



線構造によるネットワークの表現とその応用*

近谷 英昭** 岡原 正幸**

Abstract

In this paper we propose a new network representation method which facilitates the description of problems where line structures are dominant factors. By line structures we mean such structures as lines, train paths on a railroad network, street, avenue or blvd. on a road network, chains or lists in a data structure, etc.

As an extension of a graph, a notion called line network is defined as a hypergraph each edge of which is a totally ordered set of nodes. Four basic data structures for a line network and their applications to those for a graph and to a train diagram are stated. And two routing problems on a transportation network are treated by line networks.

1. ま え が き

鉄道網や道路網などの実際の輸送網においては、線的な概念あるいは線構造と呼ぶべきものが顕著にみられる。例えば、線区(鉄道網)、号線、街道、通り(道路網)、列車の運行経路(列車運行図)、バスの運行経路(バス運行系統図)、列車のスジ、駅の時間軸(列車ダイヤ)などがこれである。

これらの線構造は我々の現実世界で非常に大きな役割をはたしており、輸送企業における多くの業務や処理もこれら線構造に密に結ばれている。このため輸送網に関連する多くの問題はそれ自身線構造を含んでおり、また線構造が本質的である場合もある。通常、このような線構造を含む問題を扱おう場合、モデルとして線構造を含まないグラフの形に変換し、そこで問題を解いた後、線構造を含む形に逆変換するのが一般的である。また問題によっては、グラフ上での処理の過程で頻りに線構造との対応をとる必要が生ずる。このようなアプローチには(1)変換のためのオーバーヘッドが大きい、(2)モデルが自然でなく、複雑になる場合が多い、(3)線構造との対応をとるため処理効率が悪

くなる場合がある、などの問題点がある。

グラフを拡張してネットワークの表現に線構造を直接とりこむことができれば、これらの問題点の多くは解決されると期待される。グラフの1つの拡張としてハイパーグラフ¹⁾があるが、これに適当な順序を導入することによって線構造を直接的に表現することができる。

本論文では線構造によるネットワーク表現としてライン・ネットワークと呼ぶ新たな概念を導入し、その表現形式について述べるとともに、それを2つの実際の経路問題に応用した結果を述べる。

2. 線構造によるネットワークの表現

2.1 ライン・ネットワークの定義

X を任意の有限集合とする。 X の任意の部分集合の族を ε とすると、 X と ε の組 $H=(X, \varepsilon)$ をハイパーグラフ(hyper graph)という。ハイパーグラフは無向グラフの拡張となっている。

ハイパーグラフ $H=(X, \mathcal{L})$ において、各

$$l = \{x_1, x_2, \dots, x_k\} \in \mathcal{L} \quad (x_i \in X) \quad (1)$$

が全順序集合をなすとき、 H をライン・ネットワーク(line network)と呼ぶことにする。 X の要素 x をノード、 \mathcal{L} の要素 l をラインと呼ぶ。ライン l を構成するノード間の順序関係を l におけるノード順位という。また、ライン l が含むノード数を l の位数、ノード x

* A representation method of network by line structures and its applications by Hideaki KONYA and Masayuki OKAHARA (Systems Engineering Lab., Railway Technical Research Institute, Japanese National Railways).

** 日本国有鉄道鉄道技術研究所システム研究室

を含むラインの個数を x の位数と呼ぶ。特に、位数 1 のノードを中間ノード、2以上のノードを分岐ノードと呼ぶことにする。ライン・ネットワークはグラフの拡張であり、ラインの位数がすべて 2 であるようなライン・ネットワークはグラフとなる。

Fig. 1 にライン・ネットワークの例を示す。図ではノードを○印、ラインを直線または曲線でつらねられたノード列で表わしている。線区の設定されている鉄道網はライン・ネットワークの好例を与える。その他、列車の運行経路図、バスの運行系統図、列車ダイヤ、データ構造などライン・ネットワークとして表現できるものが数多くある。

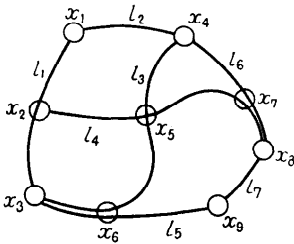


Fig. 1 Line network with nine nodes and seven lines.

2.2 ライン・ネットワークにおける双対性

ライン・ネットワーク $H=(X, \mathcal{L})$ において、ライン l に含まれるノードの集合を $X(l)$ 、ノード x を含むラインの集合を $\mathcal{L}(x)$ と表わす。 $X(l)$ にノード順位が与えられていると同様に、 $\mathcal{L}(x)$ 内のラインにも全順序が定義されているものとしよう。これを x におけるライン順位と呼ぶ。このときライン・ネットワークにおいては、ノードとラインを同等に取り扱いうることを示すことができる。

今、 $H=(X, \mathcal{L})$ に対してもう 1 つのライン・ネットワーク $H^*=(\mathcal{L}^*, X^*)$ を次のように定義する。すなわち、ライン $l \in \mathcal{L}$ に対してノード $l^* \in \mathcal{L}^*$ 、ノード $x \in X$ に対してライン $x^* \in X^*$ を

$$x \in l \Leftrightarrow x^* \ni l^*$$

であるように対応させる。このとき l におけるノード順位と x におけるライン順位がそれぞれ l^* におけるライン順位と x^* におけるノード順位に写されるとする。 H^* を H の双対表現と呼ぶ。(Fig. 2 の(a), (b) 参照)。

次に、 $H=(X, \mathcal{L})$ において $x \in l$ であるようなノード・ラインの組 (x, l) に対してノード $y_{x,l}$ を対応せしめ、その全体を Y とし、 $x \in X$ および $l \in \mathcal{L}$ に対して

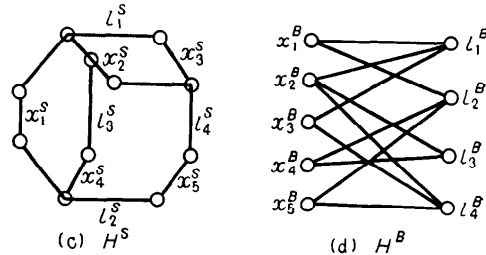
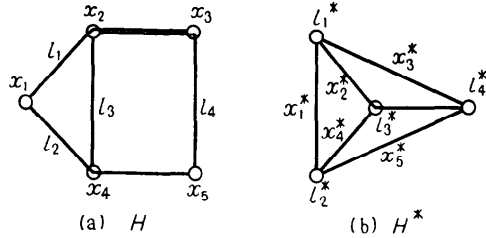


Fig. 2 Four representations of a line network.

それぞれライン

$$x^S = \{y_{x,l} \mid l \in \mathcal{L}(x)\} \tag{2}$$

$$l^S = \{y_{x,l} \mid x \in X(l)\} \tag{3}$$

を対応させ、それらの全体を各 X^S, \mathcal{L}^S とする。これらにおけるノード順位にはそれぞれ x, l におけるライン順位、ノード順位が写されるものとする。 $H^S=(Y, X^S \cup \mathcal{L}^S)$ によって定義されるライン・ネットワークを H の対称表現と呼ぶ。(Fig. 2(c) 参照)。

最後に、 $x \in X, l \in \mathcal{L}$ に対してそれぞれノード x^B, l^B を対応させその全体を X^B, \mathcal{L}^B とし、 $x \in l$ なるノード・ラインの組 (x, l) に対してライン

$$x_{x,l} = (x^B, l^B) \tag{4}$$

を対応させそれらの全体を Z とする。 x^B, l^B におけるライン順位は上に準じて与える。このとき $H^B=(X^B \cup \mathcal{L}^B, Z)$ によって定義されるライン・ネットワークを H のバイパータイト表現と呼ぶ。(Fig. 2(d) 参照)。 H^B はライン順位を無視するとバイパータイト・グラフとなる。

定義からわかるように、 H, H^* ではノードとラインの役割が相互に入れかわり、また H^S と H^B では H のノードとラインは全く対等の形で表わされている。これら 4 つのライン・ネットワークの間には Fig. 3 (次頁参照) に示すような関係が成立することを容易に示すことができる。

すなわち

$$(H^*)^* = H \tag{5}$$

$$(H^*)^S = H^S, (H^*)^B = H^B \tag{6}$$

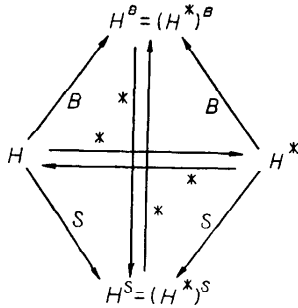


Fig. 3 Relations among H, H^*, H^S and H^B .

$$(H^S)^* = H^B, (H^B)^* = H^S \quad (7)$$

がなりたつ。

H をグラフとする場合、 H^* と H^S はグラフとはならないから、この関係はグラフの範囲では完結しない。この関係は H の問題を他の3つの表現のいずれで取り扱ってもよいことを示している。例えば、我々はグラフ論の結果から H^B を媒介として H におけるある種の結果を導くことができる²⁾。

2.3 ライン・ネットワーク上のパス

Fig. 4 に示すようなノード、ラインの列

$$\mu = [x_0, l_1, x_1, l_2, x_2, \dots, x_{i-1}, l_i, x_i] \quad (8)$$

但し、 $x_i \in l_i \cap l_{i+1}, l_i \neq l_{i+1}$

を x_0 から x_i へのパスという。この定義は、実際の応用において経路を指定するとき、その経路上でラインが変わるノードのみを指定すればよく他の途中のノードは指定する必要がないことを反映するものである。

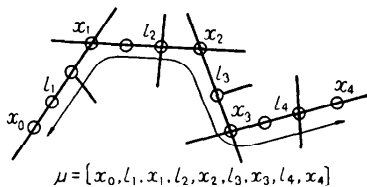


Fig. 4 Path definition on line network.

x_0 から x_i へのパス μ (8)を単にノードのみの列

$$\mu_X = [x_0, x_1, x_2, \dots, x_{i-1}] \quad (9)$$

で表わそうとする場合、これが x_0 から x_i へのパスを一意的に決めるための必要十分条件は、すべての i について x_{i-1}, x_i を含むラインが1つだけ存在することである。これに対してパス μ をラインの列

$$\mu_L = [x_0, x_i : l_1, l_2, \dots, l_i] \quad (10)$$

で表わそうとする場合、これが x_0 から x_i へのパスを一意的に決めるための必要十分条件は、各 i について l_{i-1}, l_i に共通に含まれるノードが1つだけ存在す

ることである。明らかに、ノード列表現 μ_X とライン列表現 μ_L がパスを一意的に決めないようなケースは、互いに双対の関係にある。

H^S においてはパスのノード列表現は常に一意的である。

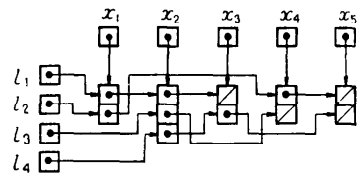
3. ライン・ネットワークのデータ構造

グラフ構造をもつ問題の計算機処理に関連して、グラフの計算機内における表現すなわちデータ構造の重要性が指摘されている^{3), 4)}。ここではライン・ネットワークに対する基本的なデータ構造を示すと同時にその応用として列車ダイヤの表現とグラフのデータ構造に対する1つの見方について述べる。

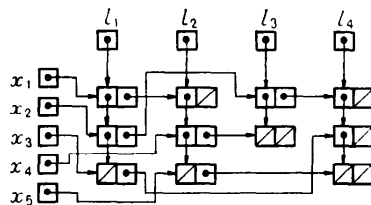
3.1 基本的なデータ構造

ノードを結合セル、ラインをポイントのチェインで表わすものとすれば、1つのライン・ネットワーク H に対して、 H, H^*, H^S, H^B に対応する4種の基本的なデータ構造が存在する。

Fig. 5(a)にFig. 2の H に対応するデータ構造を示す。ノード・セルはそのノードを含むラインのライン・チェインに対応するポイントを含む。図の上側および左側ははれぞれノード・セルおよびライン・チェインのヘッドを示すポイントのリストである。図には構造を記述する要素のみを示したが、これらの各要素には応用により何らかの値を対応させることができる。双対表現 H^* に対応するデータ構造はノードとラインの役割が入れかわるがFig. 5(a)と同様の構造となる。



(a) H -構造



(b) H^S -構造

Fig. 5 H and H^S structures of the line network in Fig. 2.

Fig. 5(b) は対称表現 H^S に対応するデータ構造である。各結合セルは H において $x \in l$ であるようなノード・ラインの組 (x, l) に対応し、当然ながらすべてノード・チェイン, ライン・チェインを示す2つのポイントから成る等長セルである。したがって、この場合結合セルを示すリストは実質的に不要であり、図でも省略してある。パイパータイト表現 H^B に対応するデータ構造は Fig. 5(b) の双対として理解できる。

3.2 基本構造の変形と応用

上記の基本構造から各種の変形構造を導くことができる。例えば、基本構造のチェインは必要によって環状リストや2方向リストにすることができる。また、応用によっては結合セルの物理的配置に拘束を加えて、配列あるいはテーブル形の構造を作りうる。このような構造の1例は 4. で示される。

列車ダイヤは列車をライン, 駅をノードとすることによって1つのライン・ネットワーク H_D となる。したがって、我々は H_D, H_D^*, H_D^s, H_D^B に対応する列車ダイヤのデータ構造をうることができる。 H_D^s は列車と駅の時間軸をラインとし、列車の駅における停車という事象をノードと考えたものになる。更に、この事象を着事象と発事象に区別すれば、Fig. 6 に示すようなライン・ネットワークが得られる。これに対応するデータ構造は列車ダイヤの極めて自然な表現となっている。

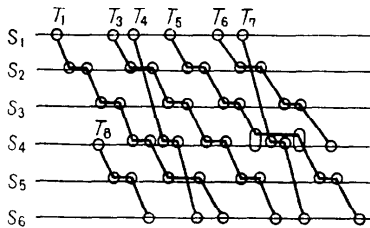


Fig. 6 Line network representing train diagram.

グラフはライン・ネットワークの特別な場合とみなせるから、上記の4つの構造はグラフに対する4つの基本的なデータ構造を導く。Fig. 7 に簡単なグラフについて例を示す。但し、チェインは環状リストとした。このことは前述のようにグラフ自身はこれら4つに対応する構造をもち得ないが、ライン・ネットワークまで拡張すればそのデータ構造を記述できることを意味している。すなわち、我々はグラフのデータ構造に対して1つの新たな視点を得たことになる。これはまた一般のデータ構造をライン・ネットワークあるいは

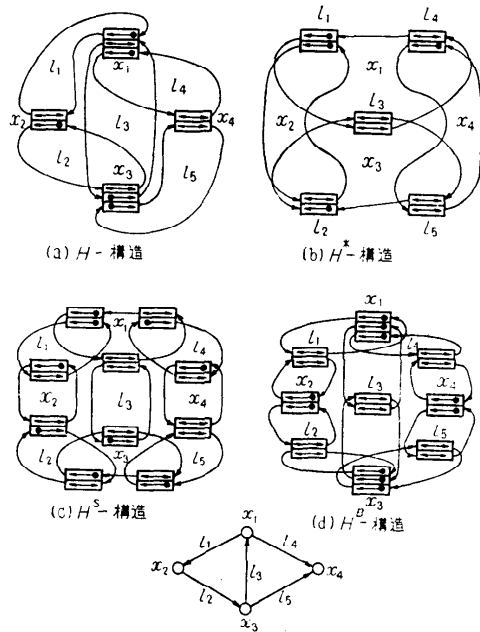


Fig. 7 Four data structures for a graph.

はハイパーグラフとして取り扱いうる可能性を示唆している。

4. 大規模な輸送網における順路選出問題への応用

4.1 問題の概要

輸送網における2点間のパスのうち特定の指定された条件を満たすものを順路という。例えば最短経路は1つの順路であり、また例えば国鉄では旅客順路, 貨物運賃計算経路, 貨物輸送経路などの順路が目的に応じて用いられる。以下では、いずれか1種の順路について議論を進める。

問題は2つのノードが与えられたとき、その間の順路を選びだすことであるが、大規模な輸送網を対象とするリアルタイム的な応用では、要求時に直接順路を算出することは処理効率上の問題が大きすぎる。このため、あらかじめ処理した何らかの順路情報をファイルしておき、要求時にこれを参照して順路を得る方法が現実的である。この種の方法としては、各ノードを中心とする経路木をファイルする方法と各ノードを中心として順路で到達可能な方向別の領域(順路領域という)をファイルする方法がある。これらの方法にライン・ネットワークの表現を適用すれば、グラフとして考えた場合に較べてかなり効率のよい方法を得ること

とができる。

4.2 輸送網の表現

以下、順路について次の条件が成りたつものとする。

順路の条件 ノード x, y 間の順路 $r(x, y)$ 上の任意の2つのノード x', y' 間の順路 $r(x', y')$ は $r(x, y)$ に含まれる。

今、与えられた輸送網をライン・ネットワーク $H_N = (X_N, \mathcal{L}_N)$ として表現する。このとき任意のライン $l \in \mathcal{L}_N$ がそれ自身両端ノード間の順路を成し、且つ、パスのノード列表現が1意的であるようなライン・ネットワークとする。例えば、鉄道網で実際の線区をラインとする場合、必要なら線区を分割することによって、このようなライン・ネットワークをうることができる。

$x \in l (x \in X_N, l \in \mathcal{L}_N)$ なるノード・ラインの組 (x, l) に対して値 $(c_l, d_l(x))$ を対応させる。ここに c_l は l の識別名、 $d_l(x)$ は l の先頭ノードから x までの l 上の距離であり、この値を x のコードという。中間ノードに対しては1個の、分岐ノードに対しては複数個のコードが与えられる。 H_N の全分岐ノードに関するコードを Fig. 8(a) に示すように値の小さい順に並べたテーブルを作り、同一ノードに対応する項を環状リストとして結合する。これをライン・ネットワーク・テーブル (LN テーブル) と呼ぶ。これは本質的にはライン・セルの物理的配置を除いて H_N^* に対応するデータ構造に同じである。

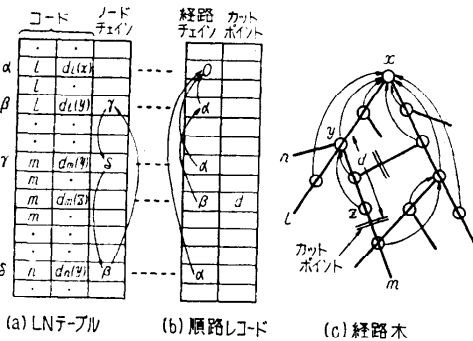


Fig. 8 Line network table (LN table) and route record of node x .

4.3 経路木による方法

順路の条件から、1つのノードに到るすべての順路は Fig. 8(c) に示すような経路木として表現できる。このとき順路は図中矢印で示すようにパスのノード列表現をとる。このような経路木を (b) に示すように LN テーブルと項ごとに対応する構造の順路レコード

として表わす。ここでカットポイントは経路木における切断点の位置を示すものであるが、数値としては切断点に対してもノードに対する同様のコードを与え、その距離部分を用いるのが都合がよい。格納位置はその切断点をはさむ2つの分岐ノード・コードの小さい方に対応する項とする。

以上のような順路レコードおよび LN テーブルを用いて、与えられた2つのノード x, y 間の順路を求める方法は以下のとおりである。但し、 x, x_i 間の順路 $r(x, x_i)$ をあるラインにそって x_{i+1} まで延長することを $r(x, x_i) \cup (x_{i+1})$ と表わし、また x, x_i 間の順路上の距離を $d(x, x_i)$ と記す。

1° $\mathcal{L}(x) \cap \mathcal{L}(y) \ni l$ のとき

$$r(x, y) = (x, y)$$

$$d(x, y) = |d_l(x) - d_l(y)|$$

この場合ファイル参照を要しない。

2° $\mathcal{L}(x) \cap \mathcal{L}(y) = \phi$ のとき

y の順路レコードを読み出し、LN テーブル上で x の位置を決め、そこから経路ポイントをたどる。すなわち、 $r(x, x_i)$ がすでに得られているとき、 x_i の経路ポイントで x_{i+1} を得、LN テーブルにより $l \in \mathcal{L}(x_i) \cap \mathcal{L}(x_{i+1})$ をきめて

$$r(x, x_{i+1}) = r(x, x_i) \cup (x_{i+1})$$

$$d(x, x_{i+1}) = d(x, x_i) + |d_l(x_i) - d_l(x_{i+1})|$$

とすればよい。

以上において $\mathcal{L}(x) \cap \mathcal{L}(y) \ni l$ の操作は LN テーブルで x, y に対応する2つのノード・チェーンを同時にたどることによって行う。

4.4 順路領域による方法

Fig. 9 に示すようにノード x からこれを含むライン上の分岐ノード x_i を経由して順路で到達可能な領域を D_i とする。この方法では x の順路レコードは x_i と D_i の組から構成される。領域 D_i の表現方法としては、 D_i を LN テーブル上にマップしたとき得

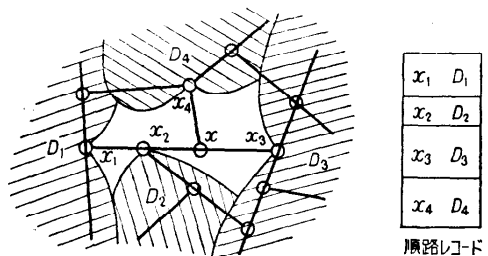


Fig. 9 Areas reachable from x via x_i along the routes satisfying a prescribed condition.

られるコードの区間和によればよい。区間の境界がラインの中間にある場合は前述の切断点コードを用いる。

この方法における順路選出の手順は次のとおりである。 x, y が与えられたとき、 $z=x$ として

1° $\mathcal{L}(x) \cap \mathcal{L}(y) \neq \emptyset$ なら

$$r(x, y) = r(x, z) \cup (y)$$

$$d(x, y) = d(x, z) + |d_i(x) - d_i(y)|$$

として完了。

$$\mathcal{L}(x) \cap \mathcal{L}(y) = \emptyset \text{ なら } 2^\circ \text{ へ。}$$

2° x の経路レコードを読み、 y を含む領域への出口ノード z' を見出し、 $m \in \mathcal{L}(x) \cap \mathcal{L}(z')$ を決めて、

$$r(x, z') = r(x, z) \cup (z')$$

$$d(x, z') = d(x, z) + |d_m(x) - d_m(z')|$$

$z=x'$ として 1° へ。

4.5 考 察

これらの方法における処理量は、要求された順路が含むライン数を k とするとき、 $k-1$ に比例的である。実際の応用では、 k の値の小さな順路に対する要求発生頻度が非常に高いため、これらの方法の処理効率は見かけよりかなりよいものとなる。例えば国鉄における各種の順路選出については、必要なファイル参照回数平均は経路木法で 0.6~0.7 回、順路領域法で 1.5~2.0 回程度となる。これらにおける経路の順送り回数は、線構造を無視し経路の全分岐ノードを順送りする方法の場合のほぼ 1/10 ですむであろう。

これらの方法の大きな利点として、順路の表現が外側の環境におけるものと同じであること、および大量に存在する中間ノードの位置づけが LN テーブル上で容易に達せられることがあげられる。なお、この方法(経路木法)は国鉄の貨物輸送経路選定システム⁶⁾として実用化されている。

5. 自動出改札システムにおける経路判定への応用

5.1 経路判定問題と基本的な考え方

自動出改札システムにおける基本的な問題として、改札ゲートにおいて乗車券等に記録された乗車経路の正当性を判定する問題がある。これは次の3つの部分問題を含んでいる。

- (1) 対象鉄道網を構成する駅の集合にどのようなコード体系を与えるか(ネットワークの符号化)
- (2) 券面上に乗車経路をどのように記述するか

(経路の記述)

- (3) 改札装置は乗車経路の正当性をどのように判定するか(判定論理)

経路の記述と判定論理がネットワークの符号化に直接依存することはいうまでもない。

ネットワークの符号化において最も本質的なことは、サイクルを含まない適当な部分ネットワークを選び、与えられた特定の点とその部分ネットワーク内の任意の2点間のパス上にあるか否かをその2点のコードのみから決定しようにすることである。このような部分ネットワークとして、ネットワークを張る木を用いる方法と、ネットワークをおおうパスの集合を用いる方法がある。前者に属するものとして白川らの方法⁶⁾、後者に属するものとして筆者らの方法⁷⁾があり、いずれも実用化されている。後者はライン・ネットワークをベースにした方法とみなすことができる。ここではライン・ネットワークに基づく更に簡単な方法について述べる。

5.2 ネットワークの符号化と経路の記述

対象鉄道網をライン・ネットワーク $H_N = (X_N, \mathcal{L}_N)$ として表現する。ラインとしては実際の線区を用いてもよいし、また、京浜東北線、横須賀線などの列車運行経路を用いてもよい。但し、環状線は分割する必要がある。 H_N の対象表現 $H_N^s = (Y, X_N^s \cup \mathcal{L}_N^s)$ を考え、 H_N^s の各ノードについてコードを与える。すなわち、4.2 におけると同様に、 H_N において $x \in l$ であるときノード・ラインの組 (x, l) に対して $(c_l, d_l(x))$ なるコードを与える。ここに c_l はライン l のコード、 $d_l(x)$ は l 内における x の位置を示すコードであり、 l におけるノード順位の単調増加関数なら何でもよい。以下では c_l を単に l と記す。

2ノード間の乗車経路の記述は H_N^s におけるパスのノード列表現を用いる。これはパスの1意表現である。例えば Fig. 10(a) (次頁参照) に示すような H_N における乗車経路は(b)の黒丸で示すように H_N^s におけるノード列で表わされる。券上には(c)に示すようにこれらのノードのコードを対応する順序に記録すればよい。

5.3 判定論理

各駅すなわち H_N の各ノードの改札装置がもつべき情報はそのノードに対応するコード(分岐ノードの場合複数個)のみである。

任意のノード $x \in X_N$ で判定を行う場合を考えよう。 x のもつコードを $(m_1, d_{m_1}(x)), (m_2, d_{m_2}(x)), \dots,$

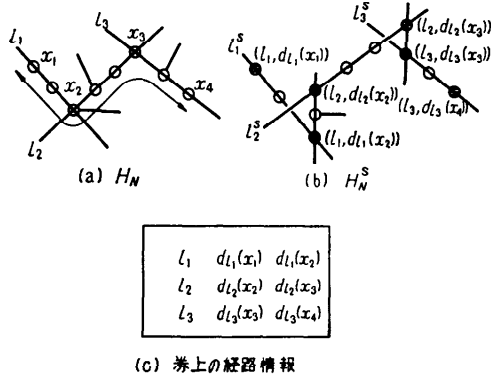


Fig. 10 Route representation on a ticket in automatic ticket examination system.

$(m_k, d_{m_k}(x))$ とする。また、改札装置は Fig. 10(c) に示したような乗車経路情報を券から順々に読みながら判定するものとする。今、 l_i まで判定が進んだとすると以下次のように判定を行う。

- 1° いずれかの j について $l_i = m_j$ なら 2° へ
さもなければ $i+1$ として 1° へ
- 2° $d_{l_i}(x_i) \leq d_{l_i}(x) \leq d_{l_i}(x_{i+1})$ または
 $d_{l_i}(x_i) \geq d_{l_i}(x) \geq d_{l_i}(x_{i+1})$
なら可と判定
さもなければ $i+1$ として (1°) へ。

以上の結果可と判定されなかった場合不可である。

5.4 考 察

この方法は以下に示すようにいくつかの優れた特性をもっている。(1)改札装置の判定論理が非常に単純である。(2)改札装置の記憶すべき情報が極めて少なく、且つ自己ノード以外の部分のネットワーク構造に全く依存しない。このため新線の建設、駅の新設改廃などのネットワーク構造の変化に極めて強い。(3)旅客の流動分布は線区や列車運行に依存しており、発生頻度の高い経路ほど少ない経路情報ですみ、また判定時間も短い。更に、必要なら仮想的な線区や列車運行経路をラインとして加えることができる。(4)2社以上の鉄道にまたがる場合にも極めて自然に拡大できる。

6. あとがき

線構造を直接ネットワーク表現にとりこむ試みとしてライン・ネットワークなる新たな概念を導入し、いくつかの応用についてその有効性を示した。この表現あるいは考え方は線構造の優越している他の多くの問題、特に輸送などの現実面で生ずる問題についてその効果を確かめることができよう。例えば、列車の乗継ぎ問題や貨物の中継輸送問題のように線構造間の関係が本質的であるような問題は、ライン・ネットワークによって非常に自然な形に記述することができる。

ライン・ネットワークあるいはハイパーグラフはグラフを包含するかなり広いクラスの情報構造に対応するものと考えられる。これらを記述し処理しうる言語が開発されれば、多くの現実的な問題に対して有力な道具を提供することになると期待される。

最後に、本研究を進めるに当たり多くの面でご援助をいただいた大野 豊京大教授、鉄研システム研究室並びに国鉄本社情報システム部の諸氏に深く感謝致します。

参 考 文 献

- 1) C. Berge: Graphes et hypergraphes, Dunod (1970).
- 2) 近谷: 輸送網の線区向き表現について, 第 15 回情報処理学会大会予稿集 (1974).
- 3) 伊理: グラフ的構造を有する情報の処理技法について, 昭和 47 年度電気四学会連合大会, 217 (1972).
- 4) 日本オペレーションズ・リサーチ学会: ネットワーク構造を有するオペレーションズ・リサーチ問題の電算機処理に関する基礎研究, 報文シリーズ T-73-1 (1973).
- 5) 国鉄コンピュータ部: 貨物輸送経路選定システム (1975).
- 6) 白川, 他: 定期乗車券の自動改札における通用経路の符号化問題, 情報処理, Vol. 6, No. 2, (1965).
- 7) 等々力, 近谷: 自動改札における乗車経路の判定法, 計測自動制御学会論文集, Vol. 8, No. 2, (1972).

(昭和 51 年 7 月 22 日受付)

(昭和 52 年 2 月 23 日再受付)