Ⅳ-25　大規模最適ネットワーク問題への適用を考慮した近似解法

神戸大学工学部　正員　森津　秀夫

1．はじめに

ある目的関数を設定し、最適な交通網案を求めようとする最適ネットワーク構成手法については、すでに多くの研究がなされている。近似解法を用いて実際問題を解いている例もある。しかし、対象とするネットワークの規模がさらに大きくなれば、従来の解法では近似解法でさえ計算時間が膨大になるため適用が困難である。そこで、ここでは大規模な最適ネットワーク問題に適用することができる近似解法について述べる。

2．大規模最適ネットワーク問題に対する近似解法

最適ネットワーク問題の規模が大きくなったときに計算時間が急激に長くなる原因は、目的関数を計算する回数がふえることと、目的関数の計算に含まれる最短経路探索に要する計算時間が増加することにある。これは近似解法においても同じである。単純なbackward法でも、各ステップでネットワークに含まれるリンクを除いた状態を調べるために、目的関数を繰り返し計算しなければならない。そこで計算時間を大幅に短縮するには、最短経路探索を含む繰り返し計算を必要としない解法を作成しなければならない。そのひとつの方法として、最大ネットワークで各リンクの評価値を求め、それを使って解を決定することが考えられる。

簡単のために、ここではネットワークの総延長の制約の下で、最短距離の合計を最小化する問題を対象とする。これは式(1)～(3)のように定式化できる。

$$\min\ Z = \sum_i \sum_j d_{ij}(x) \qquad (1)$$

$$\text{s.t.}\quad \sum_k l_k x_k \leqq L \qquad (2)$$

$$x_k = 0 \text{ or } 1 \qquad (3)$$

ここに、$d_{ij}(x)$は解xのネットワークにおけるノードi j間の最短距離，l_kはリンクkの長さ，Lは制約長すなわちネットワーク長の上限，x_kはリンクkがネットワークに含まれるとき1，含まれないとき0とする。

いま最大ネットワークにおけるノードi j間の最短距離をd^1_{ij}，第2最短経路の長さをd^2_{ij}とする。そうすれば、i j間の最短経路を成すリンクの少なくとも1本が存在しないとき，i j間の最短距離はd^2_{ij}以上になる。すなわち、i j間の最短経路は$d^2_{ij} - d^1_{ij}$の価値を持つと考えることができる。これを単位長さあたりの値に直し、各リンクの最短経路としての評価値を次式で定めることにする。

$$f'_k = \sum_i \sum_j \delta_{ijk}(d^2_{ij} - d^1_{ij}) / d^1_{ij} \qquad (4)$$

ここに，δ_{ijk}はリンクkが最大ネットワークのi j間の最短経路に含まれるとき1，含まれないとき0とする。

あるリンクがネットワークに含まれないときは，他のリンクが代わりにその機能を果たすことになる。そこでネットワークに含まれないリンクがあるとき，そのリンクの両端のノード間の最大ネットワークにおける第2最短経路が機能を代替するものとする。式(4)に修正を加え，式(5)でリンクの評価値を定める。

$$f_k = f'_k + \sum_{k'} (1 - x_{k'}) \gamma_{kk'} f'_{k'} l_{k'} / d^2_{i_{k'} j_{k'}} \qquad (5)$$

ここに，$\gamma_{kk'}$はリンクkがリンクk'の両端のノード$i_{k'}$，$j_{k'}$間の第2最短経路に含まれるとき1，含まれないとき0とする。

第2最短経路だけでなく，もっと多くの経路を考慮したり、ネットワークの状態に応じてさらに緻密な評価値を作成することも可能である。しかし，大規模ネットワークへ適用する場合，計算時間と記憶容量の増加を避けるには、このような簡単な評価値にすることが好ましいと考えられる。

基本的なアルゴリズムには，リンクを順次除去，あるいは付加するだけのbackward法とforward法を用い，それぞれ簡易backward法，簡易forward法と呼ぶことにする。各ステップでは，そのときのネットワークの状態に対して式(5)によりリンクの評価値を求め，除去あるいは付加するリンクを決める。ただし簡易forward法ではすべてのノードがつながるまではループをつくらないようにする。

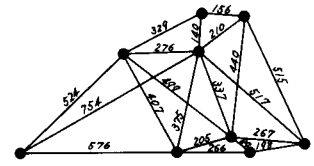
これらのアルゴリズムでは，最短経路探索を行うのは最大ネットワークでリンクの評価値を求めるときと，得られた解の目的関数を計算するときの合計2回だけである。

3．計算例

図－1の最大ネットワークでいくつかの制約長に対して解を求め，計算結果を示したものが表－1である。表－1では簡易backward法，簡易forward法のどちらも計算時間がbackward法より短く，得られた解は同程度であることがわかる。図－1以外の2つの最大ネットワークに対しても同じように計算し，それぞれの解法を用いてネットワーク案を決定したとき，最適解に対してどれだけの損失を被るかを求めてみた。その結果はbackward法が0.85%，簡易forward法が0.72%，簡易backward法が4.18%であった。簡易backward法が悪いのは，主として制約長が極めて小さいときに良い解が得られなかったためである。

以上の計算結果からは，得られる解の精度は簡易forward法ではbackward法とほとんど変わらず，簡易backward法も制約のきつい場合を除けば，ほぼ同じであるといえる。

次にネットワークの規模が大きくなったときに，計算時間がどのようになるかを調べてみる。図－2に示すネットワークパターンで，ノード数が4のときをネットワーク規模1とし，ネットワーク規模を変えて解を求める。その結果を示したのが図－3である。ただし簡易backward法は簡易forward法と大差ないので省略した。また比較のために1回の最短経路探索に要する計算時間も示した。これから明らかなように，簡易forward法の計算時間の増加はbackward法に比べるとはるかにゆるやかである。backward法を適用することが困難な規模のネットワークでも，簡易forward法では容易に解を求めることができるのがわかる。

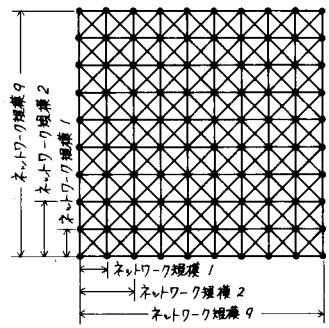


図－1　例題1の最大ネットワーク

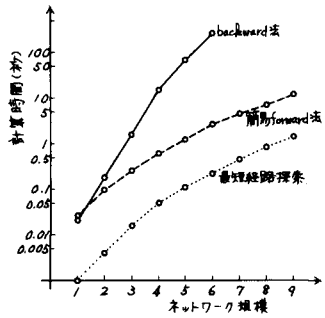
表－1　例題1の計算結果

（計算時間：ミリ秒）

制約長	backward法 目的関数値	backward法 計算時間	簡易backward法 目的関数値	簡易backward法 計算時間	簡易forward法 目的関数値	簡易forward法 計算時間	最適解 目的関数値
2500	22464	315	28050	82	21664	72	20720
3100	19868	302	19972	84	20561	82	19370
3700	19019	282	19123	80	19141	86	18942
4300	18677	248	18802	76	18662	91	18543
4900	18372	239	18463	80	18323	99	18196
5500	17982	189	17982	77	17986	102	17982
6100	17749	124	17749	69	17781	106	17749
6700	17749	126	17749	71	17709	111	17709



図－2　例題2の最大ネットワーク



図－3　例題2の計算時間

4．おわりに

ここで示した近似解法は，backward法と同程度の解を短い計算時間で求めることができるものである。ここでは簡単な最適ネットワーク問題をとり上げたが，他の最適ネットワーク問題についても同じような考え方を適用することができるであろう。そしてこれらの解法により，最適ネットワーク構成手法の適用範囲は大きく広がるであろう。