

# 居住候補地発見支援のための鉄道経路探索システムの提案

## Proposal of Railway Route Search System for Finding Living Place

菊池 圭祐  
Keisuke Kikuchi\*<sup>1</sup>

小林 賢一郎  
Kenichiro Kobayashi\*<sup>2</sup>

橋本 武彦  
Takehiko Hashimoto\*<sup>2</sup>

高間 康史  
Yasufumi Takama\*<sup>1</sup>

\*<sup>1</sup>首都大学東京大学院システムデザイン研究科  
Graduate School of System Design, Tokyo Metropolitan University

\*<sup>2</sup>株式会社 GA technologies  
GA Technologies

This paper proposes a route search system for supporting users to find their living place based on railway information. When we want to find a living place, the access to the place of work or school is usually considered. However, usual railway search systems are designed to search a route from departure station to destination at a specified time. Given the destination station and conditions such as commuting time and the number of connection, the proposed method finds multiple stations satisfying the conditions. This paper describes the search algorithm and explains the feedback about a prototype system from the salespersons in a real-estate company.

### 1. はじめに

本稿では、ユーザが居住地を探す作業の支援を目的として、目的地志向の路線探索手法を提案する。一般に、居住地を選択する際には、最寄り駅から勤務地や学校などの目的地までのアクセスの良さを考慮することが多い。しかし、現在、広く普及している路線検索システムは、出発地・目的地間の経路を求めることを目的としているため、居住候補地を探す際のような、指定した目的地に到達可能な候補地を複数探索する用途には適していないという問題点がある。

この問題点に対して、提案手法では、ユーザの目的地から、所要時間および乗換回数などの条件を満たす全ての駅および使用路線を検索可能とする。また、提案手法を用いた検索システムのプロトタイプを実装し、不動産会社の営業担当者とのヒアリングにより、路線検索システムの有効性を検証する。

### 2. 関連研究

一般的な路線探索は、基本的に出発駅から目的駅への最短経路問題として捉えることができる。そのため、探索はダイクストラ法 [Dijkstra 59] などのコストを考慮した探索アルゴリズムに基づいて行われる。ダイクストラ法とは、グラフネットワークにおいて、単一始点最短経路問題を解くための最良優先探索のアルゴリズムである。ダイクストラ法を用いた路線探索に関する研究として、運賃設定の異なる複数の鉄道会社を含む鉄道ネットワーク上での運賃計算アルゴリズムの報告 [池井 08] などがある。現状、一般に普及している乗換案内サービスで用いられている具体的な探索アルゴリズムやノウハウは、公開されていない部分が多いとされているが、研究では、所要時間、乗換回数、料金などの複数の実用的な制約条件から、優良経路を高速検索する手法などが提案されている [半田 05]。

### 3. 提案手法

#### 3.1 提案システム

路線探索は、東京都都内の 654 駅、80 路線のデータを対象として行う。提案する路線探索システムでは、ユーザの目的地の最寄り駅、移動時間上限および路線の乗換回数上限の設定を入力と

して、条件を満たして目的地に到着可能な全ての駅を居住候補地として求め、経路情報とともにユーザに提示する。また、提案手法は、乗換案内サービスのように特定日時での移動経路を調べられることを目的としたものではないため、出発時刻・到着時刻の時間指定は行わず、朝の通勤時における一般的な所要時間に基づき探索する。具体的には、朝の通勤通学時間を想定して、平日 AM8:00 から AM9:00 の間に運行している路線ダイヤの一般的な移動時間を設定する。

#### 3.2 路線ネットワーク

駅間の接続情報および路線情報をもとに、図 1 のような路線ネットワークを構築する。路線ネットワークでは、ノードを駅、エッジを移動可能な路線、エッジコストを駅間移動の所要時間とする。また、隣接駅ノード間で複数路線が運行する場合には、基本的に最短の移動時間となる路線とコストを設定する。また、同一路線区間のエッジには、路線の駅順に沿った向きの情報を付与する。これは、以下の探索アルゴリズムにおいて、不正乗車となる経路を判別することを目的として設定している。また、同一路線のみで、乗換なしで移動できる駅間には、その区間の所要時間の合計値をコストとして、直接エッジを接続する。図 1 に、路線ネットワークの例を示す。図 1 では、A 駅から C 駅までが同一路線で移動できる駅となっている。この場合、隣接する駅間のエッジに加えて、A 駅から C 駅間を所要時間の合計値である 7 分をコストとして、エッジを接続する。これによって、路線ネットワーク上での移動に関して、エッジ数と乗換回数が一致する。このネットワーク構造より、以降に説明する探索アルゴリズムにおいて、所要時間と乗換回数を識別することが可能になる。

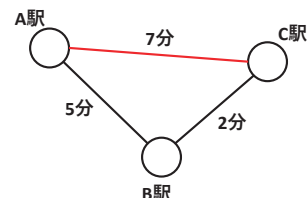


図 1: 路線ネットワーク例

#### 3.3 探索アルゴリズム

3.2 節の路線ネットワークを利用した探索アルゴリズムを説明する。探索は始点となる駅ノードを指定し、所要時間の上限エッジコストおよび乗換回数の上限回数を設定して、条件を満たす

連絡先: 高間康史, 首都大学東京大学院システムデザイン研究科,  
〒191-0065 東京都日野市旭が丘 6-6, ytakama@tmu.ac.jp

到達可能な駅ノードへの最短経路探索となる。最短経路探索はダイクストラ法を用いて行う。なお、ダイクストラ法による探索は Python のグラフ分析パッケージ networkX のライブラリを用いて、始点ノードから各ノードまでの最短経路および最短距離を求める。以下に、アルゴリズムの擬似コードを示す。PathsA, PathsB, ResultPaths は経路を要素とするリストであり、重複した要素は持たない。PathsA, PathsB は事前に計算しておく。また、アルゴリズム上での枝切りおよび再探索の処理内容の例を図 2 に示す。図 2 では、A 駅から D 駅間に B, C 駅を経由する各駅停車が運行しているほか、B, C 駅に停車しない快速があるような状況を想定している。この時、B 駅から A 駅に戻って快速を利用する経路は、A-B 間と A-D 間の向きが逆となるため候補外となり、再探索となる。A-D 間の経路を枝切りし、再探索された B-D 間の経路コストは 12 分から 14 分となる。

#### Algorithm 1 探索アルゴリズム

```

1: G ← グラフデータ読み込み
2: PathsA ← 始点から各ノードまでの所要時間最短経路
3: PathsB ← 始点から各ノードまでのエッジ数最短経路
4: ResultPaths = []
5: while PathsA do
6:   if エッジ数 ≤ 乗換回数上限 then
7:     ResultPaths に経路を追加
8:   end if
9: end while
10: while PathsB do
11:   if 経路の総コスト ≤ 所要時間上限 then
12:     ResultPaths に経路を追加
13:   end if
14: end while
15: while ResultPaths do
16:   if 経路に逆向き移動のエッジが含まれる then
17:     対象エッジをグラフ G から枝切り
18:     再探索 ⇒ 再探索結果の総コスト・総エッジ数を取得
19:     if (再探索の総コスト > 所要時間上限) ∨
20:       (再探索の総エッジ数 > 乗換回数上限) then
21:       経路を ResultPaths から削除
22:     end if
23:   end if
24: end while

```

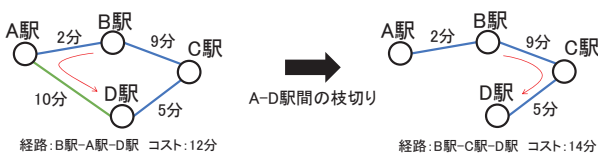


図 2: 枝切りと再検索の例 (B 駅 → D 駅の経路)

## 4. 実装結果

3 節で述べた路線探索手法を用いて、実装した検索システムのインタフェース画面を図 1、検索を行った例を表 2 に示す。検索例では、東京駅から乗換回数 2 回以下で 40 分以内の条件を入力とした。また、検索結果は一部を抜粋して表記している。

不動産企業の営業担当者とのヒアリングでは、路線探索の提案システムを実際に使用してもらい、実際の営業現場での観点から利用可能性を評価した。

営業担当者からは、現状の現場で使用している検索手法よりも提案システムの方が効率的に条件を満たす駅を探すことが可能だと評価された。また、営業担当者だけでなく、顧客側の利用を視野に入れた場合、居住候補地として選択可能な駅および地理的な範囲を、地図上で可視化できた方が直感的な理解につな

がるという意見を頂いた。また、提案システムでは、目的駅の設定は 1 駅となっているが、複数駅を入力として、条件を満たす駅を AND 検索できた方が、一人暮らしの個人ユーザだけではなく、夫婦などのユーザでも効率的な検索を行うことができる、という意見も頂いた。

### 沿線検索アプリケーション

目的駅: 八王子 所要時間: 40 分以内 乗換回数: 2 回以内 検索

**八王子駅から40分以内・乗換回数2回以内の駅**

ルート1  
西八王子駅  
所要時間:3分 乗り換え回数:0回  
JR中央線(快速): 西八王子  
ルート詳細

ルート2  
高尾駅  
所要時間:7分 乗り換え回数:0回  
JR中央線(快速): 西八王子, 高尾  
ルート詳細

ルート3  
豊田駅  
所要時間:5分 乗り換え回数:0回  
JR中央線(快速): 豊田

図 3: 提案システムのインタフェース

表 1: 提案手法での検索例

到達駅	所要時間	乗換回数	使用路線
有楽町駅	2分	0回	JR 山手線
神田駅	2分	0回	JR 中央線
⋮	⋮	⋮	⋮
末広町駅	10分	1回	JR 中央線, 東京メトロ銀座線
⋮	⋮	⋮	⋮
泉岳寺駅	20分	1回	JR 京浜東北線, 都営浅草線
⋮	⋮	⋮	⋮
沼袋駅	40分	2回	東京メトロ丸ノ内線, 東京メトロ東西線, 西武新宿線

## 5. おわりに

本稿では、ユーザが居住地を探す作業の支援を目的として、目的地志向の路線探索手法、統計データに基づく類似地域探索手法を提案した。路線探索手法について検証を行った結果、不動産企業の営業現場にて、顧客の設定条件に基づき居住候補地となるを探す際に有効であることを確認した。居住候補地の選択には、その街の印象や家賃相場、住みやすさなどの情報も重要であるため、今後は地域の印象や特性も考慮した検索機能を検討する予定である。

## 参考文献

- [Dijkstra 59] E.W. Dijkstra, "A note on two problems in connection with graphs," Numerische Mathematik, Vol. 1, No. 1, pp. 269-271, 1959.
- [池井 08] 池上敦子, 森田隼史, 山口拓真, 菊地丞, 中山利宏, 大倉元宏, 鉄道運賃計算のための最安運賃経路探索: 複数の鉄道会社を含む場合, 日本オペレーション・リサーチ学会論文誌, Vol. 51, pp. 1-24, 2008.
- [半田 05] 半田恵一, 田中俊明, 乗換え案内サービスにおける経路探索手法, 電子情報通信学会論文誌, Vol. J88-D-I, No. 10, pp. 1525-1533, 2005.