システムにおける SA および ROA に関しては、株式会社駅探の乗換 ASP サービス (標準時分版)^(注2) を用いており、主にインタフェース部を実装した。途中駅を含めた駅情報に関しても本サービスのデータを用いている。なお、本実装では具体的なダイヤ情報を用いない平均移動時間を用いるサー

ビスのみを契約しており、候補経路はその範囲内から検索されるが、インタフェース利用時の効果の本質は同様である。

4.2 構成画面

4.2.1 検索画面

検索画面においては、一般的な乗換案内サービスと同様に、乗車駅 S_0 と下車駅 S_d 、あわせて区間 $[S_0, S_d]$ における順位付け条件 O_c ((1) 式) を入力する。

図2 検索画面

4.2.2 経路分割

検索を実行すると、順位付け条件をもとにした全区間 $[S_0, S_d]$ の候補経路集合 R_1 ((2) 式) を得る。

次に、 $r_{1y} \in R_1$ を経路途中の駅 S_y 以降で異なる経路を検索、表示させたい場合には、 S_y と区間 $[S_y, S_d]$ における O_c の選択を行う。

S_y は r_{1y} 上にある経路途中の駅となるが、元経路において乗り換えを要する乗換駅である場合と、乗り換えを要さない途中駅である場合が想定できる。本システムでは、乗換駅である場合と途中駅である場合において S_y の選択方法が異なるため、それぞれの場合で手順を述べていく。

S_y が乗換駅である場合は、図3(a)に示すような r_{1y} の右側にある「早」「安」「楽」ボタンのいずれかのボタンを押す。それぞれのボタンは O_c を選択するものであり、「早」は t_o 、「安」は f_o 、「楽」は c_o を示している。 O_c を選択した後、図3(b)にあるような S_y を選択する画面へ遷移する。ここで、各乗換駅の上に表示される「→」ボタンを押すことで、 S_y を選択する。例えば、図3で横浜駅を選択したい場合には、図3(b)の横浜駅の上にある「→」をクリックする。



図3 S_y の選択画面（乗換駅の場合）

S_y が途中駅である場合には、 r_{1y} のうち、 S_y が存在する区間の路線のラインを押す。路線ラインを押すと、画面上に図4のように途中駅が描画される。^(注3) それぞれの途中駅の右側に「早」「安」「楽」といった O_c を選択するボタンが配置されているため、 S_y の右側にある O_c のボタンを選択する。

このように異なる手順としているのは、画面表示上の制約と操作回数の抑制とを考えた結果である。

(注2) : <https://go.ekitan.com/service/asp/transit/index.shtml>

(注3) : 実装の都合上、途中駅が取得されない場合や、通過駅が表示される場合がある。



図 4 S_y の選択画面（途中駅の場合）

4.2.3 多数経路の描画

4.2.2 節の操作により、 $[S_y, S_d]$ での部分区間において、選択された O_c をもとに再検索を行うことで、追加候補経路集合 $R_2((3) \text{ 式})$ を得る。再検索を行い、得られた R_2 は部分区間 $[S_y, S_d]$ の経路しかないため、 $[S_0, S_y]$ においては r_{1y} における同区間と結合を行い、描画を行う。

併せて、 R_1 上での検索順序近傍経路 ((4) 式) についても比較したい場合はある。そこで、 R_1 のうち $R_3((4) \text{ 式})$ を簡略化して描画を行う。

$$R_3 = \{r_{1y-2}, r_{1y-1}, r_{1y+1}, r_{1y+2}\}, R_3 \subset R_1 \quad (4)$$

これは、 R_1 は利用者の選択した O_c をもとに候補経路が並んでおり、例えば t_o で検索を行うと概ね所要時間が短い順に $\{r_{11}, r_{12}, \dots, r_{1y}, \dots, r_{1n}\}$ となる。なお、ROA のアルゴリズムの関係上、必ずしもそのように並んでいるとは限らない。そのため、 r_{1y} の近傍経路は所要時間が r_{1y} に近い値の経路となっている。

なお、 R_3 以外の R_1 中の経路については本実装では描画領域の関係上非表示となっているが、描画領域を拡張することで表示が可能である。

以上の操作により、それぞれの部分区間ごとに所望経路を発見していく。経路を 2 分割した際における本システムの流れとしては以上の操作の通りであるが、3 分割以上の操作を行わないと利用者の所望経路が得られない場合もある。本システムでは、このような再帰的分割操作も可能ではある。しかしながら、実利用上においては操作や表示が煩雑になるため、さらに工夫しないと実装は難しい。このため、当該機能は未実装であり、今後の課題である。

R_2, R_3 の候補経路集合を描画した試作システムの実装画面を図 5 に示す。図 5 では、まず富士駅から多摩動物公園駅まで t_o で検索を行った後、途中駅である沼津駅で分割を行い、沼津駅から多摩動物公園駅を f_o で探索を行った後の表示画面である。

4.3 実装上の制約

再検索によって得られた R_2 は、部分区間 $[S_y, S_d]$ の経路であり、元々の検索区間である $[S_0, S_d]$ における $[S_0, S_y]$ においては、 r_{1y} における同区間と同一であるといえる。すなわち、検索区間全体の候補経路は前半と後半の各区間の接続経路となる。このとき、区間分割を指定した S_y においては、場合によっては直通列車であったり、乗換移動が必要であったりするため、SA により適切な乗換設定が本来必要な箇所である。

しかし本稿では、このような具体的な経路を指定しての適切な経路情報を得ることに對して、試作システムの実装上の制約

としての SA が対応していないため、 S_y における乗換時間は、 S_y が r_{1y} 上の乗換駅であった場合にはその乗換時間、そうでない場合には 5 分に設定する。なお、接続経路中に不要な重複区間が含まれる場合にはできるだけ試作システムにて除外するようにしたが、これも本来 SA が対応すべき点である。

また、実装上の制約として、結合を行った経路に関しては運賃計算ができない仕様となっているが、これに関しても本来は SA が対応すべき点である。

5. 実験

提案手法によるインタフェースが、経路編集方式より迅速に経路間の比較を行い、所望経路を発見できるインタフェースであるのかを検証する。評価指標には、実験参加者が検索画面を入力し、候補経路が表示されてから、問題として提示した条件を満たす経路を発見するまでの探索所要時間を用いる。

5.1 比較対象

比較対象として、従来の経路編集方式 C 、提案手法における順位付け条件だけを拡張したインタフェース P_1 、多数経路の比較容易性を向上した提案手法によるインタフェース P_2 に関してそれぞれ試作システムを用意し、所望経路発見までの時間を比較する。各方式間の差分に関して、表 1 に示す。

表 1 比較対象の各特徴

比較対象	C	P_1	P_2
表示経路数	3	10	10 ^(注4)
区間毎の O_c の指定	×	○	○
共通部分経路の木構造的な集約	×	×	○

C では図 6 のような既存研究である経路編集方式を利用して、この方式は、 O_c を選択せずに経路を分割し、新たな候補経路を描画するというものである。このインタフェースでは多数経路を表示してしまうと経路間の比較が困難になってしまうため、経路選択の迅速性が損なわれてしまう。そのため、表示経路数が 3 に絞られている。表示経路数を少なくすることで、経路比較の容易性を保ちつつ、ある程度は迅速に経路選択を発見することができる方式となっている。しかし、候補経路が少なく、部分区間毎に O_c を選択していないため、所望経路を得られない場合が生じるという問題もある。

P_1 では図 7 のような、提案手法のうち一部を実装したものである。このインタフェースでは、それぞれの部分区間において O_c を入力し経路を分割し、得られた新たな候補経路集合を描画するというものである。このインタフェースでは表示経路数を 10 に増やしている。各部分区間毎に O_c を選択することで候補経路の多様性が増し、多くの候補経路から所望経路を得ることは可能になった。しかし、提案手法のうち、共通部分経路の木構造的な集約を行っていないため、経路間の差分の比較が容易に行うことができないため、経路発見の上での迅速性は考慮されていないという問題がある。

P_2 では P_1 を 3.4 節のように共通部分経路を木構造的に集約を行い、改善を行ったインタフェースである。順位付け条件を

(注 4)：簡約表示化した候補経路は含まない。

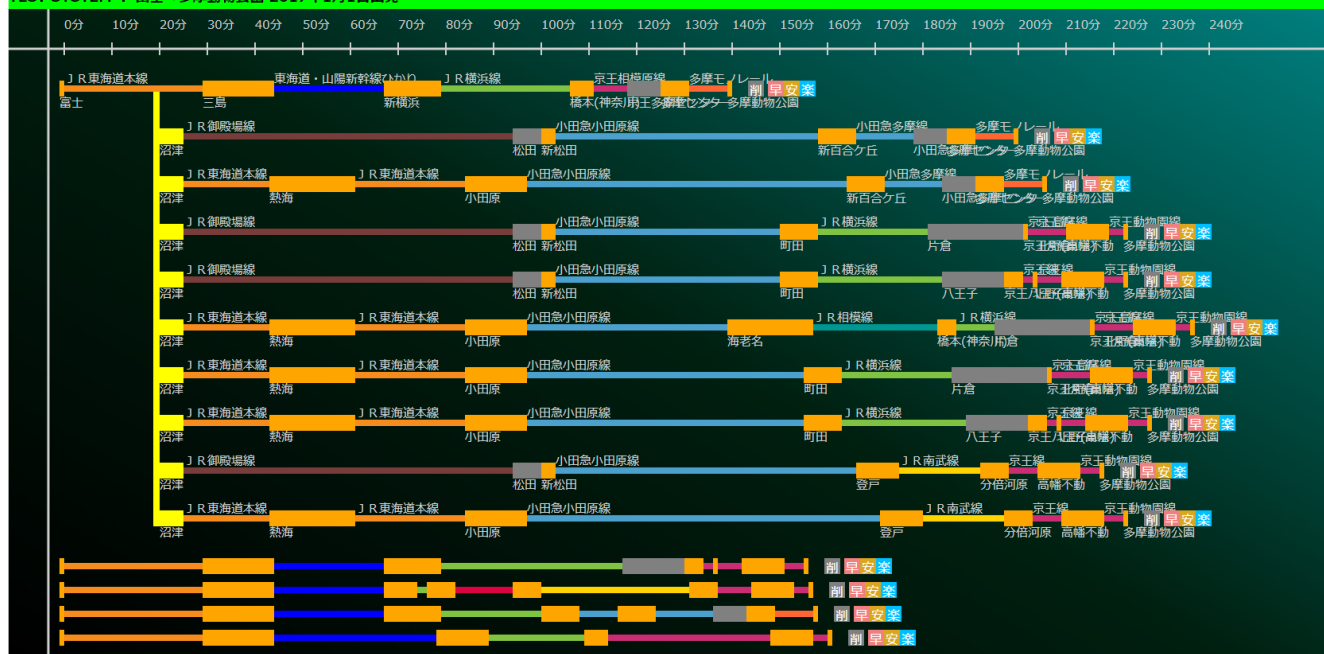


図5 分割後の結果表示画面

もとに描画を行う点は P_1 と同様であるが、木構造的な集約により経路同士の差分がより明らかとなり、迅速性を考慮したインタフェースとなっている。

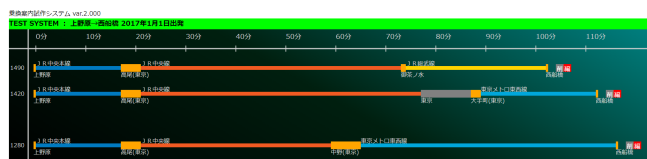
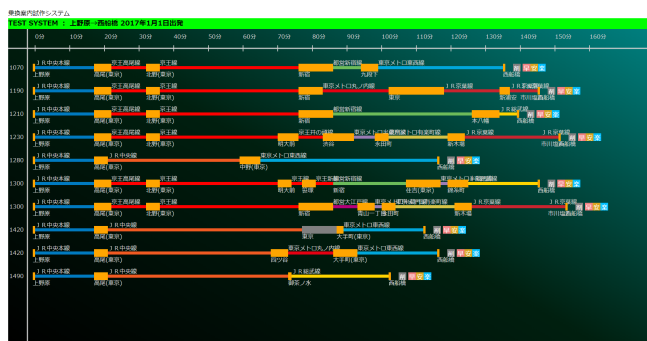


図6 経路編集方式 (インタフェース C)

図7 提案手法の一部を実装した方式 (インタフェース P_1)

5.2 実験手順

まず、実験参加者にそれぞれのインタフェースの操作方法を理解させるために、概ね 10 分程度書面でのマニュアルを見ながら表 2 の例題を実施者とともに解く。

表 2 における条件 1. では、利用者が「吉祥寺駅-新宿駅」の部分区間における知識があり、JR 中央線を所望経路としており、かつ、順位付け条件で「時間順」を用いることで所望経路を得られるという 2 種類の前提知識を持っていることを想定している。今回は、この前提知識を実験参加者に「条件」として提示をしている。なお、条件 2. は利用者が「新宿駅-茗荷谷駅」

表2 例題

乗車駅	吉祥寺 (きちじょうじ)
下車駅	茗荷谷 (みょうがだに)
条件	1. 新宿駅まで JR 中央線 (快速) で到着する。(時間順) 2. 新宿駅から乗換なしで茗荷谷へ到着する。(乗換回数順)

の部分区間における知識がなく、単純に乗換回数が少ない経路で到着する経路を所望経路としていることを想定している。

各方式インタフェースで例題を解き、参加者が操作方法を理解した後、用意した問題を計 3 問解いてもらう。参加者にそれぞれのインタフェースの検索画面で乗車駅、下車駅、順位付け条件^(注5)を入力し検索を行う。

1 問の探索所要時間は、各方式インタフェースにおいて検索区間の入力後、最初の検索候補経路画面の表示が完了したところから、問題の解答終了までの時間を指す。問題の解答終了には正答終了と誤答終了の 2 通りがある。正答終了は、問題毎に設定された正解経路が画面上に提示されており、参加者がその経路が正解経路であるとの見解を実施者に伝えることである。ただし、問題によっては正解経路が提示されない場合が存在し、その場合は正解経路が存在しないという見解を実施者に伝えることを以って正答終了とする。一方で誤答終了とは、被験者が正解経路であるとの見解を伝えたが正解経路でない場合と、正解経路である見解を伝えられずに時間切れとなる場合とを指す。それぞれの場合を表 3 に示す。表 3 では正答終了を「○」、誤答終了を「×」としている。

問題は 3 題あり、それぞれの問題を以下の表に示す。

(注5) : 経路編集方式 C においては順位付け条件の入力はない。

(注6) : 該当なしと解答したときとする。

表 3 正答終了と誤答終了

	正解経路の正誤	時間内に解答	時間切れ
正解経路の提示あり	正答	○	×
	誤答・未解答	×	×
正解経路の提示なし	正答 ^(注6)	○	×
	誤答・未解答	×	×

表 4 問題 1

乗車駅	新丸子 (しんまるこ)
下車駅	錦糸町 (きんしちょう)
条件	1. 渋谷駅まで東急東横線で到着する。(時間順) 2. 渋谷駅から乗換なしで錦糸町へ到着する。(乗換回数順)

表 5 問題 2

乗車駅	神戸空港 (こうべくうこう)
下車駅	伏見稲荷 (ふしみいなり)
条件	1. 梅田駅 (阪神) まで最短時間で到着する。(時間順) 2. 梅田駅 (阪神) から京橋駅を経由し、京阪本線で伏見稲荷へ到着する。(乗換回数順)

表 6 問題 3

乗車駅	掛川 (かけがわ)
下車駅	本山 (愛知) (もとやま)
条件	1. 金山駅 (愛知) まで JR 東海道本線のみで到着する。(時間順) 2. 金山駅 (愛知) から乗換なしで本山へ到着する。(乗換回数順)

3 題の問題のうち、表 6 における問題 3 は、 C では正解経路が提示されない問題となっている。

5.3 結果と考察

本実験の参加者は 10~40 代の男女計 18 名である。全 18 名から 3 種類のインタフェースを用い 3 題の問題を行った。本実験ではインタフェースの利用順序、また問題の順序により参加者の経路知識に差異が出ないようにするため、グレコラテン方格法を適用し実施した。

各インタフェースにおける問題毎の平均探索所要時間を表 7 および図 8 に示す。

表 7 平均探索所要時間

問題	問題 1	問題 2	問題 3
C :所要時間 [s]	30.0	46.2	N/A
P_1 :所要時間 [s]	35.1	59.4	45.9
P_2 :所要時間 [s]	30.6	46.8	38.3
所要時間比 (P_1/C)[%]	116.8	128.8	N/A
所要時間比 (P_2/C)[%]	101.7	101.3	N/A
所要時間比 (P_2/P_1)[%]	87.0	78.7	83.4

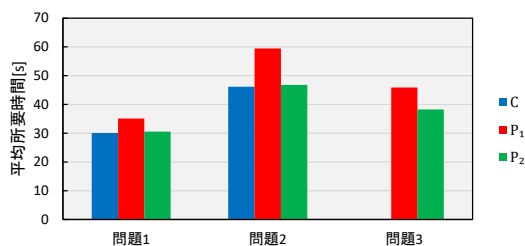


図 8 問題毎の平均探索所要時間

探索所要時間を見ると、 C と P_2 では概ね所要時間が同じとなり、 P_1 は他のインタフェースよりも所要時間が増加する結果となった。なお、今回の実験では誤答終了となる参加者はいずれの問題でも存在しない結果となった。

まず、従来法 C から提案手法のうち一部を実装した P_1 に対して比較を行うと、概ね 20%程度平均所要時間が増加している。平均では 20%程度の増加となったが、参加者によって差があり、所要時間が減少した参加者も存在した。この原因の一つとして、 P_1 では順位付け条件 O_e を入力したことで正解経路が上位に表示され、一番上の経路から正解経路を探していくと、下位の経路を見ることなく正解経路を発見してしまったことが挙げられる。その一方で、全ての経路に目を通す参加者も存在し、このような参加者は所要時間が約 2 倍に増えた。

次に、従来法 C から提案手法 P_2 に対してで比較を行うと、概ね平均所要時間が同じとなることが示された。これにより、木構造的な集約を行うことで、多数経路の比較を従来の少数の経路同士の比較とほぼ同等の迅速性を保てることが示された。

最後に、提案手法のうち一部を実装した P_1 から提案手法 P_2 を見ると、概ね 10~20%程度平均所要時間が短縮した。問題 2 において、条件 1 を満たす経路を探す際に、候補経路内に「阪急」「阪神」という似た名前の路線が存在し、条件を満たす経路を探す際に探索所要時間が増加する原因となっていた。条件 1 を満たす経路を発見し、条件 2 を満たす経路を発見するために分割操作を行うと P_1 では条件 1 で迷いがあった箇所も描画されるため、条件 1 を満たしているか再確認している参加者も存在した。一方で P_2 では、条件 1 で迷いがあった箇所を省略表示しているため、容易に正解経路を発見できたこともあり、他の問題と比べて大きく平均所要時間が減少した。また、問題 3 では問題 1 と比べ省略表示を行う区間が長いこともあり、問題 1 よりも平均所要時間が減少した。

6. おわりに

本稿では、知識条件という経路分割手法の意義を説明する利用者モデルを導入し、拡張した経路編集方式下での多数経路の比較容易性を向上させる省略表示を提案した。併せて、実験により既存手法に対する有効性を示した。さらなる迅速な所望経路発見手法の開発が今後の課題である。

文 献

- [1] 清水 祐弥, 小林 亜樹, “乗換案内における利用者知識を反映するインタラクティブな経路編集方式”, DEIM Forum 2017, D2-3, 2017.
- [2] K.Masaki, R.Otsuka, H.Fujikura and H.Shiizuka, “Change in User’s attitude by giving the unknown Information”, International Symposium for Emotion and Sensibility 2008, Proceeding, pp.130–134, 2008.
- [3] 寺部 慎太郎, 齊藤 あづさ, 郷原 翔一, “鉄道経路探索 Web サービスの利用者調査”, J-Rail 講演論文集 2009(16), pp.457458, 2009.
- [4] 寺部 慎太郎, 重里 光佑, 内山 久義, “鉄道経路探索 web サイトに関する利用実態の特徴分析”, 土木計画学研究・講演集, 37, No.341, 2008-6.
- [5] JR 東日本アプリリニューアルプロジェクト: JR 東日本, <http://www.jreast-app.jp/go/>, (2018-01-09 閲覧)