IV-26　最適ネットワーク問題に関する一考察

大阪市立大学工学部　正員　西村昂
大阪市立大学工学部　学生員　○柿木浩一

1．はじめに

最適ネットワーク問題とは，OD交通量と地点（ノード）の数とその幾何学的配置が与えられ，しかも総延長の上限（建設費などからくる制限）内で，例えば総走行台キロを最小にするネットワークを探索することである。この問題は費用便益分析において，費用（総延長キロLc）が一定のもとで便益を最大（総走行台キロTを最小）にするネットワークを探索することに相当する。最適ネットワークを求めるアルゴリズムは，すでに多くの人々によって提案されているが，ここでは限界効用を考慮して近似的に探索する方法を考えてみた。

2．これまでの最適ネットワーク問題へのアプローチ

これまでの解法では，小さなネットワークから大きなネットワークへ条件を満たすように改良していくforward手順と，逆に大きなネットワークから小さなネットワークへ改良していくbackward手順の二つに大別される。

(1) FORWARD ALGORITHM[1)]

与えられた地点に対してminimal spanning treeを作り（これに対する走行台キロをT_0とする），アークの数が増大すると総走行台キロが減少し，総延長キロの制限以内でTを最小にするネットワークを作る。このアルゴリズムは 1) treeのアークの集合から任意アークをとり，このtreeのco-graphから任意のアークを一本づつ加えてTを計算してためておく。2) 1)の操作をtreeのすべてのアークについて行ない min T > T_0ならTが最適ネットワークであり，min T < T_0ならmin Tが新しい解となる。次にtreeより一本アーク数の多いグラフについて同様の探索法をくり返し，順にネットワークを改良していく。

(2) BACKWARD ALGORITHM[1)]

与えられた地点に対して$\frac{n}{2}(n-1)$個のアークからなる完全連結グラフを作り，一つのアークを削除することによって目的関数の値の増加が最小になるようにアークの数を減らしていく。そして総延長がLcより小さくなるまで続けて最適解を得る。改良手順はFORWARD ALGORITM に類似している。

(3) 佐々木・前島論文[2)]

この論文は(1)のFORWARD手順と同じようなアルゴリズムでOD交通量の大なる地点間から順にアークを加えてゆき，n-1本のアークでtreeを構成する。このtreeに建設費が増加しないようにアークを加えゆき，建設費が減少しなくなれば，それを最適ネットワークとする。

3．最適ネットワークの考え方

最適ネットワーク問題で(1)や(2)は建設費を一定にして，道路網の評価基準を総走行台キロを選んでいる。(3)は建設費が区間交通量に比例（交通量と建設費が比例すれば走行台キロにも比例）するとして，評価基準に建設費を選んでいる。交通量と建設費がステップ関数的に比例する場合は，建設費が

最小のとき走行台キロは必らずしも最小にならないがほぼ近い値となる。このように近似的にほぼ最適ネットワークに近い近似解が、少ない計算量で得られるなら実用上さしつかえないと考えてよいであろう。本報告ではforwardとbackward手順のより近似的手法として次のアルゴリズムを考える。

(1) forward 手順

I. optimal spanning treeの探索。①地点間距離の短いアークから順に地点を結ぶ。②アークを加えることによってループを作るならば最後に加えたアークをはずして①の操作をすべてのノードが連結するまで続ける。(minimal spanning tree) このtreeの総延長キロL_0, 総走行台キロT_0を計算する。③treeのco-graphのアークを短い順にtreeに加え、できたループの中で長いアークから順に一本ずつはずし、各LとTを計算する。$L<L_c$で$T>T_0$なら③をくり返し、$T<T_0$なら限界効用$C=(T_0-T)/(L-L_0)$を計算し、③の操作をすべてについて行ないCの最大なネットワークを最適treeとする。このときのL, TをL_t, T_tとする。以後、アーク数がnの連結グラフを$g(n)$, その最適グラフを$g^*(n)$で表わす。

II. 最適tree $g^*(n-1)$のco-graphのアークを短い順に$g^*(n-1)$に加えて$g^*(n-1)$の近傍グラフ(アーク1本を入れ換えることによって得られるグラフを指すことにする)を探索する。このときのL, Tを計算し、$T\leq T_t$ならば$C=(T_t-T)/(L-L_t)$を計算してためておく。$T>T_t$なら次のアークを選んでIIをくりかえし、Cの最大を$g^*(n)$とする。以下$L<L_c$で同様に行って$g^*(n+1)$ ……を得る。

(2) backward 手順

I. 完全グラフ$g(\frac{1}{2}n(n-1))$を作り、その総延長L_0と総走行台キロT_0を計算する。

II. $g(\frac{1}{2}n(n-1))$のアークを長いものから順にとり、そのときのL, Tを計算する。$T<T_0$ならば$C=(T-T_0)/(L_0-L)$を計算してためておく。$T>T_0$ならば次のアークを選んでL, Tを計算する。すべてのアークについて行ないCの最小のものを$g^*(\frac{1}{2}n(n-1)-1)$とし、そのL, TをそれぞれL_1, T_1とする。同様にして$g^*(\frac{1}{2}n(n-1)-1)$の近傍グラフから$g^*(\frac{1}{2}n(n-1)-2)$を探索する。そして$L\leq L_c$になるまで$g^*(\frac{1}{2}n(n-1)-i)$の探索を続ける。

4. 計算例

地点数が4個でOD交通量と各地点間距離が次のmatrixで表わされている。このとき総延長キロの制限$L_c=26$キロとする。

$$
\text{OD交通量}=\begin{array}{c} \\ ① \\ ② \\ ③ \\ ④ \end{array}\!\!\begin{array}{c} ①\ ②\ ③\ ④ \\ \begin{bmatrix} 0 & 5 & 2 & 1 \\ 5 & 0 & 3 & 4 \\ 2 & 3 & 0 & 1 \\ 1 & 4 & 1 & 0 \end{bmatrix} \end{array}
\qquad
\text{地点間距離}=\begin{array}{c} \\ ① \\ ② \\ ③ \\ ④ \end{array}\!\!\begin{array}{c} ①\ ②\ ③\ ④ \\ \begin{bmatrix} 0 & 7 & 6 & 4 \\ 7 & 0 & 5 & 9 \\ 6 & 5 & 0 & 5 \\ 4 & 9 & 5 & 0 \end{bmatrix} \end{array}
$$

(1) forward 手順

(イ) $g_0(3)$はminmal spanning treeを示す。$g_0(3)$に(1, 3)のアークを加え、(1, 3, 4)のループができるのでそのループ内のアーク(3, 4), (1, 4)を一つずつとり除いたものを$g_1(3)$, $g_2(3)$に示す。

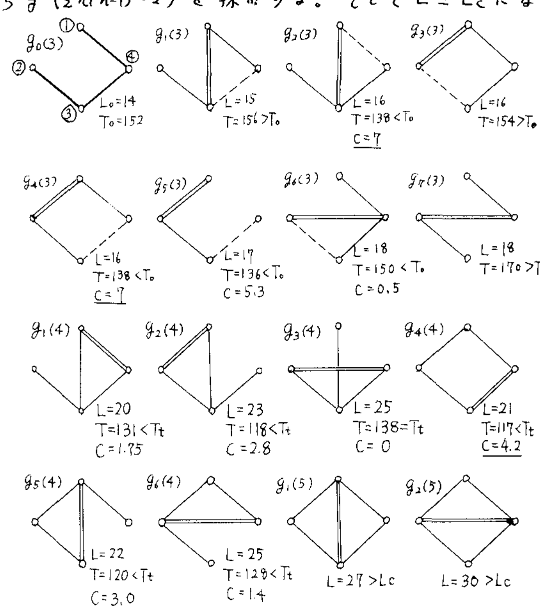

図-1 FoRWARD 手順

$T<T_0$のときCを計算しておいておく。同様にしてco-graphのすべてのアークについて行なったものが$g_3(3)$〜$g_7(3)$である。そして計算したCの最大なものは$g_2(3)$と$g_4(3)$だからこの場合2つの$g^*(3)$となり、それぞれ$g_1^*(3)$, $g_2^*(3)$とする。$L_t=16$, $T_t=138$となる。

(ロ) $g_1^*(3)$からアークを一本づつ加えて$g_1^*(3)$の近傍グラフを探索し、L, Tを計算し、$T<T_t$ならCを計算しておく。$g_1(4)$〜$g_3(4)$はその結果を示す。$g_2^*(3)$についても同様に計算したものを$g_4(4)$〜$g_6(4)$に示す。これら$g_1(4)$〜$g_6(4)$でCの最大なものは$g_4(4)$であるから$g_4(4)$を$g^*(4)$とする。

(ハ) $g^*(4)$からアークを一本づつ加えて$g^*(4)$の近傍グラフを探索し、(ロ)と同様に計算したものが$g_1(5)$, $g_2(5)$であるがともに$L>L_c$となり、最適ネットワークは$g^*(4)$となる。

(2) backward手順

(イ) ノード数が4個の完全グラフ$g(6)$を作り、そのL, TをL_0, T_0とする。(ロ) $g(6)$から各アークを一本づつとってL, T, Cを計算する。これらを計算したグラフが$g_1(5)$〜$g_6(5)$で、Cの最小のものは$g_6(5)$であるから、これを$g^*(5)$とする。このL, Tをそれぞれ$L_5=27$, $T_5=111$とおく。(ハ) $g^*(5)$の近傍グラフとして$g^*(5)$の各アークを一本づつとってL, T, Cを計算したグラフが$g_1(4)$〜$g_5(4)$である。Cの最小のものは$g_2(4)$であるからこれを$g^*(4)$とし、このとき$L<L_c$になったのでこれを最適ネットワークとする。

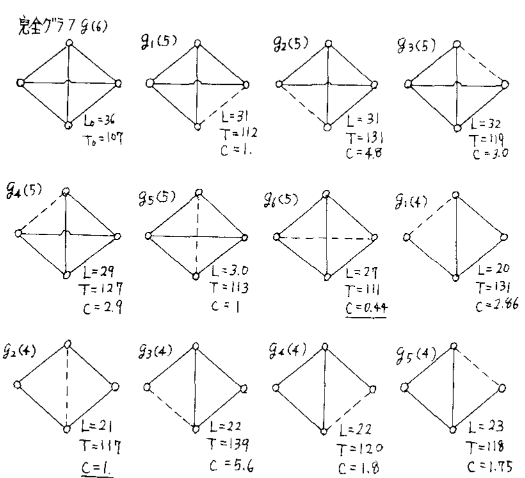

図-2 BACKWARD手順

これら(1), (2)の探索プロセスは図-3のようであり、限界効用Cは勾配を表わすから、これを利用してネットワークを改良することの有利性がうかがわれよう。なお2.(1)の手法で計算しても同じ最適ネットワークが得られた。

5、おわりに

本報告で示した近似的な手法と最初に紹介したFORWARD ALGORITHMを比べると最適treeの探索ステップ数はあまり違わないが$g(n)$, $g(n+1)$……となるにしたがってステップ数が少なくなる。計算例で$g(4)$のステップ数は前者が7ステップであるのに対し、後者は24ステップになり、計算量はかなり少なくなる。また本報告で限界効用の概念を導入したが、さらに検討する余地がある。

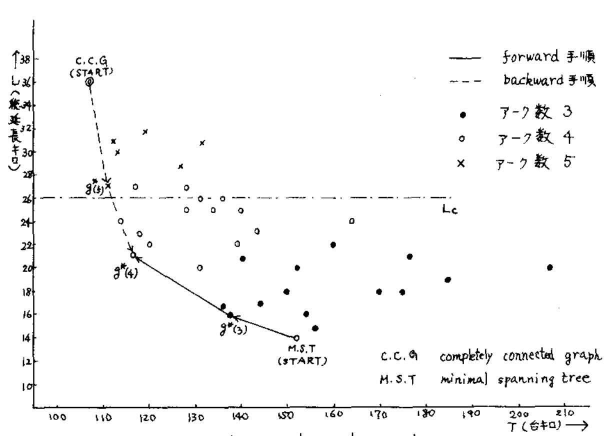

図-3 forwardとbackwardの探索プロセス

参考文献 1) Allen J. Scott, Combinatorial Programming Spatial Analysis and Planning, Methuen & Co. Ltd. (1971)

2) 佐々木綱、前島忠文、道路網形態に関する一考察、土木学会論文集 163号 (1969)