

ネットワークの分割およびバンドリング による交通量配分計算の簡略化

Network Aggregation by Decomposition and Bundling for
Large Scale Traffic Assignment

飯田 恭敬 *, 朝倉 康夫 **, 広川 誠一 ***, 鷹尾 和享 ****

Yasunori IIDA, Yasuo ASAKURA, Seiichi HIROKAWA & Kazutaka TAKAO

As well as some effective algorithms, several methods of network aggregation have been proposed in traffic assignment process. The large scale actual network should be approximately represented for numerical analysis of network flow. This paper discusses the network aggregation based on the decomposition. It is shown that the simple decomposition may not be effective if the minimum path search algorithm is greatly improved. Then the authors propose the "bundling method" in order to decrease the number of nodes at the edge of the decomposed block. The actual nodes and links at the boundary are aggregated into one or a few dummy nodes and links. Some numerical examples show that the bundling method can approximately describe the network flow.

1. はじめに

交通計画の対象地域が空間的に拡大したことやより詳細な交通網の表示が求められるようになったことを背景に、交通量配分計算に用いられるネットワークは大規模化している。大規模ネットワークに対して配分計算を実行するには、多大な計算時間と計算機容量が必要であり、データ入力・結果表示のための作業量も膨大なものとなる。配分結果の精度を確保しつつ、計算費用を減らすためには、計算の効率化とネットワーク表示の工夫が必要となる。

ネットワーク集計化(Network Aggregation)の目的は、実際の交通網よりも規模(リンク、ノード数)の小さい計算用ネットワークを合理的に作成することにあると考えられる。具体的には、実ネットワークから部分的にリンクを抽出すること、さらに何本かのリン

クを束ねて仮想的なリンクを作成することである。このような作業は、実務における交通量配分においても、経験的に行われている。しかし、実ネットワークから配分計算用のネットワークを作成することに関して、必ずしも客観的な基準があるわけではない。

実ネットワークから不必要なリンクを削除するあるいは必要なリンクだけを抽出するための方法論、さらにその実証的な研究としては、Chan(1976), Haghani & Daskin(1983), Bovy & Jansen(1983), 枝村他(1981), 溝上他(1987) などがあるが、現在までのところ抽出や削除の明確な基準は得られていないように思われる。

一方、ネットワークの階層化(上位と下位のネットワークに分ける)による配分方法は、Eash et al. (1983), 内山他(1982)などにより研究されている。最近では、階層化とネットワーク分割による計算の効率化とを組み合わせた方法論が展開されるようになった。理論的アプローチには Daganzo(1980.a)の研究や、Hearn(1984)による Transfer Decomposition と呼ばれる方法がある。集計化ネットワークと詳細ネットワ

* 正会員 工博 京都大学教授 工学部交通
土木工学科 (〒606 京都市左京区吉田本町)

** 正会員 工博 愛媛大学講師 工学部
土木工学科 (〒790 松山市文京町3)

*** 千葉県庁

**** 学生員 京都大学大学院

ークの対応関係をつけるという発想は、すでに Chan (1976)にも見られるが、Hearn は利用者均衡問題の中でこれを理論的・明示的に取り扱った点で評価されている。(Friesz,1985) この方法の実用性については、加藤他(1987)に紹介されている。なお、ネットワーク分割による最短経路探索の効率性については、既に井上(1975)が論じているのを始め、分割・階層化による交通量配分は、林他(1982) が I A 法の中で用いているし、飯田・高山(1985) のメッシュ分割による方法の中にも同様のものが提案されている。

これらに対し、ネットワークの連続体近似による交通流解析が研究されており、これも大規模ネットワークの一種の集計化であると考えられる。たとえば、田口(1978), Daganzo(1980.a), Dafermos(1980) などである。現在までのところ操作性の点で難点があるが、高密度交通圏のマクロな解析には適切なものとなろう。

ところで、ネットワーク集計化法の良否を配分計算過程で判断するときの基準は、集計化しない場合と比較して、①計算精度が確保されるか ②計算時間が短縮されるか ③メモリが節約できるか などである。たとえば、Hearn の方法をそのまま適用すると、精度確保の面での問題はないが、計算時間は集計化しない場合よりも増加する可能性がある。また、実用的な視点からは I A 法と分割・階層化の組合せが有力な方法であるが、単純なネットワーク分割は効果的でないかもしれない。このような問題について、従来の研究報告例は必ずしも豊富とはいえない。

本研究は、分割・階層化によるネットワーク集計化法の系譜に位置するものであるが、具体的な研究の狙いは以下の通りである。

- ①ネットワーク分割の効果を、最短経路探索の効率化との関連から数値的に調べること。
- ②単純な分割が有効でない場合の新しい方法(ブロック境界の集計化)を提案すること。
- ③この方法を I A 法による交通量配分計算に組み込んだ場合の効果・精度を、数値的に調べること。

2. ネットワーク分割による交通量配分

2.1 ネットワーク分割による最短経路配分の計算手順

交通量配分では、どのような手法を用いても O D ペア間の最短経路探索は必須の手順である。そして、配分計算に必要な時間の大部分は、経路探索に費やされている。最短経路探索の最も効率的な方法であるとされている Dijkstra 法をそのまま用いた場合、1つのノードから他の全てのノードへの最短 tree を作成するた

めに要する時間は、ノード数の 2 乗に比例する。したがって、配分計算に必要な時間は(セントロイド数)と(ノード数の 2 乗)の積に比例することになるが、ネットワークが大規模化すればこの値は極端に増加する。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、配分計算の時間を短縮するというアイデアが提案された。(たとえば井上,1975)

ネットワーク分割を適用した All-or-Nothing 配分の手順は以下の通りである。

step.0: ネットワークを複数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およびブロック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。仮リンクのリンク長は最短経路上の所要時間とする。

step.2: セントロイド・境界ノードおよびそれらを直結する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する。セントロイド間で最短経路を探索し、O D 交通量を All-or-Nothing 配分して、仮リンクの交通量を求める。

step.3: 仮リンクの交通量を step.1 でメモリーした経路(真のリンクからなる)に再配分する。

ブロックに分割された真のリンクからなるネットワークを下位、仮リンクによる全域のネットワークを上位とすると、両者の関係は、下位ネットから上位ネットへ仮リンクの所要時間が与えられ(step.1)、上位ネットから下位ネットへセントロイド・境界ノード間の O D 交通量が与えられる(step.3)となる。

ネットワークを分割して求めた最短経路の所要時間は、ネットワークを分割しないで求めた最短経路所要時間と等しい。なぜならば、「ある 2 点間の最短経路は、その経路上の任意の 2 点間についての最短経路で構成されている」からである。ただし、等価な最短経路が複数存在する場合には、必ずしも経路が同じになるとは限らない。

ネットワーク分割数を増やすと、1つのブロックのノード数が少なくなるので step.1 に要する計算時間は減減するがブロック境界ノードが増えるので step.2 に要する時間は増増する。また、step.3 の計算時間は無視できるほど小さい。したがって、全体の計算時間はある分割数で最小となるはずである。

ネットワーク分割による最短経路配分過程を I A 法による交通量配分計算に組み込んだ場合は、step.3

の後でリンク所要時間を更新して step.1 にもどればよい。FW法では、step.3 の後で一次元探索とリンク所要時間の更新を行ってstep.1 にもどればよい。

2.2 数値計算例

2.2.1 ネットワーク分割の効果

仮想的な格子状ネットワークとODデータを用いて、ネットワーク分割による配分計算を行った。ネットワーク規模は、ノード 441個、リンク 1680本（有向リンク）、セントロイド 25個の中規模なものである。OD交通量はネットワークの平均的混雑率が 0.7 程度となるように与えている。配分手法はIA法であり、繰り返し回数は10回である。ネットワーク分割は、個々のブロックの規模が等しくなるようにネットワーク全体を単純に等分割するものである。

分割しない場合とネットワークの分割数を4分割から4.9分割まで変化させたときの計算結果をまとめたのが表 1である。なお、ここでは改良前のDijkstra法（付録1）により最短経路探索を行っているほか、仮リンク数を減らす工夫（広川,1988）を加えている。

(1)計算時間：

CPU時間は、繰り返し1回あたりの平均計算時間で、それを先に述べた3つの段階に分けて示した。なお、

真のリンクの所要時間の更新の部分は、step.3 に含めている。

step.1 では、分割数を増やすにしたがって計算時間が減少している。これは、各ブロックのネットワーク規模が小さくなっていくためである。また、分割数が多くなるにしたがって減少の割合が小さくなっている。これは境界ノードが複数のブロックに所属しているため、分割数を多くすると、境界ノードの数が増加し、それに伴って各ブロックのネットワーク規模の縮小の効果が小さくなるためである。

step.2 では、分割数を増やすにしたがって境界ノードが増加するために計算時間が長くなる。その値は、分割数が多くなるほど加速度的に増加している。また、一旦経路を記憶する(step.1)ので、真のリンクへの配分(step.3)にはほとんど時間がかかっていない。

その結果、全体の計算時間は1.6分割で最小となっており、計算時間からみたネットワークの最適分割数は1.6となる。その時の計算時間は、分割しない場合の約4分の1にまで短縮されている。

(2)メモリーサイズ：

分割数を増やすにつれて、ブロック数・境界ノード数・仮リンク数が増加するので、必要なメモリーサイズは次第に増加している。しかし、それでも、ブロッ

クに分割した場合に必要なメモリーサイズは、分割しない場合の数倍程度に抑えられている。仮リンクの本数は、分割数を増やすにしたがって、境界ノードの増加による影響を受けて次第に増加している。

表 1 ネットワーク分割による数値計算結果

分割数	CPU TIME(ミリ秒)				メモリー (KB)	上位ネットワーク規模 リンク数 ノード数		RMSE ² ($\times 10^5$)
	計	step1	step2	step3				
分割せず	1598 (1589)	—	—	—	58	—	(25)	—
4	818 (745)	761 (50)	52 (50)	5	152	842	57	.37
9	534 (382)	400 (126)	129 (126)	5	194	1241	105	.16
1.6	414 (229)	245 (161)	164 (161)	4	203	1399	121	.37
2.5	511 (189)	207 (296)	300 (296)	4	230	1429	177	.31
3.6	565 (161)	180 (377)	381 (377)	4	270	1670	201	.19
4.9	666 (150)	170 (487)	493 (487)	4	311	1794	232	.25

CPU TIME：繰り返し部分の1回あたりの平均時間とstep1,2,3の内訳

()内は最短経路探索に要した時間

上位ネットワークのノード数：セントロイド数と境界ノード数の合計

RMSE²：リンク交通量の残差平方和平均（分割しない場合を基準）

(3)誤差：

ネットワーク分割を行っても、原則的には分割しない場合と全く同じリンク交通量が得られる。見かけ上誤差が生じているのは、時間の等しい複数の最短経路が存在したことによるものであり、すべてランダムに発生した誤差である。

2.2.2 最短経路探索のアルゴリズムを効率化した場合

最短経路探索の計算時間がノード数の2乗に比例するようなアルゴリズムを用いると、単純なネットワーク分割を行うだけでも配分計算時間は短縮された。しかし、経路探索のアルゴリズムを効率化すると、最短treeの作成時間をノード数の1.1~1.2乗に比例して増加する程度に抑えることが可能となる。(付録)

このとき、単純なネットワーク分割では、必ずしも計算時間の短縮が期待できないことになる。なぜならば、ブロックに分割された下位ネットワークでの計算時間(step.1)が、ブロック分割数の大小に関わらずほとんど変わらなくなるのに対し、上位ネットワークでの計算に必要な時間(step.2)は、分割数を増やすと確実に余計にかかるからである。具体的な計算結果は、単純なネットワーク分割の効果が全く生じないことを示している。(表2)

3. バンドリングによる交通量配分

3.1 バンドリングの考え方とその方法

2.で述べたネットワーク分割による最短経路探索法を用いた交通量配分では、分割しない場合に比較して厳密性は失われず、配分精度も保証される。しかし、ネットワークを単純に分割しただけでは、ブロック境

界ノードが多数発生し、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する。すなわち、ネットワーク分割を巧みに行わなければ、必ずしも計算時間の短縮が期待できないということである。また、境界ノードが多いと、上位ネットワークの規模が拡大し、必要なメモリーサイズも増加することになる。

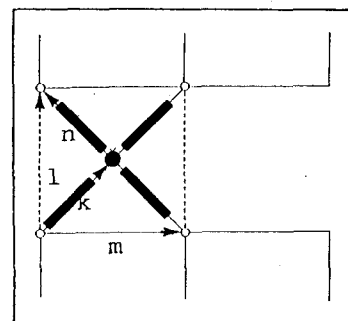
そこで、ネットワークをいくつかのブロックに分割し、上位・下位のネットワーク間でフローの受渡しを行うという基本的構造を変えることなく、配分精度を許容される範囲に収めて、計算時間の短縮とメモリーの節約を図るために、本研究では、ブロック境界ノードを減らす方法について検討を加えた。最も単純な方法は、ブロックとブロックの境界上に『集計化した境界ノード』を作成することである。

集計化した境界ノードとは、境界線と交わるリンクを束ねたものである。このノードは、2つのブロックに所属して、両ブロックの仮リンクの接点の役目を果たす。真のノードは1つのブロックだけに所属する。

(図1)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。このとき、真のノードからダミーリンクを2本以上引かないようにしなければならない。これは、ダミーリンクだけからなる経路に交通量が配分されることを防ぐためである。交通量ができるだけ真のリンクに配分されるようにすれば、ブロック境界の集計化による配分誤差が小さくなると考えられる。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一連の集計化作業のことを『バンドリング(Bundling)』と呼ぶ。

表 2 改良後のDijkstra法を用いた場合

分割数		CPU TIME(ミリ秒)			
		計	step1	step2	step3
改良前 分割せず		1598 (1589)	—	—	—
改良後	分割 せず	121 (112)	—	—	—
	4	150 (98)	113 (30)	32 (52)	5 (4)
	9	168 (92)	110 (52)	54 (58)	4 (4)
	16	166 (84)	101 (58)	61 (58)	4 (4)



○：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

ダミーリンクは、真のネットワークにおけるODペア間の経路ができるだけ変わらないように設定しなければならない。概念的には、均衡状態でのダミーリンクのフローパターンが均衡条件式を満たすようにすればよい。実際に配分を行うときの繰り返し計算の中では、フローパターンに対するダミーリンクの所要時間の設定方法が問題となる。これは、バンドリングされる真のリンクの走行時間関数を合成することであるが、リンクが直列または並列以外の場合に合成関数を明示的に求めることは難しい。

本研究では走行時間関数の合成は行わず、真のリンクの所要時間からダミーリンクの所要時間を求めることにした。このとき、繰り返し計算の中で、真のリンクの所要時間を更新するためには、ダミーリンクの交通量を境界線と交差する真のリンクに還元しなければならない。なぜならば、このリンクにはダミーリンクの交通量を負荷しない限り、交通量が流れないからである。具体的なダミーリンクの所要時間の設定法と、真のリンクへの交通量の受渡し法は次のようである。図1において、真のリンク l と m の所要時間から、ダミーリンク k の所要時間を決定する方法としては、2つの方法が考えられる。

- (a) k の所要時間を l と m の平均とする。これはODが同じブロックに所属する場合に、その経路がダミーリンクを通らないようにするためである。
 - (b) k の所要時間を l と m の大きい方の $1/2$ とする。
- 一方、ダミーリンクの交通量を真のリンクに還元する方法には、次の方法が考えられる。
- (v) リンク l の交通量は k と n の平均とする。
 - (i) 同一境界ノードへ向かうダミーリンクの交通量の和を真のリンクへ均衡配分する。すなわち、1つの境界ノードに束ねられている真のリンクについて、境界線を並列に横切っているものと仮定し、並列リンクの均衡交通量の求め方を適用して配分する。

3.2 バンドリングを用いた最短経路配分計算の手順

ネットワーク分割に加えて、ブロック境界をバンドリングしたときの、All-or-Nothing 配分の手順は以下のとおりである。

- step.0: ①ブロックの境界に境界ノード・ダミーリンクを作成する。
- ②各ブロックごとにセントロイドと境界ノードとの間に仮リンクを作成する。
- ③ダミーリンクの所要時間を定義する。

- step.1: 各ブロックごとにセントロイドおよび境界ノードの間で、最短経路探索を行い、経路となるリンクをメモリーし、その経路を仮リンクとする。
- step.2: 仮リンクから構成される上位ネットワークについて、セントロイドごとに最短経路探索を行い、仮リンクにOD交通量を配分する。
- step.3: 仮リンクのリンク交通量を、step.1でメモリーした経路(下位ネットワーク)に配分する。
- step.4: ①ダミーリンクのフローを真のリンクに負荷する。
- ②真のリンクの所要時間を更新する。
- ③ダミーリンクの所要時間を更新する。
- ④step.1へ

以上の手順は、2.で述べたネットワーク分割による最短経路探索を用いた配分の計算手順と同じであり、step.0とstep.4でダミーリンク設定の手順が追加されているだけである。そのため、この方法はIA法やFW法の中の最短経路配分手法として組み込むことができる。

バンドリングを用いた計算法では、ブロック境界ノードが少なく済むため、計算時間の短縮とメモリーの節約が期待できる。しかし、ダミーリンクがブロック境界線に沿って引かれるため、交通量がダミーリンクに集中し、ある程度の誤差が生じることは避けられない。

3.3 数値計算例

先の数値計算例で用いたものと同じネットワークを用いて、提案したバンドリングの効果を数値的に検討する。具体的には、①各ブロック1辺あたりの境界ノード数、②ネットワーク分割数、③ダミーリンク作成方法がそれぞれ異なる場合の計算時間、メモリー、計算精度を比較・考察する。交通量配分手法は、IA法(繰り返し回数10回)である。

3.3.1 1辺あたりの境界ノード数による差異

ネットワークの分割数を4分割に固定して、各ブロックの1辺あたりの境界ノード数が異なるときの計算結果をまとめたものが表3である。ダミーリンクは、いずれの場合も、所要時間を方法(a)で更新し、リンク交通量を方法(v)により還元している。

計算時間は分割しない場合に比較してかなり短縮でき、1辺あたりの境界ノード数が1個の場合は、分割しない場合の40%であった。境界ノード数を増やす

と、step.1,2ともに計算時間が増加するので、全体の時間も増加し、1辺に5個の境界ノード数を設けると分割しない場合とあまり変わらなくなる。メモリーは、単純な分割だけの場合に比べるとバンドリングにより節約されているが、分割しない場合の2倍程度は必要である。

分割しない場合のリンク交通量を基準にした計算精度を見ると、いずれの指標においても、境界ノードを減らすことによる精度低下が生じている。境界ノードが1辺に1個の場合は、相関係数が0.8に低下するほか、特定のリンクに交通量が集中しやすくなるので、残差や総走行時間の誤差も大きくなっている。誤差のほとんどは、ランダム誤差によって占められ、系統的な誤差は生じていない。

3.3.2 ネットワーク分割数による差異

表4は、ブロック境界ノードの数をブロックの大きさにかかわらず1辺に1個とし、ネットワーク分割数を変えたときの計算結果をまとめたものである。ダ

ミーリンクは、3.3.1と同様の設定である。

計算時間は16分割で最小（分割しない場合の1/3程度）となっているが、4分割に比べるとその差は小さい。これは、分割数の増加とともにstep.2の計算時間が増加することに加え、step.1の計算時間もあまり短くならないことによるものである。また、分割数の増加とともに、上位ネットワークの規模が大きくなるので、メモリーも増加している。

計算精度は、分割数を増やすにつれて、いずれの指標も悪くなっている。これは、境界部分のダミーリンクの数が相対的に増加し、しかもそれらの設定が必ずしも充分でなかったために、交通が特定のリンクに集中したことによると考えられる。

3.3.3 ダミーリンクの所要時間設定による差異

真のリンク走行時間からダミーリンクの所要時間を設定する方法(a,b)と、ダミーリンクの交通量を真のリンクに還元する方法(7,4)を組み合わせたときの計算結果をまとめたものが表5である。なお、計算時

表3 1辺あたりの境界ノード数による計算結果の差異

分割数	境界 ノード	CPU TIME(ミ秒)				メモリー (KB)	上位ネットワーク規模			精度			
		計	step1	step2	step3		リンク1	リンク2	ノード	R	RMSE ²	CV ²	TT
分割せず	—	121	—	—	—	60	—	—	25	—	—	—	—
4	1	48	35	10	3	131	152	252	29	0.800	3.2	3.0	+33.8
4	2	61	42	14	4	122	156	392	33	0.892	1.7	1.7	+11.7
4	5	106	69	32	5	146	160	1004	45	0.937	.99	.99	+8.7
4	BLせず	150	113	32	5	154	—	842	57	0.977	.37	.37	+1.0

表4 ネットワーク分割数による計算結果の差異

分割数	境界 ノード	CPU TIME(ミ秒)				メモリー (KB)	上位ネットワーク規模			精度			
		計	step1	step2	step3		リンク1	リンク2	ノード	R	RMSE ²	CV ²	TT
分割せず	—	121	—	—	—	60	—	—	25	—	—	—	—
4	1	48	35	10	3	131	152	252	29	0.800	3.2	3.0	+33.8
16	1	42	26	13	3	142	408	268	49	0.742	4.3	4.2	+64.8
49	1	75	43	28	4	241	672	588	109	0.615	6.6	6.6	+123.5

リンク1:ブロック境界ダミーリンク数(本), リンク2:ブロック内仮リンク数(本), ノード:セントロイド数+境界ノード数(個)

精度:リンク交通量について、分割しない場合と比較したときの誤差指標。

R:相関係数, RMSE²:残差平方和平均($\times 10^5$), CV²:ランダム誤差($\times 10^5$), TT:総走行時間(%)

間やメモリーはほとんど変わらないので省略している。

相関係数および残差で見ると、最も良好であるのが(a)と(7)の組合せによるものであるが、いずれの方法も計算精度が大きく異なるとはいえない。(a)と(7)の組合せによるものは、設定が最も簡単な方法でもあるので、ここに示したもののの中では良い方法といえる。ダミーリンクの設定は、バンドリング法の最も重要な箇所であるが、ここに提案した諸方法では必ずしも充分でなく、改良の余地が多く残されている。

表 5 ダミーリンクの設定方法による精度比較

所要時間 作成法	真のリンクへの 交通量還元法	精度			
		R	RMSE ²	CV ²	TT
k: (1+m)/2	1: (k+n)/2	0.800	3.2	3.0	+33.8
k: max(1,m)/2	1: (k+n)/2	0.721	4.5	4.1	+31.6
k: (1+m)/2	均衡配分	0.757	4.0	3.6	+28.9

分割数は4分割、1辺の境界ノード数は1個

4. おわりに

本研究で得られた主な成果は、以下の通りである。

- ①ネットワークの単純な分割による交通量配分法では、分割しない場合に比較して誤差は生じない。しかし、最短経路探索のアルゴリズムを改良することにより、分割しない場合の計算効率上がるので、分割による計算時間の短縮が期待できない場合も起こりうる。
- ②分割したネットワークのブロック境界ノード数を減らすことが、計算時間やメモリーの節約に寄与するため、境界上のノードと関連するリンクを束ねて仮想的なリンクとノードを作成し、交通量配分を行う方法（バンドリング法）を提案した。この方法は、IA法やFW法に組み込んで使うことができる。
- ③バンドリング法では、交通流に誤差が生じることは避けられないが、ある程度の誤差を許容すれば計算時間の節約に効果的であることがわかった。
今後に残された課題は、以下の通りである。
- ①分割数を増やすとバンドリング法の精度が落ちるので、ダミーリンク設定法を改良する必要がある。
- ②混雑率が精度に及ぼす影響を分析する必要がある。
- ③計画対象地域は分割・バンドリングせず詳細なネットワークのまま取り扱い、周辺地域は適当に分割・バンドリングするという方法が实际的である。このような方法について、検討する必要がある。

④ネットワーク交通流をマクロに解析するうえでは、ネットワークの連続体近似による方法が有力な方法（とくにデータ入力と結果の表示）である。バンドリングしながら分割数を増加させることは、連続体近似モデルの数値解析手順と類似したものとなるため、両者の関連について調べる必要がある。

【付録】Dijkstra法による最短経路探索の計算時間

最短経路を求める途中の段階で、ノードは次の3つの集合に分けられる。

集合1：起点ノード（ホームノード）からの最短経路が見つかったノードの集合

集合2：起点ノードからの経路が少なくとも1つ見つかったノードの集合（それが最短経路かどうかはまだ判らない）

集合3：残りのノードの集合

計算手順は以下の通りである。

- 7) 初期設定：全てのノードを集合3に入れる。ホームノード(H)を集合1に入れる。
- 7) 集合1に移されたノードを始点とするすべてのリンクLの終点ノードNを考える。Nが集合3に属するならば、Nを集合2に移し、Lによる経路およびHからの距離を記憶する。Nが集合2に属するならば、既知の経路と、Lを用いてHからNへ至る経路とを比較し、後者が前者よりも短ければ記憶していた経路およびHからの距離をLによるものに入れ換え、また、そうでなければLを捨てる。
- 7) 全てのノードについて、それが集合2に属しているかどうかを調べ、属しているノードのうちHからの距離が最短のノードを集合1に移す。全てのセントロイドが集合1に入るまで(7)～(7)を繰り返す。
この手順をそのままプログラミングすると、1つのホームノードからの最短トリーを作成するための計算時間は、おおむね [ノード数の2乗] に比例し、そのうち95%以上が(7)のステップで費やされる。これは、集合1へ移すノードを探すのに全てのノードを抽出するからである。
- しかし、全てのノードを抽出する必要はなく、集合2のノードだけを抽出すればよい。さらに、ノードが集合3から集合2へ移る時点、および集合2に属する段階で記憶する経路が入れ替わった時に、Hからの距離が短い順に並べておけばより効率的になる。その際、新たに集合2に入ったノードは既に入っているノード

よりもHからの距離が長いことが多いので、集合2のノード列の最後尾から距離を比較して挿入位置を決めると比較の回数が少なくて済む。このようにすれば、1つのホームノードあたりの計算時間は改良前に比較して1オーダー短縮でき、リンク長の分散にもよるが、[ノード数の1.1~1.2乗]に比例することになる。

付表1は、格子状ネットワークにおいて規模(ノード数)を拡大したときの、計算時間の比較結果である。ノード数が5000のとき、改良前に比較して改良後は計算時間が約1/100に短縮されている。なお、この方法は筆者らが独自に開発したものであるが、伊理他(1983)に示されている「ラベル確定法」の「ヒープ法」にほぼ対応するものであることがわかった。

付表1 最短経路木の作成時間(セトリド1個当たり,ミ秒)

ノード数	改良後	改良前	ノード数	改良後	改良前
36	0.2	0.5	1681	17.5	892.6
121	0.8	5.2	2116	23.0	1408.1
256	1.9	21.9	2601	29.6	2128.1
441	3.4	63.4	3136	36.6	3080.0
676	5.6	146.7	3721	50.4	4392.2
961	9.2	295.6	4356	59.4	5985.7
1296	12.8	533.0	5041	73.4	7974.2

【参考文献】

- Bovy P.H.L. & Jansen G.R.M. (1983):
Network Aggregation Effects upon Equilibrium Assignment Outcomes; An Empirical Investigation
Transp. Sci., Vol.17, pp.240-262
- Chan Y. (1976):
A Method to Simplify Network Representation in Transportation Planning,
Transp. Res., Vol.10, pp.179-191
- Dafermos S.C (1980):
Continuum Modelling of Transportation Networks
Transp. Res., Vol.14B, pp.295-301
- Daganzo C.F. (1980.a):
An Equilibrium Algorithm for the Spatial Aggregation Problem of Traffic Assignment,
Transp. Res., Vol.14B, pp.221-228
- Daganzo C.F. (1980.b):
Network Representation, Continuum Approximation and a Solution to the Spatial Aggregation Problem of Traffic Assignment,
Transp. Res., Vol.14B, pp.229-239
- Eash R.W., Chon K.S., Lee Y.J., Boyce D.E. (1983):
Equilibrium Traffic Assignment as an Aggregated Network for Sketch Planning,
Transp. Res. Rec., 944, pp.30-37
- 枝村・森津・木下・樋口(1981):
配分対象道路網作成の自動化
土木計画学研究・講演集, No. 3, pp.341-349
- Friesz T.L. (1985):
Transportation Network Equilibrium, Design and Aggregation: Key Developments and Research Opportunities,
Transp. Res., Vol.19A, pp.413-427
- Haghani A.E. & Daskin M.S. (1983):
Network Design Application of an Extraction Algorithm for Network Aggregation
Transp. Res. Rec., 944, pp.37-46
- Hearn D.W. (1984):
Practical and Theoretical Aspects of Aggregation Problems in Transportation Planning Models in Transportation Planning Models (Florian ed.)
North Holland, pp.257-287
- 林・林・野口(1982):
階層的経路探索を用いた大規模道路網配分計算の簡略化手法, JSCE年次学術講演会概要 IV, pp.395-396
- 広川誠一(1988):
大規模ネットワークの交通量配分計算の効率化に関する方法論的研究, 京都大学修士論文
- 飯田・朝倉・広川・鷹尾(1988):
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関西支部講演概要
- 飯田・朝倉・広川・鷹尾(1988):
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, JSCE年次学術講演会概要集 IV
- 飯田・高山・横山(1985):
メッシュ分割によるネットワーク表示の簡略化手法を用いた交通量配分計算法
土木計画学研究・論文集, No. 2, pp.149-156
- 井上博司(1975):
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
- 伊理編(1983):
地理的情報の処理に関する基本アルゴリズム,
OR学会報文シリーズ・T-83-1, pp.147-153
- 加藤・宮城・松葉・真野(1987):
Bendersの分解原理を応用した交通配分手法の実用性, 土木計画学研究・講演集, No.10, pp.399-406
- 溝上・松井・二宮(1987):
道路網の集約化が配分交通量の推定精度に与える影響, 土木計画学研究・講演集, No.10, pp.377-384
- 田口東(1978):
交通網の分析手法, OR学会誌, 12, pp.756-763
- 内山・林・横谷・大島(1982):
大規模交通ネットワークにおける経路探索の簡略化手法に関する研究
土木計画学研究・講演集, No.4, pp.413-419
- Wilson E.M., Matthais J.S., Betz M.J. (1974):
A Traffic Assignment Planning Model; the Load-Node Concept, Transp. Res., Vol.8, pp.75-84