論　文

線構造をもつネットワーク上の最適順路問題*

近　谷　英　昭**

Abstract

A network with line structures, called line network, is defined as a hypergraph each edge of which forms a totally ordered set of nodes. Many objects concerning transportation networks such as railroad networks, train or bus route networks, road networks, train diagrams, etc. can be represented naturally by line networks.

An algorithm for the shortest paths on a line network is presented. This algorithm is an extended version of Bellman's shortest path algorithm. Minimum time traveling problem on a train diagram is treated as an application of line network and of the shortest path algorithm on it. Moreover, the algorithm is extended so as to serve the problems of obtaining the minimum cost relaying schedules among trains.

1. まえがき

輸送網において何らかの意味での最適順路を見出す問題は輸送に関する計画，スケジュリング，制御などの分野でしばしば起こる重要な問題である．中でも，最短路問題とその変形問題は最も基本的なクラスを成している．ネットワークにおける最短路問題については，その適用範囲の広さや重要性から，解法に関する多くの研究がなされており[1]，また，すでにそのうちのいくつかはある意味で最適解法であることが示されている[2]．しかし，実際的な問題に対する適用という立場からみると，少なくとも2つの問題点を指摘できる．1つは，解法の“良さ”に対する評価が一般に完全グラフを対象としてなされているのに対して，現実的なネットワークは極めて sparse だという点である．実際に，最適解法とされる Dijkstra のアルゴリズム[3]は，sparse なネットワークに適用した場合，Bellman のアルゴリズムより[5]低速であるという実験的結果が報告されており[5]，また筆者らの実験でも同様の結果がえられている．

もう1つの問題点は，現実に生ずる問題あるいはその実際面での表現とグラフによる問題の記述との間に差があり，両者間の変換という問題が存在することである．このうち，最も一般的でかつ重要と思われる点は，輸送網においては，鉄道網における線区や列車運行系統，バス路線図における運行系統，道路網における通り，街道，号線，列車ダイヤにおける列車を表わす折線（スジ）や駅の時間軸などのような線的な概念が極めて優越しており，これらが我々の現実世界で問題の表現や記述に重要な役割をもっていることである．以下，このような線的な概念を単に線構造と呼ぶことにする．

例えば，上野—秋田間の最短路が必要な場合，我々の期待するのは

上野—(東北本線)—福島—(奥羽本線)—秋田

という線構造に依存した形式の答である．これに対して，分岐駅のみからなるグラフあるいはネットワークで最短路問題を解いた答は

上野—日暮里—赤羽—……—大曲—秋田

（中間の分岐駅数 17）

という形式であり，線構造による形式への変換が必要となる．また，このためネットワークの外に線構造を示すデータが必要であるが，一般に線構造を示すデータはネットワーク構造に関する情報を含むため，データとしても冗長になる．これは，例えば列車ダイヤに

* Optimal Path Problems on Networks with Line Structures by Hideaki KONYA (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

おいて最短時間の列車乗継ぎプランを求める問題や，列車間の中継コストを含むような最小コスト中継問題など，線構造間の関係が基本的であるような問題においては一層本質的な問題となる．これらにおいてはグラフによる記述は一般に複雑となり，また，問題の自然な表現を与えない．

我々は線構造そのものによってネットワークを表現する試みとして，グラフを拡張したラインネットワークと呼ぶ概念を導入した[6),7)]．上述のような線構造の優越した問題に対しては，その線構造を表現するラインネットワークを作り，その上で問題および解法を記述する方がより直截・簡明であり，変換のオーバヘッドがなく総合的な効率がよいと期待される．

本論文においては，まずラインネットワーク上の最短路問題に対して Bellman のアルゴリズムを拡張する．次に，このアルゴリズムを最小時間列車乗継ぎ問題に適用し，また，最小コスト中継問題に対してこのアルゴリズムを更に拡張する．

2. 線構造をもつネットワーク

任意の有限集合 X に対して，X の任意の部分集合族を $\mathcal{L}$ とするとき，X と $\mathcal{L}$ の組 $H=(X,\mathcal{L})$ をハイパーグラフ (hypergraph) という[8)]．これは無向グラフの一つの拡張である．

ハイパーグラフ $H=(X,\mathcal{L})$ において，特に各 $l\in\mathcal{L}$ が X の要素の順序づけられた列をなすとき，すなわち，

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

が全順序集合をなすとき，H をラインネットワークと呼ぶことにする[7)]．X の要素をノード，$\mathcal{L}$ の要素をラインと呼ぶ．また，ノード x を含むラインの部分集合を $L(x)$ と表わし，その要素数 $|L(x)|$ を x の位数という．

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，ラインを直線あるいは曲線でつらねたノード列として表わす．線区の定義された鉄道網，列車の運行系統図，バス路線図，列車ダイヤなどラインネットワークとして表わされる対象は多い．

与えられたラインネットワーク $H=(X,\mathcal{L})$ に対して，次のように距離を導入する．まず，各ラインについてその始端ノードを定める．$x\in l$ であるようなノード $x\in X$ とライン $l\in\mathcal{L}$ に対して，l の始端ノードから x までの上の距離を $d_l(x)(\geq 0)$ と表わす．このとき，同一ライン上の2つのノード $x_i,x_j\in l$ 間の

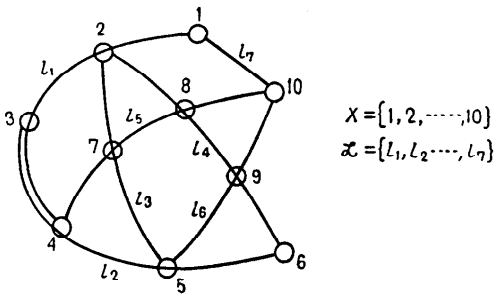

Fig. 1 Line network $(X,\mathcal{L})$ with ten nodes and seven lines

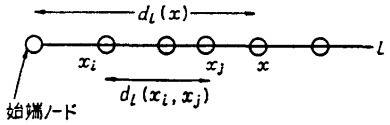

Fig. 2 Distance in line network

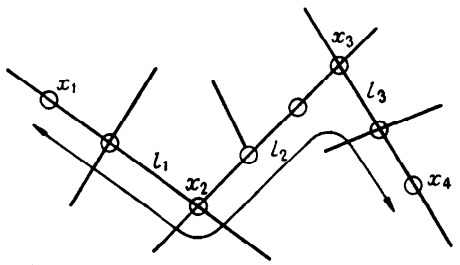

Fig. 3 Path definition in line network

距離 $d_l(x_i,x_j)$ は

$$d_l(x_i,x_j)=|d_l(x_i)-d_l(x_j)|\qquad(2)$$

で与えられるものとする (**Fig. 2** 参照)．

$H=(X,\mathcal{L})$ において，次のようなノード，ラインの交代列 $\mu[x_1,x_k]$

$$\mu[x_1,x_k]=[x_1,l_1,x_2,l_2,\cdots,x_{k-1},l_{k-1},x_k]\qquad(3)$$

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

を x_1 から x_k へのパス (path) という．H に対して上述のような距離が定義されているとき，パス $\mu[x_1,x_k]$ の長さ $p(\mu)$ を次のように定義する (**Fig. 3** 参照)．

$$p(\mu)=\sum_{i=1}^{k-1}d_{l_i}(x_i,x_{i+1})\qquad(4)$$

3. ラインネットワーク上の最短路問題

3.1 最短路問題とアルゴリズム

ここではネットワークの特定のノードから他のすべ

```
1  DL[x]:=∞ for all x≠s;
2  NL[x]:=0 for all x;
3  LD[s]:=0;
4  QLIST:={s};
5  while QLIST≠φ do
6  begin
7    choose x∈QLIST;
8    QLIST:=QLIST−{x};
9    for each (x, y)∈A & y≠NL[x] do
10     if DL [y]>DL[x]+d(x, y) then
11     begin
12       DL[y]:=DL[x]+d(x, y);
13       NL[y]:=x;
14       QLIST:=QLIST∪{y};
15     end
16 end;
```

Fig. 4 Shortest path algorithm on network (X, A)

てのノードへの最短路を求めるいわゆる $1-n$ 問題を考える．通常のネットワークに対する最短路アルゴリズムは，一つの一般化した形式では **Fig. 4** のようになる．ここで $d(x, y)$ はエッジ (x, y) の長さ，DL，NL はそれぞれ距離，経路を示すラベルであり，QLIST はそこからラベリング操作を続ける必要のあるノードのリストである．多くの場合アルゴリズムの差は文 7, 8 および 14 における QLIST の取り扱いの違いにある．例えば，Dijkstra のアルゴリズム[3]では 7 において QLIST 中のノードのうち DL の値の最も小さいノードを選ぶ．この場合一度 QLIST から取りだされたノードは再度 QLIST に入れられることはない．Dijkstra のアルゴリズムには 14 において sorting し，7 で先頭ノードを取りだすもの，QLIST の構造を binary heap[9] とするものなどの変形がある．これに対して，QLIST の取り扱いの最も単純なものは 7 で QLIST の任意のノードを選ぶ方法である．これは Bellman の方法[4]の変形であるが，以下単に Bellman のアルゴリズムという．この方法では各ノードは 2 度以上 QLIST に入れられる可能性がある．

この 2 つの典型的な方法はいずれもラインネットワーク上のアルゴリズムとして拡張できる．しかし，Dijkstra のアルゴリズムではラベリング操作の拡大が中心の特定ノードからの最短距離のみによって原則として 1 意的にきまるため，ラインの存在がアルゴリズム上何ら有効でなく，逆に効率上ネガティブに働くと考えられる．これに対して Bellman の場合は，ラベリング操作の拡大に制約がないため，ライン上のノードに対して一括してラベリングを行うように拡張できる．また，一方，現実的な問題におけるネットワークはほとんどの場合極めて sparse であるため，この種のネットワークに対する Bellman のアルゴリズムの優位性[5]も考慮して，ラインネットワークに対する最短路アルゴリズムとしては Bellman の方法を拡張する．

3.2 ラインネットワークに対するアルゴリズム

距離の定義されたラインネットワーク $H=(X, \mathcal{L})$ と 1 つのノード $s\in X$ が与えられているとする．このアルゴリズムでは，s から始めてライン単位にラベリングを行い，ラベル変更を受けたノードを後のラベリング続行点の候補としてリストする．このリストからのノードの取りだしは任意であり，また，リストが空になればアルゴリズムは終了する．このアルゴリズムを以下 SPATH と呼ぶ．

アルゴリズム SPATH を ALGOL 風の記法で具体的に示すと **Fig. 5** のようになる．ここでは見やすくするためにライン単位のラベリングを 1 つの procedure として分けた．ラベルとして次の 3 つを用いる．DL [x] は解法の各時点できまっている $s\to x$ のパスの長さを示すラベルであり，初期的には十分大きな値が与えられる．LL[x] と NL[x] は各時点できまっている $s\to x$ のパスを示すラベルであり，前者はそのパスが x に到達したライン，後者はそのパスが最後にライン LL[x] に入ったノードを示す（**Fig. 6**（次頁参照）参照）．

QLIST は最初 $\{s\}$ とされ，以下ラベル変更を受けたノードが追加される．QLIST は具体的には FIFO

```
   procedure SPATH;
1  begine
2    DL[x]:=∞ for all x∈X;
3    DL[s]:=0;
4    NL[x]:=0 for all x∈X;
5    LL[x]:=0 for all x∈X;
6    QLIST:={s};
7    while QLIST≠φ do
8    begin
9      choose x∈QLIST;
10     QLIST:=QLIST−{x};
11     for l∈L(x) & l≠LL[x] do
12       SCAN (x, l);
13   end
14 end SPATH;

   procedure SCAN (x, l);
1  begin
2    for y∈l & y≠x do
3      if DL[y]>DL[x]+d_l(x, y) then
4      begin
5        DL[y]:=DL[x]+d_l(x, y);
6        NL[y]:=x;
7        LL[y]:=l;
8        QLIST:=QLIST∪{y};
9      end
10 end SCAN;
```

Fig. 5 Shortest path algorithm SPATH on line network $(X, \mathcal{L})$

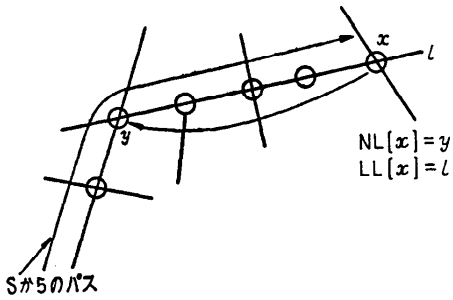

Fig. 6 Routing labels in SPATH

の Queue タイプのリストでよい．QLIST から取りだされたノードを x とするとき，x を含むライン $l\in L(x)$ のうち LL[x] 以外の各ラインに対してラベリングの操作を加える．これが procedure SCAN である．SCAN (x, l) では l 上の x 以外のノード y についてラベリング操作を行い，ラベル変更を受けた y のみを QLIST に入れる．l 上のノードへのラベリングの順序には制約はないが，以下では l にそって x からそれぞれの方向に順次ノードをたどって行われるものとする．このとき，SCAN の文2を次の 2′ に置きかえて，ラベリングの範囲を限定してもよい．

2′ **for** $y\in l$ & $y\neq x$ such that DL[y] $\geq$DL[x]+$d_l(x, y)$ **do**

アルゴリズム SPATH の正当性の証明は後で与える．

4. 最短時間列車乗継ぎ問題

4.1 問題と列車ダイヤのラインネットワーク

列車ダイヤあるいは時刻表Dが与えられているものとする．問題は発駅 s_0，着駅 s_k および発時間帯 $[T_1, T_2]$ が指定されたとき，$[T_1, T_2]$ 間に s_0 を出発し，最短時間で s_k に達するようなD上の列車乗継ぎプラン

$$s_0, t_1, s_1, t_2, \cdots, t_k, s_k \qquad (5)$$

を求めることである．ここに，$t_1, t_2, \cdots t_k$ は列車，$s_1, s_2, \cdots, s_{k-1}$ は乗継ぎ駅である．この問題は列車乗継ぎ問題の中で最も基本的なものであり，旅客の旅行行程の作成や案内などで起きる．

列車ダイヤは次のように極めて自然な形でラインネットワークとして表現できる．すなわち，列車ダイヤDにおける駅の集合を $\mathcal{L}_s$，列車の集合を $\mathcal{L}_t$ とするとき，$\mathcal{L}=\mathcal{L}_s\cup\mathcal{L}_t$ でラインの集合を定義し，列車 $t\in\mathcal{L}_t$ の駅 $s\in\mathcal{L}_s$ における着，発の事象をそれぞれノード x_{sti}, x_{st2} とし，それらの全体を X とする

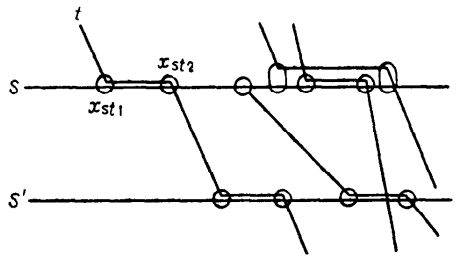

Fig. 7 Line network of train diagram

(**Fig. 7** 参照)．このラインネットワーク $H_D=(X, \mathcal{L})$ において $x_{sti}\in X$ に対して

$d_s(x_{sti})=d_t(x_{sti})$：事象 x_{sti} の生起時刻　(6)

によって距離を与える．すなわち，列車の着発時刻そのものによってラインネットワーク H_D に距離を導入する．この H_D において，発駅 $s_0\in\mathcal{L}_s$ および時間区間 $[T_1, T_2]$ によってきまる発事象に対応するノードの集合を X_0 とする．このとき，上述の問題はラインネットワーク H_D において X_0 の任意のノードから s_k の任意のノードへ至るパスのうちで長さの最も短かいものを求める問題となる．

4.2 アルゴリズム

この問題は，原理的には前述のアルゴリズム SPATH を一部修正したもので解くことができる．すなわち，Fig. 5 の SPATH において文3, 6 を

3′ DL[x]:=0 for $x\in X_0$

6′ QLIST:=X_0

で置きかえ，各ラインにおけるラベリングの方向を1方向に限定すればよい．勿論，列車ダイヤは 24 時間周期で巡回的であるから距離の計算は modulo 24 (時間) で行う必要がある．アルゴリズムの終了時において，経路を示すラベル LL に(5)の形式の最適列車乗継ぎプランが得られる．

なお，実際的な観点からすると，このアルゴリズムはラベリングの対象範囲が通常明らかに不要と思われる範囲まで拡大するという問題を含んでおり，列車ダイヤの対象が大きくなると本質的な問題となる．しかし，この問題はネットワークの表現論とは基本的に別個の問題であるから，ここではふれないことにする．

5. 最小コスト列車中継問題

5.1 問　題

1つのネットワークとその上を運行する列車が与えられているとする．このとき列車運行パターンとは，同一発着，同一経路の同種列車を1つの列車運行系統

として代表させたものである．駅をノード，列車運行パターンをラインとすることによって，列車運行を表わすラインネットワーク $H_T=(X, \mathcal{L})$ をうる．各駅 $x\in X$ ではそこを通るどの列車運行パターン相互間でも，貨物の中継(旅客の乗継ぎ)が可能であるとする．

この H_T に対して，列車運行パターン $l\in\mathcal{L}$ 上の駅 x, y 間の輸送コスト $d_l(x, y)$ $(>0, x\neq y)$ および駅 $x\in X$ における中継コスト $r(x)$ (≥ 0) が定義されているものとする．但し，$d_l(x, y)$ については線形性を仮定する．すなわち，$x, y, z, \in l$ がこの順に並んでいるとき

$$d_l(x, z)=d_l(x, y)+d_l(y, z) \qquad (7)$$

であるとする．また，H_T 上のパス

$$\mu[x_1, x_k]=[x_1, l_1, x_2, l_2, \cdots, l_{k-1}, x_k] \qquad (8)$$

が与えられたとき，パス μ のコスト $c(\mu)$ を

$$c(\mu)=\sum_{i=1}^{k-1} d_{l_i}(x_i, x_{i+1})+\sum_{i=2}^{k-1} r(x_i) \qquad (9)$$

と定義する．

問題は列車運行のラインネットワーク $H_T=(X, \mathcal{L})$ が与えられたとき，指定されたノード $s\in X$ から他の任意のノードへの最小コストパスを求めることである．このような問題は貨物の中継や旅客の乗継ぎの問題として起きるが，前者においては中継作業量や平均中継時間などが中継コストの主成分となり，後者においては例えば平均乗継ぎ時間などが中継コスト，駅間所要時間が輸送コストとなる．後者は **4.** の最短時間列車乗継ぎ問題に対してその対象範囲を限定するための予備的問題と考えることもできる．

5.2 アルゴリズム

この問題は上述のラインネットワーク H_T において，各ノード $x\in X$ をそれを通るラインに対応したノードに分割し，それらの間を長さ $r(x)$ のラインで相互に結合したラインネットワーク H_T' を作り，これにアルゴリズム SPATH を適用して解くことができる．しかし，ここでは H_T そのものを用いたより効率のよいアルゴリズム SPATH-X について述べる．

SPATH-X は SPATH の SCAN ルーチンを一部拡張することによって得られる．**Fig. 8** に SPATH-X のための SCAN を示す．ここでは Fig. 5 の SCAN の文 2 を 2′ で置きかえたものを拡張する．すなわち，**Fig. 9** に示すように，x を中心とするライン l のラベリング操作において，

$$\mathrm{DL}[y]+r(y)\geq \mathrm{DL}[x]+d_l(x, y)+r(x) \qquad (10)$$

を満たす範囲のノードのうち，

```
    procedure SCAN (x, l);
1   begin
2     for y∈l & y≠x such that
          DL[y]+r(y)≥DL[x]+d_l(x, y)+r(x) do
3       if DL[y]>DL[x]+d_l(x, y)+r(x) then
4       begin
5         DL[y]=DL[x]+d_l(x, y)+r(x);
6         NL[y]=x;
7         LL[y]=l;
8         QLIST=QLIST∪{y};
9       end
10  end SCAN;
```

Fig. 8 Procedure SCAN revised for SPATH-X

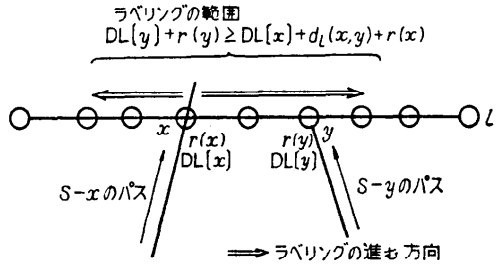

Fig. 9 Labeling in SPATH-X

$$\mathrm{DL}[y]>\mathrm{DL}[x]+d_l(x, y)+r(x) \qquad (11)$$

となるもののみラベル変更をするようにする．このラベリングのプロセスは，x から l にそって(10)を満たさないノードに遭遇するまで続けるものとする．

5.3 アルゴリズムの正当性

本節においては，アルゴリズム SPATH-X が有限回の演算で終了することおよびアルゴリズムの終了時に最適パスを与えることを証明する．SPATH-X において $r(x)=0(x\in X)$ とすれば SPATH そのものとなるから，アルゴリズム SPATH の正当性も同時に示される．

定理 1 アルゴリズム SPATH-X は有限回の演算で終了する．

証 明 あらゆる場合において，SCAN ルーチンは明らかに有限回の演算で終了する．SCAN は QLIST から取りだされたノードについてその (位数 −1) 回だけ call されるから，ノードが QLIST に入れられる延べ回数が有限であることを示せばよい．

各 $d_l(x, y)$ および $r(x)$ は非負の有理数，したがって，また，整数と仮定しても一般性を失なわない．また，各 DL[x] の初期値は有限の値 $M(<\infty)$ と仮定してよい．1つのノード $x\in X$ が QLIST に入れられるのは DL[x]の値が変化したたときのみであり，そのとき DL[x] の値は確実に 1 だけ減少する．したがって，いずれかのノード $y\in X$ が QLIST に入れ

られる延べ回数が M を越えるとすれば，DL[y]<0 となる．しかるに，このアルゴリズムで DL[y] の値が変化を受けるのは非負の値による置きかえのみによってであるから，これは矛盾である．　(Q. E. D.)

補助定理　アルゴリズム SPATH-X の終了時において，同一ライン上の任意の2つのノード x, y に対して次の関係が成りたっている．

$$\mathrm{DL}[y]\leq \mathrm{DL}[x]+d_l(x,y)+r(x)\quad x,y\in l \tag{12}$$

証　明　アルゴリズムの終了時において(12)を満たさない $x, y\in l$ が存在するものとする．すなわち，

$$\mathrm{DL}[y]>\mathrm{DL}[x]+d_l(x,y)+r(x) \tag{13}$$

とする．

x が最後に QLIST から取りだされた状態を考える．このときまず x と y の中間に

$$\mathrm{DL}[z]+r(z)<\mathrm{DL}[x]+d_l(x,z)+r(x) \tag{14}$$

を満たす $z\in l$ が存在することを示す．実際，そのような z が存在しなければ，x からのラベリング操作が y までとどく．しかるに，この時点でも(13)が成りたっており，したがって，当然

$$\mathrm{DL}[y]+r(y)\geq \mathrm{DL}[x]+d_l(x,y)+r(x) \tag{15}$$

も成りたっている．これは x からのラベリングによって y がラベル変更を受けることを意味し，この時点で

$$\mathrm{DL}[y]=\mathrm{DL}[x]+d_l(x,y)+r(x) \tag{16}$$

とされる．これはアルゴリズムの終了時において(13)が成りたつことに反する．したがって，(14)を満たす $z\in l$ が x, y の中間に存在する．この関係(14)は，右辺がそれ以後変化しないから，終了時においても成りたつ．したがって，これと関係(13)から，終了時において

$$\mathrm{DL}[y]>\mathrm{DL}[x]+d_k(x,z)+d_l(z,y)+r(x) \tag{17}$$

$$>\mathrm{DL}[z]+d_l(z,y)+r(z) \tag{18}$$

となる．したがって，x と y の代りに z, y について上と同じ議論が成りたつ．すなわち，この議論は巡回的である．しかるに，x と y の間の l 上のノードは有限個しかないから，これは矛盾である．　(Q. E. D.)

定理 2　アルゴリズム SPATH-X は，終了時において最小コストパスを与える．

証　明　SPATH-X の終了時において，任意のノード $x\in X$ に対して

$$\begin{array}{ccccccccc} x & \mathrm{LL}[x_1] & \mathrm{NL}[x_1] & \mathrm{LL}[x_2] & \cdots & \mathrm{LL}[x_{k-1}] & \mathrm{NL}[x_{k-1}] \\ \| & \| & \| & \| & & \| & \| \\ x_1 & l_1 & x_2 & l_2 & \cdots & l_{k-1} & x_k=s \end{array} \tag{19}$$

は x—s のパス μ_x を決め，DL[x] はそのパスのコストを与える．

今，あるノード $x\in X$ に対して DL[x] が最小コストとでない（μ_x が最小コストパスでない）とする．このとき x—s の最小コストパス $\bar{\mu}_x$ を

$$x=\bar{x}_1, l_1, \bar{x}_2, l_2, \cdots l_{k-1}, \bar{x}_k=s \tag{20}$$

とするとき，$\bar{\mu}_x$ 上で少なくとも DL[$\bar{x}_1$] は最小コストでなく，また，DL[$\bar{x}_k$] は最小コストである．したがって，パス $\bar{\mu}_x$ 上に

$$\mathrm{DL}[\bar{x}_i]\text{：最小コストでない}\ \&$$
$$\mathrm{DL}[\bar{x}_{i+1}]\text{：最小コスト}$$

となるようなノード $\bar{x}_i$ が存在する．しかるに，$\bar{x}_i$, l_i, $\bar{x}_{i+1}$ は最小コストパスの1部であるから

$$\mathrm{DL}[\bar{x}_{i+1}]+dl_i(\bar{x}_{i+1},\bar{x}_i)+r(\bar{x}_{i+1}) \tag{21}$$

はノード $\bar{x}_i$ に対するパスの最小コストを与える．DL[$\bar{x}_i$] は最小コストでないから

$$\mathrm{DL}[\bar{x}_i]>\mathrm{DL}[\bar{x}_{i+1}]+dl_i(\bar{x}_{i+1},\bar{x}_i)+r(\bar{x}_{i+1}) \tag{22}$$

となる．これは補助定理により，SPATH-X の終了という仮定に反する．したがって，すべての $x\in X$ に対して，μ_x は最小コストパスであり，DL[x] はそのコストを与える．　(Q. E. D.)

6. あとがき

以上，ラインネットワーク上の最短路アルゴリズムおよびその拡張について述べ，同時に線構造の優越した問題に対してラインネットワークが自然な表現手段を与えることを示した．

線構造との間の変換オーバヘッドを無視した純粋にアルゴリズムのみの効率に関していえば，sparse なネットワークに対して実際的なラインの設定をした場合，上記のアルゴリズム SPATH は一般に Bellman のアルゴリズムより多少効率が落ちるが Dijkstra のアルゴリズムより高速であることが実験的に確かめられる．更に，SPATH は線構造に関する変換オーバヘッドをほとんど含まないため，実際面における総合的な効率は高い．

ラインネットワークはその表現力においてグラフのそれを含むものであり，実際的な問題により接近した処理を可能にするとともに，ネットワーク構造をもつ問題のための言語あるいはそのプロセッサの開発に1つの有望なアプローチを与えるものと考えられる．

参考文献

1) S.E. Dreyfus : An Appraisal of Some Shortest Path Algorithms, Operations Research, Vol. 17, pp. 395～412 (1969).
2) M. Nakamori : A Note on the Optimality of

Some All-Shortes-Path Algorithm, J. Oper. Res. Soc. Japan, Vol. 15, pp. 201～204 (1972).
3) E. W. Dijkstra : A Note on Two Problems in Connection with Graphs, Numer. Math., Vol. 1, pp. 269～2711 (1959).
4) R.E. Bellman : On a Routing Problem, Quart. Appl. Math. Vol. 16, pp. 87～90 (1958).
5) B. Golden : Shortest Path Algorithms-A Comparison, Operations Research, Vol. 24, pp. 1164～1168 (1976).
6) 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).
7) 近谷，岡原：線構造によるネットワーク表現とその応用，情報処理，Vol. 19, No. 1, pp. 9～15 (1978)
8) C. Berge : Graphes et hypergraphes, Dunod (1970).
9) E. L. Johnson : On Shortest Paths and Sorting, Proc. ACM 25th Annual Conf., Aug. pp. 510～517 (1972).

(昭和52年5月6日受付)
(昭和52年9月5日再受付)