

# 乗り換え案内サービスエンジン

Engine for Finding Train and Airplane Transfer Sequences

唐崎 幸弘  
KARASAKI Yukihiro

半田 恵一  
HANDA Keiichi

鈴木 孝弘  
SUZUKI Takahiro

乗り換え案内サービスは、当社が行っている“駅前探険倶楽部”の主要なサービスの一つである。このサービスは、当社が1997年5月にわが国で最初に実現した実際の時刻表や運行データに基づいたもので、外出する際に適した経路を実用的なレベルで案内することが可能である。現在のサービス範囲は全国を対象とし、約500路線、約9,000駅の鉄道のほか、航空路線をサポートし、時刻表改定の当日から最新の情報を提供している。エンジン部は最適な解をリアルタイムに提供するために独自方式を採用している。また、エンジン部をAPI(Application Programming Interface)により分離し、パソコン(PC)、携帯情報端末など多彩なユーザーインタフェースのサポートや、ASP(Application Service Provider)提供を行っている。検索結果はXML(eXtensible Markup Language)形式で出力し、機能拡張を考慮している。

The engine for finding train and airplane transfer sequences is the key component of Toshiba's route guidance service, which is one of the main services of the "Ekimae-Tanken Club", Internet-based navigation service for train passengers. The route guidance service, based on actual timetable and operation data, was realized by Toshiba for the first time in Japan in May 1997. The service provides practical guidance on the optimum course to be taken, covering the nationwide railway network consisting of about 500 lines and 9,000 stations as well as the main air routes in Japan. Timetable information is constantly updated and users can obtain the latest data from the day on which any change occurs.

To support this service, Toshiba has developed an original engine which is designed as a module that is independent from applications. We have defined a route guidance application program interface (API) to enable flexible services, such as support for various types of terminals (PCs, mobile phones, etc.) and supply of application service provider (ASP) service. The output of this engine is in the form of Extensible Markup Language (XML), allowing further extension.

## 1 まえがき

外出、出張、旅行などで鉄道などの交通機関を利用する場合、特に時間や行き先が不定形なとき、どのような経路で行けば効率がよいのか、どの時間に運行されているのかを効率よく検索したいというニーズが高まっている。従来、PCなどの検索ソフトウェアが存在していたが、路線や駅の新設/廃止、あるいはダイヤ改定情報が最新でなく、あくまで目安としてしか利用できなかった。

近年、インターネットの普及により、乗り換え案内が提供され普及してきている。当社は、1997年5月に全国初の実際の時刻データに基づいた乗り換え案内の提供を首都圏1,500の駅で開始した。その後、関西、東海、九州エリアへの拡大を経て、現在は全国約500路線、約9,000駅と航空路線をサポートした乗り換え案内サービスを提供し、Webで1日に15万件を超える利用がある。

当社の乗り換え案内は独自の方法により、経路探索、時間割当て、運賃計算処理を行い、出発時刻、到着時刻の指定、有料特急指定など、使いやすい機能を備えている。このほか、終電案内(目的駅まで到達できる終電を検索、案内)、時

刻表表示(路線、駅、方向、曜日指定による時刻検索)が可能で、PC、携帯情報端末などで行っている駅前探険倶楽部で利用することができるほか、ASPサービスへ提供を行っている。乗り換え案内の概念を図1に示す。

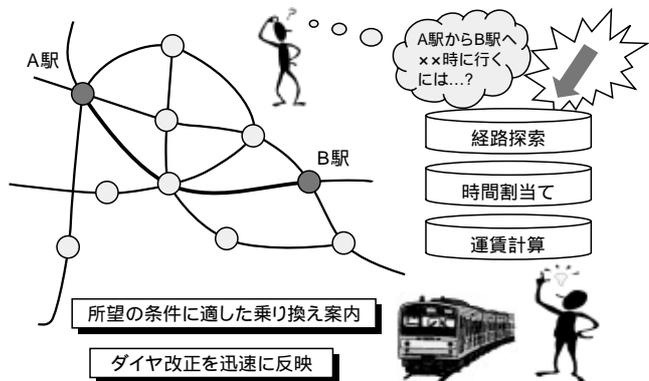


図1. 乗り換え案内 発駅から着駅まで条件に合った乗り換えを実際の時刻表データにより料金とともに提示する。

Finding train and airplane transfer sequences based on actual timetables

ここでは、乗り換え案内のエンジンの特長と処理の概要について述べる。

## 2 経路探索

経路探索では、路線図を拡張したネットワーク上でよい経路の候補を複数求める。経路の良さは所要時間や乗り換え回数などを基準にし、多様性のある経路群を探索する。

### 2.1 ネットワーク $N, N'$

全国の駅(空港も駅とみなす)をノードに持ち、駅間の路線又は徒歩乗り換えをアークに持つネットワークを記号  $N$  で表す。ただし、複数の路線が接続している駅は複数のノードで表現され、ノード間に乗り換えアークと呼ばれるアークが設けられている。異なる駅間での乗り換えは徒歩アークと呼ばれる。各アークには平均所要時間が重みとして与えられ、路線名などの情報がラベルとして与えられている。図2にネットワークの一部を示す。破線が徒歩アーク、太線が乗り換えアークを表す。

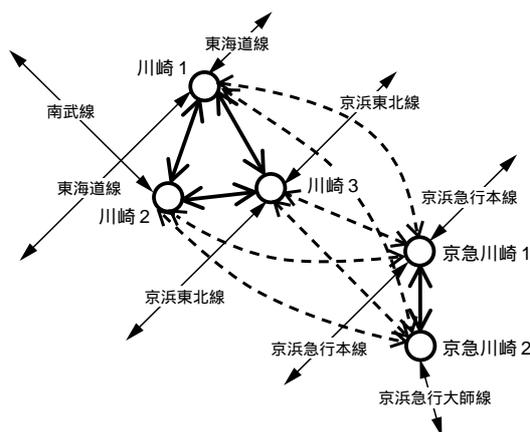


図2. ネットワーク  $N$  の一部 乗り換えが必要な駅は路線ごとに別ノードとし、乗り換えをアークで表現する。

Part of network  $N$

経路探索では基本的にこのネットワーク  $N$  を用いるが、処理の高速化のために、 $N$  から徒歩アーク又は乗り換えアークの接続していないノードをすべて除去したネットワーク  $N'$  も併用する。 $N'$  のアークには、快速などの高速列車を優先した平均所要時間を重みとして与えることにより、より現実に沿った所要時間を反映するようにしてある。徒歩アークや乗り換えアークの重みにはホームにおける待ち時間も考慮してある。

### 2.2 アルゴリズム概要

ネットワーク上で出発ノードから目的ノードに至る複数のパスを、距離(重みの和)の小さい順に列挙する問題は第

1~ $K$  最短パス問題<sup>(注1)</sup>として古くから研究され、様々なアルゴリズムが提案されている。われわれは文献(1)を参考に、ループやキセル乗車を含まない複数の経路を求めるプログラムを開発した(パス上の乗り換えアーク及び連続する同一路線名のアークを短縮したものを経路と呼ぶ)。また、飛行機、新幹線、特急、普通列車などを利用した経路がバランスよく得られるようにアークをグループ分けし、ネットワーク上で各グループのアーク群を取捨選択して探索するようにした。経路探索の処理概要は次のとおりである。

- (1) 乗、下車駅に対して、それらの近傍にある  $N'$  のノード(近傍ノードと呼ぶ)とそこに至る路線名をそれぞれ求める。
- (2) 乗、下車駅の近傍ノードの各ペア  $(u, v)$  に対して、 $u$  から  $v$  に至るパスを  $N'$  上で複数求める。
- (3) (2)の各パスを短縮して経路化し、その前後に乗車駅からの経路と下車駅への経路を接続する。

図3に例を示す。乗車駅は大森、下車駅は小作である。まず(1)より大森の近傍ノードとして大井町と蒲田が、小作の近傍ノードとして拝島が求まる。路線名はそれぞれ京浜東北線、青梅線である。次に、(2)より大井町から拝島に至るパスと、蒲田から拝島に至るパスがそれぞれ複数求まる。図3では蒲田からのパスは省略してある。また、大井町と拝島はそれぞれ一つのノードにまとめてある。最後に(3)より、例えば“大森;京浜東北線;品川;東海道線;東京;中央線;立川;青梅線;小作”のような経路が複数得られる。

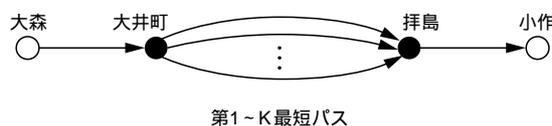


図3. パスの探索 発着駅近傍の単純なパスと乗り換えが発生するパスを組み合わせ探索を行う。

Search of paths

### 2.3 経路の質の向上

経路の質を高めるために種々の工夫をした。以下に、そのいくつかを述べる。

- (1) 特急料金が必要な短距離区間の置換え 上野~東京間に新幹線を利用したり、新大阪~大阪間に在来線特急を利用したりするのは冗長なため、そのような短距離区間は普通路線に置き換えるようにした。
- (2) 乗り換え回数の少ないパスを再探索 経路探索のアルゴリズムは、基本的に平均所要時間の短いパスから順に探索していく。乗り換えの少ないパスを求める

(注1) ネットワーク  $G$  において、出発ノード  $s$  から目的ノード  $t$  へのもっとも短いパス(第一パス)から  $K$  番目に短いパス(第  $K$  パス)を求める問題。

ために、乗り換え時間を適宜調整することにより、乗り換えの少ないパスも求めるようにした。

そのほか、高速化のためにネットワーク上の探索範囲の限定や、明らかに冗長な経路を削除するなどの工夫を行った。

### 3 時間割当て

時間割当てでは、経路探索で求められた経路の候補に対して、各路線の運行データに基づき列車の時刻を検索する。このとき、乗り継ぎや同一路線の列車種別に依存した乗換駅(停車パターン)の組合せについても考慮している。

#### 3.1 反復探索

経路探索で得られた経路候補に対して、出発指定時刻から下車駅に向けて一方向の時間割当てを行った例を表1に示す。この例では9:04に大江(愛知)駅へ到着するが、大江(愛知)~東名古屋港間は朝、夕の列車しかない。このため16:14発の列車まで430分待ちとなり、出発指定時刻に忠実な列車を案内すると全体の所要時間が8時間9分になる。

一方、表2は下車駅での到着時刻16:17を基に乗車駅に向けて逆方向に時間割当てを行った結果である(反復探索結果)。8:00発で検索した結果に対して小泉駅15:04発の列車を案内し、いずれも到着時間は16:17となる。

これは当社乗り換えエンジンで採用している「反復探索法」の特長であり、時間割当てを出発時刻に一番近い列車から始め、目的駅に達した時刻に基づいて、目的駅から逆方向にたどって出発列車を求める。すなわち「同じ時刻に着くのならできるだけ遅く出発する列車を案内する」ことが可能となり、ユーザーのホームでの待ち時間をできるだけ少なくして真冬や真夏にも快適な列車を利用して外出ができ

表1. 単純な時間割当ての例  
Example of simple time-assignment

検索条件 2001年08月10日 8:00発 小泉 東名古屋港間の経路 所要時間489分				
小泉	08:08	JR 太多線(普通)	多治見	08:13
多治見	08:18	JR 中央本線(快速)	金山(愛知)	08:49
金山(愛知)	08:52	名鉄名古屋本線(普通)	神宮前	08:54
神宮前	08:55	名鉄常滑線(普通)	大江(愛知)	09:04
大江(愛知)	16:14	名鉄築港線(普通)	東名古屋港	16:17

表2. 逆方向の時間割当て  
Improvement by reverse time-assignment

検索条件 2001年08月10日 8:00発 小泉 東名古屋港間の経路 所要時間73分				
小泉	15:04	JR 太多線(普通)	多治見	15:09
多治見	15:23	JR 中央本線(快速)	金山(愛知)	15:58
金山(愛知)	16:02	名鉄名古屋本線(普通)	神宮前	16:05
神宮前	16:05	名鉄常滑線(普通)	大江(愛知)	16:11
大江(愛知)	16:14	名鉄築港線(普通)	東名古屋港	16:17

るよう配慮している。

なお、表の“ ”は乗り入れ列車、すなわち複数路線をまたがって走行する列車を示すものである。

#### 3.2 巡回路線

図4に示すユーカリが丘線は手鏡のような巡回路線である。列車はユーカリが丘駅 公園駅~公園駅 ユーカリが丘駅という一方向だけ走行する。逆方向である中学校駅から女子大駅へ行く場合は、いったん公園駅まで乗車して乗り換える必要がある。このような巡回路線に対しても、表3のように実時間に基づいた列車を割り当てることができる。

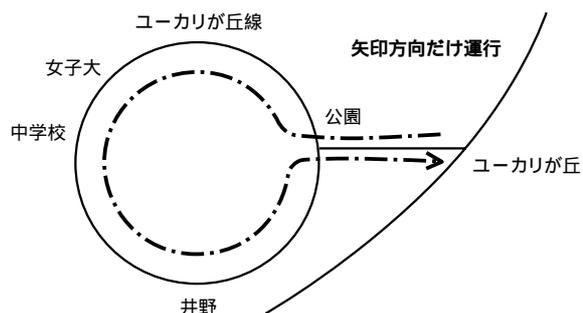


図4. 一方向の巡回経路の例 一方向だけ運行している路線では最短パスではなく、実際の運行データに基づいた経路の提示が必要となる。

Example of one-way cyclic line

表3. ユーカリが丘線の時間割当ての例  
Time-assignment for Yucarigaoka Line

検索条件 2001年08月10日 12:00発 中学校 女子大間の経路				
中学校	12:06	ユーカリが丘線	井野(千葉)	
井野(千葉)		ユーカリが丘線	公園	12:10
公園	12:23	ユーカリが丘線	女子大	12:24

## 4 運賃計算

運賃と料金の違いは、運賃が出発地から目的地までの移動に関する対価であり、一方特急や指定席などの付加価値に関するものは料金と呼ばれる。ここでは運賃計算について述べる。運賃は電鉄会社や路線ごとに異なり、乗り換え案内では多数の規則に基づいて運賃の計算を行っている<sup>(2)</sup>。

運賃規定の代表的なものが営業キロ換算であり、乗車した路線距離によって運賃が決まる。これ以外に、①均一区間、②営業キロに依存しない駅対駅、③特殊規則(特別加算区間、特別減算区間、特定区間運賃)、などがある。

通常は、上記の規則に従って営業キロ、均一区間、加算/減算区間の計算を行うが、それ以外に特例と言われる規則が存在する。

図5の場合は、「次の区間の左側の駅から枝分かれする一方の線区から他方の線区まで乗車する場合で、列車が左側の駅を通過するため、左側の駅と右側の駅との間を折り返し乗車する場合は、同区間のキロ数を含めないで運賃計算をします」という特例の具体例である。この特例に対応する区間上での「折り返し」は、その区間自体を通らなかったように扱い、本来は「志井公園～小倉～行橋」分のキロ数となるが、特例区間であるため「城野 小倉 城野」を除いたキロ数で運賃計算することになる。

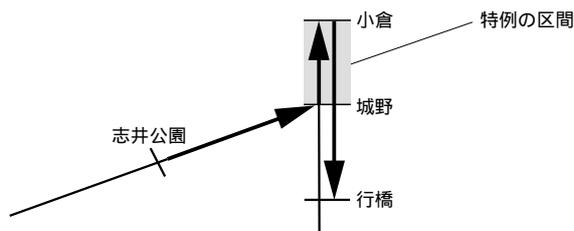


図5 . 分岐駅を通過する列車に乗車する場合の特例 特例区間の運賃は計算対象外となる。  
Fare exception rule

## 5 XML-API

乗り換え案内エンジンの経路探索結果は、アプリケーション部とのインタフェースをXML-API(図6)により実現している。

インタフェースのXMLフォーマット化は、以下を目的としている。

- (1) 機能拡張による新規情報追加の容易性 機能拡張による新規タグ追加の場合、既存アプリケーション側は不明(新規)タグは無視できる。
- (2) ASP利用者へ差異化情報提供 出力情報の追加が容易になるため、将来的にA社向けシステムでは時刻表へのリンク情報だけ、B社向けのシステムでは予約

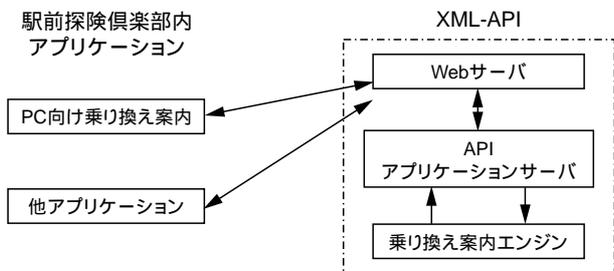


図6 . 乗り換えエンジンXML-API構成 XML-APIにより、ユーザーアプリケーションを分離し、拡張性・保守性を高めている。  
Outline of engine for finding transfer sequences for XML-API

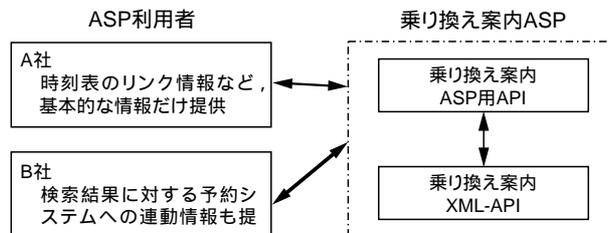


図7 . ASP利用者への差異化 XML-APIによりASP事業者への提供機能を選択的に行うことが可能である。  
Value-added application service

システムと連動するための情報を付加するなどサービスの差異化を図れる(図7)。

- (3) 各種端末向けサービス展開への容易性 経路探索結果情報に対する意味づけが可能となるため、アプリケーション側が各種端末の表示性能に即した画面作成が容易になる。
- (4) 乗り換え案内エンジン負荷分散の容易性 アクセス数に合わせたエンジンサーバ増設が可能となる。

## 6 あとがき

乗り換え案内サービスエンジンの概要について述べた。乗り換え案内は利用価値のあるデータに基づき、的確な検索結果をリアルタイムに提供することが生命線である。また、ユーザーニーズの高い検索機能の強化実現が必要である。今後、更に「使える」サービス実現を目指しエンジン性能向上とコンテンツ整備を行っていく。

## 文献

- (1) E.Q.V. Martins, et al. "An algorithm for ranking loopless paths," Research Report, Univ. de Coimbra, 1999.
- (2) 弘済出版 JR時刻表



唐崎 幸弘 KARASAKI Yukihiro  
iバリュークリエーション社 技術部主務。  
乗り換え案内の開発に従事。  
iValue Creation Co.



半田 恵一 HANDA Keiichi, D. Sc.  
研究開発センター システム技術ラボラトリー 研究主務、  
理博。グラフ応用、最適化手法の研究に従事。  
System Engineering Lab.



鈴木 孝弘 SUZUKI Takahiro  
東芝デジタルメディアエンジニアリング(株) ソフトウェアセン  
ター e&Sソリューション担当チームマネージャー。乗り換え  
案内の開発に従事。  
Toshiba Digital Media Engineering Corp.