

最短ツリーの記憶による交通量配分の経路探索の効率化*

Acceleration of Shortest-Path Search in Traffic Assignment by Memorizing Shortest-Path Trees*

鷹尾和享**・朝倉康夫***

By Kazutaka TAKAO**・Yasuo ASAKURA***

1. はじめに

交通量配分には種々のアルゴリズムがあるが、その多くは多数回の繰り返し計算を必要とする。その過程で最短経路探索が何度も行われるため、配分計算全体のパフォーマンスは最短経路探索の実装方法に強く影響される。実務においては一般にネットワークのサイズが大きく、長い計算時間を要する。そのうえ、たとえば、高速道路のハーフランプや、リンクコストの足し算では表現できない(non-link-additive)料金体系のような、実際のネットワークの様子を正確に表現することがしばしば要求され、そのためにネットワーク表現が複雑になり、それが配分計算をさらに重くしている。

本稿は、前回の iteration での最短ツリーを記憶することで、最短経路探索を高速化する方法について述べる。ラベル修正法は最短経路探索のアルゴリズムの一つであり、探索の過程でノードに付いた長距離ラベルを、より短い距離のラベルに修正していくことで、ある根ノードからの最短ツリーを得る。記憶しておいた前回の最短ツリーを利用し、各ノードのコストを今回の iteration 用に更新すると、その値は、長距離のラベルが付くのを阻止するための足切りラインとして用いることができ、ラベル修正の回数を減らすことができる。

これまで、配分計算を高速化する種々の試みが行われてきたが、既存の報告は、繰り返し回数が少ない場合や、小さなネットワークでの計算例しか報告していない場合が多い。しかし、実務では、配分計算をできるだけ短時間で終わらせることは切迫した問題である。したがって、本稿では、実際のネットワーク上で実際の繰り返し回数で提案手法を適用した場合のパフォーマンスを調べることを重視する。すなわち、本稿の目的は、最短ツリーの記憶法を実際のネットワーク表現に適用し、

そのパフォーマンスを報告することである。今日では、PC のメモリサイズが大きくなり、1GB 規模のメモリを確保することも難しくない。したがって、豊富なメモリをいかに上手に使うかを議論する良い時期である。

2. 関連研究

Frank-Wolfe(FW)法は、大規模なネットワークに簡単に適用できることから、広く用いられている配分手法である。いくつかの研究例が FW 法の改良を試みている。たとえば、補助解のフローパターンを改良する方法(Lee and Nie, 2001)¹⁾や、ステップサイズを改良する方法(Weintraub *et al.*, 1985)²⁾があり、また、別の新しいアルゴリズムを用いる方法(Dial, 2006)³⁾もある。

FW 法の実行過程では最短経路探索が何度も行われる。これまでに多くの最短経路探索のアルゴリズムが開発されているが、その多くは最短経路探索の問題をおのの探索別に単独で取り扱っており、iteration 間で組み合わせることに着目しているのものはあまりない。

本稿で述べる方法に似た方法を Dial (2006)⁴⁾が提案しており、また、Mahmassani *et al.* (1993)⁵⁾も短く触れている。それは、マルチクラスの最短経路探索において、あるクラスの最短ツリーを用いて他のクラスの最短経路探索を高速化するものである。Dial の方法は、クラス間で走行時間が異なるリンクは全体うちのごくわずかしかなことを利用している。この、クラス間でのツリーの記憶という方法は、iteration 間での最短経路探索の高速化に簡単に応用できる。ただし、FW 法が収束していないうちはステップサイズは 0 ではないため、ほとんどのリンクの走行時間が前回の iteration と異なる。

3. ラベル修正法とツリーの記憶

ラベル修正法は効率のよい最短経路探索アルゴリズムである。探索の間、ノード n にはラベルと先行ポイントが付いている。ラベルとは、現在探索されているうちで最短の、根ノードからの所要コストであり、先行ポイントとはそれに対応する経路を表す。「スキャン」操作とは、新しく見つかった経路が現在のラベルより低コスト

* キーワード：交通量配分、ネットワーク、最短経路、アルゴリズム、ラベル修正法

** 正員 博士(工)

(社)システム科学研究所 (〒604-8223 京都市中京区新町通四条上ル小結棚町 428 新町アイエスビル TEL: 075-221-3022, FAX: 075-231-4404)

*** 正員 工博

神戸大学大学院 工学研究科 教授

かどうかを調べる操作であり、nの下流ノードmの現在のラベルと、新しく見つかったnからの経路のコストを比較し、新しい方が低コストであれば、mのラベルを修正し、mをキュー（下流ノードをスキャンする必要のあるノードのリスト）に加える。この操作を繰り返すことで最短ツリーを求める。

この時、mが既にキューから取り出されていた場合、mは再度キューに加わることになる。このような場合、mの下流ノードが連鎖的にキューに再加入することになり、その多くがラベル修正を引き起こし、それがさらにキューへの再加入を引き起こす。したがって、最短でないラベルが一時的に付いてしまうと、非効率的である。

そこで、前回の iteration の最短ツリーを記憶しておくことで、この現象を緩和することができる。リンク走行コストは前回の iteration と異なるため、各ノードの根ノードからの所要コストは、今回の iteration 用に更新しなければならない。その結果、記憶したツリーは最短ツリーではなくなる可能性がある。しかし、最短ではないにせよ、そのコストの経路が存在すると解釈することができる。したがって、更新後の所要コストを足切りラインとして利用し、それより大きいコストのラベルが付くのを阻止することができる。

手順をまとめると次のようになる。

- (1) 初回は普通に最短経路探索を行う。ただし、求めたツリーは記憶する。
- (2) 次回以降は足切りラインを利用しながらラベル修正法を行う。全ての根について次の操作を行う。
 - (2-1) ツリーの所要コストを更新する。これが足切りラインとなる。
 - (2-2) もしスキャンされたコストが足切りラインよりも大きければ、無視する。そうでなければ、通常通りのラベル修正法を行う。

足切りラインを利用する段階で、所要時間が全く同じ経路が複数存在した場合、スキャンの順序が通常のラベル修正法と異なる場合がある。その結果、たとえばFW法の場合、収束が不十分だと、配分結果のリンクフローは異なる場合がある。

FW法に関して言えば、繰り返しの初期の頃は一次元探索のステップサイズが大きい傾向がある。そのため、記憶したツリーの更新後の所要コストは、現在の iteration の最短コストと比べてかなり大きくなる傾向があると言える。その場合、足切りラインが効率よく働かない可能性がある。しかし、繰り返しの重ねるにつれてステップサイズは小さくなっていくので、足切りラインは効率的に働くようになり、計算時間は劇的に改善していくと期待できる。

一方、ツリーの記憶のために追加的に必要となるメモ

リは次のように見積もることができる。

$$\text{追加メモリ量} = \text{先行ポイントのバイト数} \times \text{ノード数} \times \text{セントロイド数}$$

4. 実際的なネットワーク表現

(1) 交差点展開

現実の道路網を正確にモデル化したい場合、ネットワーク表現も複雑になり、最短経路探索も重くなる。

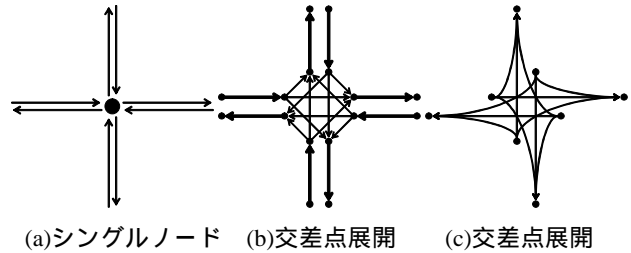


図 - 1 交差点展開

現実の道路網では、片方向しか出入りできないようなランプ（ハーフランプ）や、右左折制限のある交差点が存在する。このような様子を正確に表現したい場合、図 - 1 (a) のようなシングルノードの形では最短ツリーが単純なツリーにならない場合がある。そこで、(b) のように交差点展開をすれば、この問題を解決することができる。さらに、(c) のように、より簡略化された形にすることもできる。詳細は Takao *et al.* (2006)⁶⁾ を参照されたい。

(2) Entry-Exit モデル

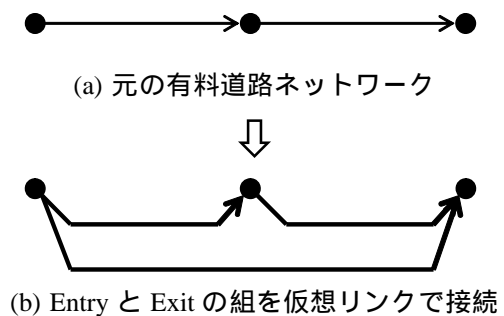


図 - 2 Entry-Exit モデル

現実の有料道路は link-additive ではない料金体系になっている場合がある。たとえば、長距離の利用区間の料金が幾分逓減されている場合がある。その場合、長距離トリップの場合は極力有料道路を走り続ける方が最短コストになることがある。そのような場合、最短ツリーは単純なツリーにならない場合があり、通常の方法では長

距離の利用区間の料金や最短経路をうまく表現できない。そこで、Yang *et al.* (2004)⁷⁾は、各利用区間を仮想リンクに置き換える Entry-Exit モデルを提案した(図 - 2)、詳細はTakao and Higashi(2006)⁸⁾を参照されたい。

5. 数値計算例

(1) テストしたモデル

提案手法の性能を見るため、実際のネットワークを用いて配分計算を行った。次の3つのネットワーク表現モデルを用いて評価を行った。それぞれについて、提案手法のツリー記憶法と、通常のラベル修正法とで FW 法の利用者均衡配分を行って性能を比較した。

Link-Additive モデル...交差点展開, link-additive 料金
Entry-Exit モデル...交差点展開, entry-exit モデル
単純モデル...シングルノード, link-additive 料金

単純モデルは各ノードの流入リンクと流出リンクの全ての組み合わせは通行可能とするモデルである。

用いたネットワークは近畿地方のネットワークであり、その規模は表 - 1 に示す通りである。セントロイドの数は 1140 である。

FW 法の繰り返し回数は 100 回とした。初期実行可能解が 1 回分あるので、セントロイドあたりの最短経路探索の回数は 101 回である。Pentium 4 (630) 3GHz の PC を使用し、プログラムは Visual C++ 6.0 でコンパイルし、Windows XP 上で実行した。

表 - 1 ネットワーク規模

指標	モデル		
	Link-Additive	Entry-Exit	単純
ノード数(元 network)	10850		
ノード数(経路探索)	31680	31465	10850
リンク数(元 network)	15328		
仮想リンク数	0	34114	0
リンク数(経路探索)	67218	75211	30529

(2) Link-Additive モデルの結果

表 - 2 に Link-Additive モデルの結果を示す。「通常の」とは、ツリーを記憶しない従来的方法である。比較のため、Dijkstra 法(ラベル確定法)による結果も示した。通常のラベル修正法でも Dijkstra 法より高速であるが、提案手法はそれよりもさらに 3.4 倍高速である。

これは、ラベル修正回数が劇的に減少するからである。繰り返しが進むにつれてステップサイズが小さくなり、足切りが効率的に働くようになり、ラベル修正回数の減少に寄与する(図 - 3)。

表 - 3 にラベル修正プロセスの内訳の回数を示す。これは iteration 1 回あたりの回数を出したものである(後半の 51 ~ 100 回目の平均)。足切りの結果、ラベル修正によるキューへの再加入の回数が 1/700 に劇的に減少している。また、その結果、ラベル修正回数のみならず、スキャンの回数の減少にもつながっている。

表 - 2 Link-Additive モデルの結果

用いた最短経路探索	計算時間(分)	メモリ(MB)
通常の Dijkstra 法	197.6	25.0
通常のラベル修正法	104.4	25.0
ツリー記憶ラベル修正法	30.3	166.5

繰り返し 100 回

表 - 3 内訳回数 (Link-Additive モデル)

	通常	ツリー記憶
足切り	0	7,721,977
ラベル修正, queue 再加入	533,398,211	767,187
queue 内でラベル修正	51,390,646	532,578
スキャンのみ	599,297,270	29,488,624

1 回あたり

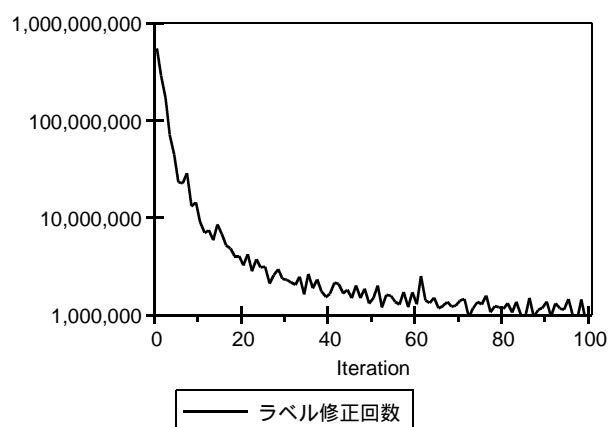
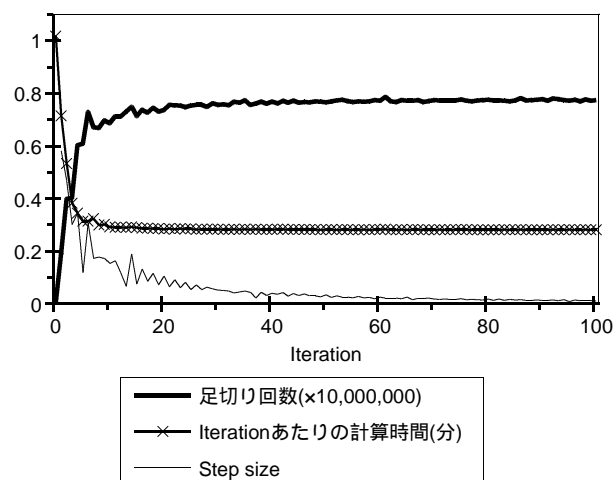


図 - 3 ラベル修正回数・関連指標の推移 (Link-Additive モデル)

その一方で、必要メモリはかなり増加することになるが、166.5MB は最近の PC で十分実行可能なサイズである。

(3) Entry-Exit モデルの結果

Entry-Exit モデルの結果を表 - 4 に示す。ツリー記憶法は通常のラベル修正法より 1.7 倍高速である。なお、仮想リンクの効果で通常のラベル修正法でも幾分速くなっており、ツリー記憶の効果は幾分小さくなっている。

表 - 4 Entry-Exit モデルの結果

用いた最短経路探索	計算時間(分)	メモリ(MB)
通常の Dijkstra 法	219.5	72.8
通常のラベル修正法	54.9	72.8
ツリー記憶ラベル修正法	32.0	217.3

繰り返し 100 回

(4) 単純モデルの結果

紙面の都合で詳細は省くが、単純モデルでもツリー記憶法は通常のラベル修正法より 2.5 倍高速になった。

6 . 考察

ツリー記憶の効果はネットワーク形状によって異なると考えられる(詳細は Takao and Asakura (2007)⁹⁾を参照されたい)。たとえば、高速道路の長いリンクは遠方のノードを直接結ぶため、通常のラベル修正法では根から遠いノードが比較的早い時点でキューに入る傾向がある。このため、高速道路を持つネットワークは提案手法の効果が大きいと思われる。

また、提案手法は一見 Dijkstra 法にも適用可能なように思われるが、高速化の効果はない。その理由は、Dijkstra 法の最小コストのノードを選ぶ操作が、大きいコストのラベルが付くのを既に防いでいるからである。したがって、提案手法との組み合わせによって、ラベル修正法が速度の面でより有利になると言える。

7 . 結論

本稿では、最短ツリーを記憶することでラベル修正法を高速化し、その効果を実ネットワーク上で確認した。本稿の成果は次のように要約される。

- FW 法をラベル修正法で組む場合、前回の iteration の最短ツリーを記憶することで高速化が図れる。
- 記憶したツリーは、長距離のラベルが付くのを防ぐための足切りラインとして利用できる。
- FW 法の計算時間は大幅に改善した。本稿の計算例では 1.7 ~ 3.4 倍高速化した。

本稿は FW 法について述べたが、本手法を分割配分法に適用しても効果があることを確認している。

なお、最近、FW 法と転換率法を組み合わせる配分法の採用が検討されているが、転換率配分法も別の意味において経路の記憶を必要とする方法であるため、実際的な制約により、本稿の提案手法とは両立できない恐れがある。

参考文献

- Lee, D.H. and Nie, Y.: Accelerating Strategies and Computational Studies of the Frank-Wolfe Algorithm for the Traffic Assignment Problem, Transportation Research Record, 1771, pp. 97-105, 2001.
- Weintraub, A., Ortiz, C., and Gonzalez, J.: Accelerating Convergence of the Frank-Wolfe Algorithm, Transportation Research Part B, 19(2), pp. 113-122, 1985.
- Dial, R.B.: A Path-Based User-Equilibrium Traffic Assignment Algorithm That Obviates Path Storage and Enumeration, Transportation Research Part B, 40(10), pp. 917-936, 2006.
- Dial, R.B.: An Efficient Algorithm for Building Min-Path Trees for All Origins in a Multi-Class Network, Transportation Research Part B, 40(10), pp. 851-856, 2006.
- Mahmassani, H.S., Hu, T.Y., Peeta, S., and Ziliaskopoulos, A.: Development and Testing of Dynamic Traffic Assignment and Simulation Procedures for ATIS/ATMS Applications, Technical Report DTFH61-90-R-00074-FG, http://www.itsdocs.fhwa.dot.gov/JPODOCS/REPTS_TE//7884.pdf, 1993.
- Takao, K., Higashi, T., Yasuda, K., and Asakura, Y.: Network Representation and Shortest Path Reflecting the Ramp Entry or Exit Direction Limitation, in Proceedings of the 13th ITS World Congress, London, U.K., 1787.pdf, 2006.
- Yang, H., Zhang, X., and Meng, Q.: Modeling Private Highways in Networks with Entry-Exit Based Toll Charges, Transportation Research Part B, 38(3), pp. 191-213, 2004.
- Takao, K. and Higashi, T.: Traffic Assignment in a Real Network with Non-Link-Additive Toll Charged Expressway, in Proceedings of the 11th HKSTS, Hong Kong, pp. 411-420, 2006.
- Takao, K. and Asakura, Y.: Acceleration of Traffic Assignment with Practical Network Representations by Shortest-Path Tree Memorization, in Proceedings of the 12th HKSTS, Hong Kong, pp. 165-174, 2007.