

# 時刻表および平均移動時間を考慮した 列車・航空便の最適乗継系列探索システム

菊池 新 加藤 誠巳 高木啓三郎  
(上智大学理工学部) (NEC 情報メディア研)

近年、我国の公共交通網は一段と整備されて来ているが、それに伴い時刻表を考慮に入れて任意の出発地から任意の目的地に至る最適な経路を求めることは困難になりつつある。筆者らは従来からパーソナルコンピュータを用いて、新幹線特急と航空便の時刻表を考慮して最適な乗継系列を探索するシステムを作成したが、出発地、目的地は新幹線の駅または空港に限定されており、最寄りに新幹線の駅あるいは空港がないところが出発地または目的地である場合にはどこの新幹線の駅あるいは空港にアクセスするのが適当であるか分からない欠点があった。本論文では、日本全国の私鉄を含む 9,015 の鉄道駅ならびに空港を出発地あるいは目的地として指定したとき、航空機、新幹線特急、JR 在来線特急の時刻表を考慮に入れて、第  $k$  最適乗継系列 ( $k = 1, 2, \dots$ ) を探索する方法ならびに実際に最適乗継系列をパーソナルコンピュータ PC-9821 を用いて Windows 3.1 の下で探索した結果について述べている。

## An Optimal Transfer Sequence Scheduling System for Airline and Train Passengers with Timetables and Average Access Times taken into Account

Shin KIKUCHI\*, Masami KATO\*, Keizaburo TAKAGI\*\*

\*Faculty of Science and Technology, Sophia University  
7-1, Kioi-cho, Chiyoda-ku, Tokyo, 102 Japan

\*\*Information Technology Research Laboratories, NEC Corporation  
1-1, Miyazaki 4-chome, Miyamae-ku, Kawasaki, Kanagawa, 216 Japan

In recent year, the public transportation networks of Japan have made remarkable progress, so that one cannot easily find the optimal route from his origin to his destination. The authors already developed a system which provides the optimal transfer sequence of the Shinkansen limited expresses and the airplanes with the timetables taken into account. The defect of this system is that only the Shinkansen stations and the airports can be designated as an origin or a destination. As a result, a person whose nearest access point is the local station does not know where to go in order to take the limited expresses or the airplanes. In this paper, an optimal transfer sequence searching method and system is presented, in which any one of 9,015 stations and airports can be chosen as an origin or a destination. This system treats not only the airplanes and the Shinaknsen limited expresses but also the limited expresses of the conventional JR lines. The time require to obtain the optimal transfer sequences is quite reasonable, which reveals that the system running on a personal computer under Windows 3.1 is of practical use.

## 1 まえがき

近年、山形新幹線の開通、羽田空港の移転・拡張、新大阪国際空港の開港予定など鉄道、航空路等の公共交通網は一段と整備増強され、複雑化してきている。それに伴い、時刻表を引いて任意の出発地から任意の目的地に至る最適な乗継経路を手で見出すことは次第に困難になりつつある。筆者らは従来からこの問題を解決するためパーソナルコンピュータを用いて最適な乗継経路を探索するシステムの開発を行ってきた<sup>[1],[2]</sup>。即ち、新幹線特急と航空便の時刻表を考慮して最適な乗継系列を探索するシステムを作成したが<sup>[1]</sup>、出発地、目的地は新幹線の駅または空港に限定されており、最寄りに新幹線の駅あるいは空港がないところが出発地または目的地である場合にはどこの新幹線駅あるいは空港にアクセスするのが適当であるか分からない欠点があった。また、平均的な駅間所要時間をリンクのコストとするネットワークを形成し、乗換え所要時間を考慮に入れて出発駅と目的駅の間第 $k$ 番目までの最短時間経路を求め、求めた経路に対して時刻表のデータを当てはめて使用する列車を定めることも試みた<sup>[2]</sup>。このシステムでは任意の出発駅と目的駅を指定することが可能であるが、求めた乗継系列が最適な乗継系列である保証はない。

本論文では、日本全国の私鉄を含む9015の鉄道駅ならび空港を出発地、目的地として指定したとき、航空機、新幹線特急、JR 在来線特急の時刻表を考慮に入れて、ここで定義した第 $k$ 最適乗継系列( $k=1,2,\dots$ )を探索する方法、ならびにパーソナルコンピュータを用いて最適乗継系列を探索した結果について述べている<sup>[3]</sup>。

## 2 最適乗継系列の定義

ここで、第1最適乗継系列を以下のように定義する。

1. 出発地を出発希望時刻以降に出発し、目的地に最も早く到着する系列の中で出発地を最も遅く出発する系列。
2. 出発時刻と到着時刻が同じ系列が複数存在する場合は、乗継回数が最小のものを

最適乗継系列とする。

3. 乗継回数が同じ場合は、乗車時間の最小のものを最適乗継系列とする。

第 $k$ 最適乗継系列は第 $k-1$ 最適乗継系列の出発時刻以降に出発地を出発し、最も早く目的地へ至る系列の中で出発地を最も遅く出発する系列として定義する。

## 3 探索法の概念

第 $k$ 最適乗継系列を探索するアルゴリズムについて述べる前に探索法の基本概念について説明する。

まず、出発地および目的地が共に空港、新幹線特急の停車駅またはJR 在来線特急の停車駅(これを以下では乗継ノードと呼ぶ)であって、表1に示す列車、航空機が運行されている場合を例にとって考える。ここで、出発地、目的地の乗継ノード番号は各々①,④であるものとし、便宜上これらは空港と駅の両者の機能を兼ねたノードであるものとする。更に出発希望時刻は10:00以降とし、探索時間範囲は2時間、即ち目的地に12:00までに到着する系列を探索するものとする。各ノードには、そこに到着する”運行リンク”と”出発地出発時刻”の組(これを運行情報と呼ぶ)を要素とし、この要素の”出発地出発時刻”の遅い順にソートされたリスト(これを運行情報リストと呼ぶ)が逐次形成されるのであるが、出発ノードを除き初期状態は空集合 $\phi$ とする。出発ノードには常に出发可を与える集合 $\infty$ が付与される。表1に示す如く到着時刻順に予めソートされた時刻表を用い、到着時刻が10:00以降の列車または便を順に調べるのであるが、最初の便JAL3便は①を9:50に出発し、出発希望時刻を満たさないで棄却する。次に10:14に②に到着する便は①を10:00以降に出発するので運行リンク②を図1(a)のようにノード①とノード②の間に付加し、ノード②にリンク番号②と①の出発時刻10:02の組、即ち運行情報(②,10:02)が空集合 $\phi$ である運行情報リストに加えられる。次の運行リンク③はノード③から出発するものであるが、ノード③には出発地①より未到達(ノード③の運行情報リストが空集合 $\phi$ )であるので棄却する。運行リンク④は条件を満たすので

表 1 運行ダイヤの例

運行リンク番号	列車または便名	出発乗継ノード番号	到着乗継ノード番号	出発時刻	到着時刻
0	JAL1 便	①	②	9:43	9:55
1	JAL3 便	①	②	9:50	10:02
2	JAS101 便	①	②	10:02	10:14
3	ANA7 便	③	④	10:05	10:15
4	ひかり 3 号	①	③	10:01	10:22
5	JAL5 便	①	②	10:20	10:32
6	ひかり 3 号	③	④	10:23	10:40
7	JAL17 便	②	④	11:30	11:45
8	JAL19 便	②	④	11:35	11:50
9	こだま 101 号	①	②	11:45	12:05
...	...	...	...	...	...

ネットワークに付加され、ノード③の運行情報リストは  $[(4, 10:01)]$  となる。同様に運行リンク⑤も付加され、ノード②の運行情報リストの要素は運行情報の”出発地出発時刻”の遅い順にソートされるので  $[(5, 10:20), (2, 10:02)]$  となる。

次の運行リンク⑥は、ノード③発、ノード④行、ノード③の発時刻は 10:23 であるが、ノード③の運行情報リスト  $[(4, 10:01)]$  を調べると、運行リンク④はノード③に 10:22 に到着しており、かつ運行リンク⑥と同一列車であるので乗継時間は 0 で良いので運行可<sup>[4]</sup>となり、ネットワークに付加される。このとき運行リンク⑥の”出発地出発時刻”は 10:01 となるのでノード④には運行情報リストとして  $[(6, 10:01)]$  が登録される。

更なる運行リンク⑦はノード②発、ノード④行き 11:30 発の航空便である。ノード②の運行情報リストは  $[(5, 10:20), (2, 10:02)]$  であるが、このリストを前の方から調べていくと、運行リンク⑤はノード②着 10:32 なる航空便であり、空港での他便への乗継時間が 40 分であるとすると、運行リンク⑦は運行可<sup>[4]</sup>となりネットワークに付加される。このとき、運行リンク⑤の出発地①の出発時刻は 10:20 であるのでノード④の運行情報リストは  $[(7, 10:20), (6, 10:01)]$  となる。この場合ノード③の運行情報リストの 2 番目の要素  $(2, 10:02)$  は出発地①の出発時刻が早いので無視される。

次の運行リンク⑧も同様にして運行可であって

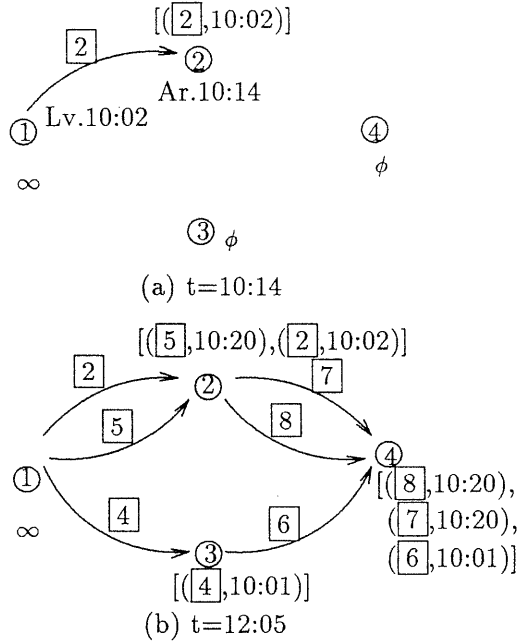


図 1 運行リンクの付加による動的ネットワークの形成

ネットワークに付加され、ノード④に  $[(8, 10:20), (7, 10:20), (6, 10:01)]$  なる運行情報リストが登録される。

次の運行リンク⑨の到着時刻は 12:05 であるがこれは 12:00 までに④に到着したいという条件を満たさないのでこの時点で運行リンクの付加による動的ネットワークの形成は終了し、図 1(b) に示すような動的ネットワークが得られることになる。

簡単な例を用いて説明した動的ネットワークの形成法を要約すると次のようになる。時刻表に相当する到着時刻順にソートされた出発希望時刻以降の運行リンクが逐次運行可かどうか調べる。運行可か否かは、その運行リンクの発ノードの運行情報リストの要素を先頭から順に取り出し、その要素を構成する運行リンクの着時間と乗継時間を加味して現在考えている運行リンクが運行可か否かを判定すれば良い。運行可と判定された運行リンクの着ノードにはその運行リンクと出発地出発時刻の組が出発地出発時刻の遅い順にソートされたリストの位置に登録されることになる。

次に図1(b)をもとにして最適系列を探索する方法について説明する。まず、目的地④の運行情報リスト  $[(8,10:20), (7,10:20), (6,10:01)]$  の各要素の運行リンクの着時間を調べ、着時刻の早い順にソートする。着時刻が同じ場合には、出発地出発時刻の遅い順にソートする。この例では、⑥の着時刻10:40、⑦の着時刻11:45、⑧の着時刻11:50であるので  $[(6,10:01), (7,10:20), (8,10:20)]$  となる。このリストの先頭要素は第1最適乗継系列の候補であり、これは出発地①を10:01に出て、運行リンク⑥で目的地④に10:40に着くことを示している。このとき⑥の発ノードは③であり、ノード③には  $[(4,10:01)]$  のみが登録されているので、ノード③に運行リンク④で着く①発10:01で来れば良いことが分かる。次に目的地に⑦で着く便は出発地を10:01より遅い10:20に出るので、この系列”①発10:01-ひかり3号-④着10:40”が第1最適乗継系列となる。次にリストの第2の要素  $(7,10:20)$  を取り出すと、これは第1最適乗継系列の出発地出発時刻10:01より遅い10:20であるので第2最適乗継系列の候補であり、出発地①を10:20に出て、運行リンク⑦で目的地11:45に着くことを示している。このとき⑦の発ノードは②であり、ノード②には  $[(5,10:20), (2,10:02)]$  が登録されているのでノード②に運行リンク⑤で着く①発10:20で来れば良いことが分かる。即ち乗継系列は、”①発10:20-JAL5便-②着10:32, ②発11:30-JAL17便-④着11:45”となる。

リストの第3の要素は  $(8,10:20)$  であるが、これは出発地①の出発時刻が10:20で前出の乗継系列と同じであるが、目的地④への着時刻が11:50と遅いので最適乗継系列とはなり得ず前出の系列が第2最適乗継系列となる。

一般に最適系列の探索は次のように行なわれる。目的地のノードに登録されていた運行情報リストの各要素を形成する運行リンクの着時刻の早い順に、着時刻が同じ場合には出発地出発時刻の遅い順にソートしたリストを形成する。このようにして形成されたリストの先頭要素を形成する運行リンクから順に出発地に至る運行リンク系列より成る木を形成し、最適乗継系列の定義を満たす乗継回数最小、乗継時間最小の木を分岐限定法を用いた縦型探索<sup>[5]</sup>で行なえば良いことは容易に理解されよう。

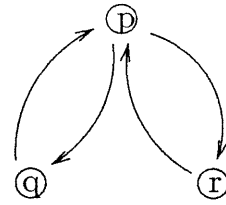


図2 連絡リンクの導入

上述の例では、乗継ノード間は運行リンクのみで接続されていたが、例えば最寄りの空港、新幹線特急停車駅、JR在来線特急停車駅の間をリムジンバス等で移動することが考えられる。このことを表現するため乗継ノード間に連絡移動時間をコストとして有する連絡リンクを導入する。図2は乗継ノード  $p$  と  $q$ 、 $p$  と  $r$  相互に連絡リンクがあることを示す図であり、 $p$  から  $q$  への連絡リンクを  $l(p, q)$  で表すことにすると、例えば乗継ノード  $q$  を発ノードとする運行リンク  $[a]$  は、たとえ  $q$  の運行情報リストが空集合  $\phi$  の場合でも次の条件が成立するとき、運行可となる。即ち乗継ノード  $p$  の運行情報リストが空集合でなく、その要素を構成するいずれかの運行リンク  $[b]$  に対し、その着時刻に運行リンク  $[b]$  から連絡リンク  $l(p, q)$  への乗継時間、連絡リンク  $l(p, q)$  の連絡移動時間、連絡リンク  $l(p, q)$  から運行リンク  $[a]$  への乗継時間を加えたものが、運行リンク  $[a]$  の発時間より早い時である。

本論文では、空港73ヶ所、新幹線停車駅66ヶ所、新幹線も停車する駅を除くJR在来線特急の停車駅474ヶ所を対象としたので乗継ノード総数は613となる。運行されている航空便は定期便1,232便、臨時便260便、新幹線は定期便596本、臨時238本、JR特急は定期1,154本、臨時145本であり、これを運行リンク本数で表現すると定期15,956本、臨時2,916本となる。また連絡リンク本数は140本である。

これまでの説明では出発地、目的地は空港、新幹線またはJR在来線の特急停車駅である乗継ノードであるとしていた。出発地、目的地が乗継ノードでない場合(これをローカル出発ノード、ローカル目的ノードと呼ぶ)には、最寄りの複数個の乗継ノードを見出し、図3に示すように連絡リンクを

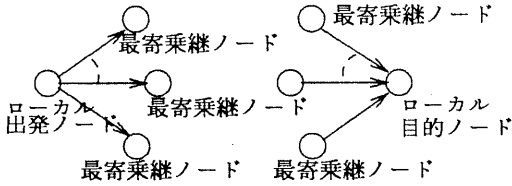


図3 ローカル出発ノードおよびローカル目的ノード

形成すれば前述の手法がほぼそのまま適用できる。ここでは最寄乗継ノードの算出およびそこへの連絡リンクの連絡所要時間は次のようにして求めている。

図4に示すネットワークにおいてリンクは、JR在来線、私鉄線、リムジンバスおよび徒歩連絡経路を表し、ノードは前述の乗継ノード、3本以上のリンクの分岐点である分岐ノード、1本または2本のリンクのみが接続されている端点・中間点ノードより成り、リンクには平均移動時間がコストとして付与されている。本論文で対象としたネットワークでは乗継ノード数は前述の如く613であり、分岐ノード数は1,253、端点・中間ノード数は7,149である。出発点または目的点が分岐ノードであるときは、端点・中間点ノードを無視し、乗継ノードおよび分岐ノードのみより成るネットワークに対して出発または目的分岐ノードからリンク間の乗継時間を考慮に入れて時間的に近い上位M個の乗継ノードを見出すことは容易に行なえる<sup>[4]</sup>。この場合ネットワークは613の乗継ノードと1,253の分岐ノードのみから成るので計算は極く短時間で実行でき、しかも出発地または目的地が指定されたとき1回おこなうだけでよい。出発地または目的地が端点・中間点ノードである場合には、これをネットワークに組入れて同様な計算を行なえばよいことは明らかである。

本論文ではローカル出発ノードまたはローカル目的ノードの最寄乗継ノードの数Mの標準値は5としたが、必要に応じて適当に削除したり、逆に特定の乗継ノードを取り入れることも出来ることは明らかである。

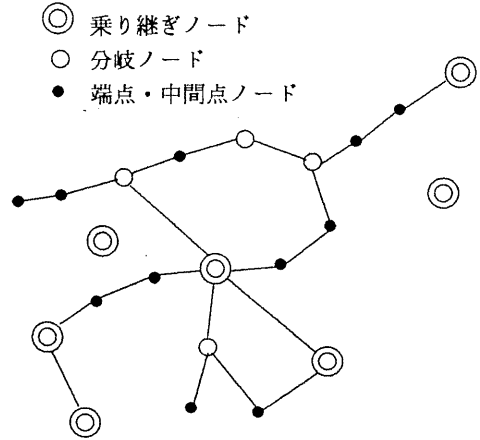


図4 平均移動時間ネットワーク

## 4 探索の手順および探索アルゴリズム

最適乗継系列探索の手順は次の通りである。

**手順1** 出発地、目的地、出発希望時刻、探索時間範囲を指定する。

**手順2** 出発地、目的地が乗継ノードでない場合には、平均移動時間ネットワークを用いて最寄りの複数の乗継ノードを見出し、連絡リンクを形成する。

**手順3** 到着時刻順にソートされた時刻表データを用いて探索時間範囲内で運行リンクの付加により動的ネットワークを形成する。これにより乗継ノードには到着した運行リンク番号と出発地出発時刻の組(運行情報)を要素とし、出発地出発時刻の遅い順にソートされた運行情報リストが登録される。

**手順4** 目的地ノードに登録された運行情報リストの要素を構成する運行リンクの目的地ノードへの到着の早い順に、到着時刻が同じ場合には、出発地出発時刻の遅い順にソートする。この再ソートされたリストの要素を先頭から順番に用いて、最適乗継系列の定義を満たす乗継回数最小、乗車時間最小となる出発地に

至る運行リンク系列より成る探索木を分岐限定法を用いた縦型探索により形成する。

#### 4.1 用語の説明

- **ノード**: 動的ネットワーク内の点を言い、 $1, 2, \dots, n$  と番号付けされている。出発ノードを  $O_0$ 、目的ノードを  $D_0$  とする。
- **リンク**: ノード間を結ぶ有向枝を言い、 $1, 2, \dots, m$  と番号付けされている。リンク  $a$  の始点ノードを  $B(a)$ 、終点ノードを  $E(a)$  で表す。リンク  $a$  が運行リンクの場合には発時刻を  $l_v(a)$ 、着時刻を  $a_r(a)$  で表し、リンク  $a$  が連絡リンクの場合には非負の連絡移動時間を  $d(a)$  で表す。尚時刻はすべて出発地の出発希望時刻を原点として分単位で表すものとする。
- **探索時間範囲**: 探索時間範囲  $T$  とは、出発希望時刻を時間の原点とする探索時間幅を言う。
- **リンク間遷移コスト**: あるノードに対する入リンク  $a$  と出リンク  $b$  に対して定まる乗継所要時間であり  $C(a, b)$  で示す。
- **リンクの出発地出発時刻**: リンク  $a$  の”出発地出発時刻”  $\mu(a)$  とは、リンク  $a$  の始点ノード  $B(a)$  までの単純経路（後述）のうち出発地  $O_0$  を最も遅く出発する時刻のことを言う。
- **ノードに対する運行情報リスト**: 各ノード  $x$  には、運行情報と呼ばれるそのノードを終点とする”運行リンク”と、その運行リンクの”出発地出発時刻”の組（運行情報）を要素とし、”出発地出発時刻”の遅い順にソートされた運行情報リスト  $\Delta(x)$  が登録される。 $\Delta(x)$  の先頭要素に相応するリンクを  $H(x)$  で示す。また、あるリンク  $a$  に対し、 $\Delta(E(a))$  において、リンク  $a$  を含む運行情報の直後にある運行情報のリンクを  $N(a)$  で示す。
- **仮想ノードおよび仮想リンク**: 系列探索木の形成時に仮想ノード  $n+1$  から出発ノード  $O_0$  に、目的ノード  $D_0$  から仮想ノード  $n+2$  に至るそれぞれコスト 0 なる仮想リンク  $m+1$ 、および  $m+2$  を便宜上設ける。
- **連絡リンク**: ノード  $x$  とノード  $y$  間の連絡リンクを  $l(x, y)$  で表す。
- **単純経路**: 閉路を含まない経路を単純経路と呼ぶ。
- **単純経路の木**: ネットワーク上の  $D_0$  までの単純経路を体系的に表したものであり、各節点（ネットワーク上のノードと区別するため、木のノードは節点と呼ぶことにする）はネットワーク上のリンクに対応し、根は仮想リンク  $m+2$  に対応する。節点番号は根を 1 とし、以後は発生された順に番号を付与し、第  $i$  番目に生成された節点の対応するリンクを  $\lambda_i$  と記す。任意の節点  $i$  から根までさかのぼった経路は、ネットワーク上の一つの単純経路を示す。この単純経路を  $P_i$  とし、 $i$  から 1 までの節点番号を  $i, p_2, p_3, \dots, p_l, 1$ 、とすると  $P_i$  は次のようになる。 $P_i = \lambda_i, \lambda_{p_2}, \lambda_{p_3}, \dots, \lambda_{p_l}, m+2$
- **親節点**: 節点  $i$  の親節点を  $F(i)$  で表す。これによりツリー構造が表現される。
- **乗継可能**: 乗継ノード  $x$  に入ってくる運行リンク  $a$  と出ていく運行リンク  $b$  は、 $a_r(a) + C(a, b) \leq l_v(b)$  のとき乗継可能であるという。
- **連絡乗継可能**: 乗継ノード  $x$  に入ってくる運行リンクを  $a$ 、乗継ノード  $y$  から出ていく運行リンク  $b$  があるとき、乗継ノード  $x$  と  $y$  が連絡リンクで結ばれているとき、 $a_r(a) + C(a, l(x, y)) + d(l(x, y)) + C(l(x, y), b) \leq l_v(b)$  のとき、連絡乗継可能であると言う。
- **発芽リンク**: 節点  $i$  に対応するリンク  $\lambda_i$  に乗継または連絡乗継可能なリンクのうち、節点  $i$  から下方に未だ生成されていないリンクのことを言い、これを  $\beta_i$  で表す。もしそのような入リンクがなければ  $\beta_i = \phi$  とする。
- **連絡乗継ノードリスト**: 乗継ノード  $x$  と連絡リンクで接続されている乗継ノードのリストを  $\delta(x)$  で表す。

## 4.2 運行リンクの付加による動的ネットワークの形成

運行リンクの付加による動的ネットワークの形成は次のようにして行なわれる。

[ステップ1] 出発ノード  $O_0$  は常に出発可 ( $\Delta(O_0) \leftarrow \infty$ ) とし、出発ノード以外のすべてのノードを未探索 ( $\Delta(i) \leftarrow \phi, i \neq O_0$ ) とし、時刻表テーブルの着時間ポインタを出発希望時刻以上の最初の位置に設定する。

[ステップ2] 時刻表テーブルを引き、着時刻が探索時間範囲  $T$  を越していれば終了。そうでなければ運行リンク  $a$  を取り出す。

[ステップ3] 運行リンク  $a$  の始点ノード  $B(a)$  の運行情報リスト  $\Delta(B(a))$  に乗継可能な運行リンク  $b$  が存在するか、またはノード  $B(a)$  の連絡乗継ノードリスト  $\delta(B(a))$  に登録されているノード  $x$  の運行情報リスト  $\Delta(x)$  に連絡乗継可能な運行リンク  $b$  が存在する場合には、ステップ4へ進む。それ以外は、時刻表テーブルのポインタを1つ進め、ステップ2に戻る。

[ステップ4] 条件を満たす運行リンク  $b$  の内、 $\mu(b)$  の最も大きい値を有するものを運行リンク  $a$  の”出発地出発時刻”として継承 ( $\mu(a) \leftarrow \mu(b)$ ) し、運行リンク  $a$  の終点ノード  $E(a)$  の運行情報リスト  $\Delta(E(a))$  に  $(a, \mu(a))$  を”出発地出発時刻”の遅い順に成るようソートして登録し、時刻表テーブルのポインタを1つ進めステップ2に戻る。

## 4.3 系列探索木の形成

目的地ノードに登録された運行情報リストの要素を構成する運行リンクの目的地ノードへの到着時刻の早い順に、到着時刻が同じ場合には、”出発地出発時刻”の遅い順にソートを行なった後、系列探索木を形成するのであるが、そのアルゴリズムを以下で述べる。ここで用いる変数の意味は次の通りである。

- $l$ : 現在対象としているリンク
- $i$ : 現在の節点番号
- $p$ : 新しく発生する節点の番号,  $u$ : 節点数
- $k$ : 求めた最適乗継系列数
- $q$ : 乗換回数,  $r$ : 乗車時間
- $q_0$ : 求めている経路の最低乗換数

$r_0$ : 求めている経路の最短乗車時間

$p_k$ : 第  $k$  最適乗継系列に相当する探索木の最後の節点

[ステップ1]  $u \leftarrow 1, i \leftarrow 1, p \leftarrow 1, k \leftarrow 1, \lambda_0 \leftarrow 0, \lambda_1 \leftarrow m + 2, F(1) \leftarrow 0, \beta_1 \leftarrow H(D_0), q \leftarrow 0, q_0 \leftarrow \infty, r \leftarrow 0, r_0 \leftarrow \infty, \mu(m + 2) \leftarrow \mu(\beta_1)$  とし、 $D_0$  以外のすべてのノードを未探索とする。

[ステップ2]  $l \leftarrow \beta_i$  とし、 $l = \phi$  ならばステップ4に進む。そうでなければ、節点  $i$  の発芽リンクを求め  $\beta_i$  とし、 $u \leftarrow u + 1, p \leftarrow u, \beta_p \leftarrow l, F(p) \leftarrow i, B(\lambda_i)$  を既探索、 $r \leftarrow r + d(\lambda_p), \lambda_p$  と  $\lambda_i$  が異なる列車または便であるとき  $q \leftarrow q + 1$  とする。  $\lambda_p$  と  $\lambda_i$  が連絡乗継とき、 $E(\lambda_p)$  を既探索、 $r \leftarrow r + d(l(E(\lambda_p), B(\lambda_i)))$ 、 $p \leftarrow p + 1$  とする。節点  $p$  の発芽リンクを求め  $\beta_p$  とする。  $q > q_0$  または、 $q = q_0$  かつ  $r > r_0$  であれば、 $\beta_p = \phi$  とする。

[ステップ3]  $i \leftarrow p$  とし、 $B(\lambda_p) = O_0$  または  $B(\lambda_p)$  の始点ノード  $B(\lambda_p)$  の連絡乗継ノードリストに  $O_0$  が登録されていればステップ5へ進む。そうでなければ、ステップ2に戻る。

[ステップ4]  $p \leftarrow i, i \leftarrow F(p)$  とし、 $O(\lambda_i)$  を未探索、 $r \leftarrow r - d(\lambda_q)$ 、 $\lambda_q$  と  $\lambda_i$  が異なる列車または便の時、 $q \leftarrow q - 1$  とする。  $\lambda_p$  と  $\lambda_i$  が連絡乗継の場合、 $B(\lambda_p)$  を未探索、 $r \leftarrow r - d(l(E(\lambda_p), B(\lambda_i)))$ 、 $p \leftarrow p - 1$  とする。  $i = 1$  ならばステップ6に進み、それ以外はステップ2に戻る。

[ステップ5]  $O_0$  を始点とする単純経路が求まったので、 $p_k \leftarrow p, r_0 \leftarrow r, q_0 \leftarrow q$  としてステップ2へ戻る。

[ステップ6]  $\beta_1 = \phi$  なら終了。そうでなければ、 $k \leftarrow k + 1, \mu(m + 2) = \mu(\beta_1)$  とし、ステップ2に戻る。

## 5 探索実行例

上述のアルゴリズムを用いて最適乗継系列を探索するシステムをパーソナルコンピュータ PC-9821 を使い、Windows 3.1 の下で実現した。この場合、最適乗継系列探索に要する CPU 時間は十分実用的なものであった。

- 出発地=甲府駅 (山梨県)
- 目的地=三沢駅 (青森県)
- 出発希望時刻=14:00

探索時間範囲=9時間

なる条件の下で求めた第1および第2最適乗継系列を図5および図6に示す。

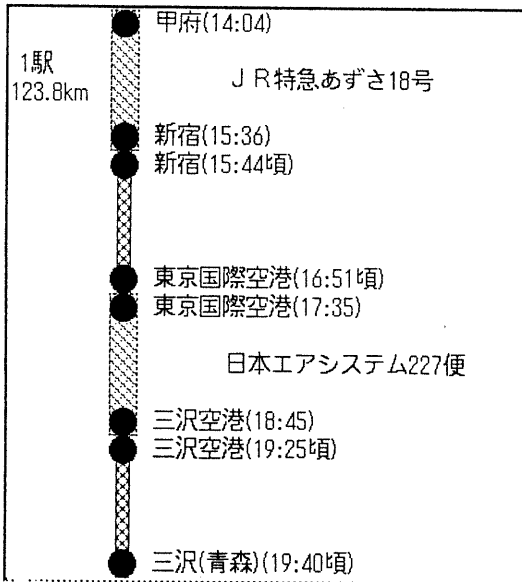


図5 甲府駅-三沢駅間第1最適乗継系列

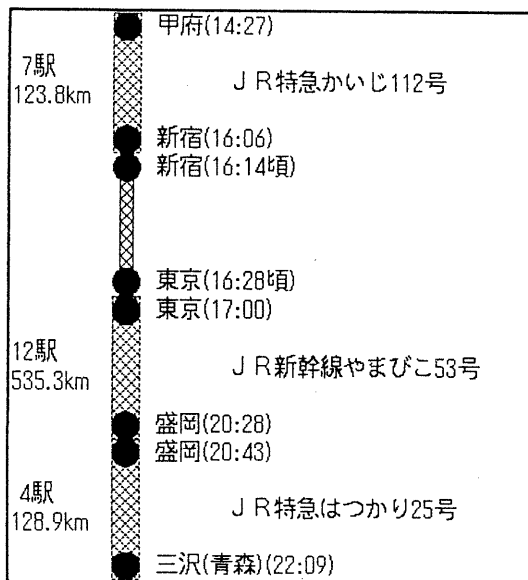


図6 甲府駅-三沢駅間第2最適乗継系列

## 6 むすび

日本全国の9,015ヶ所の駅または空港を出発地あるいは目的地として、航空機・新幹線特急・JR在来線特急の時刻表を考慮した列車・航空便の最適乗継系列を探索する手法ならび探索実行例について述べた。今後、座席予約システム等と組み合わせて、更に実用的なシステムにすることが考えられる。

最後に、御討論戴いた本学マルチメディア・ラボの諸氏に謝意を表する。

### 参考文献

1. 加藤誠巳, 高木啓三郎: 航空機・新幹線の最適乗継案内システム, 情処学会情報システム研, IS15-2(昭 62-07).
2. 加藤誠巳, 中條有規, 菊池新: 鉄道旅行スケジューリングシステム, 情処学会第44回全大, 4U-4(平 04-03).
3. 加藤誠巳, 菊池新, 高木啓三郎: 時刻表および平均移動時間を考慮した列車・航空便の最適乗継系列探索システム, 情処学会第48回全大, 2T-8(平 06-03).
4. 大西啓介, 加藤誠巳: 交差点内コストを考慮した道路網における経路探索の手法とそのマルチメディア型経路案内システムへの応用, 情処学論, Vol.33, No.7, pp.970-979(1992).
5. 今田俊明: 人工知能, オーム社(昭 62).