



## 乗車券自動発行システムにおける径路距離計算の一方式\*

石田真也\*\* 影浦秀蔵\*\* 白川 功\*\*\*

### Abstract

In the design of an automatic railroad ticket issuing, of primary importance is an efficient ticket fare calculation scheme, especially when a railroad network is of such a large scale that the conventional fare table method is not available.

The present paper considers an efficient algorithm for calculating the path distances in a weighted graph representing a railroad network, which is applied to the ticket fare calculation scheme for a railroad ticket issuing system developed by Kinki Nippon Railway Co., Ltd. This algorithm is distinctive mainly in that for a specified spanning tree, a given path is decomposed into the tree path portion and the links on it, for which a simplified distance calculation scheme is accessible.

### 1. ま え が き

鉄道における各種乗車券発行を自動化する<sup>1)</sup>に際して、

(i) 与えられた乗車径路を計算機へいかに効率良く入力するか、

(ii) 入力された乗車径路データ(符号)に基づき計算機内でいかに能率良くその運賃計算を行うか、という重要な問題が生じる。

一般に鉄道における乗車径路は、同一駅を2度通過しない径路に限定され、その運賃は乗車距離により決定されることから、これらの問題は、対象路線を重み付きグラフと考えるならば、

(i) グラフ上の道を効率良く符号化する、

(ii) 道の符号よりその道の重み(距離)を能率良く算出する、という問題になる。

このように与えられた乗車径路をグラフ上の一つの道としていかに効率良く表現するかという問題に対し

ては、すでに定期乗車券の自動改札システムにおける径路の符号化の問題に関連して、いくつかの手法が提案され、実用化されている<sup>2),3)</sup>。文献2)に示されている手法においては道はその両端に節点と通過する補木を指定することにより簡潔に符号化され、上記の(i)の問題に対して一つの有効な手段を提供するものであると考えられる。

本文は、この手法による道の符号より効率的に重み(距離)を算出する方法について述べるものであり、文献2)における道の符号化方式が(ii)の問題に対しても効率的な手法であることを示すものである。

### 2. 基本的考察

駅を節点に、駅間を結ぶ路線を枝に対応させたグラフを鉄道網といい、これを $G$ で表わす。以下では一般性を失わずに鉄道網は連結であると仮定する。与えられた鉄道網 $G$ の節点のうち、線度1の節点を $G$ の端点、線度3以上の節点を $G$ の分岐点と呼ぶ。 $G$ の枝 $e$ が節点 $v_i, v_j$ を両端点とするとき、 $e=(v_i, v_j)$ で表わす。さらに、枝の系列 $[(v_1, v_2), (v_2, v_3), \dots, (v_{n-1}, v_n)]$ を辺列といい、 $v_i \neq v_n$ のとき、これを開辺列といい、 $v_1 = v_n$ のとき、これを閉辺列という<sup>4)</sup>。同一節点を2度以上通らない開辺列、閉辺列をそれぞれ、道、閉路と呼ぶ。 $G$ の各枝 $(v_i, v_j)$ に正の重み $l(v_i, v_j)$

\* An Algorithm for Calculating Route Distances in Automatic Railroad Ticket Issuing System by Shinya ISHIDA, Shuzo KAGEURA (Research Laboratory, Kinki Nippon Railway Co., Ltd.), and Isao SHIRAKAWA (Faculty of Engineering, Osaka University).

\*\* 近畿日本鉄道(株)技術研究所

\*\*\* 大阪大学工学部電子工学科

が与えられているものとし、これをその枝の距離という。また、辺列  $[(v_1, v_2), (v_2, v_3), \dots, (v_{n-1}, v_n)]$  の距離を、

$$l[(v_1, v_2), (v_2, v_3), \dots, (v_{n-1}, v_n)] \\ \triangleq l(v_1, v_2) + l(v_2, v_3) + \dots + l(v_{n-1}, v_n) \quad (2.1)$$

で表わす。

$G$  のすべての節点を含み、しかも閉路を含まない連結部分グラフ  $T$  を  $G$  の木という。与えられた木  $T$  の上では、任意の節点  $v_i, v_j$  の間の道は一意的に定まり、これを木道と呼び  $P_T(v_i, v_j)$  で表わす。また、 $G$  の木  $T$  の補木を  $C_T$  と表わし、 $C_T$  に含まれる枝を特にリンクといい、 $c_i (i=1, 2, \dots, \mu)$  で表わす。

グラフ  $G$  の任意の道  $P_i$  は、その道の両端の節点  $v_0, v_D$  (発駅と着駅) と、その道の通るリンク  $c_{i_1}, c_{i_2}, \dots, c_{i_n}$  を指定することにより、一意的に決定され<sup>2)</sup>、したがって、そのような道を、

$$P_i = P(v_0, v_D; c_{i_1}, c_{i_2}, \dots, c_{i_n}) \quad (2.2)$$

で表わすことにする。

いま、この道  $P_i$  に対して、その端点  $v_0, v_D$  および各リンク  $c_{i_k}$  の端点の集合を  $Z_i = \{z_i\}, (|Z_i| = 2n + 2)$  とする\*。このとき  $P_i$  が通る順に  $Z_i$  の節点を並べてできる系列を、

$$\tilde{Z} = [\tilde{z}_1 (= v_0), \tilde{z}_2, \dots, \tilde{z}_{2n+2} (= v_D)] \quad (2.3)$$

とすれば、 $P_i$  を  $T$  上の木道と  $C_T$  のリンクとを交互に連ねた系列

$$P_i = [P_T(\tilde{z}_1, \tilde{z}_2), (c_2, \tilde{z}_3), P_T(\tilde{z}_3, \tilde{z}_4), \dots, \\ P_T(\tilde{z}_{2n+1}, \tilde{z}_{2n+2})] \quad (2.4)$$

の形に分解して表現することができる。ここで  $P_T(\tilde{z}_1, \tilde{z}_2), P_T(\tilde{z}_3, \tilde{z}_4), \dots, P_T(\tilde{z}_{2n+1}, \tilde{z}_{2n+2})$  は互いに節点を共有しない木道であることに注意しよう。したがって、鉄道網  $G$  の任意の道を式(2.2)のように、その両端点と通過するリンクの集合によって指定することとすれば、その距離の算出は、

1° 道  $P_i$  の両端と、通過するリンクの集合で指定される節点の集合  $Z_i$  を式(2.3)のように道の通る順序に並べかえる。

2° 並べかえられた節点系列  $\tilde{Z}_i$  をもとに道  $P_i$  を木道と、リンクとに分解し、それぞれの距離を算出する。

この手順に従って行うことができる。このうち、2° は後で述べるように比較的簡単に行うことができる。1° に

ついては、実際には、道  $P_i$  について与えられている情報はその両端  $v_0, v_D$  とリンク  $c_{i_1}, \dots, c_{i_n}$  のみであり、これらの情報のみに基づいて  $Z_i$  を順序付けねばならないという点に問題があり、そのため、1° に替わるものとし、次の手順を考えこととする。

(I) 集合  $Z_i$  を  $n+1$  個の節点の対に分割し、かつ各対の節点を両端とする木道が互いに節点を共有しないようにする。

(II) (I) で得られた木道をリンクにより結ぶことにより式(2.4)のように道を構成する。

節点集合  $Z_i$  が1つの道  $P_i$  により決定される限り、(I), (II) の手順により構成される道は必ず存在するから、この手順により得られた道がもとの道  $P_i$  と一致することを保障するためには、これらの手順による道の構成が一意的であることが示されれば十分である。また実際の応用に際しては、操作の誤りなどに起因し、指定された両端点とリンクが道を構成し得ない場合も生じ、距離計算においては、指定された両端点とリンクが実際に道を構成するか否かの判定も合せて行う必要がある。こうした点に留意しつつ、以下(I)の分割に注目し、その一意性と具体的な分割手法について述べることにする。

### 3. 木道分解

一般に、与えられた相異なる偶数個の節点の集合  $U = \{u_1, u_2, \dots, u_{2m}\}$  が、 $m$  個の節点の対  $(u_{11}, u_{12}), (u_{21}, u_{22}), \dots, (u_{m1}, u_{m2})$  に分割され、かつそれぞれの対により指定される木道  $P_T(u_{11}, u_{12}), P_T(u_{21}, u_{22}), P_T(u_{m1}, u_{m2})$  が互いに節点を共有しないとき、木  $T$  が  $U$  によりこれらの木道に分解されるといい、このような  $U$  による木道への分解を  $U$  の木道分解という。

[定理1] 木  $T$  の任意の偶数個の節点の集合  $U = \{u_1, u_2, \dots, u_{2m}\}$  の木道分解は一意的である。

(証明) 2つの相異なる分解が存在すると仮定する。いま、一方の分解に対応する木道の集合を  $Q_A$ 、他方のそれを  $Q_B$  とし、

$$Q_A - Q_B = \{P_T(u_{a1}, u_{a2}), P_T(u_{a3}, u_{a4}), \dots\} \quad (3.1)**$$

$$Q_B - Q_A = \{P_T(u_{b1}, u_{b2}), P_T(u_{b3}, u_{b4}), \dots\} \quad (3.2)$$

とおく。 $Q_A - Q_B, Q_B - Q_A$  は仮定より空集合ではなく、いま  $Q_A - Q_B$  の1つの道、たとえば  $P_T(u_{a1}, u_{a2})$  について考えると、 $Q_A$  は節点を互いに共有しない道の集合であるから、 $Q_A$  には  $P_T(u_{a1}, u_{a2})$  以外にもはや  $u_{a1}$ 、あるいは  $u_{a2}$  はを端点としてもつような道は存

\*  $|Z_i|$  は  $Z_i$  の元の個数を表わす。また厳密には  $Z_i$  はいわゆる multi-set である。

\*\* 集合  $X, Y$  に対して  $X - Y \triangleq \{x | x \in X, x \notin Y\}$

在しない。一方、 $P_T(u_{a_1}, u_{a_2}) \in \Omega_A - \Omega_B$  は  $P_T(u_{a_1}, u_{a_2}) \notin \Omega_B$ , すなわち、 $\Omega_B$  には  $u_{a_1}, u_{a_2}$  を端点としてもつような  $P_T(u_{a_1}, u_{a_2})$  以外の2つの道が存在することを意味する。従って  $\Omega_B - \Omega_A$  の中に  $u_{a_1}, u_{a_2}$  のそれぞれを端点としてもつ、 $P_T(u_{a_1}, u_{a_2})$  以外の2つの道が必ず存在することになる。 $\Omega_B - \Omega_A$  の任意の道についても同様に、その両端の節点のおのおのを端点とする2つの道が  $\Omega_A - \Omega_B$  に含まれる。このようにして順次  $\Omega_A - \Omega_B, \Omega_B - \Omega_A$  の道で、端点が互いに一致するものを交互に連ねることにより、一つの閉辺列を構成することができる。 $\Omega_A - \Omega_B$  (あるいは  $\Omega_B - \Omega_A$ ) は互いに節点を共有しない道の集合であるから、この閉辺列中に同一の枝が2度現われるならば、その枝は  $\Omega_A - \Omega_B$  と  $\Omega_B - \Omega_A$  の双方に同時に含まれる枝である。従ってもし全ての枝が2度現われるならば、 $\Omega_A - \Omega_B$  (または  $\Omega_B - \Omega_A$ ) のみに属する道の枝のみでこの閉辺列を含むことになり、 $\Omega_A - \Omega_B$  (または  $\Omega_B - \Omega_A$ ) が互いに節点を共有しない道の集合であることに反する。いいかえればこの閉辺列の中には1度しか現われない枝が含まれていることになるが、これも  $T$  が木であることに矛盾する。(証明終り)

次に木道分解の具体的方法を述べるが、まずその準備として基本的な2,3の事柄を考察しておく。

任意の1節点  $v_i$  を選び、 $T$  上で  $v_i$  から  $T$  の各端点に向うようにすべての木枝に方向付けを行うことによって有向木を作ることができる。このようにして作られた有向木を  $\vec{T}(v_i)$  で表わし、 $v_i$  をその木の根という。 $\vec{T}(v_i)$  上で任意の2節点  $v_i, v_j$  に対し、 $v_j = v_i$  であるか、または  $v_i$  から枝の方向に沿って  $v_j$  に到達できるとき  $v_j$  は  $v_i$  の下位にあるといい  $v_i \rightarrow v_j$  で表わし、 $\hat{F}(v_i) \triangleq \{v_j | v_i \rightarrow v_j\}$  とおく、また、 $v_j$  が  $v_i$  の下位にないとき、すなわち、 $v_j \notin \hat{F}(v_i)$  のとき、 $v_i \rightarrow v_j$  で表わすことにする。さらに、 $\vec{T}(v_i)$  上の任意の2節点  $v_i, v_j$  に対して、 $P_T(v_i, v_j), P_T(v_j, v_i)$  に共通に含まれる節点のうちで、最も下位にある節点を  $v_i, v_j$  の合流点ということにし、 $\xi(v_i, v_j)$  で表わす。ここで、 $v_i \rightarrow v_j$ , または  $v_j \rightarrow v_i$  であれば、 $\xi(v_i, v_j)$  はそれぞれ  $v_i$  または  $v_j$  に一致し、そうでなければ  $\xi(v_i, v_j)$  は二つの分岐点となっていることに注意しよう。

さて、 $T$  の任意の相異なる偶数個の節点の集合  $U = \{u_1, u_2, \dots, u_{2m}\}$  の木道分解が存在するとすれば、次の定理が成立する。

【定理2】 偶数個の節点の集合  $U = \{u_1, u_2, \dots, u_{2m}\}$

の木道分解によって得られる  $m$  個の木道の集合を

$$\Omega = \{P_T(u_{i_1}, u_{i_1}), P_T(u_{i_2}, u_{i_2}), \dots, P_T(u_{i_{2m-1}}, u_{i_{2m}})\} \quad (3.3)$$

とすれば、 $\Omega$  の中の一つの道  $P_T(u_{i_j}, u_{i_{j+1}})$  で  $\hat{F}(\xi(u_{i_j}, u_{i_{j+1}}))$  が  $u_{i_j}, u_{i_{j+1}}$  以外の  $U$  の節点を含まないものが必ず存在する。

(証明) このような道が存在しないと仮定し矛盾を導く。いま、木道分解して得られた式(3.3)の  $\Omega$  に対して合流点の集合

$$E \triangleq \{\xi(u_{i_1}, u_{i_1}), \xi(u_{i_2}, u_{i_2}), \dots, \xi(u_{i_{2m-1}}, u_{i_{2m}})\} \quad (3.4)$$

を定義すると、これらの各合流点は互いに相異なる。この  $E$  においては、すべての合流点に対してそれ自身以外の  $E$  の合流点をその下位節点として含むことはできない。いいかえれば、 $E$  の合流点  $\xi(u_{i_j}, u_{i_{j+1}})$  でその下位節点集合  $\hat{F}(\xi(u_{i_j}, u_{i_{j+1}}))$  に自分自身以外の  $E$  の合流点を含まないものが必ず存在する。このような  $u_{i_j}, u_{i_{j+1}}$  に対する木道  $P_T(u_{i_j}, u_{i_{j+1}})$  に対しても、仮定より、その下位節点集合は、 $u_{i_j}, u_{i_{j+1}}$  以外の  $U$  の節点を含む。つまり  $\Omega$  の2つの道  $P_T(u_{i_j}, u_{i_{j+1}}), P_T(u_{i_k}, u_{i_{k+1}})$  ( $j \neq k$ ) で  $u_{i_k} \in \hat{F}(\xi(u_{i_j}, u_{i_{j+1}}))$ ,  $\xi(u_{i_k}, u_{i_{k+1}}) \in \hat{F}(\xi(u_{i_j}, u_{i_{j+1}}))$  なるものが存在する。このことは、すべての  $\Omega$  の道は節点を共有しないという仮定に反している。(証明終り)

このようにして得られた節点の対  $(u_{i_j}, u_{i_{j+1}})$  を  $U$  から取り除き、木道  $P_T(u_{i_j}, u_{i_{j+1}})$  を  $\Omega$  から取り除いてできる新しい節点对、木道の集合をそれぞれ  $U', \Omega'$  とするならば、 $\Omega'$  は  $U'$  の木道分解によって得られる木道の集合となる。従って、 $U', \Omega'$  についてやはり定理2が成立することになる。このようにして、 $U, \Omega$  に対してこの操作を順次繰返し適用することにより、次の定理が成立することがわかる。

【定理3】 偶数個の節点の集合  $U = \{u_1, u_2, \dots, u_{2m}\}$  の木道分解によって得られた  $m$  個の木道の集合を  $\Omega$  とし、 $\Omega$  の木道を上記操作によって取り除かれる節点对の順に添字を並べかえ、改めて

$$\Omega = \{P_T(u_{i_1}, u_{i_1}), P_T(u_{i_2}, u_{i_2}), \dots, P_T(u_{i_{2m-1}}, u_{i_{2m}})\} \text{ と表わすならば}$$

$$\left. \begin{aligned} u_{i_1}, u_{i_1}, \dots, u_{i_{2m-1}}, u_{i_{2m}} &\in \hat{F}(\xi(u_{i_1}, u_{i_1})) \\ u_{i_2}, u_{i_2}, \dots, u_{i_{2m-1}}, u_{i_{2m}} &\in \hat{F}(\xi(u_{i_2}, u_{i_2})) \\ \dots &\dots \\ u_{i_{2m-1}}, u_{i_{2m}} &\in \hat{F}(\xi(u_{i_{2m-1}}, u_{i_{2m-1}})) \end{aligned} \right\} \quad (3.5)$$

が成立する。(証明略)

$U$  の2節点  $u_p, u_q$  について、 $\hat{F}(\xi(u_p, u_q))$  が、 $u_p,$

$u_q$  以外の  $U$  の節点を含まないなら,  $P_T(u_p, u_q)$  は  $U$  の  $u_p, u_q$  以外の任意の 2 節点を結んでできる道と節点を共有しない. 従って節点集合より式 (3.5) を満足するように順序付けられた節点を作ることができるならば, そのような各節点対に対応する木道はすべて互いに節点を共有しないことになる. 従って次の定理が得られる.

**【定理 4】** 任意の偶数個の節点集合  $U$  が与えられたとき,  $U$  の木道分解が可能であるための, 必要十分条件は,  $U$  が式 (3.5) を満たすような節点対  $(u_{i_1}, u_{i_1}), (u_{i_2}, u_{i_2}), \dots, (u_{i_{2m-1}}, u_{i_{2m-1}})$  に分解されることである. (証明略)

定理 2~4 により, 節点集合  $U = \{u_i\}$  の木道分解は  $U$  の節点対  $u_p, u_q$  で  $\hat{f}(\xi(u_p, u_q))$  が  $u_p, u_q$  以外の節点を含まないものを見つけ出す操作の繰返しに帰着される. そこでこの操作を能率良く行うため, まず  $T$  の各節点  $v_i$  に対して節点番号  $N(v_i)$  を次のような深さ優先技法 (DFS)<sup>5)</sup> の手順により与える.

- 0° スタック  $S$  を空にし,  $k \leftarrow 0$  とする.
- 1° 木  $\bar{T}$  の根  $v_r$  を  $S$  に入れる.
- 2°  $S$  が空であれば操作終了, そうでなければ  $k \leftarrow k+1$  とする.
- 3°  $S$  の最上段にある節点の下位に隣接する節点があれば 4° へ, なければ 5° へ行く.
- 4°  $S$  の最上段にある節点の下位に隣接する節点を (複数個存在するならそのうちの任意の一つを)  $S$  に入れ, 3° へ戻る.
- 5°  $S$  より (最上段の) 節点を取り去り, かつその節点に番号  $k$  を与えて 2° へ戻る.

このようにして各節点  $v_i$  に番号  $N(v_i)$  を割り当てた後に, この  $N(v_i)$  の順に節点を並びかえてできる系列を

$$\bar{V} = [\bar{v}_1, \bar{v}_2, \dots, \bar{v}_i], \quad (i \leq G \text{ の節点の個数}) \quad (3.6)$$

とする. このとき, 節点番号の付け方から明らかなように次の性質が成り立つ.

- (性質 1)<sup>5)</sup>  $\bar{T}$  上で  $\bar{v}_i \rightarrow \bar{v}_j$  であれば,  $j \leq i$  である.
  - (性質 2)<sup>5)</sup>  $\bar{T}$  上で  $\bar{v}_i \rightarrow \bar{v}_j$  であれば,  $j \leq k \leq i$  なるどの  $k$  に対しても  $\bar{v}_i \rightarrow \bar{v}_k$  である.
- また,  $\xi(\bar{v}_i, \bar{v}_j) \rightarrow \bar{v}_i, \bar{v}_j$  と性質 1, 2 より次の性質 3 が導かれる.
- (性質 3)  $i \leq k \leq j$  なるどの  $k$  に対しても  $\xi(\bar{v}_i, \bar{v}_j) \rightarrow \bar{v}_k$  である.

この性質 3 を用いて次の定理が証明される.

**【定理 5】** 2 つの節点  $\bar{v}_i, \bar{v}_j (i < j)$  の合流点  $\xi(\bar{v}_i, \bar{v}_j)$  に対して,

- (i) ある  $\bar{v}_k (k < i)$  が  $\xi(\bar{v}_i, \bar{v}_j) \rightarrow \bar{v}_k$  であれば,  $\xi(\bar{v}_i, \bar{v}_j) \rightarrow \bar{v}_1, \bar{v}_2, \dots, \bar{v}_{k-1}$  であり,
- (ii) ある  $\bar{v}_l (j < l)$  が  $\xi(\bar{v}_i, \bar{v}_j) \rightarrow \bar{v}_l$  であれば,  $\xi(\bar{v}_i, \bar{v}_j) \rightarrow \bar{v}_{l+1}, \bar{v}_{l+2}, \dots, \bar{v}_i$  である.

(証明) まず (i) を証明する. ある  $h (1 \leq h \leq k-1)$  に対して,  $\xi(\bar{v}_i, \bar{v}_j) \rightarrow \bar{v}_h$  であるとすれば,  $\hat{f}(\xi(\bar{v}_h, \bar{v}_j)) \subset \hat{f}(\xi(\bar{v}_i, \bar{v}_j))$  であるが,  $\bar{v}_h \notin \hat{f}(\xi(\bar{v}_i, \bar{v}_j))$  より  $\bar{v}_h \notin \hat{f}(\xi(\bar{v}_h, \bar{v}_j))$ , すなわち  $\xi(\bar{v}_h, \bar{v}_j) \rightarrow \bar{v}_h$  となり, これは性質 3 に矛盾する. よって (i) が成立する. 次に (ii) を証明する. ある  $p (l+1 \leq p \leq i)$  に対して,  $\xi(\bar{v}_i, \bar{v}_j) \rightarrow \bar{v}_p$  であるとすれば,  $\hat{f}(\xi(\bar{v}_i, \bar{v}_p)) \subset \hat{f}(\xi(\bar{v}_i, \bar{v}_j))$  となるが,  $\bar{v}_i \notin \hat{f}(\xi(\bar{v}_i, \bar{v}_j))$  から  $\bar{v}_i \notin \hat{f}(\xi(\bar{v}_i, \bar{v}_p))$ , すなわち  $\xi(\bar{v}_i, \bar{v}_p) \rightarrow \bar{v}_i$  となりこれも性質 3 に矛盾する. (証明終り)

そこで, いま節点集合  $U (|U| = 2m)$  の各節点をその節点番号順に並べてできる系列

$$\bar{U} = [\bar{u}_1, \bar{u}_2, \dots, \bar{u}_{2m}] \quad (3.7)$$

に注目する. このとき性質 3 により, 式 (3.5) の第 1 式を満たすような  $U$  の 2 節点の対  $(u_{i_1}, u_{i_2})$  を見出すためには, 式 (3.7) の系列において, 相隣り合う各節点対  $(\bar{u}_i, \bar{u}_{i+1}), (i = 1, 2, \dots, 2m-1)$  について  $\xi(\bar{u}_i, \bar{u}_{i+1})$  が  $\bar{u}_i, \bar{u}_{i+1}$  以外の  $2m-2$  個の節点を下位集合  $\hat{f}(\xi(\bar{u}_i, \bar{u}_{i+1}))$  に含むかどうかを調べればよいということがわかる. ところが定理 5 により, このためには,  $\hat{f}(\xi(\bar{u}_i, \bar{u}_{i+1}))$  が,  $\bar{u}_{i-1}, \bar{u}_{i+2}$  を含むかどうかをそれぞれ調べるだけで十分であるということが保障される. よって式 (3.5) の第 1 式を満たす節点対は, Fig. 1 で示されるような手続きを適用することによって求められる. このような対  $(u_{i_1}, u_{i_2})$  が求まると,  $U$  から  $u_{i_1}, u_{i_2}$  を除去して, 同様の手続きを適用することに

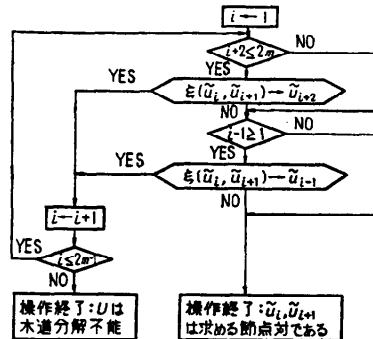


Fig. 1 Fundamental process for tree decomposition.

より、式(3.5)の第2式を満たす  $(u_i, u_i)$  が求まる。同様の手続きを繰り返して、 $(u_i, u_i), \dots, (u_{i2m-1}, u_{i2m})$  が順次求められる。従って  $U$  の木道分解を実行する上で残された問題は、 $\bar{U}$  の3節点  $\bar{u}_p, \bar{u}_q, \bar{u}_r$  に対して  $\xi(\bar{u}_p, \bar{u}_q) \rightarrow \bar{u}_r$ 、または  $\xi(\bar{u}_p, \bar{u}_q) \rightarrow \bar{u}_r$  の判定をどのようにして行うかということになる。

4. 節点の符号化と木道分解のアルゴリズム

木道分解を能率良く実行するため、各節点に対して新しい節点コード  $\gamma(v_i)$  を次式

$$\gamma(v_i) = \begin{matrix} \alpha(v_i) \\ \beta(v_i) \end{matrix} \quad (4.1)$$

のように2つのコード—分岐コード  $\alpha(v_i)$ 、順序コード  $\beta(v_i)$ —で構成する。以下に、 $\alpha(v_i), \beta(v_i)$  の符号について述べる。

(a) 分岐コード  $\alpha(v_i)$

$\bar{T}$  の根  $v_r$  以外の端点および分岐点に対して、節点番号の昇順に左から(上位ビットから)1ビットずつ割当て、各節点  $v_i$  の分岐コードとしてそれらの各ビットを次のように定める。

(i) 根  $v_r$  以外の端点または分岐点である各節点  $v_i$  に対しては、 $v_r$  から  $v_i$  に至る木道に含まれる分岐点、端点に対応する各ビットに0を与え、他のすべてのビットを1とする。

(ii) 根  $v_r$ 、端点、分岐点以外の各節点  $v_i$  に対しては、 $v_r$  から  $v_i$  を通り、 $v_i$  の下位にある最初の分岐点または端点に至る木道に含まれる各分岐点、端点に対応するビットに0を与え、他はすべて1とする。

(iii) 根  $v_r$  に対してはすべてのビットを1とする。

(b) 順序コード  $\beta(v_i)$

$\bar{T}$  上の分岐点または端点間の道で、途中に分岐点を含まないようなもののなかで、節点個数が最大なものを見出し、その個数を  $\nu$  とするとき、 $\beta(v_i)$  として、 $\lceil \log_2(\nu-1) \rceil$  個のビットを割り当て(ただし、 $\lceil x \rceil$  は  $x$  より小でない最小の整数を表す)、各ビットの値を以下のように定める。

(i) 根  $v_r$ 、分岐点、または端点である各節点  $v_i$  に対しては  $\beta(v_i)$  のすべてのビットを0とする。

(ii) それ以外の各節点  $v_i$  に対しては、 $v_r$  から  $v_i$  の下位にある最初の分岐点または端点に至る木道に含まれる( $v_i$  を含め、分岐点または端点を除いた)節点数を表す2進コードを  $\beta(v_i)$  に割り当てる。

以上のように構成された節点コード  $\gamma(v_i)$  は、次に述べるようないくつかの基本的性質を備えている。

(性質1) 節点番号  $N(v_i)$  によって作られた節点

系列(3.6)と節点コード  $\gamma(v_i)$  により作られる節点系列とは一致する。すなわち式(3.6)の系列の任意の2節点  $\bar{v}_i, \bar{v}_j$  について  $\gamma(\bar{v}_i), \gamma(\bar{v}_j)$  を2進符号として比べると

$$i < j \text{ ならば } \gamma(\bar{v}_i) < \gamma(\bar{v}_j)$$

が成立する。

また、 $\alpha(v_i) = [\alpha_1^i, \alpha_2^i, \dots, \alpha_p^i]$ 、 $\alpha(v_j) = [\alpha_1^j, \alpha_2^j, \dots, \alpha_p^j]$  に対して

$$\alpha(v_i) \vee \alpha(v_j) \triangleq [\alpha_1^i \vee \alpha_1^j, \alpha_2^i \vee \alpha_2^j, \dots, \alpha_p^i \vee \alpha_p^j]$$

とする。ただし、右辺の“ $\vee$ ”は論理和を表す。さらに、 $\alpha(v_i)$  が0をもつ各ビットにおいて  $\alpha(v_j)$  も0をもつとき、 $\alpha(v_i) \supset \alpha(v_j)$  と表すならば、次の性質2, 3, 4が成立する。

【性質2】 任意の2節点  $v_i, v_j$  に対して  $\xi(v_i, v_j)$  が分岐点であるならば、その分岐点は、分岐コード  $\alpha(\xi(v_i, v_j))$  において最上位の0のビット位置に対応する分岐点と一致する。

【性質3】  $v_i \rightarrow v_j (v_j \in \bar{T}(v_i))$  であるのは、

$$(i) \alpha(v_i) = \alpha(v_j), \text{ かつ } \beta(v_i) \geq \beta(v_j) \quad (4.2)$$

$$(ii) \alpha(v_i) \neq \alpha(v_j), \text{ かつ } \alpha(v_i) \supset \alpha(v_j) \quad (4.3)$$

のいずれかが成立するとき、かつそのときに限る。

【性質4】 任意の2節点  $v_i, v_j$  およびそれらの合流点  $\xi(v_i, v_j)$  に対して次式が成立する。

$$\alpha(\xi(v_i, v_j)) = \alpha(v_i) \vee \alpha(v_j) \quad (4.4)$$

性質1は新しい節点コード  $\gamma(v_i)$  によって作られた節点系列によっても Fig. 1 に示すアルゴリズムがそのまま適用できることを保障し、性質2は後で述べる木道の距離計算を行う上で重要な役割を果たす。さらに性質3, 4により、Fig. 1 における、 $\xi(\bar{u}_i, \bar{u}_{i+1}) \rightarrow \bar{u}_{i+2}$ 、 $\xi(\bar{u}_i, \bar{u}_{i+1}) \rightarrow \bar{u}_{i-1}$  の判定を以下のようにして行うことができる。

(I)  $\xi(\bar{u}_i, \bar{u}_{i+1}) \rightarrow \bar{u}_{i+2}$  の判定

この判定は性質3の式(4.2), (4.3)に対応して

$$\alpha(\xi(\bar{u}_i, \bar{u}_{i+1})) = \alpha(\bar{u}_{i+2}) \text{ かつ}$$

$$\beta(\xi(\bar{u}_i, \bar{u}_{i+1})) \geq \beta(\bar{u}_{i+2}) \quad (4.5)$$

$$\alpha(\xi(\bar{u}_i, \bar{u}_{i+1})) \neq \alpha(\bar{u}_{i+2}) \text{ かつ}$$

$$\alpha(\xi(\bar{u}_i, \bar{u}_{i+1})) \supset \alpha(\bar{u}_{i+2}) \quad (4.6)$$

のいずれかが成立するかどうかを調べればよい。特に式(4.5)に注目すると、合流点  $\xi(\bar{u}_i, \bar{u}_{i+1})$  が分岐点でないならば、 $\xi(\bar{u}_i, \bar{u}_{i+1})$  は  $\bar{u}_{i+1}$  と一致し、 $\alpha(\xi(\bar{u}_i, \bar{u}_{i+1})) = \alpha(\bar{u}_{i+1})$ 、 $\beta(\bar{u}_i, \bar{u}_{i+1}) = \beta(\bar{u}_{i+1})$  が成立する。従って式(4.5)の判定は、 $\alpha(\bar{u}_{i+1}) = \alpha(\bar{u}_{i+2})$  かつ  $\beta(\bar{u}_{i+1}) \geq \beta(\bar{u}_{i+2})$  の判定となるが、系列  $\bar{U}$  の作り方より、この判定の第1式、第2式が同時に成立することはなく、

従ってこの場合は式(4.5)の判定は必要でないことになる。また合流点  $\xi(\tilde{u}_i, \tilde{u}_{i+1})$  が分岐点であるならば、 $\beta(\tilde{u}_{i+1}, \tilde{u}_{i+1})=0$  であるから式(4.5)第2式の判定は  $\beta(v_{i+2})=0$  の判定に帰着され、従って式(4.5)の判定に替え

$$\alpha(\xi(\tilde{u}_i, \tilde{u}_{i+1})) = \alpha(\tilde{u}_{i+2}) \text{ かつ } \beta(\tilde{u}_{i+2}) = 0 \quad (4.7)$$

の判定を行えば十分である。

(II)  $\xi(\tilde{u}_i, \tilde{u}_{i+1}) \rightarrow \tilde{u}_{i-1}$  の判定

この判定も性質3の式(4.2), (4.3)に対応して

$$\alpha(\xi(\tilde{u}_i, \tilde{u}_{i+1})) = \alpha(\tilde{u}_{i-1}) \text{ かつ } \beta(\xi(\tilde{u}_i, \tilde{u}_{i+1})) \geq \beta(\tilde{u}_{i-1}) \quad (4.8)$$

$$\alpha(\xi(\tilde{u}_i, \tilde{u}_{i+1})) \neq \alpha(\tilde{u}_{i-1}) \text{ かつ } \alpha(\xi(\tilde{u}_i, \tilde{u}_{i+1})) \supset \alpha(\tilde{u}_{i-1}) \quad (4.9)$$

のいずれかが成立するかどうかの判定となる。特に、式(4.8)に注目すると、系列  $\tilde{U}$  の作り方より  $\alpha(\xi(\tilde{u}_i, \tilde{u}_{i+1})) = \alpha(\tilde{u}_{i-1})$  ならば  $\alpha(\tilde{u}_i) = \alpha(\tilde{u}_{i+1}) = \alpha(\tilde{u}_{i-1})$  が成立し、 $\xi(\tilde{u}_i, \tilde{u}_{i+1}) = \tilde{u}_{i+1}$ 、 $\beta(\xi(\tilde{u}_i, \tilde{u}_{i+1})) = \beta(\tilde{u}_{i+1})$  となるから、 $\beta(\xi(\tilde{u}_i, \tilde{u}_{i+1})) \geq \beta(\tilde{u}_{i-1})$  は常に成立する。従って式(4.8)第2式の判定は無視でき、式(4.8), (4.9)の判定は式(4.9)第2式の判定のみに帰着される。

以上の考察と性質4から、Fig. 1の処理手続きは、 $\tilde{U}$  の各節点コード  $\gamma(\tilde{u}_i)$  を用いれば Fig. 2 のように表わされる。

### 5. 木道の距離計算

木道の分解により得られた  $P_T(\tilde{v}_i, \tilde{v}_j)$ , ( $i > j$ ) の距離  $l[P_T(\tilde{v}_i, \tilde{v}_j)]$  は

$$l[P_T(\tilde{v}_i, \tilde{v}_j)] = \begin{cases} l[P_T(v_i, \tilde{v}_j)] - l[P_T(\tilde{v}_i, \tilde{v}_i)]: & \tilde{v}_i \rightarrow \tilde{v}_j \text{ のとき} \\ l[P_T(v_i, \tilde{v}_i)] + l[P_T(v_i, \tilde{v}_j)] - 2l[P_T(v_i, \xi(\tilde{v}_i, \tilde{v}_j))]: & \tilde{v}_i \rightarrow \tilde{v}_j \text{ のとき} \end{cases} \quad (5.1)$$

で与えられる。従って、 $T$  のすべての節点  $\tilde{v}_i$  に対して距離  $l[P_T(v_i, \tilde{v}_i)]$  を与えておこなうならば、 $T$  上の任意の2節点  $\tilde{v}_i, \tilde{v}_j$  ( $i > j$ ) の距離の算出は、 $\tilde{v}_i \rightarrow \tilde{v}_j$  かどうかの判定と、 $v_i$  から合流点  $\xi(\tilde{v}_i, \tilde{v}_j)$  への距離の算出に帰着される。 $\tilde{v}_i \rightarrow \tilde{v}_j$  の判定は、先に述べた節点コードに関する性質3により容易に行える。一方、 $\tilde{v}_i \rightarrow \tilde{v}_j$  ( $i > j$ ) の場合、 $v_i$  から合流点  $\xi(\tilde{v}_i, \tilde{v}_j)$  への距離算出については、節点コードに関する性質2から、合流点  $\xi(\tilde{v}_i, \tilde{v}_j)$  が一つの分岐点  $\tilde{v}_k$  として容易に求められ、 $l[P_T(v_i, \xi(\tilde{v}_i, \tilde{v}_j))]$  はこの  $\tilde{v}_k$  に対して  $l[P_T(v_i, \tilde{v}_k)]$  によって与えられる。従って、任意の節点对  $\tilde{v}_i, \tilde{v}_j$  間の木道  $P_T(\tilde{v}_i, \tilde{v}_j)$  の距離は Fig. 3 のような手

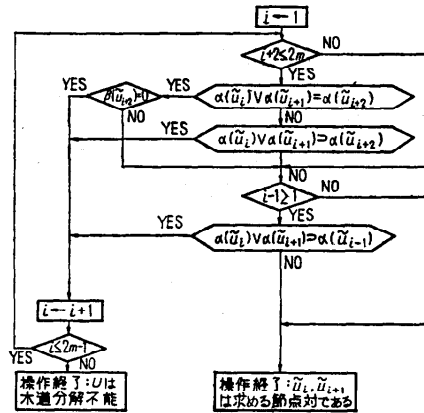


Fig. 2 Flow chart of the tree decomposition with the use of vertex codes.

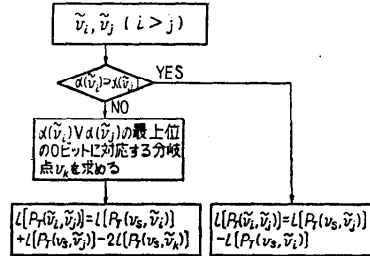


Fig. 3 Process for tree path distance calculation.

順で容易に求めることができる。

### 6. 木の選び方と道の表現の変換

乗車券自動発行端末装置は、 $G$  の各節点(駅)に対応する発駅、着駅を指定するためのボタン群と、各乗車区間が通るリンクを指定するための経路ボタン群とをもち、乗車区間の指定は発駅、着駅に対応するボタンと経路するリンクに対応するボタンをセットすることにより行われる。従って、経路ボタンのセットの回数を必要最小限にとどめるためには、リンクは端末装置設置駅よりなるべく“遠い”ところにある方が望ましく、木  $T$  はそのような観点より選ばれる必要がある。従って端末装置を鉄道網上に散在させ運賃計算などの処理をセンターコンピュータで一括して行う場合には、端末装置設置場所ごとに異なる木が選ばれることになるが、センターコンピュータでは、距離計算、木道分解に必要な節点コード、距離テーブルなどを、異なる木ごとに用意するのは効率が悪いため、各木ごとに異なる道の表現を、単一の標準的な木に基づく道の表現に変換する必要が生じる。

センターコンピュータにおける標準的な木の選び方は、センターコンピュータでの処理の効率化が重視される。本方式による距離計算では、道がリンクを含まなければ木道分解する必要がないこと、また木道の距離計算では、道の両端点  $v_0, v_D$  が  $v_0 \rightarrow v_D$ , または  $v_D \rightarrow v_0$  の関係にあれば、合流点の距離を検索する必要がないことなどから、標準的な木およびその根は、センターコンピュータで扱う乗車経路全体を考え、それらができるだけリンクを含まず、しかもできるだけ発着駅的一方が他方の下位にあるように選ばれるべきであろう。

このことから、一般的には、木の根に対しては、対象となる節点（駅）の中で最も乗降客の多い駅が選ばれることになり、木は、その駅からリンクに対応する区間ができるだけ“遠く”にあるように選ばれることになる。

このような異なる木に基づく道の表現を相互に変換することは、文献 2) の手法を用いて次のように行われる。

一般に、1つの木  $T$  に関して、この補木の各リンク  $c_i$  ( $i=1, 2, \dots, \mu$ ) が決定する基本閉路を  $L_i$  とすれば、任意の道  $P_i = P(v_0, v_D; c_{i_1}, c_{i_2}, \dots, c_{i_n})$  は

$$P_i = P(v_0, v_D; c_{i_1}, c_{i_2}, \dots, c_{i_n}) = Pr(v_0, v_D) \oplus (L_{i_1} \oplus L_{i_2} \oplus \dots \oplus L_{i_n}) \quad (6.1)$$

で与えられる<sup>2)</sup>。ただし、グラフ  $G_1 = [V_1, E_1], G_2 = [V_2, E_2]$  に対して  $G_1 \oplus G_2$  は  $E_1, E_2$  の対称差  $E_1 \ominus E_2$  によって生成されるグラフを表わす<sup>4)</sup>。従って、任意の枝  $e$  が  $P_i$  上にあるかどうかの判定は、 $e$  が木道  $Pr(v_0, v_D)$  上にあるかどうかと  $L_{i_1} \oplus L_{i_2} \oplus \dots \oplus L_{i_n}$  上にあるかどうかとの判定に帰着される。

以上の考察から、道の変換の問題は、 $T, T'$  をそれぞれ端末装置の木、センターコンピュータの木とし、それらのリンクの集合をそれぞれ  $C_T = \{c_1, c_2, \dots, c_\mu\}$ ,  $C_{T'} = \{c'_1, c'_2, \dots, c'_\mu\}$  として、指定された道  $P_i$  が

$$P_i = P(v_0, v_D; c_{i_1}, c_{i_2}, \dots, c_{i_n}) \quad (6.2)$$

$$P_i = P(v_0, v_D; c'_{i'_1}, c'_{i'_2}, \dots, c'_{i'_m}) \quad (6.3)$$

と2通りに表わせるとき、前者の表現から後者の表現を得る問題となる。

そこで、 $C_{T'}$  のリンクで  $C_T$  のリンクでないものを  $f_1, f_2, \dots, f_l$  とすると、式(6.2)から式(6.3)への変換は、 $f_1, f_2, \dots, f_l$  の各リンクが道  $P_i$  上にあるかどうかを調べることによって得られることになる。従って式(6.2)が与えられたとき、 $f_1, f_2, \dots, f_l$  のなかで、 $P_i$  上にあるものを  $f'_1, f'_2, \dots, f'_p$  ( $p \leq l$ ) とし

て求められたとすれば、これらのリンクに  $c_{i_1}, c_{i_2}, \dots, c_{i_n}$  のなかで  $C_{T'}$  に含まれるリンク  $f_{p+1}', \dots, f_{p+q}'$  ( $p+q=m$ ) を付け加えることにより、式(6.3)に対応する道の表現

$$P_i = P(v_0, v_D; f'_1, f'_2, \dots, f'_p, f_{p+1}', \dots, f_{p+q}') \quad (6.4)$$

が得られる。よって、 $C_T$  のリンクのうち  $C_{T'}$  のリンクに一致するものをあらかじめセンターコンピュータに記憶しておくならば  $f_{p+1}', \dots, f_{p+q}'$  は容易に選出することができる。従って問題となるのは、 $f'_1, f'_2, \dots, f'_p$  を求めること、すなわち  $f_1, f_2, \dots, f_l$  の各リンクが道  $P_i$  上にあるかどうかの判定をいかに行うかということになるが、これを行うためには式(6.2)により、各  $f_k$  ( $k=1, 2, \dots, l$ ) に対して、 $f_k \in Pr(v_0, v_D)$  かどうかの判定と、 $f_k \in L_{i_1} \oplus L_{i_2} \oplus \dots \oplus L_{i_n}$  かどうかの判定の2つの判定を行えばよい。この判定を行うためにはセンターコンピュータは木  $T$  の構造をあらかじめ知っておく必要があり、以下に述べる判定方法においてはこれらの情報は各節点  $v_i$  およびリンク  $f_k$  ( $k=1, \dots, l$ ) に対する付加符号  $\rho(v_i), \sigma(f_k)$  として記憶されている。

(I)  $f_k \in Pr(v_0, v_D)$  の判定

$f_k$  を  $T$  から開放除去して得られる2つの部分木を  $T_k^{(1)}, T_k^{(2)}$  とし、各節点  $v_i$  に対して  $l$  ビットのコード

$$\rho(v_i) = [\rho_1^{(i)}, \rho_2^{(i)}, \dots, \rho_l^{(i)}] \quad (6.5)$$

を設け、各ビット  $\rho_k^{(i)}$  の値は

$$\rho_k^{(i)} = \begin{cases} 0: v_i \text{ が } T_k^{(1)} \text{ 上にあるとき,} \\ 1: v_i \text{ が } T_k^{(2)} \text{ 上にあるとき,} \end{cases} \quad (6.6)$$

で与えるものとする。このとき  $f_k \in Pr(v_0, v_D)$  の判定は、次式によって行われる。

$$\rho_k^{(i)} + \rho_k^{(j)} \equiv 1 \pmod{2} \iff f_k \in Pr(v_0, v_D) \quad (6.7)$$

(II)  $f_k \in L_{i_1} \oplus L_{i_2} \oplus \dots \oplus L_{i_n}$  の判定<sup>2)</sup>

各  $f_k$  に対して  $\mu$  ビットのコード

$$\rho(f_k) = [\sigma_1^{(k)}, \sigma_2^{(k)}, \dots, \sigma_\mu^{(k)}] \quad (6.8)$$

を設け、各ビット  $\sigma_i^{(k)}$  の値は

$$\sigma_i^{(k)} = \begin{cases} 0: f_k \text{ が } L_i \text{ 上にないとき} \\ 1: f_k \text{ が } L_i \text{ 上にあるとき} \end{cases} \quad (6.9)$$

で与えるものとする。このとき  $f_k \in L_{i_1} \oplus L_{i_2} \oplus \dots \oplus L_{i_n}$  の判定は、次式によって行われる。

$$\sigma_{i_1}^{(k)} + \sigma_{i_2}^{(k)} + \dots + \sigma_{i_n}^{(k)} \equiv 1 \pmod{2}$$

$$\iff f_k \in L_{i_1} \oplus L_{i_2} \oplus \dots \oplus L_{i_n} \quad (6.10)$$

以上のように、センターコンピュータでは式(6.2)から式(6.3)への道の表現の変換に付随して、各節点

に対するコード  $\rho(v_i)$  および各リンク  $f_a$  に対するコード  $\sigma(f_a)$  と式(6.7), (6.10)による判定操作が余分に必要となる。

## 7. あとがき

実際の乗車券発行システムにおいては、乗車区間の距離計算のほか、自動改札機等に必要な磁気コード類の生成、乗車券面上の乗車径路の印刷表示、発売乗車券の分類集計などを行う必要がある。本文において述べた道の表現法、あるいは道を木道に分解し、それぞれの木道において必要とする“計算”を行うという手法は、単に距離計算のみならず、こうした乗車券発行における様々の“道”に関する計算処理においても、一つの有効な手段を提供するものである<sup>5)</sup>。

最後に、日頃御指導頂いている大阪大学工学部電子工学科尾崎弘教授、近畿日本鉄道(株)中井実監査役、同技術研究所中川利雄所長に謝意を表します。

## 参考文献

- 1) 石田：画像伝送式乗車券発行システムの概要，近鉄技報，Vol. 6, No. 1, pp. 44~53 (1974).
- 2) 白川，嵩，尾崎，小田，井上：定期乗車券の自動改札における通用径路の符号化問題，情報処理，Vol. 6, No. 2, pp. 81~88 (1965).
- 3) 近谷，岡原：線構造によるネットワークの表現とその応用，情報処理，Vol. 19, No. 1, pp. 9~15 (1978).
- 4) 尾崎，白川，翁長：グラフ理論，コロナ社，東京 (1975).
- 5) R. Tarjan: Depth-first Search and Linear Graph Algorithms, SIAM, Vol. 1, No. 2, pp. 146~160 (1972).
- 6) 石田，影浦：定期乗車券発行システムにおける乗車区間の指定と距離計算，システムと制御，Vol. 20, No. 12, pp. 19~24 (1976).

(昭和52年9月12日受付)

(昭和53年2月13日再受付)