

IV-103 ネットワーク分割とバンドリングによる交通量配分の数値計算

京都大学大学院 学生員 鷹尾和享
 京都大学工学部 正 員 飯田恭敬
 愛媛大学工学部 正 員 朝倉康夫
 建設省 正 員 広川誠一

1. はじめに

大規模ネットワークを対象とした交通量配分には、膨大な計算時間が必要であり、従来から、計算手順の効率化やネットワーク表示の簡略化方法について多くの研究成果が報告されている。本研究では、I A法（Incremental Assignment）による交通量配分の計算時間短縮を目的とした『ネットワーク分割』の考え方に基づいて、さらに計算時間を短縮するためのネットワーク集約の方法を提案し、その数値計算例を示す。

2. ネットワーク分割による最短経路探索

交通量配分に要する計算時間の大部分は、最短経路探索に費やされている。最短経路探索法の中で、最も効率的であるとされている Dijkstra 法をそのまま用いると、配分計算時間は概ね（セントロイド数）と（ノード数の2乗）の積に比例して増加する。そこで、ネットワークをいくつかのサブネットワーク（ブロック）に分割し、ブロックごとに最短経路探索を行うことによって計算時間の短縮を図る方法が提案されてきた。¹⁾ この考え方を I A法に適用すると、その概略の手順は以下のとおりである。

Step.0 ネットワークをいくつかのブロックに分割する。

Step.1 各ブロックごとにセントロイド及び境界ノードの間で最短経路探索を行って経路をメモリーし、その経路を一本の仮リンクで置き換える。（リンク長：最短経路上の所要時間）

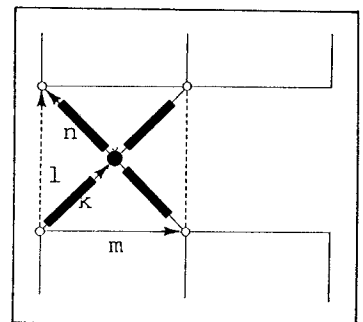
Step.2 仮リンクから構成される全域での仮ネットワークを作成し、All-or-Nothing配分を行う。

Step.3 仮リンクの交通量を Step.1 でメモリーした経路（真のリンクから成る）に再配分し、真のリンク所要時間を更新する。Step.1 へ戻り、必要回数だけ Step.1～3 を繰り返す。

ネットワーク分割数を増加させると、Step.1 に要する計算時間は減減し、Step.2では増増する。これらは互いにトレードオフの関係にあるので、全体の計算時間はある分割数で最小となる。²⁾ また、この方法では、正確に最短経路が求められるため、原則的にネットワーク分割を行わない場合と全く同じネットワーク交通流が得られる。しかし、Dijkstra法の計算手順におけるノード探索作業を改良すると、経路探索に要する時間が1オーダー短くなる。このとき、ネットワーク分割をよほど巧みに行なわないと、計算時間の短縮が期待できないこともありうる。

3. バンドリングの考え方と計算手順

ネットワークを単純にブロック分割すると、ブロック境界ノードが多数発生する。その結果、ブロック単位での最短経路探索回数の増加によって計算時間の短縮効果が半減したり、仮ネットワークの規模の拡大により、必要なメモリーサイズが増加する。そこで、境界ノードの数を減らすために、ネットワークを分割した際のブロック境界において『集計化した境界ノード』を作成し、計算時間短縮とメモリ節約を図ることを考えた。（図-1）集計化した境界ノードは、ブロック境界線と交わるリンクを束ねたものである。これは、2つのブロックに所属して、各ブロックの仮リンクの接点の役目を果たす。集計化した境界ノードと境界線に隣接する真のノードとの間は、束ねられた真のリンクの代わりにダミーリンクで結ばれる。このように境界線と交わるリンクを束ね、境界ノードとダミーリンクを作成する集計化作業のことをバンドリング



○ : 真のノード
 ● : 境界ノード
 破線 : 束ねられた真のリンク
 太線 : ダミーリンク

図-1 ブロック境界のバンドリング

（Bundling）と呼ぶ。ダミーリンクの所要時間は、束ねられたリンクとその近傍のリンクの所要時間を基にして定義する。したがって、束ねられた真のリンクの所要時間を順次更新するために、ダミーリンクの交通量を該当するリンクに負荷する必要がある。この作業は、先の Step. 3 の手順の中に含まれるが、その方法として、たとえば、ダミーリンク k の所要時間は、真のリンク l と m の平均とし、逆にダミーリンクのフローを真のリンクに還元する場合、リンク l のフローは k と n の平均とする方法が考えられる。

4. 数値計算例

まず、格子状ネットワークを用いて、その規模を順次拡大させ（最大ノード数5000個）、ノード数と最短経路探索に要する時間を調べた。その結果、Dijkstra法の手順を改良することにより、ネットワーク規模の1.1~1.2乗に比例する程度にまで計算時間を減らすことが可能であることがわかった。次に、ネットワーク規模をノード数 441個、セントロイド数25個、有向リンク数1680本に固定し、ネットワークを格子状のブロックに等分割し、その分割数を4, 16, 25, …… と順次増加させて分割数と計算時間の関係を調べた。その結果、単純に分割するだけでは、必ずしも計算時間が短くならないことが示された。

そこで、分割とバンドリングを併用した場合について計算し、計算時間と誤差を調べた。（表一1）なおIA法の繰り返し回数は、どのケースでも10回に固定して比較している。その結果は、次のようにまとめることができる。（1）ブロックに分割しバンドリングを行うと、分割しない場合に比べて、いずれも必要な記憶容量は増加し、計算精度が低下するものの、計算時間は20%~60%短縮される。CPU時間のうち 約2/3 が Step. 1、1/3 が Step. 2 で消費される。残差に占めるランダム誤差の割合が大きく、バンドリングによる系統的な誤差は生じていない。（2）ブロック分割数を4に固定して境界ノード数を変化させたとき、境界ノード数が2個程度（5本のリンクをバンドリングする）であれば精度の低下も小さく、計算時間も短く効率的である。（3）境界ノード数を1個に固定して分割数を増やすと、分割数16で計算時間は最小となるものの、誤差が増大することが避けられない。これは、バンドリングして作成したダミーリンクに関する諸設定が必ずしも十分ではなく、特定のリンクにフローが集中したためであり、改良の余地が残されている。

5. 今後の課題

（1）分割及びバンドリングによる方法を同じ計算時間で精度比較、（2）Frank-Wolfe法の計算手順に組み込んだ場合の数値計算 については、現在検討を進めている段階であり、稿を改めて発表する予定である。

表一1 数値計算の結果（CPU時間、メモリおよび精度）

分割数	境界 ノード	CPU TIME	内訳 Step1 Step2 Step3	メモ (KB)	相関 係数	RMSE ²	CV ²	総走行 時間
分割せず	—	121	— — —	60	—	—	—	—
4	5	106	69 32 5	146	0.937	.99	.99	+8.7
4	2	61	42 14 4	122	0.892	1.7	1.7	+11.7
4	1	48	35 10 3	131	0.800	3.2	3.0	+33.8
16	1	42	26 13 3	142	0.742	4.3	4.2	+64.8
49	1	75	43 28 4	241	0.615	6.6	6.6	+123.5

CPU TIME:繰り返し1回あたりの平均CPU時間（ミリ秒） 相関係数, RMSE², CV² :

リンクフローについて、分割しない場合の配分結果と比較した相関係数, 残差平方和（%）

ランダム誤差 ($\times 10^6$) 総走行時間:分割しない場合と比較した誤差（%）

参考文献 1)井上博司：道路網における等時間原則による交通量配分に関する基礎的研究，京都大学学位論文, pp.110, 1975 2)飯田・朝倉・広川・鷹尾：ネットワーク分割による交通量配分簡略化法の数値的検討，JSCE関西支部63年度講演会概要集，1988