

名古屋大学工学部 正員 林 良嗣
愛知県 正員 林由紀夫
名古屋大学大学院 学生員 野口宏一

1. はじめに

交通需要分析において、全体の計算時間のかなりの部分が配分計算に費やされている。大都市圏における大規模な交通網を対象とする実際の交通計画においては、多数の計画代替案を比較分析し十分に検討を加えることは決して今日までの配分モデルや計算機の性能の向上をもってしても容易なことではない。本研究では、道路網での交通配分のために、階層的経路探索法を用いて許容誤差レベルに応じて簡略計算を行う方法を提案する。この方法は、多数の代替案を比較する場合には特に有効なものである。なお、ネットワークを階層的に分割して解く方法は、実際の計画ではいくつか試みられているが、それらのほとんどはかなり恣意的な方法であり、計算は簡略化されるがそのためにどの程度の誤差が生じるのかは不明である。¹⁾

2. 階層的経路探索法

2-1 経路探索手法

- (1) 対象となるネットワーク(図1-a)を、境界を横切る道路本数になるべく少なく、また境界上になるべく多くのセントロイドが位置するように分割する(図1-b)。
- (2) 各小ネットワーク毎に、内部のセントロイドとスクリーンノード(ここで、小ネットワークの境界となるノードもスクリーンノードと呼ぶことにする)の間の最短経路探索を *Dijkstra* 法により求める(図1-c)。
- (3) 各小ネットワーク内で求めた最短経路を、新たにリンク置き、セントロイドとスクリーンノードのみをノードとしたネットワークを構成する。そして、セントロイド間の最短経路を求める(図1-d)。

2-2 計算作業量の従来の方法との比較

計算作業量は、従来の方法では式(1)、本方法では式(2)によって概算される。²⁾

$$F_1 = C N^{K_1} \quad (1)$$

$$F_2 = \sum_{i=1}^m \{ (C_i + S_i) N_i^{K_1} \} + C(C+S)^{K_2} \quad (2)$$

ここに、 C ：セントロイド数、 N ：ノード数、 N_i 、 C_i 、 S_i ：ゾーン i のノード数、セントロイド数、スクリーンノード数(セントロイドを除く)、 m ：ゾーン数、 $1 < K_1 < 2$ 、 $1 < K_2 < 2$

次に実際に、名古屋市道路網(図2)を例にとって計算作業量を比較してみる。このネットワークは125ノード、224リンク、16セントロイドで、1ノードから出る平均リンク数は3.6本である。いま、このネットワーク全体を *Dijkstra* 法で解く場合(ケース0)と、階層的探索法で厳密な最短経路を求める場合(ケースA)とを比較したものが表1である。表中のノード数 a とCPU時間 e の関係より、CPU時間は a^k ($1 < k < 2$) に比例していることがわかる。その結果、16本の最短経路樹を作るのに要するCPU時間は195ms → 127msと35%短縮されているのがわかる。

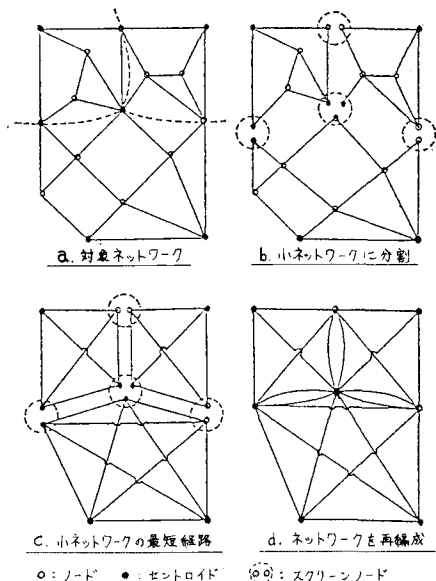


図1 階層的探索法の手順

3. 階層的経路探索法を用いた配分簡略化法

階層的経路探索法を用いると、厳密な経路所要時間を求める過程においてある程度のCPU時間の短縮が図れることを前節で示した。本節では、上述の階層的経路探索法の特徴をいかして許容誤差レベルに応じて経路所要時間の近似解をより短時間で求め、配分計算を簡略化する方法について述べる。ここでは前節で用いたのと同じ名古屋市の道路網において配分誤差とCPU時間も比較する。

表2は、図2のネットワークのスクリーンノードのうち、セントロイドと重っていない幾つかのノードを、スクリーンノードとせず、に最短経路を求めたときのflow-independentな場合とflow-dependentな場合の近似計算例である。ケースAごう平均トリップ長は33分、平均リンク交通量は48591台/日である。ここでは、各ケースについて誤差が得られている。flow-dependentな場合のケースCでも誤差平均は平均トリップ長の15%でしかない。このことは交通量についても同様なことがいえる。これは、サブネットワーク内のODには所要時間の誤差がなく、ゾーン境界を通過するODについても一部は影響がない場合があるためである。

次に、多数の路線代替案の比較について考える。このためには、上述の誤差以外に、一部のリンクを修正した場合に誤差がどの範囲にまで及ぶかを予め調べておくことにより(図3はその例)、影響が許容誤差以下となるサブネットワークについては経路探索の再計算が不要となる。大規模ネットワークではこのような再計算の省略できる部分が相対的に多くなり、分析全体として大幅な節約になる。

4. まとめ

本稿では、名古屋市の幹線道路網を例として示したが、本方法はその解法からわかるように、本来はより大規模で、ネットワークの何箇所かにネック(例えば、橋など)のある実際の広域道路網に対してさらに有効であると言える。なお、flow-independentな鉄道網の場合には、より有効であることが既に確かめられている³⁾最後に、貴重な助言をいただいた河上省吾教授に感謝の意を表する次第である。

<参考文献>

- 1) たとえば、久保田・斎藤・大井：交通量の路線配分手法の改善に関する調査研究、日本道路協会 1981。
- 2) 伊理正夫他編：ネットワーク構造を有するレゾナンス問題の電算機処理に関する基礎研究、日本オペレーションズリサーチ学会 1973.3
- 3) 林・橋本・大島・中村：大規模鉄道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、土木学会論文報告集(投稿中)

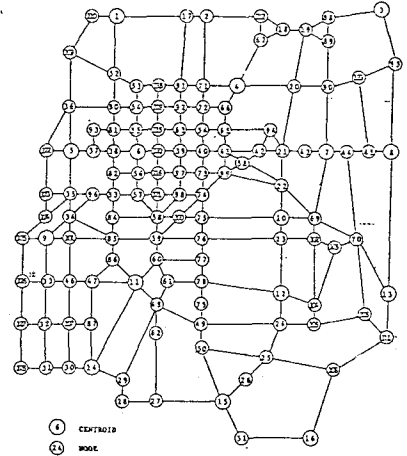


図2 名古屋道路ネットワーク

表1 ノード数によるCPU時間

	ノード数 a(個)	リンク数 b(本)	セントロイドとスクリーンノード間のリンク数 c(個)	1ノード5出るリンク数 d=2b/a(本)	スクリーンノード間のリンク数 e(個)	CPU時間 f=cwe(MS)
ケース0	125	224	16	3.6	12	195
ケースA	ゾーン1	14	17	6	2.4	0.1
	ゾーン2	17	24	9	2.3	0.1
	ゾーン3	22	28	9	2.5	1
	ゾーン4	25	41	7	3.3	1
	ゾーン5	69	114	17	3.3	4
総合ネット	26	224	16	18.8	2.3	127

表2 CPU時間と誤差

ケース	OD間所要時間				交通量		CPU (MS)
	flow-independent		flow-dependent				
	誤差平均 (min)	誤差標準偏差 (min)	誤差平均 (min)	誤差標準偏差 (min)	誤差平均 (台)	誤差標準偏差 (台)	
B	0.54	2.10	0.75	1.16	2870	3655	93
C	1.73	3.39	4.28	4.21	7731	9903	72

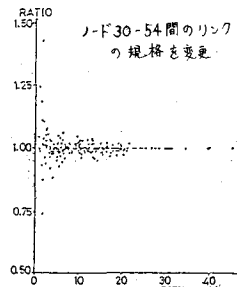


図3 リンク規格の変更による所要時間の変動の分布