

降下方向ベクトル計算へのランダム性の導入による 利用者均衡配分の演算効率化

Acceleration of User Equilibrium Assignment by Introduction of Randomness into Descending Vector Calculation

北海道大学工学部環境社会工学科 ○学生員 神谷奏有 (Kanau Kamiya)
北海道大学大学院工学研究院 正員 杉浦聡志 (Satoshi Sugiura)

1. はじめに

利用者均衡配分は道路ネットワーク分析において実務上でも重要な役割を果たしている。大規模ネットワークへの応用においては、計算負荷の軽減が課題となっており、ネットワークデザイン問題など利用者均衡配分を繰り返し解く必要がある問題への応用を踏まえれば、効率的に求解できる手法に対する要請は小さくない。利用者均衡配分の代表的な解法は Frank and Wolfe¹⁾, Leblanc et al.²⁾による Frank-Wolfe 法(以下, FW 法)がある。しかし, FW 法は解の近傍では振動が生じることから解の収束が緩慢となることが知られている。FW 法以外の解法では起点別の非巡回有向グラフ内でのフロー調整によって収束を目指す Class bush アルゴリズムとよばれる解法があり, Dial³, Bar-gera⁴⁾が提案, Nie⁵⁾によって整理されている。この他にも Bar-gera⁶⁾による TAPAS など FW 法と比較して解の収束が高速な手法が提案されてきた。しかしながら, これらの手法はマルチクラス配分に対応していない。道路施策を分析する上で, 車種を考慮する必要がある場面は多く, マルチクラス配分が可能なアルゴリズムの効率化が望まれる。

FW 法の効率化の既往研究では, Conjugate direction Frank-Wolfe Methods⁷⁾が提供されている。これは解の更新時に降下方向を最急勾配方向とするのではなく, 接ベクトルの情報を用いて降下方向を決定している。この手法は単一繰り返し内での質的な探索の向上を図る手法ととらえることができる。一方で, 単一繰り返し内における計算量の量的な削減により効率化を図る手法もある。鷹尾・朝倉⁸⁾は最短経路探索においてひとつ前の反復の最短ツリーを記憶することで探索時間を削減し高速化を実現している。杉浦⁹⁾は勾配ベクトルの算出において, ランダムに選択する起点のみ部分的に更新する方法を提案している。この手法は数値実験により効率向上が確認されている。この手法は解の降下方向ベクトルの修正により解の振動抑制に資すること, 最短経路探索の量的削減を同時に実現することを企図している。数値実験では計算時間を低減した一方で, 降下方向ベクトルへのランダム性の導入の理論的妥当性については言及されていない。また, 起点の選択において一様分布を用いており, この修正によりさらなる演算効率の向上が期待できる。

本研究では杉浦⁹⁾の着想に基づき, 一様分布に基づいた起点選択から, 求解中に得られる情報を利用して選択する確率を与える手法を考える。提案手法はテストネットワークに適用され, そのパフォーマンスを確認する。

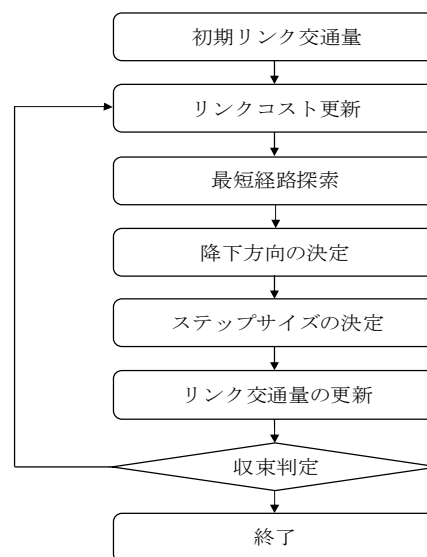


図-1 FW 法のフローチャート

2. 提案手法の概要

FW 法はリンク交通量ベクトルを変数として, 均衡状態のリンク交通量を求解するアルゴリズムである。FW 法のフローチャートを図-1に示す。まず, 交通量のない状態で各 OD ペアについて最短経路探索を行い, 全ての交通量を最短経路に配分しリンク交通量の初期解を得る。次に, 配分されたリンク交通量に基づき, BPR 関数によりリンクコストを更新する。更新されたリンクコストに基づいて最短経路探索を行い, 全ての交通量を最短経路に配分しなおし補助解を得て降下方向を決定する。次に降下方向にどれだけ進むか(ステップサイズ)を決定し, 解を更新する。解が収束しているならば終了し, そうでなければリンクコスト更新に戻り, 同様に繰り返す。

提案手法では, FW 法の最短経路探索と降下方向の決定の部分の修正を行う。FW 法のように全ての OD ペアに対する最短経路探索による降下方向ベクトルの計算は行わず, 確率的に選択された起点についてのみ部分的に最短経路探索を行い, 解を更新する。この方法は杉浦⁹⁾の手法と同等である。提案手法は, 杉浦⁹⁾が一様分布により効果方向ベクトルを計算する起点を選択するのに対して, 繰り返し演算中に得られる情報によって与えた重みに基づいて確率的に選択する。この操作により, 更新の必要性の高いと思われる起点を優先的に選択することで均衡状態へ効率的に収束することを目指す。

3. 提案手法

3.1 記号の定義

本研究で用いる記号を以下のように定義する.

A	: リンク a の集合
S	: 起点 s の集合
T	: 終点 t の集合
ϕ	: 起点集合 S の部分集合
d_{st}	: OD ペア (s, t) における交通需要
C_a	: リンク a の交通容量
n	: 繰り返し回数
δ_s^n	: 繰り返し回数 n において起点 s が ϕ に含まれないとき 1, それ以外で 0 をとる変数
$\delta_{st,a}^n$	: 繰り返し回数 n においてリンク a が OD ペア (s, t) の最短経路に含まれるとき 1, それ以外で 0 をとる変数
t_{a0}	: リンク a のゼロフロー時のリンクコスト
t_a^n	: 繰り返し回数 n のリンク a のリンクコスト
x_a^n	: 繰り返し回数 n のリンク a のリンク交通量
$x_{a,s}^n$	: 繰り返し回数 n のリンク a の起点 s 別のリンク交通量
p_a^n	: 繰り返し回数 n の降下方向ベクトルの要素
u_{st}	: OD ペア (s, t) における最短経路コスト
α, β	: BPR 関数のパラメータ
$P_{link,a}^n$	: 繰り返し回数 n においてリンク a が選択される確率
$P_{origin,s}^n$	: 繰り返し回数 n において起点 s が選択される確率

3.2 提案手法における起点選択方法

提案手法では, 均衡が実現した時には混雑リンクにおける交通集中が繰り返し演算中の解と比べて緩和されることに着目した手法を考える. 起点の選択は 2 段階の手順で構成される. まず各繰り返しにおいて交通量を減少させるべき混雑したリンクを選択する. 次に, その選択されたリンクを利用している起点のいくつかを確率的に選択する.

まず, 混雑したリンクを選択するためには, BPR 関数の傾き, すなわち微分値を用いて評価する. リンク容量やゼロフロー時のリンクコストはリンクによって異なるため, これらは相対的な基準として利用しにくい. 各繰り返し時点におけるリンク交通量に対応する BPR 関数の傾きが大きいほど交通量の減少に対しコストを効率的に減少させることが可能である. リンクを選択する確率を以下のように設定する.

$$P_{link,c}^n = \frac{\frac{d}{dx_c^n} t_c(x_c^n)}{\sum_{a \in A} \frac{d}{dx_a^n} t_a(x_a^n)} \quad (1)$$

$$\exists c \in A$$

where

$$\frac{d}{dx_a} t_a(x_a) = \alpha \beta \frac{t_{a0}}{C_a} \left(\frac{x_a}{C_a} \right)^{\beta-1} \quad (2)$$

$$\forall a \in A$$

次に選択されたリンク c について考える. リンク c のリンク交通量について, 起点別のリンク交通量を用いて重み付けを行い, 確率的に更新起点を選択する. ある起点 o が選択される確率は以下のように設定する.

$$P_{origin,o}^n = \frac{x_{c,o}^n}{x_a^n} \quad (3)$$

$$\forall a \in A, \exists c \in A, \exists o \in S$$

最後に, 選択された起点についてのみ再配分を行う. 選択された起点集合 ϕ に含まれる起点についてのみ最短経路探索を行い, 降下方向ベクトルのリンク別の要素を求める.

$$p_a^n = \sum_{s \in S} \left(\delta_s^n \cdot x_{a,s}^n + \sum_{s \in \phi} \sum_{t \in T} \delta_{st,a}^n \cdot d_{st} \right) \quad (4)$$

$$\forall a \in A$$

式(4)の前半の項では, あるリンクについて更新する起点が用いている場合にその起点別リンク交通量を 0 としている. 用いない場合は更新されないため, 現在の解のままとする. 後半の項では, 更新した交通量を各リンクに配分する.

4. 数値演算

4.1 演算条件

試算には Anaheim ネットワークを利用する. (出典: <https://github.com/bstabler/TransportationNetworks/tree/master/Anaheim>). ノード数は 416, リンク数は 914, OD ペアは 1406, トリップ数は 104,694.40, 起点数は 38 である. FW 法と杉浦⁹⁾手法, 提案手法のいずれも次元探索には黄金分割法を用いる. また, 収束条件には式(5)の relative gap⁵⁾を用い, その値が 10^{-2} , 10^{-3} , 10^{-4} に達した時の演算時間を記録する.

$$r = 1 - \frac{\sum_{s \in S, t \in T} u_{st} d_{st}}{\sum_{a \in A} x_a t_a} \quad (5)$$

提案手法, 杉浦⁹⁾手法では各繰り返しにおいて選択する起点の数を設定する必要があるが, ここでは総起点数の 10% 程度となる 4 つの起点を選択することとする. 演算は筆者の環境(CPU: AMD Ryzen 3400G, RAM: DDR4-2666, 8GB, 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 を用いる.

4.2 評価方法

提案手法, 杉浦⁹⁾の手法は, ランダム性を含むため, 1 回の試算で比較・評価することは適切ではない. このため, 100 回繰り返し演算を行い, 得られた解の統計量を用いて演算結果を評価する.

まず, 提案手法と FW 法, 杉浦⁹⁾手法の 3 手法を演算時間・繰り返し回数の観点から比較し, 提案手法の効率性を評価する. 次に, 更新起点数と演算時間との関係性についても言及する.

表-1 100 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	0.30	0.84	2.12
最小値	0.09	0.34	1.06
平均値	0.17	0.56	1.55
中央値	0.17	0.56	1.54
標準偏差	0.04	0.10	0.22
(参考)FW 法	0.20	1.18	3.77
(参考)杉浦手法	0.21	0.79	2.21

表-2 100 回演算における繰り返し回数の統計指標

	$r < 10^{-2}$	$r < 10^{-3}$	$r < 10^{-4}$
最大値	26.0	90.0	240.0
最小値	4.0	33.0	116.0
平均値	12.5	57.6	173.6
中央値	12.0	58.5	173.0
標準偏差	4.9	11.7	25.4
(参考)FW	2.0	11.0	49.0
(参考)杉浦手法	17.4	88.7	266.3

4.3 演算時間・繰り返し回数

まず、提案手法で100回繰り返し演算を行い、relative gap がそれぞれ 10^{-2} 、 10^{-3} 、 10^{-4} に達した時の演算時間の統計指標を表-1に示す。比較のため、FW法で演算した1回の値、杉浦⁹⁾手法を100回繰り返し演算した時の平均値を同時に示している。提案手法において relative gap が 10^{-4} を下回った時の平均演算時間は1.55秒であり、これはFW法の3.77秒と比べ2.4倍程度、杉浦⁹⁾手法の2.21秒と比べ1.4倍程度の演算速度を得られている。次に表-2では、100回繰り返し演算を行い、relative gap がそれぞれ 10^{-2} 、 10^{-3} 、 10^{-4} に達した時点での繰り返し回数を表している。提案手法において relative gap が 10^{-4} を下回った時の平均繰り返し回数は173.6回であり、FW法の49回と比べ3.5倍ほど要している。繰り返し1回あたりの演算時間を減少させる代わりに繰り返し回数が増えているが収束までの演算時間の高速化を実現している。一方で、杉浦⁹⁾手法の266.3回と比べ0.65倍程度の繰り返し回数で到達している。更新起点の数は提案手法・杉浦⁹⁾手法ともに同じであるから、提案手法は杉浦⁹⁾手法と比較し、繰り返し1回あたりの質を向上させていると解釈できる。

次にそれぞれの手法の試行の1回を取り出して比較する。relative gap の推移の比較を図-2に示す。提案手法は他の手法に比べ早い段階で relative gap の小さな値に達し、他の手法よりも小さな値を常に維持できており、精度が高い。図-3の目的関数の推移の比較においても提案手法が他の手法よりも速く、提案手法が0.5秒程度で達している値にFW法では4秒以上、杉浦⁹⁾手法でも2

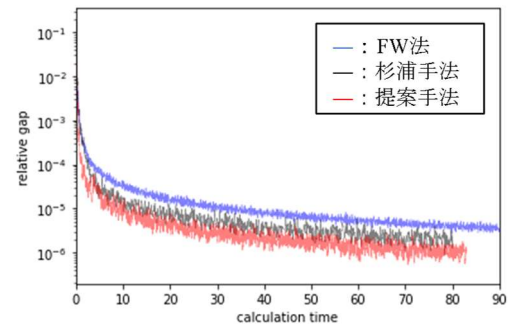


図-2 relative gap の推移

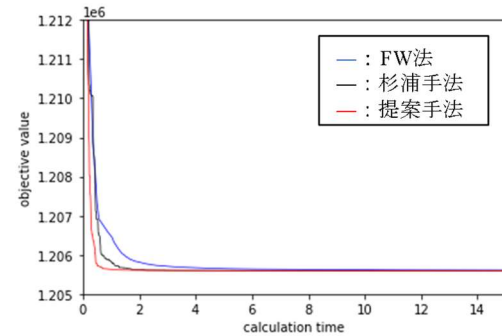


図-3 目的関数の推移

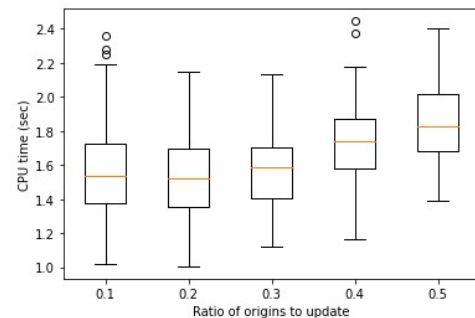


図-4 更新起点割合別の演算時間

秒程度かかっている。以上より、提案手法が他の手法より効率的に求解出来ていると確認できる。

4.4 選択する起点数によるパフォーマンス比較

これまでの数値演算は更新する起点数を総起点数の10%程度となる4起点としていた。更新割合が小さい場合、解の更新量が小さく、前回の更新とほとんど変わらないために多くの繰り返しを必要とし演算時間が増加すると考えられる。一方で、更新割合が大きくなると、最短経路探索にかかる時間が増加し、従来のFW法と同様に解の振動も大きくなるために演算時間も増加する。これらのことから起点の更新割合にも演算時間を小さくするための最適な値が存在することが考えられる。そこで、更新起点数が演算時間に与える影響についても確認する。更新割合を10%、20%、30%、40%、50%とした時の演算時間の比較を箱ひげ図を用いて図-4に示す。演算時間は更新割合によって変化し、更新割合が20%の時に最も演算時間が小さくなっている。このことから、比較的选择する起点の割合を比較的小さくするとき、性能が高いことがわかる。この値は対象ネットワーク等によって

変化することが考えられ、一般的な知見とはならないが、杉浦⁹⁾では Sioux-falls ネットワークにおいて全体の 20% 程度の時に最も高いパフォーマンスを示しており、類似の結果が得られている。これらの経験を重ねることで、安定的に良好なパフォーマンスを示す起点の選択割合を設定する必要がある。

5. おわりに

本稿では、起点への重み付けを用いたランダムな選択手法を FW 法に導入することで他の手法よりも高速に演算できることを数値演算によって示した。一方で、降下方向ベクトルの降下方向、解の収束といったランダム性の導入による有効性や影響について理論的な説明は与えられていない。今後は、提案手法を理論的に裏づける。また、更新起点数による演算時間の変化についても述べたが、ネットワークの形状に依存するとみられ、これらの関連性も確認する。

参考文献

- 1) Frank, M., and Wolfe, P.: An algorithm for quadratic programming. Naval Research Logistics Quarterly, 3 (1-2), pp.95-110, 1956.
- 2) 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.
- 3) 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.
- 4) Bar-Gera, H.: Origin-based algorithm for the traffic assignment problem. Transportation Science, 36 (4), pp.398-417, 2002.
- 5) Nie, Y.: A class of bush-based algorithms for the traffic assignment problem, Transportation Research Part B, 44 (1), pp.73-89, 2010.
- 6) Bar-Gera, H.: Traffic assignment by paired alternative segments, Transportation Research Part B, vol. 44 (8-9), pp.1022-1046, 2010.
- 7) 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.
- 8) 鷹尾和亭, 朝倉康夫: 最短ソリーの記憶による交通量配分の経路探索の効率化, 土木計画学研究・論文集, 26 巻, pp.527-533, 2009
- 9) 杉浦聡志: 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, 第 62 回土木計画学研究発表会・講演集, 2020