Transactions of the Operations Research Society of Japan
Vol. 54, 2011, pp. 1–22

鉄道運賃計算アルゴリズム
—Suica/PASMO 利用可能範囲の JR 東日本 510 駅の運賃を対象とした場合—

森田 隼史
成蹊大学
(現在: 日本信号)

池上 敦子
成蹊大学

菊地 丞
日本信号

山口 拓真
日本信号

中山 利宏
日本信号

大倉 元宏
成蹊大学

(受理 2009 年 7 月 2 日; 再受理 2010 年 10 月 25 日)

和文概要　鉄道運賃は，基本的に乗車距離が長くなればなるほど高くなるように設定されているが，同じ距離でも，会社によって，さらには同じ会社内でも地域や路線によって異なる料金が設定されている．さらに，乗車区間によっては割引ルールや特定の運賃が設定されていることなどから，最短経路の運賃が最安になるわけではない．運賃計算では，利用者の乗車経路が明確でない場合，乗車可能経路の中から最も安い運賃となる経路を利用したとみなし，その運賃を採用するルールが設定されている．そのため，与えられた 2 駅間の正しい運賃を計算するためには，その 2 駅間の乗車可能経路の運賃を全て，もしくはその 1 部を列挙して判断する必要があると考えられてきた．これに対し，我々は 2008 年，複数の鉄道会社を含む鉄道ネットワークにおける最安運賃経路探索用ネットワーク Farenet と探索アルゴリズムを提案し，これを利用した自動改札機用運賃計算エンジンの実用にいたった．本論文では，Farenet 構築の基盤となった 1 会社内の運賃計算，具体的には，首都圏エリアで利用可能である IC カード乗車券 Suica/PASMO の適用範囲に含まれる JR 東日本 510 駅の全 2 駅間（129,795 組）に対して行った運賃計算について報告する．4 つの対キロ運賃表と複数の運賃計算ルールが存在するこの運賃計算において，異なる地域・路線を考慮した部分ネットワークとダイクストラ法を利用することにより，多くの経路を列挙する従来の運賃計算方法において数時間要していた計算を，約 1 秒で処理することに成功した．論文の最後では，アルゴリズムの効率を示すとともに，対象ネットワークが持つ運賃計算上の特徴についても報告する．

キーワード: 交通，鉄道ネットワーク，運賃計算，最短経路問題

1. はじめに

鉄道サービスを運営する鉄道会社では，利用者の通った経路に対し，正確に運賃を計算しなければならないという強いニーズを持つ．そのため，各駅に自動改札機などを設け，利用者の通った経路を判断して運賃を計算することを行っている．しかし，自動改札機は，利用者の乗降車駅を判断することはできるものの，その 2 駅間に多くの経路がある場合には，実際の乗車経路までを特定することはほとんどできない．そこで，このようなケースに対応するため,「2 駅間に多数存在する到着可能な経路の中で，運賃が最も安い経路（以下，最安経路）を利用したとみなす」というルール [5] を設定し，利用者にとって損のないような運賃を計算するようにしている．つまり，自動改札機などの機器を生産する立場からすると，各 2 駅間における最安運賃を計算できなければならない．現在，駅に設置されている自動改札機では，あらかじめ求めた各駅間の最安運賃の情報を保持することにより，正確な運賃を計算できるようになっている．

運賃は，基本的に乗車距離が短いほど安くなるように設定されている．しかし，鉄道会社によって運賃体系が異なることや多くの運賃計算上のルールが存在することにより，最短経路が最安経路になるとは限らない．そのため，最安運賃を計算するためには，多数の乗車可能経路を列挙し，その中から最安経路を選択しなければならないといわれてきた．

野末 [11] は，与えられた 2 駅間の運賃の計算方法として，複数の経路を列挙し，各経路の運賃を計算して最も安い運賃を選択することを提案している．我々が運賃計算システムを扱う現場において使用してきた従来の運賃計算方法についても同様に，経路を出来る限りたくさん列挙する方法を用いてきた．この経路の列挙方法については，企業秘密のため具体的な内容は公表していないが，長年の経験をもとに数多くの経路を列挙している．そして，それぞれの経路に対して運賃計算上のルールなどを考慮しながら運賃を算出し，その中で最も安い運賃を選択することを行っていた．この方法を用いれば，乗車可能な経路を十分な数だけ列挙することによって，正しい最安運賃を求めることができていた．しかし，この方法には問題があり，運賃計算システムを扱う現場を悩ませていた．それは処理時間が非常にかかることである．首都圏の鉄道網においては，ほとんどの 2 駅間において乗車可能な経路が数多く存在するため*，この経路列挙に処理時間が非常にかかっていた．また，列挙した複数の経路に対し，運賃計算上のルールを考慮した運賃を算出する場合においても，列挙した経路と関係のあるルールかどうかを事前に判断することができない場合が多いため，結果的に多くのルールと照らし合わせる処理が発生してしまい，処理時間がかかっていた．

現在，首都圏における鉄道網は，多くの路線が敷かれ，多くの駅で他の路線と乗り換えることができるようになっている．また，異なる鉄道会社の路線間の相互乗り入れも増え，非常に複雑化している．2007 年には，IC カード PASMO[8] が導入され，Suica[6] との共通化が行われたことにより，今後鉄道会社間の相互乗り入れが増え，ますます鉄道網は複雑化していく可能性がある．その場合，現状の多数の経路を列挙する運賃計算方法では，上記で挙げた問題が開発面や保守面において，より大きな障害となる可能性が考えられる．そこで，我々は，現状の列挙に頼る運賃計算の方法とは視点を変え，列挙をほとんど必要とせず，少ない手間で最安運賃を見つけることができる運賃計算アルゴリズムの構築を目指し，研究を行った．運賃計算を考えた際，駅ノードと駅間アークで表現できる物理的ネットワーク上での処理が向いているものとそうでないものに分けられると我々は考え，1 会社内の運賃計算と複数社を含む鉄道ネットワーク上の運賃計算に対して異なるアプローチ方法を考えた．そして，後者に関しては，Suica/PASMO が利用可能な範囲（以下，関東 IC 範囲）を対象に，物理的な構造に基づくネットワークを利用する代わりに，運賃計算用の意味的ネットワーク（以下，Farenet）を構築し，最短経路問題を解けば最安運賃を求めることができる仕組みを作ることにより，処理時間の大幅な削減に成功した [7]．

しかし，Farenet を構築するためには，各鉄道会社について，同じ会社内に属する 2 駅間についての最安運賃が正しく設定されて（すでに求められて）いることが前提となる．1 会社内についての運賃計算を考える場合，他社との関係を意識する必要がないため，基本的には最短経路が最安経路になると考えられる．しかし，鉄道会社によっては，利用する区間や路線によって料金体系が異なったり，複雑な運賃計算上のルールが存在することにより，最短経路が最安経路にならないことが起こる．そのため，運賃計算システムを扱う現場では，経路を数多く列挙する方法から逃れることができず，1 会社内の運賃計算においても経路の列挙に多くの処理時間を費やしていた．本論文では，その中でも最も駅数が多く，かつ考慮すべき運賃計算上のルールも多く存在した東日本旅客鉄道（以下，JR 東日本）の例を対象に，これまで経路列挙の方法を取らざるを得なかった理由を明確にするとともに，JR 東日本に属する全駅間の最安運賃を高速に計算することができるアルゴリズムを提案する．ま

*宇都宮駅と伊東駅の間には，首都圏に存在する JR 東日本内の路線だけの利用に絞っても，約 374 万通りの経路が存在している．

た，提案アルゴリズムの効率とともに，適用した際に明らかになった運賃設定の特徴についても報告する．

2. 鉄道運賃計算

各鉄道会社では，運賃計算のために，表 1 に示すような「鉄道を利用した距離」に対して運賃を定めた対キロ運賃表（鉄道を利用した距離を x，それに対する片道運賃を $f(x)$ とする）を持っている．そのため，乗車経路が明確な場合には簡単に運賃を計算することができる．ちなみに,「鉄道を利用した距離」に端数が含まれる場合には，その端数は切り上げて考える．つまり，利用距離が 3.5km の場合には 4km として運賃を計算することになる．この対キロ運賃表は，単調増加の性質を持つことから，基本的に運賃は乗車した距離が短いほど安くなる．そのため，最短経路が最安経路になると考えることができる．最短経路を求める方法としては，代表的な方法として，Dijkstra 法 [1] が知られている．

表 1: 対キロ運賃表

距離（km） x	片道運賃 (円) $f(x)$
1 〜 3	130
4 〜 6	150
7 〜 10	160
11 〜 15	190
16 〜 20	250
21 〜 25	330

しかし，この対キロ運賃表が複数存在すると，最短経路が最安経路になるとは限らなくなる．その状況をわかりやすく説明するために（対キロ運賃表が異なる複数社を含むネットワークを利用して），図 1 にその例を示す．

図 1 では，吉祥寺駅と新宿駅周辺の路線図と，2 駅間の最安経路と最短経路を示している．最短経路は，荻窪駅で JR 東日本の路線から東京メトロに乗り換えて新宿駅に行く経路であり，最安経路は，京王電鉄の路線だけを利用して行く経路である†．他社の路線に乗り換えて新宿駅に向かう方が短い距離で辿り着くことができるが，乗り換えることにより使用する対キロ運賃表が変わり，JR 東日本と東京メトロそれぞれの運賃を足し合わせることになるため，運賃は高くなる．また，図 1 に示した対キロ運賃表の違いの他にも，複数の鉄道会社の路線を短い距離で乗り継いだ場合に適用される乗継割引などが存在することから，最安経路を求めることは容易ではない．そのため，我々の従来の運賃計算では，まず多数の経路を列挙し，各経路に対して運賃を計算してから一番安いものを選択する方法を採用していた．つまり，最安経路になる可能性のある経路を数多く列挙することで，求めた運賃が最安である保証を持たせようとしていたことになる．

これに対し，我々が 2008 年に提案した運賃計算方法 [7] では，各社の各駅間をつなぐアークのコストに運賃を設定したネットワーク Farenet を構築し，Dijkstra 法などの動的計画法を利用できる基盤を作った．そして，分割購入経路（途中で下車し，また乗車することで，

†多くの人が利用すると思われる JR 東日本のみの経路は，12.2km で 210 円である．

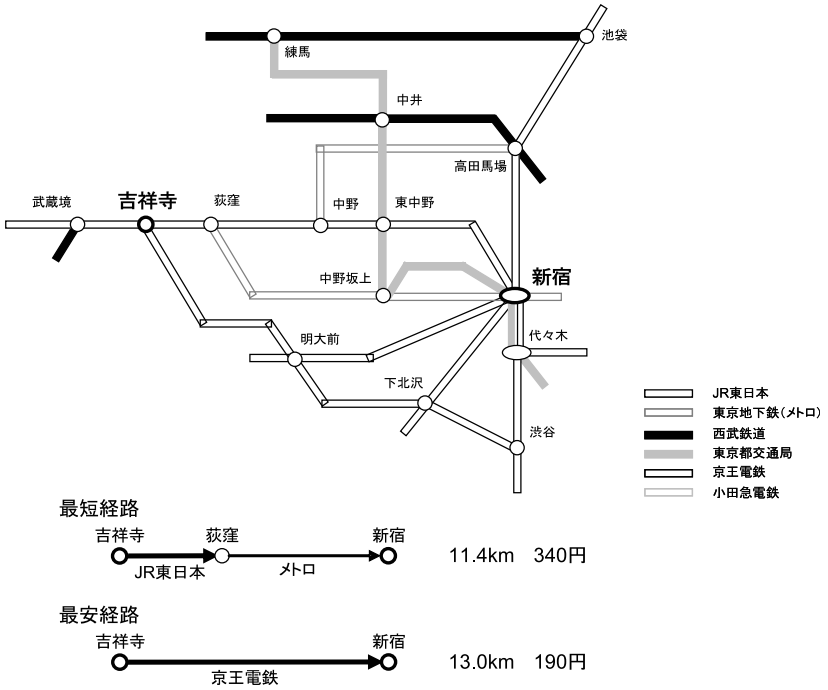

図 1: 最短経路≠最安経路

安く目的地まで行くことができる経路）を排除できる Directnet や，乗継割引アークを加えた DDnet を構築して利用することにより，利用会社に制約のない（利用会社数に上限がない，同じ会社の複数利用を許す）最安経路探索問題に対するアルゴリズムを提案し，正確かつ高速な運賃計算を可能にした‡．これに基づく運賃計算エンジンは，自動改札機の運賃計算のために実用化されている．

我々が構築した運賃計算のためのネットワーク（Farenet，Directnet，DDnet）を構成するアークのコストは，1 会社内の全駅間の運賃に対応する．つまり，これらの正確な運賃が得られていることが大前提であり，各会社内の運賃計算が正確かつ高速に行われない限り意味を持たない．1 会社内を対象とした運賃計算を考えた場合，他社との乗換を考えなくてよいので，鉄道会社ごとの料金体系の違いや乗継割引などの存在を考慮する必要がなくなる．そのため，基本的には最短経路と最安経路が一致する．しかしながら，本研究で対象とした関東 IC 範囲上に存在する鉄道会社の中には，会社内で複数の対キロ運賃表を持ち，その対キロ運賃表を区間や路線の種類によって使い分けて計算したり，運賃計算の特例§を考慮して計算するという運賃計算上のルールの関係から，「最短経路の運賃が最安である保証」がない会社が存在している．表 2 に，関東 IC 範囲上に存在する鉄道会社 27 社の駅数，利用している対キロ運賃表の数，そして，定められている特例のあり・なしを（それぞれ○と—で）示す¶．関東 IC 範囲においては，対キロ運賃表を複数利用している，あるいは特例が定められている会社が 10 社も存在する．そのため，この 10 社の運賃計算では，最安である保証が最短経路の運賃を求めるだけでは得られなくなる．本論文では，その中でも，我々の従来の運賃計算方法では 10 数時間要していた JR 東日本の運賃計算を対象にする．関東 IC 範

‡利用会社に制約のある場合の運賃計算については，別報にて報告しているので参照されたい [12].
§運賃計算の特例とは，通常の運賃計算のルールを適用すると利用実態にそぐわない場合などに対処するため，設けられたルール.
¶2007 年 3 月 18 日までの情報である.

囲上の JR 東日本では，対キロ運賃表を 4 種類持ち，駅数が最多の 510 駅存在する上，日本の鉄道の中心とも言える東京駅も有していることから，東京駅に関わる特別な運賃計算ルールも存在する．本論文では，全 2 駅間の組合せ（129,795 組）に対して，最安であるという保証のある経路を早期に見つけることにより，経路列挙数を最小限に抑えるアルゴリズムを提案する．

表 2: 関東 IC 範囲 27 社の駅数，対キロ運賃表の数，運賃計算特例

会社名	駅数	対キロ運賃表の数	運賃計算の特例
JR 東日本	510	4	○
江ノ島電鉄	15	1	—
箱根登山鉄道	11	1	—
相模鉄道	25	1	○
横浜市交通局	32	1	—
東武鉄道	204	1	—
京成電鉄	64	2	○
新京成電鉄	24	1	○
西武鉄道	92	1	—
京王電鉄	69	1	○
東京急行電鉄	87	2	○
京浜急行電鉄	72	1	○
小田急電鉄	70	1	—
東京メトロ	139	1	○
東京都交通局	99	1	○
東京モノレール	10	1	—
北総鉄道	15	1	—
伊豆箱根鉄道	12	1	—
埼玉新都市交通	13	1	—
横浜新都市交通	14	1	—
ゆりかもめ	16	1	—
多摩モノレール	19	1	○
東京臨海高速鉄道	8	1	—
東葉高速鉄道	9	1	—
埼玉高速鉄道	8	1	—
横浜高速鉄道	6	1	—
首都圏新都心鉄道	20	1	—

3. JR 東日本の運賃計算ルール

本研究は，JR 東日本における運賃計算の方法について，時刻表に載せられている情報 [4] を利用して行った．この節では，この情報を説明する．図 2 に，本研究で対象とした関東 IC 範囲における JR 東日本 510 駅について，おおよその位置を示す．

3.1. 運賃体系の異なる 3 つの区間と 2 種類の路線

関東 IC 範囲において，JR 東日本だけを対象にした場合のネットワークは，JR 時刻表などで定義されている東京近郊区間と同じになる．そして，この東京近郊区間内には，東京近郊区間用に定義された運賃体系とは異なる山手線内区間，電車特定区間と呼ばれる区間が存在し，地理的には，次ページに示すような包含関係がある．

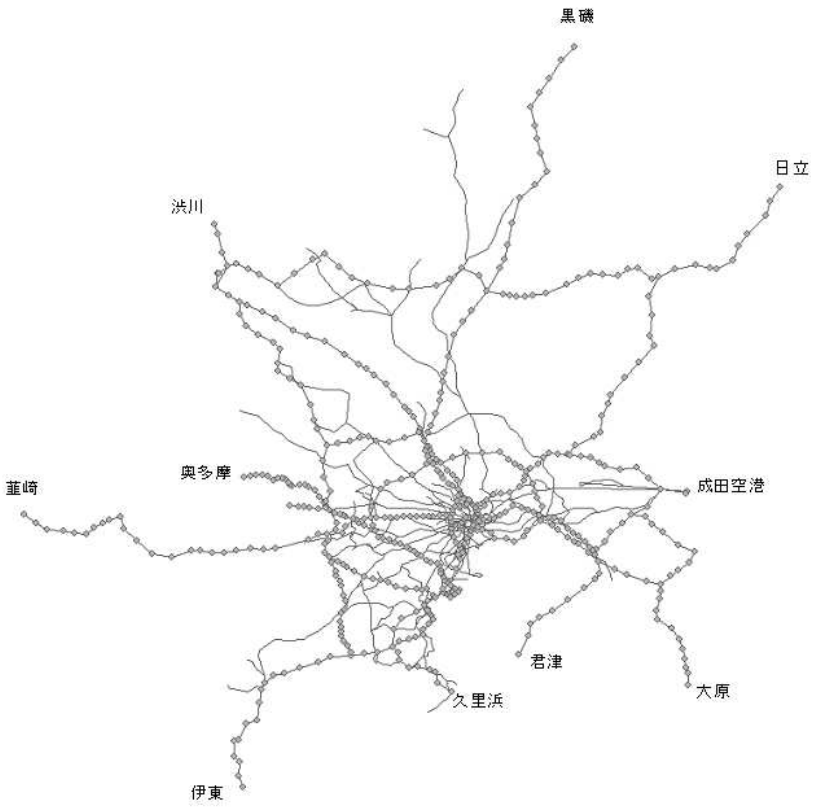

図 2: 関東 IC 範囲における JR 東日本 510 駅

山手線内区間（**36** 駅）⊂ 電車特定区間（**272** 駅）⊂ 東京近郊区間（**510** 駅）

また，JR 東日本の中にある路線は，幹線と呼ばれるものと地方交通線（以下，地交線）と呼ばれるものの 2 種類に分けられ，地交線を利用した際の運賃の方が幹線を利用した際の運賃より高く設定されている．図 3 は，鉄道ネットワーク上における 3 つの区間と路線の関係を示したものであり，表 3 には，各区間に各路線がどのように存在するかをまとめたものを示す．

表 3: 各区間における路線状況

	山手線内区間	電車特定区間	東京近郊区間
幹線	○	○	○
地交線	—	—	○

ここで，電車特定区間（山手線内区間を含む）の路線は全て幹線である．つまり，幹線は 3 つの区間全てに存在しているが，地交線は東京近郊区間内の電車特定区間の外側において，2 路線 (八高線と東金線) が存在するだけとなっている．

3.2. 4 種類の対キロ運賃表

JR 東日本では，表 3 に示した 4 つの○印の範囲に対して，それぞれ異なる対キロ運賃表が用意されている（付録を参照）．表 4 に 4 つの対キロ運賃表の適用対象範囲，名称，距離 x に関する運賃関数を示す．

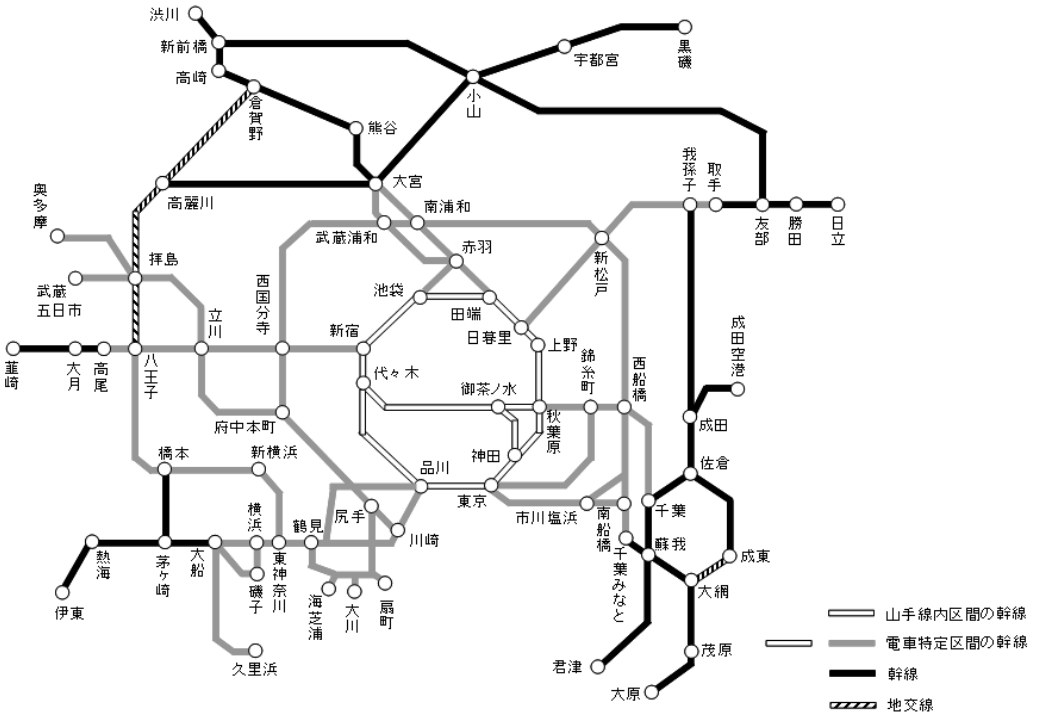

図 3: 東京近郊区間：電車特定区間（山手線区間を含む）と電車特定区間外の幹線と地交線

表 4: 4 つの対キロ運賃表の適用対象範囲と名称および関数名

	山手線内区間	電車特定区間	東京近郊区間
幹線	山手線内区間用：$f_{\text{山手}}(x)$	電車特定区間用：$f_{\text{特定}}(x)$	幹線用：$f_{\text{幹線}}(x)$
地交線	—	—	地交線用：$f_{\text{地交}}(x)$

この 4 つの対キロ運賃表には，任意の距離 x に対して，以下の関係がある．

$$f_{\text{山手}}(x) \leq f_{\text{特定}}(x) \leq f_{\text{幹線}}(x) \leq f_{\text{地交}}(x) \tag{3.1}$$

利用者が幹線のみを利用した場合，山手線内区間だけの利用であれば山手線内区間用の対キロ運賃表，電車特定区間だけの利用であれば電車特定区間用の対キロ運賃表，そして，電車特定区間外の利用を含めば幹線用の対キロ運賃表を使って運賃を求めることになる．また，地交線だけを利用した場合には，地交線用の対キロ運賃表を使って運賃を求める．しかし，幹線と地交線を混合利用した場合には，次で示すように，その経路がどのような構成になっているかにより，適用する対キロ運賃表が異なったり，乗車距離を換算することが行われる．

3.3. 幹線と地交線を混合して利用する経路の運賃計算

幹線と地交線を混合して利用する経路の運賃計算では，次のことを考慮する必要がある．表 5 に，説明に利用する「利用キロ数」の定義を示す．

幹線と地交線を混合して利用し，その混合経路の営業キロが 10km 以内の場合には，そのまま営業キロを地交線用の対キロ運賃表に適用して運賃を求める．また，幹線と地交線を混合して利用し，その混合経路の営業キロが 10km を超えた場合には，運賃計算キロを使い，

表 5: 運賃計算のための利用キロ数の定義

営業キロ	運賃計算のために，駅間に設定されているキロ数
換算キロ	営業キロの擬似キロ数．一般に，営業キロより長く設定される
運賃計算キロ	混合経路に対し，地交線部分を換算キロに置き換え，幹線部分の営業キロと足し合わせたキロ数．幹線と地交線を連続利用した場合の運賃計算に用いる

幹線用の対キロ運賃表に適用して運賃を求めるように決められている．このルールをまとめたものを表 6 に示す．

表 6: 幹線と地交線を混合利用した際の運賃計算ルール

混合利用した距離	使用する対キロ運賃表	使用する距離
10km 以下	地交線用対キロ運賃表	営業キロ
11km 以上	幹線用対キロ運賃表	運賃計算キロ

3.4. 運賃計算の特例

運賃計算を行う上で，考慮しなければならない特例ルールが存在するので以下に示す．

特定運賃

特定の運賃（特定運賃）があらかじめ設定されている 2 駅の組合せが存在している．つまり，その 2 駅を発駅，着駅として利用した場合，その 2 駅間の距離に関係なく，あらかじめ決められた特定運賃が採用されることになる．例えば，東京駅と西船橋駅間がその 1 つの例である．東京駅から西船橋駅までの運賃は，その最安経路の距離を対キロ運賃表に適用して求めれば 380 円になるが，この特定運賃が適用されることから 290 円と設定されている．このように，あらかじめ設定されている 2 駅の組合せは合計 110 組存在する．

山手線内区間の駅を発着する場合の特例

この特例は,「山手線内区間に属する各駅」と「東京駅からの営業キロが 101km から 200km である駅」に関しての運賃を計算する場合，その 2 駅間の距離に関係なく東京駅からの最安運賃に振り替えるというものである．例えば，山手線内区間に属する新宿駅から韮崎駅までの運賃を求める場合，韮崎駅は東京駅からの距離が 142.2km であるため，この特例ルールが適用される．つまり，新宿駅から韮崎駅の運賃はこの 2 駅間の距離 136.7km から考えれば，2,210 円になるのだが，東京駅からの運賃 2,520 円に振り替えられることになる．この特例を考慮するためには，東京駅からその他の駅への最短距離と最安運賃をあらかじめ求め，知っておく必要がある．

新幹線と在来線が並行する区間の特例

この特例は，新幹線と在来線が並行する区間の特例であり，在来線を利用していたとしても，新幹線経路を利用したと考えてもよいこととなっている（逆も言える）．つまり，新幹線と在来線が並行している区間において，新幹線経路の方が運賃が安い場合にはそちらの経路を採用する必要がある．

正確な運賃を計算するためには，上記に挙げた 3 つの特例ルールを考慮して計算する必

要がある．しかし，3つ目に示した特例ルールについては，関東IC範囲においてどの区間が「新幹線と在来線が並行する区間」なのか時刻表では明文化されていなかったため，今回はこの特例ルールを考慮することはせずに，運賃計算を行うアルゴリズムを考えることにした．

4. 運賃計算を複雑にする要因

前節で挙げた運賃計算ルールにより，最短経路が最安経路になるとは限らない例を示す．図4は，電車特定区間に属する拝島駅，横浜駅の2駅間の経路を意識して作成したものであるが，この2駅間には（ネットワーク上，少し離れた場所は省略してあるものの），多くの種類の経路が存在していることがわかる．地交線と幹線を混合利用する経路が複数あり，その中には電車特定区間や山手線内区間まで利用した経路もある．また，幹線だけを利用した経路も複数存在し，その中には電車特定区間だけを利用した経路や山手線内区間まで利用した経路もある．

これらの数多く存在する経路のうち，どの経路の運賃が安くなるのかは，4つの対キロ運賃表の関係式 (3.1) より，それぞれの経路の運賃を計算して比較してみないとわからない．もちろん「最短経路の運賃が最安になる」という保証はない．

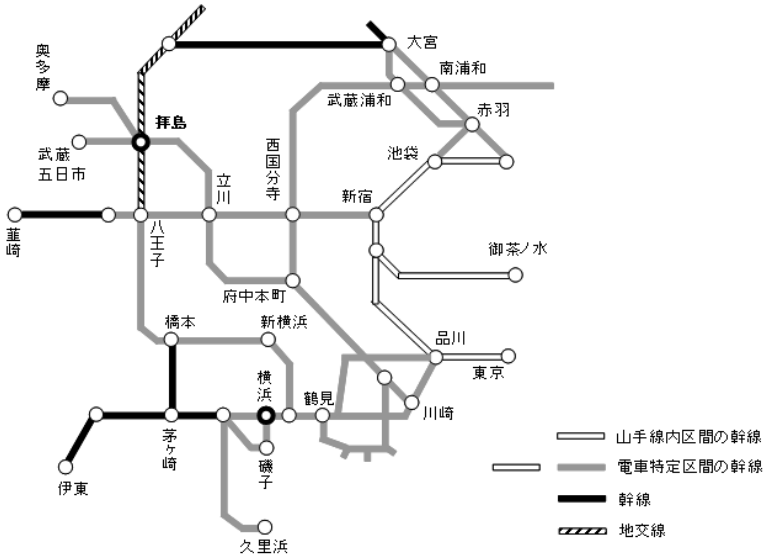

図 4: 拝島，横浜間に存在する経路

運賃計算の難しさは，対象の2駅間に存在する「運賃計算の方法が異なる数多くの経路」の中から，「最安運賃の保証がある経路」を求めることにあり，どの領域に存在する経路までを対象にすればよいか等の判断を間違えれば，正しい運賃を算出できない可能性も起きてくる．当然，2駅間に存在する経路を全列挙し，各経路の運賃を比較すれば，最安保証を得ることができる．実際，従来我々が使用してきた運賃計算方法は，必ず最安運賃である保証を持たなければならないため，乗車可能経路を数多く列挙することを行っていた．しかし，この経路列挙にとても時間を費やしていたことは前にも述べた．

経路列挙を速く行う方法としては，k-shortest paths 問題を少ない計算量で解くことができるアルゴリズム [2, 3, 9, 10, 13] が知られている．しかし，これらのアルゴリズムは，アークコストの和で表される総距離を評価値とし，列挙しなければならない経路（距離が短い

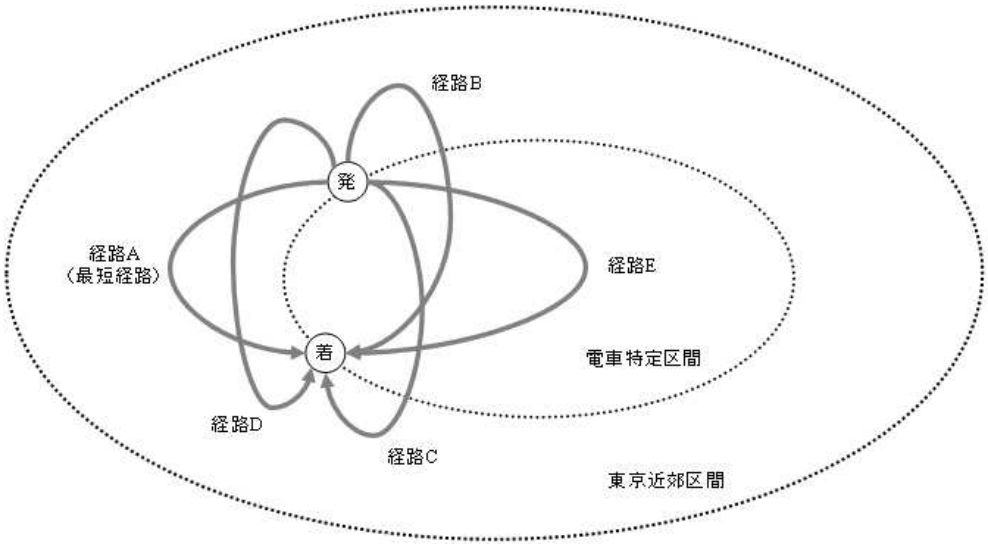

図 5: 最安経路になりうる経路

順）の数がわかっているときなどに有効なものである．そもそも我々の問題では，経路を（例えば距離の短い順に）何本列挙すれば最安運賃となる経路を含む保証が得られるのか明らかでないため，これらのアルゴリズムを直接利用することは適切ではない（列挙速度が速くとも，従来の方法と同様に経験的に列挙の終了条件を決めることになり，最適である保証を得ることができない）．

そこで，我々は，経路列挙を速く行う方法に頼らず，最適保証のある解を非常に少ない手間で見つけることを可能にする運賃計算アルゴリズムを考えた．

5. 運賃計算アルゴリズム

運賃計算ルールから生じる「最短経路の運賃が最安になる保証がない」という問題点を解決するにあたり，我々は経路の列挙による方法とは視点を変え，出来る限り列挙に頼らない (出来れば経路を 1 本求めるだけ) で計算することを目指した．

5.1. 部分ネットワークの設定

各区間や路線の種類が入り交じった複雑なネットワークを，図 5 のような簡単な形に表現して，運賃計算を考えてみる．図 5 に示した経路は，発駅と着駅の両方が電車特定区間内に存在する例を対象に，その 2 駅間に存在する数多くの経路のうち,「最安経路になりうる経路」を示したものである．経路 A と経路 B は，幹線と地交線を混合利用する経路であり，幹線部分と地交線部分を利用する距離が異なるものである．経路 C は，幹線だけを利用する経路であり，経路 D は地交線だけを利用する経路である．そして，経路 E は電車特定区間内の幹線を利用する経路である．説明の便宜上，この 2 駅間の最短経路が経路 A，電車特定区間の最短経路が経路 E であったとする．表 7 に，図 5 に示す 5 つの経路が利用する路線の種類とその距離を示す．

経路 E は，電車特定区間用の対キロ運賃表を使って運賃を求める経路であることから，対キロ運賃表の関係式 (3.1) により，電車特定区間外を利用する各経路より安くなる可能性がある．経路 A と経路 B は，幹線と地交線を混合利用し，かつ利用距離が 10km を超す経路である．そのため，地交線部分を換算キロに置き換えた距離と，幹線部分の営業キロとを足

表 7: 各経路の情報

経路	幹線部分	地交線部分	合計 (営業キロ)	合計 (運賃計算キロ)
A	1.3km	26.3km	27.6km	30.3km
B	25km	4km	29km	29.4km
C	29.1km	0km	29.1km	-
D	0km	28km	28km	-
E	33km	0km	33km	-

し合わせた運賃計算キロを使い，運賃を計算することになる．ここで示す例では，運賃計算キロで考えると，経路 B の方が経路 A より短くなる．このことから，地交線部分の利用距離により，経路 B も最安経路になる可能性が出てくる．また，経路 C のように，地交線部分を利用しない幹線だけを利用する経路についても，運賃を計算するための距離に営業キロを用いるため，経路 A と経路 B の運賃を計算するために使う運賃計算キロより短くなる可能性が出てくる．このことから，経路 C も最安経路になる可能性がある．地交線部分だけを利用する経路 D も同様に，運賃を利用する距離に営業キロを用いるため，利用距離によって最安経路になる可能性が出てくる．ここでは 5 つの経路を例に挙げたが，同じタイプの経路が同じ距離のものを含めて数多く存在する可能性があり，東京近郊区間全体のネットワークにおいて距離の短い順に経路を列挙したとしても何本目までに含まれるという保証を持たない．

我々は，上記で示す「最安経路になりうる経路」を効率よく算出することを行いたい．上記の例における 2 駅間に存在する全経路は，経路 A や B のような「幹線と地交線を混合利用する経路」，経路 C のような電車特定区間外の「幹線だけを利用する経路」，経路 D のような「地交線だけを利用する経路」，経路 E のような「電車特定区間だけを利用する経路」のどれかに必ず分類される．運賃を計算するために使用する各種の対キロ運賃表は，距離が短いほど運賃は安くなることがわかっているため，混合経路の場合にはその利用距離によって場合分けが必要になるが，同じタイプの経路の中では必ず距離が最短の経路が最安経路になる．そこで，表 8 に示すような区間毎のネットワークを定義した．そして, この区間毎のネットワークを利用することにより，対象区間を狭めながら最短経路問題を解くことで, 経路の列挙をほとんど必要とせずに,「最安経路になりうる経路」を算出することが可能になると考えた．

表 8: 区間毎の各ネットワーク

部分ネットワーク	ノード数	アーク数	アークのコスト
東京近郊区間のネットワーク	510	1,076	営業キロ
東京近郊区間のネットワーク	510	1,076	運賃計算キロ
電車特定区間のネットワーク	272	580	営業キロ
山手線内区間のネットワーク	36	76	営業キロ
幹線のネットワーク	488	1,024	営業キロ
地交線のネットワーク	28	52	営業キロ

ちなみに，図 5 の例では，東京近郊区間のネットワーク（コストが営業キロ）上で最短経

路問題を解くことにより，経路Aが求まる．また，東京近郊区間のネットワーク（コストが運賃計算キロ）上で最短経路問題を解くことにより，経路Bと経路Cのコストが小さい方が求まる（コストが大きい方の経路は最安経路にならない）．経路Dは地交線のネットワーク上で，経路Eは電車特定区間のネットワーク上で最短経路問題を解けば，それぞれ求まる．

5.2. アルゴリズム

5.1節で提案した部分ネットワークを利用する，正確かつ高速なアルゴリズムを構築した．我々が構築したアルゴリズムは2つのフェーズに分かれる．1つ目のフェーズでは，特例ルールを意識せず各2駅間の運賃を求めることを行う．そして2つ目のフェーズでは，存在する3つの特例ルールのうち，我々が対象にした2つのルールに対応する処理を行う．以下に示すアルゴリズムの考え方は，1つ目のフェーズで行う処理の意味を簡潔に示したものである．

アルゴリズム（フェーズ1）の考え方

各発着の組合せに対して，以下の処理を行う．
if (発着2駅が山手線内区間に属する場合)
{
　山手線内区間の最安経路，電車特定区間の最安経路，東京近郊区間の最安経路
　それぞれを比較して一番安い運賃を採用．
}
else if (発着2駅が電車特定区間に属する場合)
{
　電車特定区間の最安経路，東京近郊区間の最安経路を比較して安い運賃を採用．
}
else
{
　東京近郊区間における最安経路の運賃を採用．
}

与えられた2駅間に対してその2駅が属する区間で場合分けをし，各場合分けの中では，「山手線内区間の最安経路」,「電車特定区間の最安経路」,「東京近郊区間の最安経路」を表8に示した各ネットワークを利用して効率よく求めることを行えばよい．山手線内区間の最安経路は，地交線が存在しないため，山手線内区間上の最短経路が最安経路になる．また，電車特定区間の最安経路は，山手線内区間と同様に地交線が存在しないため，電車特定区間上の最短経路が最安経路になる．しかし，東京近郊区間には，地交線が存在することから混合利用経路に対する場合分けが必要になる．これらを踏まえて，自明ではない「東京近郊区間の最安経路」をも効率よく求める計算方法を次ページに示す．この東京近郊区間の最安経路を求める計算では，求まる経路に応じて幹線用と地交線用の対キロ運賃表を使い分けて運賃を算出する必要がある．現在, この2つの対キロ運賃表の設定では，$f_{\text{地交}}(x) \leq 200$円 ($x \leq 10$の場合)，$f_{\text{幹線}}(y) \geq 230$円 ($y > 10$の場合)(付録の対キロ運賃表を参照）であることがわかっているため，本アルゴリズムでは以下の関係を利用することとする．

$$f_{\text{地交}}(x) < f_{\text{幹線}}(y) \quad (x \leq 10 \text{かつ} y > 10 \text{の場合}) \tag{5.1}$$

最安運賃計算方法

■ 記号説明

- $N_{\text{近郊-営業キロ}}$：東京近郊区間のネットワーク（コストが営業キロ）
- $N_{\text{近郊-運賃計算キロ}}$：東京近郊区間のネットワーク（コストが運賃計算キロ）
- $N_{\text{特定}}$：電車特定区間のネットワーク
- $N_{\text{山手}}$：山手線内区間用のネットワーク
- $N_{\text{幹線}}$：幹線ネットワーク
- $N_{\text{地交}}$：地交線ネットワーク
- $f_{\text{幹線}}, f_{\text{特定}}, f_{\text{山手}}, f_{\text{地交}}$：距離 x に対する運賃を与える関数（4 つの対キロ運賃表に対応）.
- $\overline{path}$：$path$ の距離.

■ アルゴリズム（フェーズ 1）

各発着の組合せに対して，以下の処理を行う.
$N_{\text{近郊-営業キロ}}$ 上の最短経路 **path0** を求める.
if（**path0** が山手線内区間だけを通る）
{
    path0 が最安経路になるため，処理終了.
}
else if（**path0** が電車特定区間だけを通る）
{
    if（2 駅ともに山手線内区間に属する）
    {
        $N_{\text{山手}}$ 上の最短経路 **path1** を求める.
        $f_{\text{特定}}(\overline{\textbf{path0}})$ と $f_{\text{山手}}(\overline{\textbf{path1}})$ を比較して安い方が最安経路になるため処理終了.
    }
    else
    {
        path0 が最安経路になるため，処理終了.
    }
}
else
{
    if（**path0** が 10km 以下の混合利用経路もしくは 10km 以下の地交線のみの経路）
    {
        $N_{\text{幹線}}$ 上の最短経路 **path1** を求める.
        $f_{\text{地交}}(\overline{\textbf{path0}})$ と $f_{\text{幹線}}(\overline{\textbf{path1}})$ の安い方の経路を **path0** とする.
    }
    else if（**path0** が 10km を超す地交線だけの経路）
    {

28 $N_{\text{近郊-運賃計算キロ}}$ 上の最短経路 **path1** を求める．
29 $f_{\text{地交}}(\overline{\textbf{path0}})$ と $f_{\text{幹線}}(\overline{\textbf{path1}})$ の安い方の経路を **path0** とする．
30 }
31 **else if**（**path0** が 10km を超す混合利用経路）
32 {
33 $N_{\text{近郊-運賃計算キロ}}$ 上の最短経路 **path0** と，$N_{\text{地交}}$ 上の最短経路 **path1** を求める．
34 $f_{\text{幹線}}(\overline{\textbf{path0}})$ と $f_{\text{地交}}(\overline{\textbf{path1}})$ の安い方の経路を **path0** とする．
35 }
36
37 **if**（発着 2 駅ともに山手線内区間に属する）
38 {
39 $N_{\text{特定}}$ 上の最安経路 **path1**，$N_{\text{山手}}$ 上の最安経路 **path2** を求め，
40 **path0**，**path1**，**path2** の中で一番安い経路を最安経路として，処理終了．
41 }
42 **else if**（発着 2 駅ともに電車特定区間に属する）
43 {
44 $N_{\text{特定}}$ 上の最安経路 **path1** を求め，
45 **path0**，**path1** の中で安い方の経路を最安経路として，処理終了．
46 }
47 **else**
48 {
49 **path0** を最安経路として，処理終了．
50 }
51 }

上記に示すアルゴリズムでは，対キロ運賃表に関する 2 つの関係式 (3.1),(5.1) を利用して最安経路を算出しているが，冗長な計算をできる限り省く工夫を行っている．例えば，アルゴリズムの考え方で示した「発着 2 駅が山手線内区間に属する場合」において，最初に東京近郊区間のネットワーク上で最安経路を求めたものが，山手線内区間のみを利用した経路であれば，それは山手線内区間の最安経路であり，電車特定区間の経路でもあることから，これら 2 つの最安経路を求めることなく，最安経路として採用することができる．

このアルゴリズムでは，まず $N_{\text{近郊-営業キロ}}$ 上で最短経路問題を解く (この結果はフェーズ 2 でも利用するため，保持しておく必要がある)．そして，求まった経路がどの区間を通り，どの種類の路線を乗車しているかによって場合分けをし，最安運賃の保証が得られる場合には，最安経路が求まったとして処理を終了する．一方，保証が得られない場合には，対象区間をより安い運賃体系の区間に狭めて最短経路問題を解き直すことを行い，求めた経路同士の運賃を比較することを行って最安経路を求める．つまり，最安経路になりうる可能性をカバーできるように場合分けをし，場合分けされた各ケースにおいて最適解を得る仕組みを作ることで，最安経路である保証を持つ計算方法となっている．

アルゴリズムの 2 行目で求めた path0 の距離に対し，地交線用の対キロ運賃表を使って運賃を求める場合には，関係式 (3.1) から，path0 より少し距離が長い「幹線用の対キロ運賃

表」を使って運賃を求める経路の方が安くなる可能性がある．そのため，アルゴリズムの23行目と28行目において,「幹線用の対キロ運賃表」を使って運賃を求める経路の中で最も距離が短い経路としてpath1を求め，運賃を比較するようにしている．なお，現在の地交線とその運賃設定では，営業キロが10kmを超す混合経路の運賃が，10km以下の混合経路もしくは10km以下の地交線のみの経路の運賃よりも安くなることがないため(関係式(5.1))，この23行目では，path0が「10km以下の混合経路もしくは10km以下の地交線のみの経路」の場合において，運賃を比較するpath1を$N_{\text{幹線}}$上で求めている(関係式(5.1)が存在しない場合には，$N_{\text{近郊-運賃計算キロ}}$上において, 10kmを超える経路の中で最も短いものを求め, その運賃とpath0, そして$N_{\text{幹線}}$上で求めた最短経路path1の運賃と比較する必要がある).

また，アルゴリズムの2行目で求めたpath0が，10kmを超す幹線と地交線の混合利用経路だった場合には，地交線部分を営業キロではなく換算キロに置き換えたネットワーク（$N_{\text{近郊-運賃計算キロ}}$）上で，再度最短経路を求めることを行う．その結果，換算キロに置き換える前と違う経路が最短経路として求まる可能性があるが，どちらの経路にしろ「幹線用の対キロ運賃表」を使って運賃を求めることになる．この場合，2行目でpath0を求めたネットワーク（$N_{\text{近郊-営業キロ}}$）上に，そのpath0よりも少し距離が長い「地交線だけで構成される経路」が存在した場合には，その「地交線だけで構成される経路」の運賃は営業キロを用いて計算されるため，$N_{\text{近郊-運賃計算キロ}}$上で再度求めた最短経路の運賃よりも安くなる可能性がある．そのため，33,34行目において，地交線のネットワーク上で最短経路を求め，地交線用の対キロ運賃表を使って求めた運賃と比較するようにしている．

アルゴリズムのフェーズ2では，我々が考慮した2つの特例ルールに対応する処理を行う．まず全組合せの中から，運賃計算の特例の1つである「山手線内区間に属する各駅」と「東京駅からの営業キロが101kmから200kmである駅」の組合せを見つける．そして，見つけた組合せに対し，フェーズ1であらかじめ求めてある東京駅からの運賃に振り替えることを行う．次に，もう1つの特例である「特定運賃が設定されている2駅間の組合せ」も見つけ，設定されている運賃に振り替えることを行う．

5.3. アルゴリズムを利用した計算実験

ここでは，我々が提案したアルゴリズムを利用して行った計算実験結果を示す．アルゴリズムの3行目から18行目までは，$N_{\text{近郊-営業キロ}}$上で求めた最短経路が,「山手線内区間の最安経路」になる，または「電車特定区間の最安経路」になることが簡単な処理でわかり，対象とする2駅間の「保証のある最安経路」が求まる．ちなみに，129,795組のうち，35,855組（全体の約28%）がこの部分で「保証のある最安経路」を得ている．残りの93,940組については，2行目で求めたpath0を構成する経路の種類が複数存在してくるため，19行目のelseに対応するブロックの処理を行うことになる．表9に，21〜35行目の各if文により分類される経路の数を示す．

表9に示した,「幹線だけの経路」以外の3つには，それぞれで比較対象となるpath1を求め，path0の運賃と比較する処理がある．しかし，現状のJR東日本におけるネットワークでは，path0の運賃よりpath1の運賃の方が安くなる組合せは，1つもなかった．そのため，現状の最安運賃計算には，上記に示す3つのpath1を求めて運賃を比較する処理は省いても，正確な最安経路を計算することが可能である．

改札機データとの比較

提案したアルゴリズムを利用し，JR東日本510駅全ての2駅間の組合せ（129,795組）に

表 9: 21〜35 行目の各 if 文により分類される経路の数

経路の種類	2 駅間の組合せ数
10km 以下の混合利用経路または地交線	102
10km を超す地交線	214
10km を超す混合利用経路	17,698
幹線だけの経路	75,926

ついて運賃を求め，実際に改札機のデータとして使用されている正確な運賃と比較した結果を表 10 に示す．表 10 では，14 の組合せだけ改札機データと一致しなかったことがわかる．運賃計算システムを扱っている現場に確認したところ，この 14 組は我々が考慮対象としていなかった「新幹線と在来線が並行する区間の特例」が適用される 2 駅間であることが確認できた．

表 10: 現行改札機データとの比較

	2 駅間の組合せ数
一致	129,781
不一致	14

表 11 は，最短経路が最安経路にならなかった組合せ数について示したものである．表 11 から，最短経路が最安経路にならなかった組合せ数はわずか 1,998 組（全体の 1.5%）だけであり，残りの 127,797 組は最短経路が最安経路になることがわかる．最短経路が最安経路にならない組合せを計算する前から判断できれば，この 1,998 組に対してだけ経路を数多く列挙すればいいわけだが，計算前にこれらの駅間を特定することはできない．つまり，従来の計算方法（我々が利用してきた方法であり，一般的に提案されてきた方法）では，全体の運賃計算のたった 1.5%のために膨大な数の経路を列挙して，最安経路である保証を得ようとしたわけである．

表 11: 最短経路と最安経路の関係

	2 駅間の組合せ数
最短経路 = 最安経路	127,797
最短経路 ≠ 最安経路	1,998

表 12 は，我々が提案したアルゴリズムにおいて，経路を何本列挙すれば，最安経路である保証を得ることができるかについてまとめたものである．表 12 では，多くの組合せが 1 本経路を求めるだけで，その経路が最安経路である保証を得ることを示している．また，複数の経路を求めなければ最安経路である保証を得ることができない組合せについても，非常に少ない数の経路列挙で最安経路である保証を得ることができることを示している．そのため，提案したアルゴリズムを使用し，JR 東日本 510 駅全ての 2 駅間の組合せ（129,795 組）について運賃を求めることを行った場合，1 秒未満（CPU:Xeon 3.6GHz，メモリ：2G）で処理ができる．現在，運賃計算システムを扱う現場において，我々が構築したアルゴリズム

表 12: 最安経路の保証を得るために列挙する経路数

列挙する経路数	2 駅間の組合せ数
1	111,644
2	452
3	16,836
4	863

をベースに,「新幹線と在来線が並行する区間の特例」に対応したバージョンも作成され，全駅間について正確な運賃を算出できるようになっている．また，処理時間についても，従来の経路列挙による方法では 10 時間以上かかっていたのが，約 1 秒で処理できるようになっている．

山手線内区間の駅を発着する場合の特例により，運賃が振り替えられる組合せ

JR 東日本 510 駅の全組合せ（129,795 組）のうち，存在する 3 つの特例ルールのうちの 1 つである「山手線内区間の駅を発着する場合の特例」が適用される組合せ数を調べた結果，2,590 組であることがわかった．この特例ルールは,「山手線内区間に属する各駅」と「東京駅からの営業キロが 101km から 200km である駅」に関しての運賃を計算する場合，東京駅からの最安運賃に振り替えるというものである．この特例ルールが適用される 2,590 組の運賃が，東京駅からの最安運賃に振り替えられることにより，もとの運賃に対してどのように変化するのかをまとめたものが表 13 である．表 13 から，東京駅からの最安運賃に振り替えられることにより，2,590 組のうち 2,368 組は振り替えられる前の運賃より安くなるまたは同じになるが，122 組は運賃が高くなってしまうことがわかる．

表 13: 東京駅からの最安運賃に振り替えられたことによる変化

	2 駅間の組合せ数
安くなる	1,161
高くなる	122
同じ	1,307

6. おわりに

本論文では，鉄道運賃において，基盤となる 1 会社内の運賃計算の位置づけと，運賃計算を複雑にする要因について報告し，これを効率よく解決するアルゴリズムを提案した．具体的にはまず，JR 東日本の運賃計算を対象に,「最短経路の運賃が最安になる保証」がないことを示した．そして，その状況下において，適した部分ネットワーク設定の下で最短経路問題を解くことにより，非常に短い時間で最適解（最安運賃とその経路）を導き出すことができるアルゴリズムを提案した．また，そのアルゴリズムを用いて，運賃計算実験を行い，アルゴリズムの効率のよさと対象ネットワークが持つ運賃計算上の特徴についても報告した．提案したアルゴリズムは，非常に高速に最安運賃を計算できるため，我々の運賃計算システムの現場で抱えていた「処理時間がかかるという問題点」を解決でき，今後の運賃計算システムの開発コストや保守コストを大幅に削減できる可能性を示している．

本論文で報告した研究成果（各駅間の最安運賃）は，我々が複数社を含む鉄道ネットワーク上において構築したネットワーク（Farenet，Directnet，DDnet）のアークのコストになる．そして，我々はこれらのネットワークを利用することにより，「利用会社数に関する制約を考慮しない最安経路」を求めるアルゴリズムを提案し，正確かつ高速な運賃計算を可能にしている [7]．しかし，関東 IC 範囲における現行の運賃計算ルールでは，利用する鉄道会社数に関する制約（4 社以内）や同じ会社の複数回利用（他社利用を挟んでの利用）を避ける制約が存在している．我々は，この制約付き最安経路探索問題についても研究を進めており，その成果報告も行っている [12]．

謝辞

本論文について貴重な御助言を頂いた査読委員の先生方ならびに論文誌編集委員の先生方に心より感謝致します．本研究の一部は，文部科学省私立大学戦略的研究基盤形成支援事業研究費による．

参考文献

[1] E.W. Dijkstra: A note on two problems in connexion with graphs. *Numerische Mathematik*, **1** (1959), 269–271.

[2] D. Eppstein: Finding the k shortest paths. *SIAM Journal on Computing*, **28** (1998), 652–673.

[3] 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 D, **J88-D1** (2005), 1525–1533.

[4] 東日本旅客鉄道株式会社: JR 時刻表 (2005).

[5] 東日本旅客鉄道株式会社: 首都圏 IC カード相互利用時の運賃の計算方法, http://www.jreast.co.jp/suica-co/fare/index.html.

[6] 東日本旅客鉄道株式会社: Suica, http://www.jreast.co.jp/suica/.

[7] 池上敦子, 森田隼史, 菊地丞, 山口拓真, 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ズ・リサーチ学会和文論文誌, **51** (2008), 1–24.

[8] 株式会社パスモ: PASMO ホームページ, http://www.pasmo.co.jp/.

[9] N. Katoh, T. Ibaraki, and H. Mine: An efficient algorithm for k shortest simple paths. *Networks*, **12** (1982), 411–427.

[10] E.Q.V. Martins, M.M.B. Pascoal, and J.L.E. Santos: The k shortest paths problem. *Research Report, CISUC* (1998).

[11] 野末尚次: 道具箱としての OR. オペレーションズ・リサーチ, **52** (2007), 135–140.

[12] 山口拓真, 森田隼史, 池上敦子, 菊地丞, 中山利宏: 利用鉄道会社数の制約を考慮した最安運賃経路探索. 電子情報通信学会論文誌 D, **J93-D4** (2010), 426–434.

[13] J.Y. Yen: Finding the k shortest loopless paths in a network. *Management Science*, **17** (1971), 712–716.

付録:対キロ運賃表

ここでは，運賃計算に使用した 4 種類の対キロ運賃表を示す．

表 14: 山手線内区間用の対キロ運賃表

距離 (km) x	片道運賃 (円) $f_{\text{山手}}(x)$
1 〜 3	130
4 〜 6	150
7 〜 10	160
11 〜 15	190
16 〜 20	250

表 15: 電車特定区間用の対キロ運賃表

距離 (km) x	片道運賃 (円) $f_{\text{特定}}(x)$	距離 (km) x	片道運賃 (円) $f_{\text{特定}}(x)$
1 〜 3	130	101 〜 120	1790
4 〜 6	150	121 〜 140	2100
7 〜 10	160	141 〜 160	2420
11 〜 15	210	161 〜 180	2730
16 〜 20	290	181 〜 200	3050
21 〜 25	380	201 〜 220	3360
26 〜 30	450	221 〜 240	3680
31 〜 35	540	241 〜 260	3990
36 〜 40	620	261 〜 280	4310
41 〜 45	690	281 〜 300	4620
46 〜 50	780	301 〜 320	4940
51 〜 60	890	321 〜 340	5250
61 〜 70	1050	341 〜 360	5460
71 〜 80	1210	361 〜 380	5670
81 〜 90	1380	381 〜 400	5990
91 〜 100	1530		

表 16: 幹線用の対キロ運賃表

距離 (km) x	片道運賃 (円) $f_{\text{幹線}}(x)$	距離 (km) x	片道運賃 (円) $f_{\text{幹線}}(x)$	距離 (km) x	片道運賃 (円) $f_{\text{幹線}}(x)$
1 〜 3	140	521 〜 540	8190	1921 〜 1960	19110
4 〜 6	180	541 〜 560	8510	1961 〜 2000	19320
7 〜 10	190	561 〜 580	8720	2001 〜 2040	19640
11 〜 15	230	581 〜 600	9030	2041 〜 2080	19950
16 〜 20	320	601 〜 640	9350	2081 〜 2120	20270
21 〜 25	400	641 〜 680	9560	2121 〜 2160	20580
26 〜 30	480	681 〜 720	9870	2161 〜 2200	20900
31 〜 35	570	721 〜 760	10190	2201 〜 2240	21110
36 〜 40	650	761 〜 800	10500	2241 〜 2280	21420
41 〜 45	740	801 〜 840	10820	2281 〜 2320	21740
46 〜 50	820	841 〜 880	11030	2321 〜 2360	22050
51 〜 60	950	881 〜 920	11340	2361 〜 2400	22370
61 〜 70	1110	921 〜 960	11660	2401 〜 2440	22580
71 〜 80	1280	961 〜 1000	11970	2441 〜 2480	22890
81 〜 90	1450	1001 〜 1040	12290	2481 〜 2520	23210
91 〜 100	1620	1041 〜 1080	12600	2521 〜 2560	23520
101 〜 120	1890	1081 〜 1120	12810	2561 〜 2600	23840
121 〜 140	2210	1121 〜 1160	13130	2601 〜 2640	24150
141 〜 160	2520	1161 〜 1200	13440	2641 〜 2680	24360
161 〜 180	2940	1201 〜 1240	13760	2681 〜 2720	24680
181 〜 200	3260	1241 〜 1280	14070	2721 〜 2760	24990
201 〜 220	3570	1281 〜 1320	14390	2761 〜 2800	25310
221 〜 240	3890	1321 〜 1360	14600	2801 〜 2840	25620
241 〜 260	4310	1361 〜 1400	14910	2841 〜 2880	25830
261 〜 280	4620	1401 〜 1440	15230	2881 〜 2920	26150
281 〜 300	4940	1441 〜 1480	15540	2921 〜 2960	26460
301 〜 320	5250	1481 〜 1520	15860	2961 〜 3000	26780
321 〜 340	5460	1521 〜 1560	16070	3001 〜 3040	27090
341 〜 360	5780	1561 〜 1600	16380	3041 〜 3080	27410
361 〜 380	6090	1601 〜 1640	16700	3081 〜 3120	27620
381 〜 400	6300	1641 〜 1680	17010	3121 〜 3160	27930
401 〜 420	6620	1681 〜 1720	17330	3161 〜 3200	28250
421 〜 440	6830	1721 〜 1760	17640	3201 〜 3240	28560
441 〜 460	7140	1761 〜 1800	17850	3241 〜 3280	28880
461 〜 480	7350	1801 〜 1840	18170	3281 〜 3320	29190
481 〜 500	7670	1841 〜 1880	18480	3321 〜 3360	29400
501 〜 520	7980	1881 〜 1920	18800	3361 〜 3400	29720

表 17: 地交線用の対キロ運賃表

距離 (km) x	片道運賃 (円) $f_{\text{地交}}(x)$	距離 (km) x	片道運賃 (円) $f_{\text{地交}}(x)$
1 〜 3	140	83 〜 91	1620
4 〜 6	180	92 〜 100	1800
7 〜 10	200	101 〜 110	1890
11 〜 15	230	111 〜 128	2210
16 〜 20	320	129 〜 146	2520
21 〜 23	400	147 〜 164	2940
24 〜 38	480	165 〜 182	3260
29 〜 32	570	183 〜 200	3570
33 〜 37	650	201 〜 219	3890
38 〜 41	740	220 〜 237	4310
42 〜 46	820	238 〜 255	4620
47 〜 55	950	256 〜 273	4940
56 〜 64	1110	274 〜 291	5250
65 〜 73	1280	292 〜 310	5460
74 〜 82	1450		

森田隼史
日本信号株式会社
〒 346-8524 埼玉県久喜市大字江面字大谷 1836-1
E-mail: s-morita@signal.co.jp

ABSTRACT

A FARE CALCULATION ALGORITHM FOR RAILWAY NETWORKS —A CASE STUDY INVOLVING 510 JR STATIONS WITHIN THE SUICA/PASMO SYSTEM—

Shunji Morita
Seikei University
(currently affiliated with *Nippon Signal*)

Atsuko Ikegami
Seikei University

Takuma Yamaguchi
Nippon Signal

Jo Kikuchi
Nippon Signal

Toshihiro Nakayama
Nippon Signal

Motohiro Ohkura
Seikei University

A railway fare for a specific path is usually determined by the distance, i.e. the longer the distance, the higher the fare. The fare rate, however, is sometimes different by area even within the same railway company. In addition, many of companies have additional rules to be used in the fare calculation, e.g. discounts for specific paths. In order to calculate the fare between a given pair of stations, we have to find the minimum cost path, but the shortest path (in the physical sense) is not always the minimum cost path for the reasons mentioned above. In the past, most of the efforts have been focused on an enumeration of feasible paths between a pair of stations and a comparison of their resulting fares in order to choose the minimum fare for the pair. This approach has inevitably led to a problem of a very large computational time. In this paper, we propose a much more efficient algorithm which basically uses the Dijkstra's algorithm with a modification which reflects the existence of a set of subnetworks which have their own fare rates. Our algorithm dramatically reduces the computational time, because the algorithm need not enumerate a large number of paths for every pair of stations.