

起点別リンク交通量の部分的更新による利用者 均衡配分の効率的求解アルゴリズムの提案

杉浦 聡志¹

¹正会員 北海道大学准教授 工学研究院 (〒060-8628 北海道札幌市北区北 13 条西 8 丁目)

E-mail: sugiura@eng.hokudai.ac.jp

本研究は静的、確定的利用者均衡配分の求解効率化を目指して、Frank-Wolfe 法 (FW 法) における各繰り返し演算におけるリンク交通量の更新をランダムに選択した一部の起点のみに限定することで、計算負荷を軽減する方法を提案するものである。一般的に FW 法では繰り返し計算において全ての起点別リンク交通量を更新するが、一部の起点に限定することで、最短経路探索の実行回数は確実に低減できる。また、起点の部分抽出により、繰り返し演算時の振動を緩和させ、均衡解への到達を早めることも期待できる。提案手法を小さなネットワークに適用し、ランダム要因による解の安定性と FW 法と比較した計算速度の向上を確認した。

Key Words: 利用者均衡配分, 起点別リンク交通量, 効率的アルゴリズム

1. はじめに

利用者均衡(User Equilibrium: UE)による経路選択規範と仮定する交通量配分は道路ネットワーク分析の適用事例も多く、実務上の重要性は広く認知されている。Beckmann et al.¹⁾により定式化され、Frank-Wolfe 法 (以下、FW 法という) による解法 (Frank and Wolfe²⁾, Leblanc et al.³⁾ が与えられて以降、計算速度の向上を目的とした UE 配分求解アルゴリズムについては多くの研究が蓄積されてきた。

起点別に bush と呼ばれる有向非巡回グラフを構成し、bush 内でのフロー調整により全体の収束を目指す手法は Dial⁴⁾, Bar-gera⁵⁾ によって提案され、Class bush アルゴリズムとして Nie⁶⁾ により整理されている。Bar-gera により提案された TAPAS⁷⁾ は OD 間経路中に生じる交差する経路ペア (PAS) を検出し、これに対して均衡が成立するようにフローを改善することで全体の均衡を目指す。改良した iTAPAS⁸⁾ なども提案されている。これらの起点別の解法、TAPAS は FW ベースの手法と比べて高速に収束することが報告されているが、単一品種のフローにのみ対応している。実用的には大型車の交通が施策上重要になるケースも多く、未だ FW 法ベースの手法も必要される場合も多いだろう。

本研究は FW 法をベースとした UE 均衡配分の高速化を目指した手法を提案する。具体的には FW のリンクコストを更新するたびに実行される全起終点間の最短経路探索を、ランダムに選択された一部の

起点のみに限定する。これにより最短経路探索の実行回数の削減でき、直接的な演算時間の低減を期待する。また、FW 法では降下ベクトル探索において、所要時間が短くなった経路に全ての OD を付加する一方で、提案手法ではその一部を付加するため、繰り返しにおける解の振動を抑制することも期待できる。この方法は解の更新方向ベクトルを改良する手法として解釈することができ、Conjugate direction Frank-Wolfe Methods⁹⁾ や、Bi-conjugate Frank-Wolfe Methods⁹⁾ などのステップサイズの決定方法を工夫した方法とも組み合わせることが可能である。これまで蓄積されてきた手法が何度演算しても同じ解が得られるのに対して、提案手法はランダムな要素を導入するために演算の度に異なる解が現れる可能性をもつ。したがって、数値演算で得られる解の安定性を確認することで手法の妥当性を示す。実数演算では Sioux Falls ネットワークを利用することとした。

2. 提案手法

(1) 変数定義

有向リンク集合 A , ノード集合 N を持つグラフ $G = (N, A)$ を考える。その他本稿で使用する変数の定義を表-1 に示す。

(2) 一般的な Frank Wolfe 法と提案手法の比較

図-1 に FW 法、図-2 に提案手法の疑似コードを示す。FW 法では図-1 中の 2 行目に記述された繰り返

1	Find initial solution using all or nothing assignment
2	Repeat iteratively
3	Update link cost
4	$\mathbf{p} = \mathbf{0}$
5	For every $(s, t) \in W$
6	Find (s, t) shortest path k
7	$p_{ij} += d_{st} \forall (i, j) \in k$
8	Find $\zeta = \arg \min f(\mathbf{x} + \zeta \mathbf{p})$
9	$\mathbf{x} = \mathbf{x} + \zeta \mathbf{p}$
10	Convergence test

図-1 Frank Wolfe 法の疑似コード

1	Find initial solution using all or nothing assignment
2	Repeat iteratively
3	Update link cost
4	Choose set of origins $\Phi \subset S$, and set of OD pair Ψ
5	$\mathbf{p}^s = \mathbf{x}^s$
6	$p_{ij}^s = 0 \forall s \in \Phi$
7	For every $(s, t) \in \Psi$
8	Find (s, t) shortest path k
9	$p_{ij}^s += d_{st} \forall (i, j) \in k$
10	Find $\zeta = \arg \min f(\sum_s (\mathbf{x}^s + \zeta \mathbf{p}^s))$
11	$\mathbf{x}^s = \mathbf{x}^s + \zeta \mathbf{p}^s$
12	Convergence test

図-2 提案手法の疑似コード

表-1 本稿で使用する変数定義

(i, j)	: ノード <i>i</i> からノード <i>j</i> に接続するリンク $(i, j) \in A$
\mathbf{x}	: リンク交通量ベクトル (未知変数) $\mathbf{x} = (x_1, x_2, \dots, x_{ A })$
\mathbf{p}	: 下降方向ベクトルの全リンク集合, その各リンク成分を p_{ij} とする
W	: OD ペア集合, その成分は起点, 終点の組 (s, t) とする
k	: ある経路に含まれるリンク集合
d_{st}	: OD ペア (s, t) 間の交通量
ζ	: 更新ステップサイズを示す変数
S	: 起点 <i>s</i> の集合
Φ	: 起点集合 <i>S</i> の部分集合
Ψ	: OD ペア <i>W</i> のうち, 起点が <i>S</i> に含まれるものの集合
\mathbf{p}^s	: 起点別の下降方向ベクトルの全リンク集合, その各リンク成分を p_{ij}^s とする
\mathbf{x}^s	: 起点別のリンク交通量ベクトル

し (以降アウトターープと呼ぶ) において 5-7 行目に示すように全ての OD ペアの最短経路探索を実行する必要がある。大規模なネットワークにおいてはこの最短経路探索の繰り返しの計算負荷が大きくなる。提案手法では図-2 中の 4 行目に示すように起点のうち一部を適当な数だけ抽出し, それに関連する OD のみについて最短経路探索を実行し, 8-9 行目に示すように得られた最短経路探索に (s, t) 間 OD 交通量を付加する。抽出されない起点別交通量については, 5-6 行目に示すように一つ前のア

ウトターープで得られた起点別リンク交通量ベクトルをそのまま利用し, 降下方向ベクトルの計算に利用する。すなわち, Φ に選択されなかった起点についてはその繰り返し回ではリンク交通量を更新しない。したがって, 1 回のアウトターープにおける最短経路探索回数は明らかに少なくなる。しかしながら, このことによって収束に必要な繰り返し回数は大きくなることも懸念される。これについて, 本提案手法の有効性を理論的に証明する方法は見当たらないため, 数値演算の結果にて示すこととする。

(3) 解の収束について

FW 法では繰り返しにより解が単調に改善されることが保証されている。提案アルゴリズムと FW 法の違いは, 任意に選択された一部の起点に関係するリンク交通量のみを更新する手順を繰り返すことのみである。すなわち, FW 法ではすべての変数について降下方向ベクトルに従い, 解を更新するのに対して, 提案手法では一部の変数のみを更新する。したがって, 提案手法は一度の繰り返しにおける解の改善量が小さくなるものの, FW 法で保証される解の収束が成立する。

3. Sioux Falls ネットワークにおける数値演算

(1) 計算条件

試算には Sioux Falls ネットワークを利用する。リンク

数は76, ノード数は24である. FW法, 提案手法のいずれもステップサイズ決定のための次元探索は黄金分割法を採用した. 収束条件は **Relative gap** により設定することとし, その値が 10^{-2} , 10^{-3} , 10^{-4} に初めて到達したときの演算時間をそれぞれ記録することとする. **Relative gap** は Nic⁶で採用された(1)式で算出する.

$$r = 1 - \frac{\sum_{st \in W} u_{st} d_{st}}{\sum_{ij} x_{ij} t_{ij}} \quad (1)$$

ここで, r : **Relative gap**, u_{st} : (s, t) 間の最小コスト, t_{ij} : リンク (i, j) のコストである.

提案手法においては, 各アウトーループにおいて解を更新する起点を選択する必要がある. ここでは, すべての起点に対して, 一様分布により選択する. 起点の数は24であり, 10%程度となる3つの起点を各繰り返し回で選択することとする. 演算は筆者の環境 (CPU: Ryzen3800X, メモリ: DDR4-3600, 64GB, OS: Windows10) で, Python によるコーディングで実装した. なお, 最短経路探索には Python のライブラリ `python-igraph` を利用した.

(2) 評価方法

提案手法はランダムな要因を含むため, 一度の試算で得られた結果のみで評価するのは適切でないだろう. そこで, 提案手法については100回実行し, 得られた解の統計量から演算結果の安定性を検証する. また, FW法との演算速度の比較については, 表示の簡単のため100回演算したうちの1つを使用する.

(3) 100回実行時における演算安定性評価と演算速度

提案手法を100回実行し, 演算中に **Relative gap** が設定した値を下回ったときの交通量について, 以下の観点で整理する. まず, 実行により得られた100個の結果について各リンク交通量の標準偏差, 変動係数を求め, その最大値を表2に示すように整理した. 100回の実行結果における $r < 10^{-4}$ に到達したときの変動係数の最大値は0.0018と非常に小さな値となっている. つぎに, リンク交通量のアウトーループごとの変動に着目する. 提案手法では, 少数の起点のみの解を更新するため, 1回のアウトーループにおけるリンク交通量の変動は大きくないことも生じる. そこで, 設定した **Relative gap** を下回ったときから, 過去10回分のアウトーループにおけるリンク交通量変動の平均をとり, その統計量を表3に示すように整理した. $r < 10^{-4}$ に到達したときのリンク交通量変動の平均は0.76, 標準偏差は1.10と非常に小さい値になった. 最大値, 最小値, 中央値の値を見ても, 演算結果のバラツキは非常に小さいといえる. 以上のことから, 提案手法の解は安定した結果を出力しているといえる. 100回実行時において設定した **Relative gap** が設定

表2 100回演算におけるリンク交通量結果の安定性

	$r < 10^{-2}$	$r < 10^{-3}$	$r < 10^{-4}$
リンク交通量標準偏差の最大値	245.31	37.84	10.61
リンク交通量変動係数の最大値	0.0430	0.0094	0.0018

表3 100回演算における過去10回のアウターループのリンク交通量変化量の平均値の統計指標

	$r < 10^{-2}$	$r < 10^{-3}$	$r < 10^{-4}$
最大値	255.44	47.87	8.39
最小値	5.1E-10	1.4E-09	8.6E-09
平均値	52.62	5.83	0.76
中央値	34.89	3.06	0.37
標準偏差	55.97	7.57	1.10
(参考) FW法	275.59	21.39	2.25

表4 100回演算における演算時間の統計指標

	$r < 10^{-2}$	$r < 10^{-3}$	$r < 10^{-4}$
最大値	1.08	2.97	10.15
最小値	0.68	1.99	6.55
平均値	0.83	2.46	8.48
中央値	0.83	2.45	8.54
標準偏差	0.07	0.18	0.68
(参考) FW法	0.66	2.55	16.33

した値に到達したときの演算時間は表4に示す結果が得られた. $r < 10^{-4}$ に到達したときの平均演算残時間は8.48秒, FW法での演算時間が16.33秒であることから, 2倍程度の演算速度が得られていることがわかる.

解のアウターループにおける推移を調べるために, 100回演算中のうち1つを取り出し, FW法と比較した. 提案手法はアウトーループ20000回演算までを記録して示す. **Relative gap** の推移を図3に, 目的関数の値の推移を図4に示す. **Relative gap** が 10^{-3} を下回るあたりからFW法と提案手法に解の改善速度に差が生じ始め, $r < 10^{-4}$ に到達する速度が大きく異なることがわかる. 目的関数の推移をみても, 提案手法は5秒程度で20000回目と同等の値に到達しているのに対して, FW法は60秒経過時でもその値に到達できていないことがわかる. 以上のことから, 提案手法が計算速度に対する解の改善に有効であるといえる.

(4) 起点の選択量による変化

以上までの検討では各アウトターープで選択する部分起点集合の数 $|\Phi|$ を 3 つとしたが、その値を変化させたときにどのような挙動となるかを確認する。選択する起点の数を 10%-50%相当になるように設定し、整数で切り上げた{3, 5, 8, 10, 12}の各ケースについて、100 回演算し、 $r < 10^{-4}$ に到達したときの演算時間を箱ひげ図として図-5に示す。中央値、最小値、最大値のいずれも選択する起点を 8 個としたときに最も演算速度が早くなるという結果が得られた。これは、選択する起点の量が少ないと、アウトターープ一度あたりに更新される解の改善が小さく、収束に至るまでに多くの繰り返しを要し、一方で大きすぎると FW 法と同様に解が振動することで演算時間が長くなったためであると考えられる。本検討で与えた条件では起点全体の 30%程度を更新するケースが最良となったが、この値についてはネットワークや OD パターンによって異なることが予想されるため、一般性をもつものではない。今後の検討蓄積によってこの設定値が一般性をもって決定できるものか判断する必要があるだろう。

5. おわりに

本稿では確定的利用者均衡配分の FW 法をベースとして、アウトターープにおいて解を更新する起点をランダムに選択することで、収束までの演算時間を小さくする方法を提案した。提案した手法は各アウトターープでの最短経路探索の実行回数を確実に縮小させること、アウトターープ間での解の振動を抑制することを期待している。Sioux Falls ネットワークに提案手法、FW 法をそれぞれ適用し、パフォーマンスを比較した。提案手法はランダムな要素を含むため、100 回実行して解の安定性を確認し、実行解ごとのバラツキが小さいことを確認した。

FW 法と比較して、演算時間は平均的に 2 倍程度早いことを示した。提案手法は FW 法と比べて解を更新する起点別リンク交通量を限定することのみ異なるため、各アウトターープにおいてステップサイズを改良する手法と組み合わせることでさらなる高速化も期待できる。

本稿で残された課題を以下に記す。本稿では提案手法を Sioux Falls ネットワークに適用したのみにとどまっている。各アウトターープで選択する起点の部分集合の数は試算中では全起点の 30%程度が最も演算速度が早いという結果が得られたが、ネットワークによってこの値は異なるものと思われる。したがって、多くのネットワークへの適用を通じて、この値の一般性を検討したい。本稿では各アウトターープで選択する起点を一様分布で選択することとした。選択する起点を目的関数の微分値

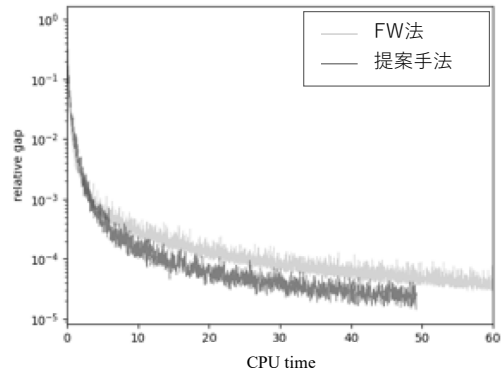


図-3 Relative gap の比較

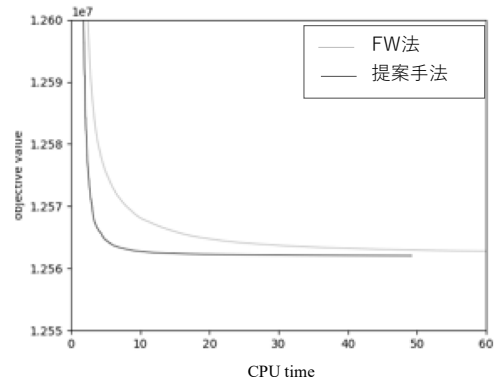


図-4 目的関数値の比較

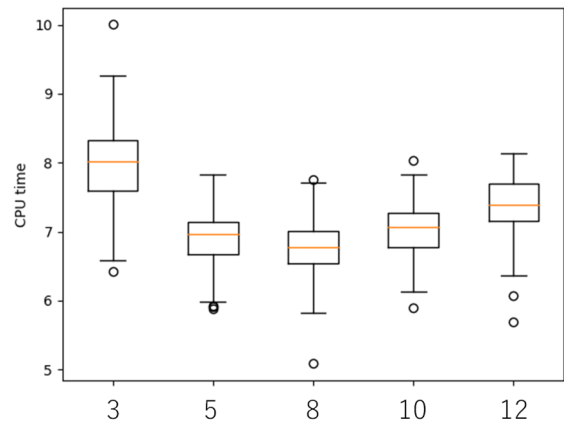


図-5 更新する起点の数による計算速度の比較

などを参照し、その重みによって確率を与える方法も考えられる。この方法の採用により収束までの計算時間の縮小も期待できる。また、class-bush アルゴリズムや Tapas 及びそれらの改良法は FW 法やその改良手法と比較して極めて高速に求解できることが示されている。これらの手法について、提案手法のように更新する変数を限定することでアウトターープ間の解の振動を防ぐ方法の開発も可能だと考えられる。この開発を今後の課題としたい。

謝辞：本研究は国土交通省「道路政策の質向上に資する技術研究開発」の令和 2 年度研究助成（テーマ名：公共交通ターミナル整備の空間経済分析に関する研究開発）

の助成を受けたものである。

参考文献

- 1) Beckmann, M., McGuire, C.B., and Winsten, C.B.: Studies in the Economics of Transportation. *Yale University Press*, New Haven, Connecticut, 1956.
- 2) Frank, M., and Wolfe, P.: An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3 (1-2), pp.95-110, 1956.
- 3) Leblanc, L.J., Morlok, E.K., and Pierskalla, W.P.: An efficient approach to solving the road network equilibrium traffic assignment problem. *Transportation Research* 9 (5), pp.309-318, 1975.
- 4) 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.
- 5) Bar-Gera, H., 2002. Origin-based algorithm for the traffic assignment problem. *Transportation Science*, 36 (4), pp.398-417, 2002.
- 6) Nie, Y.: A class of bush-based algorithms for the traffic assignment problem, *Transportation Research Part B*, 44 (1), pp.73-89, 2010.
- 7) Bar-Gera, H.: Traffic assignment by paired alternative segments, *Transportation Research Part B*, vol. 44 (8-9), pp.1022-1046, 2010.
- 8) Xie, J., and Xie, C.: An Improved TAPAS Algorithm for the Traffic Assignment Problem, *Proc., 17th International IEEE Conference on Intelligent Transportation*. Qingdao, China, 2015.
- 9) Mitradjieva, M, and Lindberg ,P.O.: The Stiff Is Moving—Conjugate Direction Frank-Wolfe Methods with Applications to Traffic Assignment, *Transportation Science*, 47 (2), pp.280-293, 2013.

(Received ??, ?)
(Accepted ??, ?)

THE EFFICIENT ALGORITHM TO SOLVE USER EQUILIBRIUM ASSIGNMENT PROBLEM BY PARTIAL UPDATING OF ORIGIN BASED LINK FLOW

Satoshi SUGIURA