

動的ネットワーク表現に基づく列車・航空便の 最適乗継系列探索の手法とその実際への応用

菊池 新[†] 高木 啓三郎^{††} 加藤 誠 巳[†]

近年、公共交通網の発達にともない、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経路を見出すこと、すなわち動的最適経路問題を人手で解くことは困難になりつつある。このような動的最適経路問題を解くため、従来からネットワークに時間の概念を導入する手法および近似解を与える方法がいくつか提案されてきている。しかし、厳密な最適解を効率良く求める手法はまだ与えられていなかった。本論文の目的はこのような問題を解決するべく空間と時間を一体化してコンパクトに表現する動的ネットワークと名付けた新しい概念を提案することにある。これによりネットワーク規模は縮小され、したがって使用するメモリ量ひいては計算所要時間を減少させることができる。最適解の探索は、本文中で定義する出発地出発時刻を逐次継承した動的ネットワークを生成した後、継承した出発地出発時刻を活用して条件を満たす最適乗継系列を分岐限定法を用いて後方向探索することにより実行される。さらに、ここで提案した探索手法を用いて、日本全国の私鉄を含む9,015の駅ならびに空港のいずれか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、最適乗継系列を実際に探索するシステムをパーソナルコンピュータを用いて試作した例について述べ、ここで提案した探索手法の有効性を示している。

A Method for Finding the Optimal Transfer Sequences of Trains and Airplanes Based on the Dynamic Network Representation and Its Application to a Practical Problem

SHIN KIKUCHI,[†] KEIZABURO TAKAGI^{††} and MASAMI KATO[†]

In recent years, the public transportation networks have made remarkable progress, so that one cannot easily find the optimal transfer sequence from his origin to his destination. While several methods for introducing the concept of time into networks and for giving suboptimal solutions have been proposed, they are not necessarily satisfactory. The authors have been engaged in developing a method and a system which provides the optimal transfer sequences of the trains and the airplanes with the timetables taken into account. In this paper, we introduce a novel concept of the dynamic network which compactly represents the spatial factor and the temporal factor together. Using this dynamic network representation, the authors give a novel and efficient algorithm for finding the strictly optimal solutions which employs the forward deduction using the timetable constraint and the backward induction using a branch and bound method. Furthermore, an optimal transfer sequence searching system for the airplanes and the limited expresses of the Sinkansen and the JR local lines has been developed using a personal computer, in which any one of 9,015 stations and airports can be chosen as an origin or a destination. The memory capacity and the CPU time required to obtain the optimal transfer sequences is quite reasonable, which reveals that the algorithm and the system is of practical use.

1. ま え が き

ノードとリンクで構成されるネットワークにおいて

リンクに非負のコストが付与されている場合、任意の出発ノードから任意の目的ノードに至るコスト最小の経路を見出す問題は静的最適経路問題として知られ、いくつかの探索手法が知られている¹⁾。これに対し、列車・航空便のように時刻表に基づいて運行される輸送網における最適経路問題は出発地を出発する時刻により最適経路が一般に異なるため動的最適経路問題と呼ばれ、従来の静的最適経路探索の手法をそのまま適用することはできない。そして、厳密な最適解を

[†] 上智大学理工学部電気電子工学科

Department of Electrical and Electronics Engineering,
Sophia University

^{††} 日本電気株式会社情報メディア研究所

Information Technology Research Laboratories, NEC
Corporation

求めることが難しいと考えられたため近似解を与える方法が従来からいくつか提案されている。たとえば、文献2)で述べられている優等列車を対象とした列車案内システムでは、直通列車あるいは1回乗継列車はすべて抽出し、もしそのような列車がないならば複数回乗継ぎを乗継ぎ駅数を制限するなどして抽出提供する方法を提案している。航空便を対象とした他のシステムでは、航空路を市外電話回線網に、空港を市外電話交換局に対応させ、空港の階位と出発空港ならびに目的空港の地理的位置情報を有効に利用することにより着信局に近い斜め回線から優先的に探索を行う市外電話の迂回中継方式である遠近回転法 (*far to near rotation*)³⁾と類似した考え方により複数個の適当な乗継ぎ航空便を選択する方法を採用している⁴⁾。さらなる例としては、鉄道網を対象として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間の第 k 番目までの最短時間経路を求め、求まった経路に対して時刻表データを当てはめて使用する列車を定める方法を提案している⁵⁾。しかし、上述のいずれの方法も求められた乗継ぎ系列が最適であるという保証はない。

このような動的最適経路問題の最適解を求めるためには、一般に空間的概念だけでなく時間的な概念をも同時に表現しうるネットワーク・モデルを導入することが要求される。そのようなモデルの1つとしてプレイスにトークンの保持時間を与えるプレイス・タイムベトリネットが提案されている⁶⁾。しかし、モデルが複雑であるため多数の便が運行される大規模な輸送網における最適乗継系列を求めるのには適していない。空間と時間を融合させた最もよく知られたネットワーク・モデルとしてダイヤネットワーク⁷⁾がある。このダイヤネットワークは本論文で提案する動的ネットワークの基本を成すものである。しかし、後で述べるようにこのダイヤネットワークを用いて連絡移動時間や乗継所要時間を表現するためにはネットワークを拡張する必要がある。そのため大規模な輸送網に適用するとダイヤネットワークのノード数、リンク数は増大し、ネットワークを計算機内部で表現するためのメモリ使用量が増大するとともに最適経路を探索するのに要する計算時間も増大する欠点があった。

本論文の特徴は空間と時間を一体化して表現するダイヤネットワークの中から関連する部分のみを動的に生成することによって得られる動的ネットワークと名付けたモデルを導入した点にある。これによりネットワークの規模は減少し、したがって使用するメモリ

量については計算所要時間を減少させることができる。さらには連絡移動時間や乗継所要時間の変更に対しても容易に対処することが可能である。最適解の探索は、本文中で定義する出発地出発時刻を逐次継承した動的ネットワークを生成した後、継承した出発地出発時刻を活用して条件を満たす最適乗継系列を分岐限定法を用いて後方向探索することにより実行される^{8),9)}。

さらに、ここで提案した手法を日本全国で運行されている航空便、新幹線特急ならびにJR在来線特急に対して適用した。そこにおいて、日本全国の私鉄を含む9,015の駅ならびに空港のいずれかを出発地あるいは目的地として指定したときに、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を用いて試作し、ここで提案した探索手法の有効性ならびに実用性を明らかにしている。

2. 求めようとする最適乗継系列

本章では、本論文で求めようとする最適乗継系列なるものの定義を与える。まず、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個存在する場合は、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。

上述の最適乗継系列の定義の意味するところは自明と考えられるが、図1に示すダイヤネットワークを用いて簡単に説明する。出発駅Aを与えられた出発希望時刻以降に出発して便1~6を用いて目的駅Eに最も早く着きたいものとする。図より駅A-B間は便1または便2、駅B-C間は便2または便3、駅C-D間は便4または便5、そして駅D-E間は便6を用いることにより、駅Eに最も早く到着できるから①により

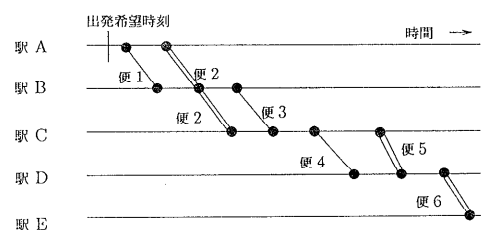


図1 最適乗継系列

Fig. 1 The best transfer sequence.

便 6 を用いるべきことが確定する。駅 A-B 間は、便 1, 便 2 のいずれを用いても駅 B に同じ時刻に到着しうが ① により便 2 が選択される。駅 B-C 間では ② により駅 B で乗り換えないうえに便 2 を使用することになる。駅 C-D 間の便は ③ により乗車時間の短い便 5 が選択される。これにより、駅 A-(便 2)-駅 B-(便 2)-駅 C-(便 5)-駅 D-(便 6)-駅 E がこの場合の最適乗継系列となる。

一般に第 k 最適乗継系列は第 $k-1$ 最適乗継系列の出発地出発時刻以降に出発地を出発し、最も早く目的地に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。また、第 k 最適乗継系列に出発時刻、到着時刻、乗換回数、乗車時間が同じである同格のものが複数個あるときには、その第 k' 番目のものを第 (k, k') 最適乗継系列と表記する。このように定義したとき本論文で求めようとする最適乗継系列とは、出発地出発希望時刻と探索時間範囲を指定し、求めたい最適乗継系列が最大 k_{\max} 番目までであるとするとき、探索時間範囲内に第 k_{\max} 番目以上の解が有る場合には第 k_{\max} 番目までの解のことをいい、探索時間範囲内では第 k_0 ($< k_{\max}$) 番目までの解しか求まらなかったときには第 k_0 番目までの解のことをいうことにする。

なお、到着希望時刻を指定して、出発地を最も遅く出発する系列を最適系列と定義することもできるが、以下で述べる探索法と基本的に同一の概念が適用可能であるので、ここでは取り扱わないこととする。さらに、ここでは航空便、新幹線特急および JR 在来線特急のみ時刻表を考慮し、それ以外は平均的所要時間を用いることとする。

3. 最適乗継系列探索のアルゴリズム

3.1 用語の定義

- 動的ネットワーク：時間とともに、関連するノードおよびリンクを逐次付加することにより形成されるネットワークをいう。
- ノード：動的ネットワーク内の点をいい、 $1, 2, \dots, n$ と番号付けされている。出発地ノードを O_0 、目的地ノードを D_0 とする。出発地ノードおよび目的地ノードは乗継ノードでなくてもよいが、その他はすべて乗継ノードより成る。
- 乗継ノード：対象とするすべての空港および駅のうち、空港、新幹線または JR 在来線特急の停車駅のことをいう。
- ローカル出発地(目的地)ノード：出発地(目的地)ノードが乗継ノードでない場合をいう。

- リンク：ノード間を結ぶ有向枝をいい、 $1, 2, \dots, m$ と番号付けされている。リンク a の発ノードを $B(a)$ 、着ノードを $E(a)$ で表す。
- 運行リンク：発ノードからの発時刻 $l_v(a)$ と着ノードへの着時刻 $a_r(a)$ を有するリンク a をいう。時刻は出発地の出発希望時刻を原点として表すものとする。
- 連絡リンク：発ノードと着ノード間の非負の連絡移動時間 $d(a)$ をコストとして有するリンク a をいう。ノード x からノード y への連絡リンクを $l(x, y)$ で表す。また、ノード x から目的地 D_0 に対する連絡リンクを $l_d(x)$ とする。
- 仮想ノードおよび仮想リンク：仮想ノード $n+1$ から出発地ノード O_0 に、目的地ノード 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。
- 出発希望時刻：出発地の出発希望時刻を t_0 とし、これを時間の原点とする。
- 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いう。
- リンク間遷移コスト¹²⁾：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、 $C(a, b)$ で表す。
- 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いう。
- ノードに対する運行情報：あるノード x に対し、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”の組のことをいう。
- ノードに対する運行情報リスト：あるノード x に対し、そのノードに対する運行情報を“出発地出発時刻”の遅い順にソートしたリスト $\Delta(x)$ をいう。 $\Delta(x)$ 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 λ_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1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。これにより木構造が表現される。
- 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 があるとする。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いう。
- 発芽リンク：節点 i に対応するリンク λ_i に乗継または連絡乗継可能なリンクであって、節点 i から下方にまだ生成されていないリンクの中で、その“出発地出発時刻”がリンク λ_i の“出発地出発時刻”と同じリンクの1つをいい、これを β_i で表す。もしそのようなリンクがなければ $\beta_i = \phi$ 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。

3.2 探索手順の概要

最適乗継系列探索の概略の手順は次のとおりである。

手順0 出発地、目的地、出発希望時刻、探索時間範囲および求めたい最適乗継系列の最大次数を指定する。

手順1 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動的ネットワークを形成し、各ノードに運行情報リストを登録する。

手順2 目的地ノードに登録された運行情報リストを用いて系列探索木を形成し、所望の最適乗継系列を求める。

以下、3.3節および3.4節において手順1および手順2をアルゴリズムの形で詳細に述べる。

3.3 動的ネットワークの形成

[ステップ1] 仮想リンク $m+1$ の着時刻 $a_r(m+1)$ を0、入リンクを $m+1$ とするリンク間遷移コスト $C(m+1, *)$ を0、出発地ノード以外のすべてのノードを未探索 ($\Delta(i) \leftarrow \phi, i \neq O_0$), $\Delta(O_0) \leftarrow [(m+1, 0)]$ とし、時刻表テーブルの着時刻ポイントを出発希望時刻以降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 を越えていれば「動的ネットワークの形成」を終了。そうでなく、発時刻が出発希望時刻 t_0 より小ならばステップ6に進み、大ならばその運行リンク 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、ステップ6に進む。

[ステップ4] 条件を満たす運行リンク b のうち、 $\mu(b)$ 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 ($\mu(a) \leftarrow \mu(b)$) する。ただし、 $b = m+1$ かつ $B(a) = O_0$ である場合は、 $\mu(b) (= \mu(m+1)) = l_v(a)$ とし、また、 $b = m+1$ かつ $B(a) \neq O_0$ である場合は、 $\mu(b) (= \mu(m+1)) = l_v(a) - C(l(E(m+1), B(a)), a) - d(l(E(m+1), B(a)))$ とする。そして、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”の遅い順になるようソートして登録する。目的地 D_0 の連絡乗継ノードリスト $\delta(D_0)$ に $E(a)$ が登録されている場合は、ステップ5に進む。それ以外は、ステップ6に進む。

[ステップ5] $a_r(a) + C(a, l_d(E(a))) + d(l_d(E(a))) \leq T$ の場合、運行リンク a の“出発地出発時刻”を継承 $\mu(l_d(E(a))) \leftarrow \mu(a)$ し、目的地 D_0 の運行情報リスト $\Delta(D_0)$ に $(l_d(E(a)), \mu(l_d(E(a))))$ を登録する。

[ステップ6] 時刻表テーブルのポイントを一つ進め、ステップ2に戻る。

3.4 系列探索木の形成^{11),12)}

ここで用いる変数の意味は次のとおりである。

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_{\max} : 求めたい最適乗継系列の最大次数

k : 求めた最適乗継系列数

k' : 求めた第 k 最適乗継系列と同格の最適乗継系列数

q : 乗換回数

r : 乗車時間

q_0 : 求めている経路の最小乗換数

r_0 : 求めている経路の最小乗車時間

$p_{k,k'}$: 第 (k, 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] $u \leftarrow 1, p \leftarrow 1, i \leftarrow 1, k \leftarrow 1, k' \leftarrow 1, \lambda_0 \leftarrow 0, \lambda_1 \leftarrow m + 2, F(1) \leftarrow 0, \beta_1 \leftarrow H(D_0), q \leftarrow 0, q_0 \leftarrow \infty, r \leftarrow 0, r_0 \leftarrow \infty, \mu(m + 2) \leftarrow \mu(\beta_1)$ 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] リンク β_i をノード i に対応するリンクとして生成, すなわち $l \leftarrow \beta_i$ とし, $l = \phi$ かつ $i = 1$ ならばステップ 6 に進み, $l = \phi$ かつ $i \neq 1$ ならばステップ 4 に進む. そうでなければ, 節点 i の未生成リンク, すなわち発芽リンクを求め β_i とし, $u \leftarrow u + 1, p \leftarrow u, \lambda_p \leftarrow l, F(p) \leftarrow i, B(\lambda_i)$ を既探索, $r \leftarrow r + d(\lambda_p), \lambda_i \neq m + 2$ かつ λ_p と λ_i が異なる列車または便であるとき $q \leftarrow q + 1$ とする. λ_p と λ_i が連絡乗継のとき, $E(\lambda_p)$ 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 β_p とする. $q > q_0$ または, $q = q_0$ かつ $r > r_0$ であれば, $\beta_p \leftarrow \phi$ とする.

[ステップ 3] $i \leftarrow p$ とし, $B(\lambda_p) = O_0$ ならば, ステップ 5 へ進む. また, λ_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, $r \leftarrow r + d(l(O_0, B(\lambda_p))), q \leftarrow q + 1$ とし, 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, $r \leftarrow r - d(\lambda_q), \lambda_i \neq m + 2$ かつ λ_q と λ_i が異なる列車または便のとき $q \leftarrow q - 1$ とする. λ_p と λ_i が連絡乗継の場合, $E(\lambda_p)$ 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, $B_p \leftarrow \phi$ とし, $r_0 = r$ かつ $q_0 = q$ ならば, $p_{kk'} \leftarrow q, k' \leftarrow k' + 1$ とし, そうでなければ $k' \leftarrow 1, p_{kk'} \leftarrow p, r_0 \leftarrow r, q_0 \leftarrow q$ とする. $B(\lambda_p) \neq O_0$ ならば, $r \leftarrow r - d(l(O_0, B(\lambda_p))), q \leftarrow q - 1$ とする. ステップ 2 へ戻る.

[ステップ 6] $\beta_1 \leftarrow N(\beta_1)$ とし, $\mu(m + 2) = \mu(\beta_1)$ ならステップ 2 に戻る. $\beta_1 = \phi$ または $k = k_{\max}$ なら「系列探索木の形成」終了. そうでなければ, $k \leftarrow k + 1, r_0 \leftarrow \infty, q_0 \leftarrow \infty, k' \leftarrow 1, \mu(m + 2) = \mu(\beta_1)$ とし, ステップ 2 に戻る.

4. 探索手法の例による説明

前章で提案した動的ネットワーク表現に基づく探索手法をその基本となるダイヤネットワーク表現と対比させながら例を用いて説明する. 図 2 において, $\textcircled{0}_0$, $\textcircled{0}_1$ はそれぞれ出発地, 目的地を表すノード, $\textcircled{1} \sim \textcircled{6}$ は空港, 新幹線または JR 在来線特急の停車駅を表す

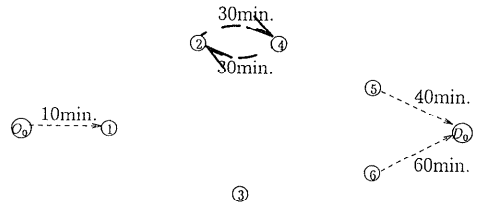


図 2 基本ネットワーク
Fig. 2 Basic network.

乗継ノードとする. 乗継ノード $\textcircled{2}$ と $\textcircled{4}$ の間のリンクはリムジンバス, 電車, 徒歩等による乗継ノード間の連絡を表す連絡リンクである. 図 2 の例では出発地, 目的地がローカル出発地ノード, ローカル目的地ノードである場合を示しており, 出発地 O_0 の最寄りの乗継ノードは $\textcircled{1}$, 目的地 D_0 の最寄りの乗継ノードは $\textcircled{5}$ および $\textcircled{6}$ である. またここで航空機の乗継所要時間は 40 分, 特急の乗継所要時間は 10 分であるものと仮定する.

表 1 に示すように到着時刻順にソートされた時刻表に従う列車, 航空機が運行されている場合について考える. ここで乗継ノード $\textcircled{1}$ および $\textcircled{3}$ は説明の便宜上空港と駅の両者の機能を兼ね備えたノードであるものとする. さらに出発希望時間は 9:00 以降とし, 探索時間範囲は 8 時間 30 分, すなわち目的地に 17:30 までに到着する系列を探索するものとする. 図 3 は表 1 に示す時刻表に従う列車および航空機の運行状況を斜めの実線で, 連絡移動および乗継移動を斜めの点線で表したダイヤネットワーク表現である. このダイヤネットワーク表現では $\textcircled{1} \sim \textcircled{6}$ の乗継ノードは乗継移動時間を表現するためそれぞれ発ノードと着ノードの 2 つに分解されていることに注意されたい.

以下このダイヤネットワーク表現を参照しながら動的ネットワークの形成法について説明する. 各ノードには運行情報と呼ぶそこに到着する“運行リンク”と“出発地出発時刻”の組を要素とし, この要素の“出発地出発時刻”の遅い順にソートされた運行情報リストが逐次形成される. この運行情報リストは, 出発地ノード $\textcircled{0}_0$ を除き初期状態では空リスト ϕ とする. 出発地ノード $\textcircled{0}_0$ には出発可を与えるリスト ∞ が付与される. 表 1 に示すように, 到着時刻順にあらかじめソートされた時刻表を用い, 到着時刻が 9:00 以降の列車または便を順に調べる. 最初の便である JAL 1 便 (運行リンク $\textcircled{0}$) は $\textcircled{1}$ を 9:20 に出発する. しかし, 図 3 に示すように乗継所要時間 40 分, 出発地からの移動時間 10 分を考慮に入れると, 出発地 $\textcircled{0}_0$ を 8:30 に出なければならず, 出発希望時刻 9:00 を満た

表 1 運行ダイヤの例
Table 1 An example of timetable.

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
⋮	⋮	⋮	⋮	⋮	⋮
[-1]	ANA 3 便	①	②	7:50	8:50
[0]	JAL 1 便	①	②	9:20	10:20
[1]	ANA 7 便	①	②	10:00	11:00
[2]	JAL 3 便	①	②	10:30	11:30
[3]	ひかり 3 号	①	③	11:30	12:30
[4]	のぞみ 1 号	①	③	12:00	13:00
[5]	ANA 12 便	③	④	13:20	13:50
[6]	JAL 8 便	④	⑤	13:40	14:10
[7]	ANA 10 便	④	⑤	14:00	14:30
[8]	のぞみ 1 号	③	⑥	13:00	15:50
[9]	JAL 12 便	④	⑤	15:30	16:00
[10]	ANA 12 便	③	④	17:10	17:40
⋮	⋮	⋮	⋮	⋮	⋮

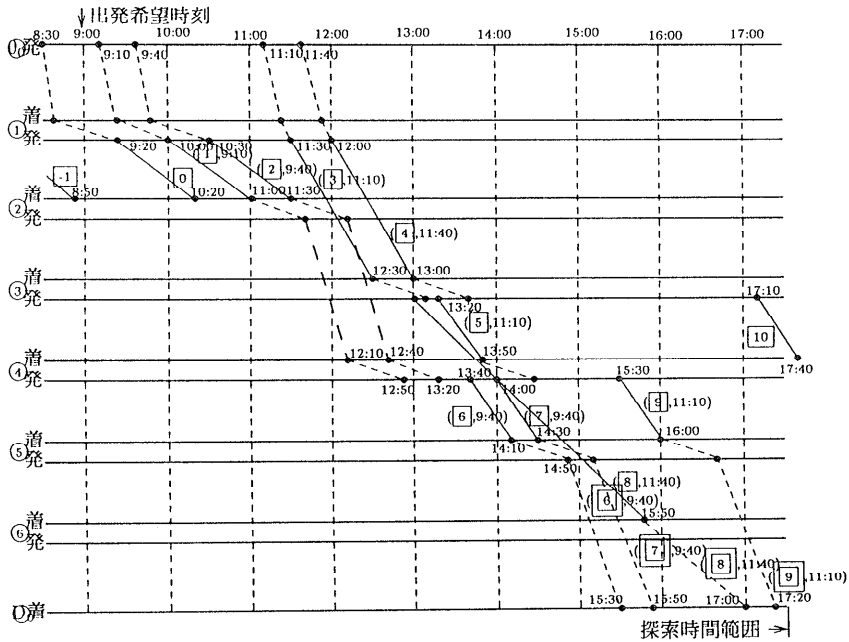


図 3 ダイヤネットワークによる表現
Fig. 3 Diagram network representation.

さないのを棄却する。次に 11:00 に ② に到着する便 ANA 7 便はノード ① を 10:00 に出発する。これに乗るには出発地 ② を 9:10 に出ればよいので、これは条件を満たす。そこでこの運行リンク [1] を図 4 (a) に示すようにノード ① とノード ② の間に付加し、ノード ② に対し運行リンク番号 [1] と出発地出発時刻 9:10 の組、すなわち運行情報 ([1], 9:10) が空リス

ト φ であった運行情報リストに加えられる。

次の運行リンク [2] も同様にしてネットワークに付加される。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ているので [(2), 9:40), ([1], 9:10)] となる。同様にして運行リンク [3] および [4] も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), 11:40), ([3],

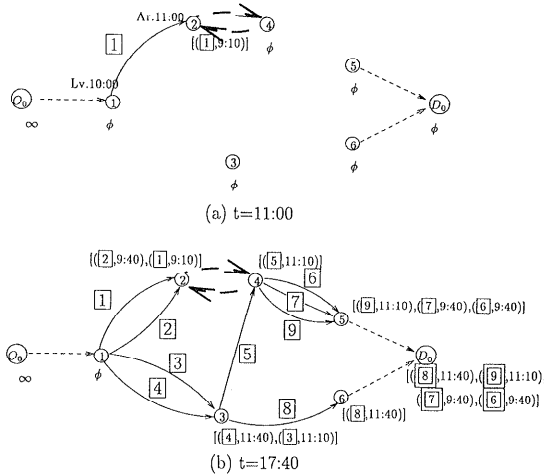


図4 動的ネットワークの形成
Fig. 4 Formation of dynamic network.

11:10) となる。

次の運行リンク [5] の発ノード ③ の運行情報リストは [(4), 11:40), (3), 11:10)] であるが、このリストを先頭要素から順に調べていく。最初の運行リンク [4] でノード ③ に着いた場合には航空機への乗継時間 40 分を加味した 13:40 以降でない航空機へは乗り継げないので棄却する。次の要素を形成する運行リンク [3] は 13:10 以降であれば航空機に乗り継可能である。したがって運行リンク [5] はネットワークに付加される。このとき運行リンク [5] の“出発地出発時刻”は運行リンク [3] の“出発地出発時刻” 11:10 が継承され、ノード ④ の運行情報リストは [(5), 11:10)] となる。

次の運行リンク [6] はノード ④ 発 13:40 である。ノード ④ の運行情報リストは [(5), 11:10)] なので、ノード ④ には運行リンク [5] が 13:50 に着いているが、時間の前後関係から運行リンク [6] には乗り継げない。しかし、ノード ④ は連絡リンクによりノード ② と連絡している。ノード ② の運行情報リスト [(2), 9:40), (1), 9:10)] を調べることで、運行リンク [2] からの連絡リンクを介しての連絡乗継が可能であることが分かる。したがって、リンク [6] にはリンク [2] の“出発地出発時刻”が継承され、ネットワークに付加され、ノード ⑤ の運行情報リストは [(6), 9:40)] となる。このとき運行リンク [6] の着ノード ⑤ はローカル目的地ノード ①₀ の最寄り乗継ノードであるので、目的地ノード ①₀ の運行情報リストを [(6), 9:40)] とする。ここで [6] は、14:50 にノード ⑤ を出て、15:30 に目的地 ①₀ に着く運行リンクを表す。同様に運行リンク [7] もネットワークに付加され、ノード ⑤ の運

行情報リストは [(7), 9:40), (6), 9:40)] となる。また、運行リンク [7] により目的地 ①₀ の運行情報リストは [(7), 9:40), (6), 9:40)] となる。同様に運行リンク [8], [8], [9] および [9] がネットワークに付加され、各ノードの運行情報リストが逐次形成される。

次の運行リンク [10] の到着時刻は 17:40 である。これは 17:30 までに目的地 ①₀ に到着したいという条件を満たさないでこの時点で動的ネットワークの形成は終了し、図 4 (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。

このようにして得られた動的ネットワークの各ノードに対して形成された運行情報リストを元にして最適乗継系列の探索は次のようにして行われる。まず、目的地ノード ①₀ に登録された運行情報リストの各要素を形成する運行リンクの目的地ノードへ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早い順にソートしたリストを形成する。この場合、運行リンク [8], [9], [7], [6] の目的地ノードへの着時刻はそれぞれ 17:00, 17:20, 15:50, 15:30 であるので [(6), 9:40), (7), 9:40), (8), 11:40), (9), 11:10)] とソートされる。このように再ソートされたリストの要素を先頭から順に用いて出発地に至る運行リンク系列より成る木を形成する。このとき、最適乗継系列の定義を満たす乗継回数最小、乗車時間最小の木を分岐限定法を用いた縦型探索¹⁰⁾で行うことになる。

第 1 最適乗継系列は第 1 の要素 [(6), 9:40) の運行リンク [6] により 15:30 に目的地ノードに着く。この運行リンク [6] は出発地出発時刻が 9:40 である運行リンク [6] を利用している。運行リンク [6] はノード ④ を発ノードとする運行リンクであるが、この例ではノード ④ はノード ② と連絡リンクで接続されている。したがってノード ② およびノード ④ の運行情報リストで出発地出発時刻が 9:40 であるものをさがすと、ノード ② の運行情報として [(2), 9:40) が唯一見つかるので運行リンク [2] により、ノード ① を出ることが分かる。ここでノード ① は出発地ノード ①₀ の最寄り乗継ノードであるのでこれで第 1 最適乗継系列が求まったことになる。すなわち第 1 最適乗継系列は “①₀ 発 9:40, ① 着 9:50, ① 発 10:30, [2] JAL 3 便), ② 着 11:30, ④ 着 12:40, ④ 発 13:40, [6] JAL 8 便), ⑤ 着 14:10, ⑤ 発 14:50, ①₀ 着 15:30” となる。この操作は図 3 のダイヤネットワークでいうと ①₀ に 15:30 に着くのに、①₀ を 9:40 に出発する条件を満たす乗継系列を時間を逆行して手繰ることに相当している。

次に目的地ノードの再ソートされた運行情報の第2要素(⑦, 9:40)を調べる。運行リンク⑦の目的地ノードへの着時刻は15:50で第1最適乗継系列の着時刻15:30より遅く、かつ出発地出発時刻は9:40であって第1最適乗継系列と同一であるので、最適乗継系列を形成し得ないことが分かる。同様にして第3の要素(⑧, 11:40)の運行リンク⑧, 運行リンク⑧, そして④を手繰ることにより第2最適乗継系列は“(②)発11:40, ①着11:50, ①発12:00, (④)のぞみ1号, ③着13:00, ③発13:00, (⑧)のぞみ1号, ⑥着15:50, ⑥発16:00, (②)着17:00”となる。最後の要素(⑨, 11:10)に関しては、運行リンク⑨の出発地出発時刻は11:10であって第2最適乗継系列のそれより早いので最適乗継系列とはなり得ない。

上述の例を用いた説明により、ここで提案する動的ネットワーク表現に基づく最適乗継系列の探索手法は、ダイヤネットワーク表現を用いることと基本的には同じであり、厳密な最適解が得られることは明らかである。

本論文で提案した動的ネットワーク表現の新規性ならびに特徴をまとめると次のようになる。

- (1) ダイヤネットワーク表現では空港または駅での便または列車の発着ごとに1つのノードが要求されるのに対し、動的ネットワーク表現では1つの空港または駅はただ1つのノードで表現される。
- (2) ダイヤネットワーク表現では、空港または駅に便あるいは列車が到着するごとに乗継移動を表現するリンクが必要とされるのに対し、動的ネットワーク表現ではリンク間の遷移コストなる概念の導入によりリンクとして表現する必要がない。
- (3) 連絡移動を表現する場合、ダイヤネットワーク表現では空港または駅に便あるいは列車が到着するごとに連絡移動を表現するリンクが必要とされるのに対し、動的ネットワーク表現ではただ1本の連絡リンクで表現できる。
- (4) 乗継移動時間あるいは連絡移動時間を変更すると、ダイヤネットワーク表現ではネットワークのトポロジー自体が変化しうるのに対し、動的ネットワーク表現ではネットワークのトポロジーには何らの変化も生じない。
- (5) 動的ネットワーク表現を用いると時刻表の変更ならびに空港あるいは駅の新設、廃止に容易に対処することができる。

5. 実際への応用⁹⁾および考察

5.1 対象とした交通網

本論文で提案した最適乗継系列探索の手法の実用性を確かめるため、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、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最適乗継系列を探索するシステムを試作した。ここで対象とした空港は73, 新幹線停車駅は66, 新幹線も停車する駅を除くJR在来線特急の停車駅は474であり、したがって乗継ノード総数は613となる。時刻表データは1993年10月当時のものを使用し、運行されている航空便は定期1,232便、臨時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すと定期15,956本、臨時2,916本となる。また使用した連絡リンク本数は140本である。

出発地、目的地が乗継ノードでない場合、最寄りの複数の乗継ノードの算出およびそこへのアクセス時間は次のようにして求めた。図5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す。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●より成る。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は前述の如く613であり、分岐ノードは1,253, 端点・中間点ノードは7,149である。出発地または目的地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ノード数計1,866個のネットワークを考える。このネットワークにおい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個の乗継ノードはリンク間の乗継時間を道路網における交差点内コストと置き換える

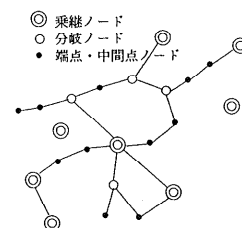


図5 平均移動時間ネットワーク

Fig. 5 Network consisting of links having average travel time costs.

ことにより文献 12) に示す方法を用いて容易に算出される。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み入れて同様な計算を行えばよい。本システムではローカル出発地ノードまたはローカル目的地ノードの最寄乗継ノードの数 M の標準値は 10 としたが、必要に応じて削除、追加が自由にできるようにしている。

5.2 試作したシステムの概要

5.1 節で述べた交通網を対象として最適乗継系列を探索するシステムをインテルの 80486 DX (33 MHz) を CPU とする 16 MB のメモリを実装したパーソナルコンピュータ PC9821 As2 を使い、Windows 3.1 の下で動作する C 言語により試作した。ソースプログラムの規模はステートメント数にして約 11,000 行、ASCII ファイル容量で約 274 kB である。システム開発に際しては探索プログラムの作成だけでなく平均移動時間ネットワークデータの作成および時刻表データの入力にかなりの労力を必要とした。

本システムのユーザが入力すべき基本パラメータは、出発地、目的地、出発希望時刻、探索時間範囲および探索系列数の最大次数 (k_{max}) であり、臨時便も使用したい場合には出発日付 (月日) も入力する必要がある。これら基本パラメータ以外に本システムでは探索条件を種々に設定して探索を実行できるようにしている。すなわち、ユーザは航空機、新幹線特急、JR 在来線特急をすべて利用することもできるし、特定のものを、たとえば航空機は利用しないことも可能である。また、航空便、列車ごとに欠航、運休状態に設定することも可能であり、乗継所要時間も標準値以外の値に自由に設定可能である。さらに前述したように出発地、目的地が乗継ノードでない場合には、標準では時間的に近い 10 個の最寄乗継ノードを自動選択している。システムが求めた最適乗継系列は図形的に表現されるだけでなく、地図上にも概略の経路が表示される。さらに付加情報として総所要時間とその内訳である乗車時間と乗換・待時間、乗換回数、乗継ノード間の停車駅数およびキロ数も表示される。なお、求めた複数個の最適乗継系列を Windows 3.1 の MDI を用いてカスケード表示、およびタイル表示により同時に見ることができるようになっている。さらなる付加機能として Windows Sound System の「テキストリーダ」機能を用いて、求めた最適乗継系列を文章化したものを合成音声で読み上げることも可能であり、また列車の途中駅名と停車時刻も表示可能である。

本システムの探索実行時間に関しては、5.3 節で示す例で分かるとおり、出発地、目的地等の条件によ

て異なるが、第 1 最適乗継系列を 1 つだけ求める場合には、メモリ内データの初期設定に要する時間およびハードディスクからの時刻表データの読み込み時間を含めて通常 3~6 秒、条件の悪い場合でも 20 秒程度で求めることが可能である。

なお、現在本システムでは出発希望時刻を指定する方式をとっており、到着希望時刻の指定はできない。しかし、出発希望時刻を早めに選び探索時間範囲を十分長くとれば実質的に到着希望時刻を指定したときに求めたい最適乗継系列と同じものが得られることになる。

5.3 探索実行例

試作したシステムを用いて実際に最適乗継系列を求めた例を以下に示す。探索に要する計算機時間には探索開始時に毎回必要とされるメモリ内データの初期設定に要する時間と実装されている RAM 容量が少ないため時刻表データをハードディスクから必要な部分を毎回読み込む時間を含んでいる。

例 1 出発地 = 甲府、目的地 = 三沢

出発希望時刻 = 14:00、探索時間範囲 = 9 時間
探索系列数の最大次数 = 2

図 6 (a), (b) に第 1, 2 最適乗継系列を示す。この場合、メモリ内データの初期化を行いディスクから時刻表データを読み込んで動的ネットワークを形成し、探索木を形成して第 1 最適乗継系列が求まる時間は約 4 秒であり、第 2 最適乗継系列まですべて求めるのに要した時間は約 6 秒であった。

例 2 出発地 = 利尻空港、目的地 = 北大東空港

出発希望時刻 = 11:00、探索時間範囲 = 26 時間
探索系列数の最大次数 = 1

図 7 (a), (b) に第 (1, 1), (1, 2) 最適乗継系列

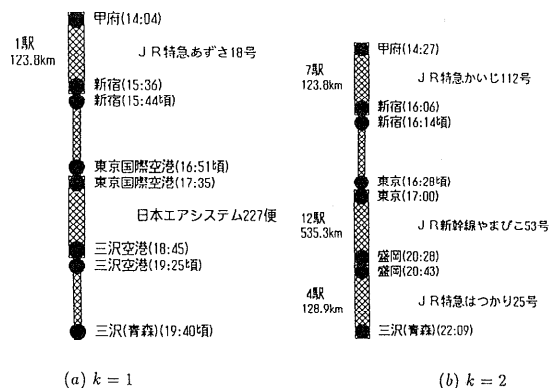
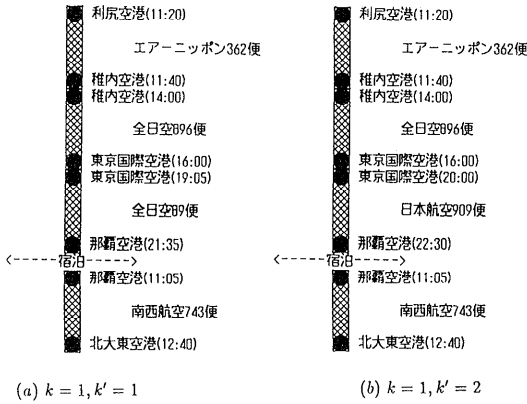
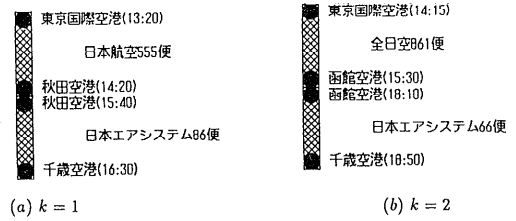


図 6 例 1 (甲府-三沢)

Fig. 6 Example 1.



例 2 (利尻空港-北大東空港)
Fig. 7 Example 2.



例 4 (羽田-千歳)
Fig. 9 Example 4.

探索系列数の最大次数 = 2

ただし、東京国際空港と千歳空港の直行便はすべて欠航という条件を設定した。

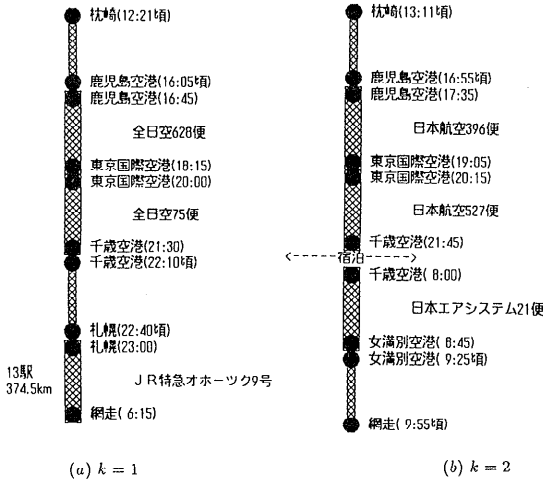
図 9 (a), (b) に第 1, 2 最適乗継系列を示す。列車も探索の対象とした場合の総計算所要時間は約 5 秒、探索の対象を航空機のみ限定すると約 3 秒であった。

5.4 考察

前述の試作システムで用いたパーソナルコンピュータは実装メモリが少なくすべてのデータを RAM 中に常駐させられない。そのため、探索実行時に時刻表データの必要な部分をディスクより読み込んでいるにもかかわらず十分実用的な計算時間で解が求められている。これは動的ネットワーク表現を用いることにより必要にして十分な情報をコンパクトに表現したこと、分岐限定法を用いて効率良く探索木を形成したことによるものである。

次にここで提案した手法のさらなる改善の可能性について考察する。4 章で与えたアルゴリズムでは、探索時間範囲までの動的ネットワークをあらかじめすべて形成してから第 1, 第 2 ... 最適系列を逐次求める探索木を形成していた。しかし、たとえば第 1 最適系列が見い出されれば、第 2 最適系列を探索するにあたり、第 1 最適系列の出発地出発時刻より以前の出発地出発時刻を有する運行リンクは動的ネットワークから取り除くことが可能である。一般に第 k 最適系列を求めるための動的ネットワークは、第 $k-1$ 最適系列をまず求めて、不要になった運行リンクを除去した後形成すれば動的ネットワークを計算機内部で表現するためのメモリ必要量はさらに減少させることができる。

系列探索木形成時における発芽リンクの選択の仕方にも改善の可能性がある。すなわち、4 章で与えたアルゴリズムでは、発芽リンクが複数個あるとき、無作為にそのうちの 1 つを選んで探索木を成長させている。しかし、探索木の形成にあたっては分岐限定法を用い



例 3 (枕崎-網走)
Fig. 8 Example 3.

を示す。この例のような場合には、航空機のみを探索の対象とすれば十分と考えられるが列車の時刻表も探索の対象に入れた場合に要した計算所要時間は約 17 秒であった。探索の対象を航空機のみ限定すると探索所要時間は約 4 秒と当然のことながら大幅な短縮が可能となる。

例 3 出発地 = 枕崎, 目的地 = 網走
出発希望時刻 = 12:00, 探索時間範囲 = 22 時間
探索系列数の最大次数 = 2

図 8 (a), (b) に第 1, 2 最適乗継系列を示す。第 1 最適乗継系列を求めるまでに約 7.5 秒を、第 2 最適乗継系列まで求めるのに約 9.5 秒を要した。

例 4 出発地 = 東京国際空港, 目的地 = 千歳空港
出発希望時刻 = 13:00, 探索時間範囲 = 6 時間

ているので、できるだけ早く良い解が見い出されれば一般に最適解を求めるための手間は削減される。そのため複数の発芽リンクが有る場合には少ない乗換回数、少ない乗車時間を与える可能性の高いもの、たとえば在来線特急より新幹線特急、新幹線特急より航空機を優先的に選ぶことが有利と考えられる。これについては今後さらに検討を加える予定である。

6. む す び

出発地および目的地を指定したとき、列車および航空機の時刻表を考慮に入れて最適な乗継系列を探索する動的最適経路問題の厳密解を与える効率の良い手法を提案した。ここで採用した動的ネットワーク表現は、空間と時間をコンパクトに一体化して表現しうするため計算機のメモリ使用量の節約が可能となる。また動的ネットワーク形成時に導出した出発地出発時刻を有利に活用して最適乗継系列を分岐限定法を用いて効率良く形成することにより、計算時間の短縮化が図られている。

さらにここで提案した手法を日本全国で運行されている航空機、新幹線特急ならびにJR在来線特急に対して適用し、日本全国の9,015の駅ならびに空港のいずれかを出発地あるいは目的地として指定したときに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を用いて試作した。その結果、きわめて実用的な時間で最適解が求めうることを示し、ここで提案した手法の有効性ならびに実用性を明らかにした。

今後の課題としては、現在は目的地にできるだけ早く着くこと、乗換回数を最小とすること、乗車時間を最小とすることを最適乗継系列の定義として用いているが、運賃等の条件を導入することも必要と考えられる。さらには、JR在来線の普通列車ならびに私鉄線の時刻表を考慮に入れたシステムに拡張することが望まれる。しかし、これにはデータ量の関係から時刻表のCD-ROM化を待つ必要がある。また、実際の観点は空席情報の入手も重要であるので、ネットワークを介して座席予約システムと有機的に結合し、実用性の向上を図ることが望まれる。

謝辞 最後に、有益なご助言、ご討論をいただいた本学マルチメディア・ラボの諸氏に謝意を表する。

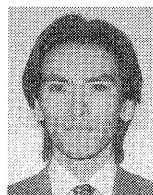
参 考 文 献

- 1) 伊理正夫, 中森真理雄: 算法の最近の進歩, 信学誌, Vol.58, No.4, pp.433-445 (1975).
- 2) 木村幸男, 古賀澄夫: 電話による列車案内システム, 鉄道技術研究報告, No.1157 (1980).

- 3) 尾佐竹尙, 秋山 稔: 交換工学, コロナ社, p.430 (1963).
- 4) 加藤誠巳, 倉部 淳: 論理型言語を用いた航空機の乗継経路探索システム, 第30回情報処理学会全国大会論文集, 6L-5 (1985).
- 5) 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, 第44回情報処理学会全国大会論文集, 4U-4 (1992).
- 6) 半田 哲: スケジュールされた輸送網における動的最短経路問題の表現法, 電気学会論文誌 C, Vol.100, No.10, pp.245-250 (1981).
- 7) 荒屋真二, 曾根 悟: 鉄道における乗客流の一解析法, 計測自動制御学会論文集, Vol.15, No.3, pp.353-359 (1979).
- 8) 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, 情報処理学会情報システム研究会報告, IS15-2 (1987).
- 9) 菊池 新, 加藤誠巳, 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, 情報処理学会情報システム研究会報告, IS48-4 (1994).
- 10) 今田俊明: 人工知能, オーム社, p.248 (1987).
- 11) 杉本克行, 加藤誠巳: 有向ネットワークにおいて閉路を含まない k 個の最短経路を求めるための手法, 情報処理学会論文集, Vol.26, No.2, pp.356-364 (1985).
- 12) 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, 情報処理学会論文集, Vol.33, No.7, pp.970-979 (1992).

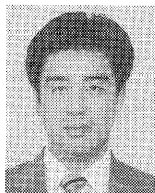
(平成8年3月13日受付)

(平成9年2月5日採録)



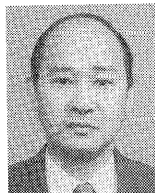
菊池 新 (正会員)

昭和44年生。平成4年上智大学理工学部電気電子工学科卒業。平成6年同大学院博士前期課程修了。同年NTTデータ通信(株)入社。平成7年(株)大西熱学入社。現在、研究開発室において、経路探索システム・遺伝的アルゴリズム・インターネットプログラミング等の研究開発に従事。



高木啓三郎（正会員）

昭和 35 年生。昭和 62 年上智大学理工学部電気電子工学科卒業。平成元年同大学院修士課程修了。同年、NEC に入社、C & C 情報研究所メディアテクノロジ研究部（現情報メディア研究所音声言語研究部）において、音声認識の研究に従事。経路探索アルゴリズム、音声認識アルゴリズムに興味を持つ。日本音響学会会員。



加藤 誠巳（正会員）

昭和 17 年生。昭和 40 年東京大学工学部電子工学科卒業。昭和 42 年同大学院工学系研究科電子工学専門課程修士課程修了。同年日本電信電話公社電気通信研究所入所。東京大学工学部研究生を経て昭和 48 年東京大学大学院工学系研究科電子工学専門課程博士課程修了。工学博士。昭和 48 年上智大学理工学部電気電子工学科講師、昭和 49 年同助教授、昭和 57 年同教授、現在に至る。都市交通情報提供システム、コンピュータ・グラフィックス、ニューラルネット、画像処理、音声信号処理、人工知能等の研究に従事。IEEE、電子情報通信学会、日本オペレーションズ・リサーチ学会、日本音響学会、日本シミュレーション学会、電気学会各会員。