

Manuscript information

Title

大規模ネットワークにおける動的利用者均衡配分の効率的解法

Efficient algorithm for solving dynamic user equilibrium traffic assignment

Authors

Name: 涌井 優尚 (Masanao Wakui)

Affiliation: 東北大学 大学院 情報科学研究科 (Graduate School of Information Sciences, Tohoku University)

Name: 酒井 高良 (Takara Sakai)

Affiliation: 東北大学 大学院 情報科学研究科 (Graduate School of Information Sciences, Tohoku University)

Name: 赤松 隆 (Takashi Akamatsu)

Affiliation: 東北大学 大学院 情報科学研究科 (Graduate School of Information Sciences, Tohoku University)

Keywords

動的利用者均衡配分, 経路選択, 線形相補性問題, Frank-Wolfe 法

(dynamic user equilibrium, route choice, linear complementarity problem, Frank-Wolfe algorithm)

Corresponding author

涌井 優尚 (Masanao Wakui)

E-mail: masanao.wakui.r4@dc.tohoku.ac.jp

CRedit

Masanao Wakui: Formal analysis; Methodology; Software; Writing original draft; Writing – review & editing.

Takara Sakai: Conceptualization; Formal analysis; Funding acquisition; Software; Writing – review & editing.

Takashi Akamatsu: Conceptualization; Funding acquisition; Supervision; Writing – review & editing.

Competing Interests

The authors declare there are no competing interests.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number JP20J21744, JP21H01448.

大規模ネットワークにおける 動的利用者均衡配分の効率的解法

涌井 優尚¹・酒井 高良²・赤松 隆³

¹学生会員 東北大学大学院情報科学研究科 (〒 980-8579 仙台市青葉区荒巻字青葉 6-6-06-408)

E-mail: masanao.wakui.r4@dc.tohoku.ac.jp (Corresponding Author)

²学生会員 東北大学大学院情報科学研究科・JSPS 特別研究員 DC (〒 980-8579 仙台市青葉区荒巻字青葉 6-6-06-408)

E-mail: takara.sakai.t1@dc.tohoku.ac.jp

³正会員 東北大学大学院教授 情報科学研究科 (〒 980-8579 仙台市青葉区荒巻字青葉 6-6-06-408)

E-mail: akamatsu@plan.civil.tohoku.ac.jp

本研究では、一起点多終点ネットワークにおける経路選択 DUE 配分の効率的解法を開発した。具体的には、まず待ち行列に point queue を仮定した上で、DUE 配分を線形相補性問題として定式化した。そしてこの問題と等価な最適化問題に対し、大規模問題に適用可能な効率的アルゴリズムを提案した。数値実験を通して提案解法は、既存研究における数値計算例を大幅に上回る規模の巨大ネットワークに対しても、素朴な解法の 100 倍から 1000 倍程度以上の計算効率を誇り、かつきわめて高精度な均衡解が計算できることが示された。

Key Words: *dynamic user equilibrium, route choice, linear complementarity problem, Frank-Wolfe algorithm*

1. はじめに

(1) 背景

交通ネットワーク上で達成される交通状態を表現する枠組みとして交通量配分の理論がある。交通量配分とは、ネットワークの形状と静的な交通需要を与件として、何らかの配分原則に基づいて実現する交通状態を出力する数理モデルである。特に配分原則として利用者個人の費用最小化原則を採用した利用者均衡 (User Equilibrium, UE) 配分は、各利用者が独立に最適な行動をとった結果実現するベンチマーク状態として重要である。実際、我が国における現行の交通施策の多くは UE 配分の結果をその根拠としている。

一方、このように静的な交通状態を扱う UE 配分には、動的な渋滞の発生・延伸・解消過程を表現することができず、時間の進行を考慮する必要のある交通施策を評価できないという欠点がある。この欠点を克服する数理モデルが動的利用者均衡 (Dynamic User Equilibrium, DUE) 配分である。DUE 配分では、時々刻々変化する交通需要を与件として、UE 配分と同じく各利用者の費用最小化行動の結果生じる動的な交通状態を出力する。そのため DUE 配分は、特に時間軸上でのピーク需要を適切に捌くための交通施策 (e.g., 動的混雑料金制度, 動的信号制御) を設計・評価する理論基盤としてきわめて重要である。

以上の重要性にもかかわらず、DUE 配分は交通施策

の根拠として広く活用されるに至っていない。その理由は、一般の場合に対して汎用的に適用できる DUE 配分アルゴリズムが存在しないことである。このようなアルゴリズムの性能として、(i) 現実的な状況設定の (ii) 大規模ネットワークにおいて (iii) 正確な DUE 配分の解を計算できること、が要求される。より具体的には、(i) には multi-commodity (一起点多終点ネットワーク) の場合に適用可能なことや待ち行列モデルが現実的であること、(ii) には、所要時間やメモリ消費量の観点から計算が効率的であること、(iii) にはアルゴリズムが収束的であることなどの要素がそれぞれ含まれる。

以上三点の要求のうち、既存の研究は主に (i) を優先して満足するように志向されてきた。ここで、(i) の multi-commodity 問題への適用を図る過程においては、その部分問題である single-commodity (一起点一終点・一起点多終点ネットワーク) の場合において正確に均衡解が求められることが必要不可欠である。しかしながら、これまで single-commodity のみを対象とした場合に限っても汎用的なアルゴリズムの提案はきわめて限定的である。にもかかわらず、近年の研究ではさらに待ち行列モデルを physical queue とするなど、増して複雑な状況設定が志向される傾向にある。結果として既存のアルゴリズムは要件 (ii), (iii) を満足しないことに加え、(multi-commodity を想定した問題の定式化はなされているものの) single-commodity の場合においても均衡解を正確に求められないものが多い^{1),2),3)} (既存研究の問題点に関して詳細は 1. (3) にて述べる)。

(2) 目的

本研究では、大規模な一起点多終点ネットワークにおける経路選択 DUE 配分を解く、効率性・正確性ともに優れるアルゴリズムを開発する。すなわち本研究では、先に提示した三要件のうち (ii), (iii) を優先して満足させる。

目的の達成のため、本研究では DUE 配分を混合線形相補性問題として、リンク・ノード変数を用いて（経路変数を用いることなく）定式化する。これにより、経路変数を用いる場合に比べて変数の次元を大幅に削減でき、大規模ネットワークへの適用が可能となる。さらに、これと等価な二次計画問題を構築し、最適化アルゴリズムを適用する。このことにより、計算した均衡解の正確性は数理計画の理論に基づいて保証される。最適化アルゴリズムには Frank-Wolfe 法ベースの解法を採用することで、実質的に線形計画問題を繰り返し解くことに落とし込むことができる。線形計画問題の効率的解法は多数存在することから、提案手法では大規模問題に対しても効率的な求解が可能となる。

(3) 既存研究と本研究の貢献

利用者の選択行動に着目すると、DUE 配分は二種類に分類できる。一つは利用者が経路の選択のみを行い、自身の旅行費用を最小化する“経路選択 DUE 配分”，もう一つは利用者が経路選択に加え出発時刻も選択する“経路・出発時刻同時選択 DUE 配分”（以降、同時選択 DUE 配分）である。両者について、一般ネットワークに対する DUE 配分アルゴリズムがこれまでに数多く提案されている。

経路選択 DUE 配分については Akamatsu⁴⁾, Lo and Szeto⁵⁾, Ban et al.¹⁾, Long et al.⁶⁾, Gentile²⁾ などの既存研究がある^{注1)}。しかしながらいずれの研究にも、先に提示した三要件の観点から問題がある。まず 1. (1) にも述べた複雑なモデルの拡張を志向した結果、既存研究の多くはヒューリスティックな不動点アルゴリズムを構築するにとどまる。この種の解法は多大な反復計算を要することに加え、均衡解への理論的な収束保証もなされていない。実際、近年の研究成果である Gentile²⁾ では、不動点アルゴリズムの一モジュールである、交通流を暫定的に配分する計算（dynamic network loading と呼ばれる）を効率化する手法を提案している。しかし不動点アルゴリズム自体の収束は依然保証されず、実際に提示されている計算結果からも正確な解を算出しているとは言えない。さらに、一部の研究^{5),6)} では経路変数を陽に用いて問題を定式化しているため、経路次元の変数を同時に扱う必要がある。このことにより、ネットワーク規模に対して変数の次元が指数関数的に増加し、大規模なネットワークへの適用には堪えない。

同時選択 DUE 配分についての既存研究（e.g., Friesz and Mookherjee³⁾, Long et al.⁷⁾, Han et al.⁸⁾, 長江ら⁹⁾）にも概ね同様の指摘が可能である。ほとんどの研究は経路変数を陽に用いるヒューリスティックな不動点アルゴリズムである^{3),7),8)}。以上に加えて、既存の解法の多く^{3),8),9)} は問題を“経路選択”と“出発時刻選択”とのそれぞれに分解することなく、複雑な条件群を全て満たすような解を一挙に求める手法がとられている。このことも、効率性・正確性の両観点から問題があろう。

本研究には、以上に提示した既存研究の問題点を踏まえ、次の三つの貢献がある：

- 本来であれば multi-commodity の DUE 配分アルゴリズムの必須モジュールである一起点多終点ネットワークにおける DUE 配分に対して、きわめて正確な均衡解が計算できること。
- 既存研究では適用例のない大規模なネットワークにおいても正確に、効率的に均衡解が計算できること。
- 同時選択 DUE 配分の効率的解法構築の基礎となる部分問題としての活用可能性を提供すること。

本研究は、ここまで論じた既存研究の問題点を体系的に解決するための最も基本的な取り組みとして位置づけることができる。

(4) 本稿の構成

本稿の構成を次に示す。まず 2. で本研究のモデルの枠組みを述べる。次に 3. で、DUE 配分を混合線形相補性問題として定式化する。4. では定式化した問題を解くアルゴリズムを提示する。続いて 5., 6. で、数値実験の結果を通して、既存解法と比較した提案手法の効率性や正確性を示す。最後に 7. で本稿のまとめを述べる。

2. モデル

(1) ネットワーク

本研究で対象とするネットワークは、唯一の起点ノードと複数の通過・終点ノード、およびそれらを結ぶ有向リンクからなる一起点多終点ネットワークとする。ネットワークに含まれるノードの集合を N 、リンクの集合を \mathcal{L} とそれぞれ定義し、それぞれの要素は $k \in N, (i, j) \in \mathcal{L}$ と表す。集合 N, \mathcal{L} の要素数は N, L とする。 N の要素のうち、唯一の起点ノードを特に o と表す。またノード $k \in N$ について、全ての流入リンクの上流側ノード集合を I_k 、全ての流出リンクの下流側ノード集合を O_k とそれぞれ表す。

ネットワーク上の全てのリンクにはボトルネックが一つずつ存在し、それ以外の区間は全て自由走行区間で

ある。リンク (i, j) の自由走行時間は \hat{c}_{ij} 、ボトルネックの容量は μ_{ij} とする。ボトルネックには、容量以上の交通量が流入しようとするとき待ち行列が発生する。待ち行列は point queue モデルで表現し、各リンクでは first in first out (FIFO) が成立する。

(2) 利用者

配分対象時間は $S \equiv [0, S]$ とする。ネットワーク上で旅行する利用者は時刻 $s \in S$ に、起点ノードから各自の終点ノード k へ向け交通流率 $q_k(s)$ で出発する。本研究では、ネットワーク上で実現する交通状態を、起点出発時刻別に定義された変数を用いて表す（このような定式化は Lagrange 座標系アプローチ^{4),9),10)} と呼ばれる）。具体的には、起点を時刻 s に出発した利用者がリンク (i, j) にて被る待ち行列遅れ時間 $w_{ij}(s)$ 、同利用者についてのリンク (i, j) への流入交通流率 $y_{ij}(s)$ 、同利用者が経験する起点からノード k への最短旅行時間 $\pi_k(s)$ 、という三種類の変数を用いる。

時刻 s に起点を出発した利用者がリンク (i, j) を通過する際に経験する旅行時間 $c_{ij}(s)$ は、同リンクの自由走行時間 \hat{c}_{ij} と待ち行列遅れ時間との和で次のように表される：

$$c_{ij}(s) \equiv \hat{c}_{ij} + w_{ij}(s) \quad \forall (i, j) \in \mathcal{L}. \quad (1)$$

また時刻 s に起点を出発した利用者の、ノード k への最早到着時刻 $\tau_k(s)$ を次式で定義する：

$$\tau_k(s) \equiv \pi_k(s) + s \quad \forall k \in \mathcal{N}. \quad (2)$$

3. 定式化

(1) 均衡条件

DUE 状態とは、各起点出発時刻ごとに、どの利用者も自分一人が経路を変更しても、自身の旅行費用をそれ以上改善できない状態である。ここで自身の旅行費用とは、利用者が起点から自身の終点まで旅行する際に通過する全てのリンクの、自由走行時間と待ち行列遅れ時間との総和である。この状態において、各変数は次に示す四種類の均衡条件を満足する。

a) 待ち行列進展条件

待ち行列進展条件は、リンク旅行時間のダイナミクスを規定する条件である。待ち行列の変化によるリンク旅行時間 $c_{ij}(s)$ の変化率は次式により表現できる¹¹⁾：

$$\frac{dc_{ij}(s)}{ds} = \begin{cases} \frac{y_{ij}(s)}{\mu_{ij}} - \frac{dw_{ij}(s)}{ds} & \text{if } c_{ij}(s) > \hat{c}_{ij} \\ \max \left[\frac{y_{ij}(s)}{\mu_{ij}} - \frac{dw_{ij}(s)}{ds}, 0 \right] & \text{if } c_{ij}(s) = \hat{c}_{ij} \end{cases} \quad \forall (i, j) \in \mathcal{L}, \forall s \in S. \quad (3)$$

ここで式 (1), (2) と、 $\frac{dc_{ij}(s)}{ds} = \frac{dw_{ij}(s)}{ds}$ を考慮すると、式 (3) は次の相補性条件に帰着する：

$$\begin{cases} w_{ij}(s) \cdot \left\{ \frac{dw_{ij}(s)}{ds} - \frac{y_{ij}(s)}{\mu_{ij}} + \left(\frac{d\pi_i(s)}{ds} + 1 \right) \right\} = 0 \\ \frac{dw_{ij}(s)}{ds} - \frac{y_{ij}(s)}{\mu_{ij}} + \left(\frac{d\pi_i(s)}{ds} + 1 \right) \geq 0, w_{ij}(s) \geq 0 \end{cases} \quad \forall (i, j) \in \mathcal{L}, \forall s \in S. \quad (4)$$

b) 最短経路選択条件

最短経路選択条件は、利用者の行動モデルを記述する条件である。DUE 配分では、利用者は常に旅行時間に関して最適行動をとる。すなわち DUE 状態では、その時々における最短経路上にある $(c_{ij}(s) - \pi_j(s) + \pi_i(s) = 0)$ リンクにのみ交通流が存在する $(y_{ij}(s) > 0)$ 。このことより次の相補性条件が成り立つ：

$$\begin{cases} y_{ij}(s) \cdot \{c_{ij}(s) - \pi_j(s) + \pi_i(s)\} = 0 \\ c_{ij}(s) - \pi_j(s) + \pi_i(s) \geq 0, y_{ij}(s) \geq 0 \end{cases} \quad \forall (i, j) \in \mathcal{L}, \forall s \in S. \quad (5)$$

c) フロー保存条件

フロー保存条件は、交通流の連続性を保証する条件である。起点を除く全てのノードについて、次の等式で表されるフロー保存条件が成り立つ：

$$\sum_{j \in O_k} y_{kj}(s) - \sum_{i \in I_k} y_{ik}(s) + q_k(s) = 0 \quad \forall k \in \mathcal{N} \setminus o, \forall s \in S. \quad (6)$$

同様の条件は相補性条件として次のように記述できる：

$$\begin{cases} \pi_k(s) \cdot \left\{ - \sum_{j \in O_k} y_{kj}(s) + \sum_{i \in I_k} y_{ik}(s) - q_k(s) \right\} = 0 \\ - \sum_{j \in O_k} y_{kj}(s) + \sum_{i \in I_k} y_{ik}(s) - q_k(s) \geq 0, \pi_k(s) \geq 0 \end{cases} \quad \forall k \in \mathcal{N} \setminus o, \forall s \in S. \quad (7)$$

d) ノード境界条件

ノード境界条件は、FIFO を含む交通流の物理的制約を表す条件である。

DUE 状態では全利用者は各自の最短経路を選択している。このことから、ある時刻 s に出発した利用者は、任意のノードへの到着時点において、それ以前の時刻に出発したどの利用者も追い越してはならない。この制約は次の不等式で表される：

$$\frac{d\pi_k(s)}{ds} \geq -1 \quad \forall k \in \mathcal{N} \setminus o, \forall s \in S. \quad (8)$$

また最短旅行時間 $\pi_k(s)$ はその定義から、自由走行時間 \hat{c}_{ij} から計算される最短旅行時間 $\hat{\pi}_k$ が最小値となる。すなわち次の不等式が常に成り立つ：

$$\pi_k(s) \geq \hat{\pi}_k \quad \forall k \in \mathcal{N} \setminus o, \forall s \in S. \quad (9)$$

(2) 出発時刻の離散化

ここからは、数値計算を行う準備として配分対象時間 $S \equiv [0, S]$ を一定の微小時間幅 ds で離散化する。これに伴い本節では、ここまで提示した各均衡条件を離散時刻系にて再表記する。

離散時刻の総数は $K + 1$ 個とし、全離散時刻の集合を $\mathcal{K} \equiv \{\kappa | 0, 1, 2, \dots, K\}$ と定義する。このようにすると、 $\kappa + 1$ 個目の離散時刻 s^κ は $\kappa \cdot ds$ と表せる。以降では出発時刻 s^κ に対応する変数を、 $w_{ij}(s^\kappa) \equiv w_{ij}^\kappa$ のように添え字 κ で表す。ただし $\kappa = 0$ は初期時刻とし、 $q_k^0 = 0, w_{ij}^0 = y_{ij}^0 = 0$ と定める。このことより、 $c_{ij}^0 = \hat{c}_{ij}$ 、 $\pi_k^0 = \hat{\pi}_k$ である。連続変数の時間微分については全て、次のように近似する：

$$\frac{dw_{ij}(s)}{ds} \simeq \frac{w_{ij}^\kappa - w_{ij}^{\kappa-1}}{ds}. \quad (10)$$

なお、以降の再表記は全てベクトル・行列形式とする^{注2)}。その際に用いる、 $w^\kappa, y^\kappa, c^\kappa, \mu$ は L 次元の、 π^κ, q^κ は $N - 1$ 次元の、対応する要素を全て並べたベクトルを指す。また A はノード・リンク接続行列を、 A_+ は A の負要素を全て 0 に置換した行列を表す。

a) 待ち行列進展条件 (離散)

$$\mathbf{0} \leq w^\kappa \perp \{\alpha(w^\kappa - w^{\kappa-1}) - y^\kappa + \alpha A_+^\top (\pi^\kappa - \pi^{\kappa-1}) + \mu\} \geq \mathbf{0}. \quad (11)$$

b) 最短経路選択条件 (離散)

$$\mathbf{0} \leq y^\kappa \perp \{c^0 + w^\kappa + A^\top \pi^\kappa\} \geq \mathbf{0}. \quad (12)$$

c) フロー保存条件 (離散)

$$\mathbf{0} \leq \pi^\kappa \perp \{-A y^\kappa - q^\kappa\} \geq \mathbf{0}. \quad (13)$$

d) ノード境界条件 (離散)

$$\pi^\kappa \geq \pi^{\kappa-1} - \mathbf{1}ds, \quad (14)$$

$$\pi^\kappa \geq \pi^0. \quad (15)$$

ただし $\alpha \equiv \text{diag} [\mu_{ij}/ds], \forall \kappa \in \mathcal{K} \setminus 0$ である。

ここで各時刻の変数・定数を縦に並べた、次に示すベクトルを定義する：

$$w \equiv \begin{bmatrix} w^1 \\ \vdots \\ w^K \end{bmatrix}, y \equiv \begin{bmatrix} y^1 \\ \vdots \\ y^K \end{bmatrix}, \pi \equiv \begin{bmatrix} \pi^1 \\ \vdots \\ \pi^K \end{bmatrix}, q \equiv \begin{bmatrix} q^1 \\ \vdots \\ q^K \end{bmatrix}.$$

さらに、このうち w, y, π を縦に並べた未知変数ベクトル $x \equiv \{w, y, \pi\}^\top$ を定義する。

以上をまとめると次に示す命題が得られる。

命題 1 DUE 配分の解は、次に示す混合線形相補性問題 [DUE-LCP] を解くことにより得られる：

[DUE-LCP]

$$\begin{aligned} \text{Find} \quad & x^* \in \mathbb{R}_+^{(2L+N-1)K}, \\ \text{such that} \quad & \mathbf{0} \leq x^* \perp (\bar{M}x^* + b) \geq \mathbf{0}, \\ \text{and} \quad & \pi^{\kappa^*} \geq \max [\pi^{\kappa-1^*} - \mathbf{1}ds, \pi^0] \\ & \forall \kappa \in \mathcal{K} \setminus 0. \end{aligned}$$

ここで、係数行列 \bar{M} は次のような、歪対称に近く、また疎な構造を持つ (\otimes は Kronecker 積を表す)：

$$\bar{M} \equiv \begin{bmatrix} \Delta_K \otimes \alpha & -I & \Delta_K \otimes \alpha A_+^\top \\ I & \mathbf{0} & I_K \otimes A^\top \\ \mathbf{0} & -I_K \otimes A & \mathbf{0} \end{bmatrix}.$$

他の定数ベクトル・行列は次のように定義される：

$$b \equiv \begin{bmatrix} \mathbf{1}_K \otimes \mu - e_K \otimes \alpha(w^0 + A_+^\top \pi^0) \\ \mathbf{1}_K \otimes c^0 \\ -q \end{bmatrix},$$

$$e \equiv \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \Delta \equiv \begin{bmatrix} 1 & & 0 \\ -1 & 1 & \\ & \ddots & \ddots \\ 0 & & -1 & 1 \end{bmatrix}.$$

(3) 起点出発時刻別の時間分解

[DUE-LCP] の未知数の個数は $(2L + N - 1)K$ である。そのため、特にネットワークの規模が大きく、かつ時刻の離散幅が細かい場合、未知数の個数は膨大となり、[DUE-LCP] を直接解くことは困難である。このことを踏まえて、本研究では時刻別分解手法^{4),10)}を採用する。具体的には [DUE-LCP] を起点出発時刻 κ について分解し、時間の進行方向に向かって逐次的に解いてゆく。ここでは、離散化した問題 [DUE-LCP] を起点出発時刻別に分解する。

起点出発時刻 κ についての均衡条件 (11) - (15) はいずれも、時刻 $\kappa, \kappa - 1$ についての変数のみから構成されている。ゆえに [DUE-LCP] は、「時刻 $\kappa - 1$ における均衡解を与件とした時刻 κ のみの問題」の繰り返しとして、前の時刻から分解して解くことができる。これに伴い、新たな定数 $\beta^\kappa \equiv w^{\kappa-1} + A_+^\top \pi^{\kappa-1} - \mathbf{1}ds$ を導入することで、式 (11) は次のように書き換わる：

$$\mathbf{0} \leq w^\kappa \perp \{\alpha w^\kappa - y^\kappa + \alpha A_+^\top \pi^\kappa - \alpha \beta^\kappa\} \geq \mathbf{0}. \quad (16)$$

ここでさらに変数 $x^\kappa \equiv \{w^\kappa, y^\kappa, \pi^\kappa\}^\top$ を定義すると、式 (12) - (15), (16) より次の命題が得られる。

命題 2 ある一つの離散時刻 $\kappa \in \mathcal{K} \setminus 0$ における DUE 配分の解は、混合線形相補性問題 [DUE-LCP-sub(κ)] を解くことにより得られる：

[DUE-LCP-sub(κ)]

$$\begin{aligned} \text{Find} \quad & \mathbf{x}^{\kappa*} \in \mathbb{R}_+^{2L+N-1}, \\ \text{such that} \quad & \mathbf{0} \leq \mathbf{x}^{\kappa*} \perp (\mathbf{M}\mathbf{x}^{\kappa*} + \mathbf{b}^\kappa) \geq \mathbf{0}, \\ \text{and} \quad & \boldsymbol{\pi}^{\kappa*} \geq \max[\boldsymbol{\pi}^{\kappa-1} - \mathbf{1}ds, \boldsymbol{\pi}^0]. \end{aligned}$$

係数行列およびベクトルは次に示す構造を持つ：

$$\mathbf{M} \equiv \begin{bmatrix} \boldsymbol{\alpha} & -\mathbf{I} & \boldsymbol{\alpha}\mathbf{A}_+^\top \\ \mathbf{I} & \mathbf{0} & \mathbf{A}^\top \\ \mathbf{0} & -\mathbf{A} & \mathbf{0} \end{bmatrix}, \mathbf{b}^\kappa \equiv \begin{bmatrix} -\boldsymbol{\alpha}\boldsymbol{\beta}^\kappa \\ \mathbf{c}^0 \\ -\mathbf{q}^\kappa \end{bmatrix}.$$

\mathbf{M} は $\bar{\mathbf{M}}$ 同様に、歪対称に近く、また疎である。

(4) サブ問題の等価変換

線形相補性問題の一般的な性質から、次に示す命題が得られる。

命題 3 [DUE-LCP-sub(κ)] の均衡解は、次に示す二次計画問題 [DUE-QP(κ)] の最適解として得られる：

[DUE-QP(κ)]

$$\begin{aligned} \min_{\mathbf{x}^\kappa} \quad & z(\mathbf{x}^\kappa) \equiv \mathbf{x}^{\kappa\top}(\mathbf{M}\mathbf{x}^\kappa + \mathbf{b}^\kappa), \\ \text{s.t.} \quad & \mathbf{M}\mathbf{x}^\kappa + \mathbf{b}^\kappa \geq \mathbf{0}, \\ & \mathbf{w}^\kappa \geq \mathbf{0}, \quad \mathbf{y}^\kappa \geq \mathbf{0}, \\ & \boldsymbol{\pi}^\kappa \geq \max[\boldsymbol{\pi}^{\kappa-1} - \mathbf{1}ds, \boldsymbol{\pi}^0]. \end{aligned}$$

なお [DUE-QP(κ)] の目的関数は、 \mathbf{M} の歪対称性から

$$z(\mathbf{x}^\kappa) = \mathbf{w}^{\kappa\top}\boldsymbol{\alpha}(\mathbf{w}^\kappa + \mathbf{A}_+^\top\boldsymbol{\pi}^\kappa) + \mathbf{x}^{\kappa\top}\mathbf{b}^\kappa \quad (17)$$

と簡略化できる。

二次計画問題は一般に相補性問題よりも数値計算上解きやすく、そのためのアルゴリズムも豊富に存在する。また、[DUE-QP(κ)] は最適解において目的関数値が 0 となる。このことを用いると、収束計算において目的関数値に基づく正確な収束判定が可能である。

4. アルゴリズム

本研究で提案するアルゴリズムは 3. (3) で述べた時刻別分解を活用したものである。すなわちその大枠は、直前時刻における DUE 配分解を与件として、均衡条件を時刻の進行方向に向かって順に解いてゆく流れである。

提案アルゴリズムは、次に示す二つのモジュールから構成される：

- [DUE-QP(κ)] を解く、Frank-Wolfe 法¹²⁾ ベースの“渋滞進展アルゴリズム”
- 均衡条件を直接解く、ダイクストラ法¹³⁾ ベースの“渋滞解消アルゴリズム”

渋滞進展アルゴリズムは、起点からどこか任意の終点ノードへ向けた交通需要が存在する (i.e., $\mathbf{q}^\kappa \neq \mathbf{0}$) 時刻 κ についての DUE 配分へ適用する。渋滞解消アルゴリズムは、起点からどこの終点ノードへも交通需要が存在せず (i.e., $\mathbf{q}^\kappa = \mathbf{0}$)、発生している渋滞が単に解消するのみの時刻 κ についての DUE 配分へ適用する。

交通需要が存在しない時刻での DUE 配分に対して、渋滞進展アルゴリズムを適用することも可能である。しかし、渋滞解消アルゴリズムは収束計算を含まないことから、渋滞進展アルゴリズムよりも高速に、正確な解が計算できる。そのため、本研究では需要の有無により上記二種類のアルゴリズムを使い分ける。

(1) 渋滞進展アルゴリズム

発生需要が存在する時刻、すなわち $\mathbf{q}^\kappa \neq \mathbf{0}$ である時刻 κ での DUE 配分解を得るためには、[DUE-QP(κ)] を Frank-Wolfe 法ベースの方法 (渋滞進展アルゴリズム) により解く。

Frank-Wolfe 法は、目的関数の線形近似をもとに次点解を逐次探索するアルゴリズムである。すなわち非線形の二次計画問題を、線形計画問題の繰り返しとして解けることが、本アルゴリズムの強みである。線形計画問題の効率的解法は数理計画の分野で広く研究されており、これを用いることで大規模な DUE 配分問題も高速に解くことができる。

渋滞進展アルゴリズムの計算手順は次に従う：

Step 0: 実行可能な初期解 $\mathbf{x}_{(1)}^\kappa$ を定め、 $n := 1$ 。

Step 1: [DUE-QP(κ)] の線形近似問題 [DUE-FW-LP] の解 $\hat{\mathbf{x}}_{(n)}^\kappa$ を計算。

Step 2: $\alpha_n := \operatorname{argmin}_\alpha z((1-\alpha)\mathbf{x}_{(n)}^\kappa + \alpha\hat{\mathbf{x}}_{(n)}^\kappa)$, s.t. $0 \leq \alpha \leq 1$ 。

Step 3: $\mathbf{x}_{(n+1)}^\kappa := (1-\alpha_n)\mathbf{x}_{(n)}^\kappa + \alpha_n\hat{\mathbf{x}}_{(n)}^\kappa$ 。

Step 4: 収束基準を満たすなら $\mathbf{x}^{\kappa'} := \mathbf{x}_{(n+1)}^\kappa$ として Step 5 へ。そうでなければ $n := n+1$ として Step 1 へ。

Step 5: $\mathbf{x}^{\kappa'}$ に整合化操作を加え、解 \mathbf{x}^κ を確定。

Step 0 から Step 4 の操作が Frank-Wolfe 法に対応する。Step 4 における Frank-Wolfe 法の収束は、[DUE-QP(κ)] の最適目的関数値が 0 と与件であることより、

$$z(\mathbf{x}_{(n+1)}^\kappa) < \varepsilon \quad (18)$$

で判定する。ここで ε は十分に小さい正の数である。

以降では各ステップでの操作の詳細を順に説明する。

a) Step 0: 初期実行可能解の計算

Frank-Wolfe 法において、初期解 $\mathbf{x}_{(1)}^\kappa$ は制約条件を満たしている必要がある。この実行可能な初期解のうち一つは、次の線形計画問題 [DUE-FW-init] の解 $\bar{\mathbf{x}}^*$ として得られる：

[DUE-FW-init]

$$\begin{aligned} \min_{\bar{\mathbf{x}}, \bar{t}_w, \bar{t}_\pi} \quad & \mathbf{1}^\top \bar{t}_w + \mathbf{1}^\top \bar{t}_\pi, \\ \text{s.t.} \quad & \mathbf{M}\bar{\mathbf{x}} + \mathbf{b}^\kappa \geq \mathbf{0}, \\ & -\bar{t}_w \leq \bar{\mathbf{w}} - \mathbf{w}^{\kappa-1} \leq \bar{t}_w, \\ & -\bar{t}_\pi \leq \bar{\boldsymbol{\pi}} - \boldsymbol{\pi}^{\kappa-1} \leq \bar{t}_\pi, \\ & \bar{\mathbf{w}} \geq \mathbf{0}, \quad \bar{\mathbf{y}} \geq \mathbf{0}, \quad \bar{t}_w \geq \mathbf{0}, \quad \bar{t}_\pi \geq \mathbf{0}, \\ & \bar{\boldsymbol{\pi}} \geq \max[\boldsymbol{\pi}^{\kappa-1} - \mathbf{1}ds, \boldsymbol{\pi}^0]. \end{aligned}$$

ここで $\bar{\mathbf{x}}$ は \mathbf{x}^κ と同次元のベクトルであり、その要素には $\mathbf{w}^\kappa, \mathbf{y}^\kappa, \boldsymbol{\pi}^\kappa$ と同次元の $\bar{\mathbf{w}}, \bar{\mathbf{y}}, \bar{\boldsymbol{\pi}}$ を持つ。

この問題は、[DUE-QP(κ)] の初期解について、 $\mathbf{w}_{(1)}^\kappa, \boldsymbol{\pi}_{(1)}^\kappa$ に関しては直前の時刻 $\kappa-1$ での均衡解と可能な限り近い値になるように、また $\mathbf{y}_{(1)}^\kappa$ に関しては最短経路選択条件を満たすように、それぞれ決定する意味を持つ。

b) Step 1: 線形近似問題

Step 1 では下記の線形計画問題 [DUE-FW-LP] を解く。ここで $\hat{\mathbf{x}}$ は \mathbf{x}^κ と同次元のベクトルであり、その要素には $\mathbf{w}^\kappa, \mathbf{y}^\kappa, \boldsymbol{\pi}^\kappa$ と同次元の $\hat{\mathbf{w}}, \hat{\mathbf{y}}, \hat{\boldsymbol{\pi}}$ を持つ。

[DUE-FW-LP]

$$\begin{aligned} \min_{\hat{\mathbf{x}}} \quad & \nabla z(\mathbf{x}_{(n)}^\kappa)^\top \hat{\mathbf{x}}, \\ \text{s.t.} \quad & \mathbf{M}\hat{\mathbf{x}} + \mathbf{b}^\kappa \geq \mathbf{0}, \quad (19) \\ & -\hat{\mathbf{w}} + \alpha^{-1}\hat{\mathbf{y}} + \mathbf{w}^{\kappa-1} \geq \mathbf{0}, \quad (20) \\ & \hat{\mathbf{w}} \geq \mathbf{0}, \quad \hat{\mathbf{y}} \geq \mathbf{0}, \quad \hat{\boldsymbol{\pi}} \geq \boldsymbol{\pi}^0. \end{aligned}$$

[DUE-FW-LP] では [DUE-QP(κ)] から、新たな制約条件 (20) が追加されたことに注意されたい。式 (20) はこの問題の制約条件 (19) を満たす範囲において次式：

$$\hat{\boldsymbol{\pi}} \geq \boldsymbol{\pi}^{\kappa-1} - \mathbf{1}ds \quad (21)$$

の十分条件であり、この問題の変数 $\hat{\mathbf{w}}, \hat{\boldsymbol{\pi}}$ の上限値を規定するために、式 (21) に代わって必要な制約条件である。両式の関係性についての証明は付録 I に記す。

[DUE-FW-LP] の目的関数に含まれる係数ベクトル $\nabla z(\mathbf{x}_{(n)}^\kappa)$ は次のように書き下せる：

$$\nabla z(\mathbf{x}_{(n)}^\kappa) = (\mathbf{M} + \mathbf{M}^\top)\mathbf{x}_{(n)}^\kappa + \mathbf{b}^\kappa. \quad (22)$$

[DUE-FW-LP] から得られた $\hat{\mathbf{x}}_{(n)}^\kappa$ を用いて、ベクトル $\hat{\mathbf{x}}_{(n)}^\kappa - \mathbf{x}_{(n)}^\kappa$ が、 n 回目の繰り返し計算における解の降下方向となる。

c) Step 2: 一次元探索

Step 2 において解く一次元探索問題の目的関数は、 α についての二次関数である。ここで、目的関数が α についての凹関数であるか凸関数であるかの情報を用いることで、この問題は解析的に解くことができる⁹⁾。

具体的には以下に示す手順で求めることができる。目的関数：

$$z((1-\alpha)\mathbf{x}_{(n)}^\kappa + \alpha\hat{\mathbf{x}}_{(n)}^\kappa) \quad (23)$$

の、 α に関する一階・二階導関数は次式である（以下、 $\mathbf{x}(\alpha) \equiv (1-\alpha)\mathbf{x}_{(n)}^\kappa + \alpha\hat{\mathbf{x}}_{(n)}^\kappa$ ）：

$$\begin{aligned} \frac{dz(\mathbf{x}(\alpha))}{d\alpha} &= \{(\mathbf{M} + \mathbf{M}^\top)\mathbf{x}(\alpha) + \mathbf{b}^\kappa\}^\top (\hat{\mathbf{x}}_{(n)}^\kappa - \mathbf{x}_{(n)}^\kappa), \\ \frac{d^2z(\mathbf{x}(\alpha))}{d\alpha^2} &= (\hat{\mathbf{x}}_{(n)}^\kappa - \mathbf{x}_{(n)}^\kappa)^\top (\mathbf{M} + \mathbf{M}^\top) (\hat{\mathbf{x}}_{(n)}^\kappa - \mathbf{x}_{(n)}^\kappa). \end{aligned}$$

ここで $\frac{d^2z(\mathbf{x}(\alpha))}{d\alpha^2}$ は α の値に依存しない定数なので、目的関数の凹凸は二階導関数から一意に特定できる。以降では凹関数の場合と凸関数の場合を分けて議論する。

目的関数が凹関数であるのは、 $\frac{d^2z(\mathbf{x}(\alpha))}{d\alpha^2} \leq 0$ の場合である。このとき、実行可能領域 $0 \leq \alpha \leq 1$ において、目的関数を最小にし得るのは $\alpha = 0$ 、もしくは $\alpha = 1$ のどちらかのみである。ゆえにこの一次元探索問題の解は次のように求まる：

$$\alpha_n = \begin{cases} 0 & \text{if } z(\mathbf{x}(\alpha=0)) < z(\mathbf{x}(\alpha=1)) \\ 1 & \text{if } z(\mathbf{x}(\alpha=0)) \geq z(\mathbf{x}(\alpha=1)) \end{cases}. \quad (24)$$

目的関数が凸関数であるのは、 $\frac{d^2z(\mathbf{x}(\alpha))}{d\alpha^2} > 0$ の場合である。このような状況は、実行可能領域内で目的関数が、1. 単調増加、2. 単調減少、3. 極値を持つ、という三つの場合が考えられる。これらを順に議論する。

1. 単調増加

$\alpha = 0$ における微係数 $\frac{dz(\mathbf{x}(\alpha=0))}{d\alpha}$ が正の値という点でその他と区別できる。このとき関数を最小にする α_n は 0 である。

2. 単調減少

$\alpha = 1$ における微係数 $\frac{dz(\mathbf{x}(\alpha=1))}{d\alpha}$ が負の値という点でその他と区別できる。このとき関数を最小にする α_n は 1 である。

3. 極値を持つ

上記二つ以外の場合全てこの場合である。このとき関数を最小にする α は、 $\frac{dz(\mathbf{x}(\alpha))}{d\alpha} = 0$ を満たす α である。ゆえにこの方程式を以下のように解け

ば, α_n は解析的に定まる:

$$\begin{aligned}
& \frac{dz(\alpha_n)}{d\alpha} = 0 \\
& \Leftrightarrow \{(M + M^T)x(\alpha_n) + b^k\}^T (\hat{x}_{(n)}^k - x_{(n)}^k) = 0 \\
& \Leftrightarrow \{(M + M^T) \left((1 - \alpha_n)x_{(n)}^k + \alpha_n \hat{x}_{(n)}^k \right) + b^k\}^T (\hat{x}_{(n)}^k - x_{(n)}^k) = 0 \\
& \Leftrightarrow \alpha_n (\hat{x}_{(n)}^k - x_{(n)}^k)^T (M + M^T) (\hat{x}_{(n)}^k - x_{(n)}^k) \\
& \quad = - \{(M + M^T)x_{(n)}^k + b^k\}^T (\hat{x}_{(n)}^k - x_{(n)}^k) \\
& \Leftrightarrow \alpha_n = - \frac{\{(M + M^T)x_{(n)}^k + b^k\}^T (\hat{x}_{(n)}^k - x_{(n)}^k)}{(\hat{x}_{(n)}^k - x_{(n)}^k)^T (M + M^T) (\hat{x}_{(n)}^k - x_{(n)}^k)}. \tag{25}
\end{aligned}$$

d) Step 5: 整合化操作

渋滞進展アルゴリズムでは [DUE-QP(κ)] を解くことにより時刻 κ における均衡解を得る。しかし、このとき単に Frank-Wolfe 法により [DUE-QP(κ)] を解くのみでは、均衡条件を全て満たしてしながらネットワーク上で物理的に不整合な解が得られる場合がある（不整合の具体的な説明は付録 II を参照されたい）。この問題を解決する Step 5 での整合化操作は、ダイクストラ法を応用した以下に示すアルゴリズムに従う:

Step 5: $x^{k'}$ に整合化操作を加え、解 x^k を確定。

Step 5-0: $\pi^k := \infty, \pi_o := 0$. 集合 \mathcal{A} に起点ノード o を追加。

Step 5-1: \mathcal{A} に含まれる“確定ノード”でないノードのうち、暫定の π_v^k が最小のもの v を \mathcal{A} から削除。 $v \neq o$ なら π_v^k をここで確定。

Step 5-2: $d_v := \pi_v^k - \pi_v^{k'}$ とし、ノード v を“確定ノード”とする。 $\forall k \in O_v$ について $w_{vk}^{k'} := \max[w_{vk}^{k'} - d_v, 0]$ 。

Step 5-3: O_v の要素をもれなく \mathcal{A} に追加。 $\forall k \in O_v$ について $\pi_k^k > \pi_v^k + w_{vk}^{k'} + c_{vk}^0$ であれば $\pi_k^k := \pi_v^k + w_{vk}^{k'} + c_{vk}^0$ 。

Step 5-4: $\mathcal{A} = \emptyset$ なら $w^k, y^k := w^{k'}, y^{k'}$ とし終了。そうでなければ Step 5-1 へ。

(2) 渋滞解消アルゴリズム

交通需要が存在しない、すなわち $q^k = \mathbf{0}$ である時刻 κ において、均衡解 x^k は、ダイクストラ法を応用することにより高速に計算できる。

交通需要が存在しない時刻 κ では $y^k = \mathbf{0}$ が成り立つ。このことを用いると、残る均衡解 w^k, π^k が満たすべき条件は、3. (1) で示した均衡条件から次の二条件に帰着

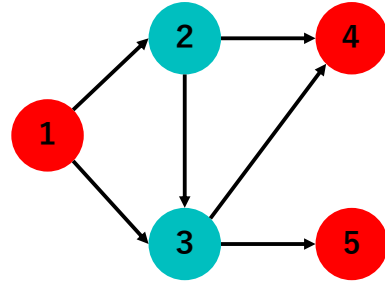


図-1: テストネットワーク

する:

$$\begin{aligned}
\mathbf{0} \leq w^k \perp \{ & (w^k - w^{k-1}) \\
& + A_+^T (\pi^k - \pi^{k-1}) + \mathbf{1}ds \} \geq \mathbf{0}, \tag{26}
\end{aligned}$$

$$\pi^k \geq \max [\pi^{k-1} - \mathbf{1}ds, \pi^0]. \tag{27}$$

これらの条件は、時刻 $\kappa-1$ と κ の間で、ネットワーク上の全てのリンク (i, j) において w_{ij} と π_i の減少量の合計が ds であること（もしくは待ち行列が完全に解消し $w_{ij}^k = 0$ であること）と、全てのノード k における π_k の減少量が ds 以下であること（もしくは $\pi_k^k = \pi_k^0$ であること）のみを要求している。

これらの要求は、最短経路探索を行いながら、たどり着いたリンクの待ち行列を物理的に可能な限り解消していく、という操作で全て満足することができる。具体的には次の渋滞解消アルゴリズムとして記述できる:

Step 0: $w^k := \mathbf{0}, \pi^k := \infty, \pi_o := 0$. 集合 \mathcal{A} に起点ノード o を追加。

Step 1: \mathcal{A} に含まれる“確定ノード”でないノードのうち、暫定の π_v^k が最小のもの v を \mathcal{A} から削除。 $v \neq o$ なら π_v^k をここで確定。

Step 2: $d_v := \pi_v^{k-1} - \pi_v^k$ とし、ノード v を“確定ノード”とする。 $\forall k \in O_v$ について $w_{vk}^{k-1} := \max[w_{vk}^{k-1} - (ds - d_v), 0]$ 。

Step 3: O_v の要素をもれなく \mathcal{A} に追加。 $\forall k \in O_v$ について $\pi_k^k > \pi_v^k + w_{vk}^{k-1} + c_{vk}^0$ であれば $\pi_k^k := \pi_v^k + w_{vk}^{k-1} + c_{vk}^0$ 。

Step 4: $\mathcal{A} = \emptyset$ なら $y^k := \mathbf{0}$ として、 $x^k = \{w^k, y^k, \pi^k\}$ と確定。そうでなければ Step 1 へ。

5. 予備実験

本章および続く 6. では、DUE 配分の数値実験を通して提案手法の計算効率性および正確性を評価する。まず本章では、提案手法による数値計算が正しく行われていることを、小規模ネットワークにおけるテスト計算により確認する。続いて次章において、提案手法に

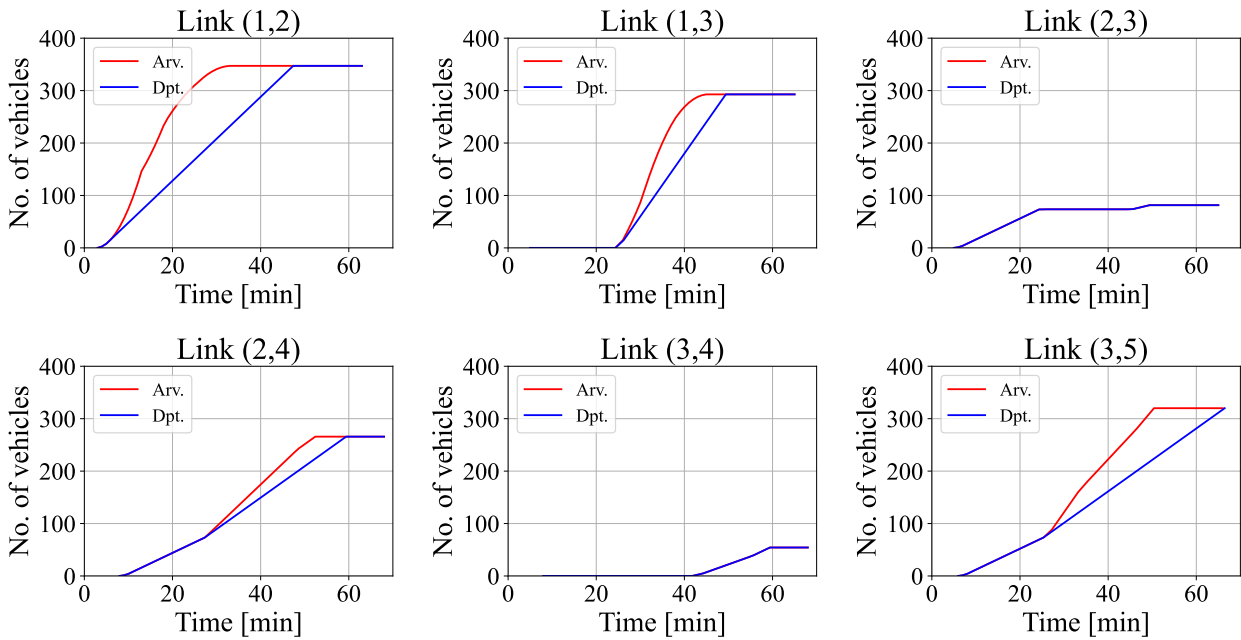


図-2: テストネットワークの累積交通量グラフ

表-1: テストケースのリンク定数

link	f.f.t.t.	capacity
(1, 2)	3.0	8.0
(1, 3)	15.0	12.0
(2, 3)	2.0	4.0
(2, 4)	5.0	6.0
(3, 4)	10.0	4.0
(3, 5)	1.0	6.0

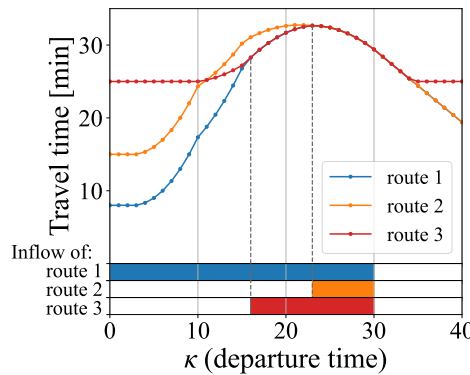


図-3: 1-4 OD ペアでの経路選択均衡

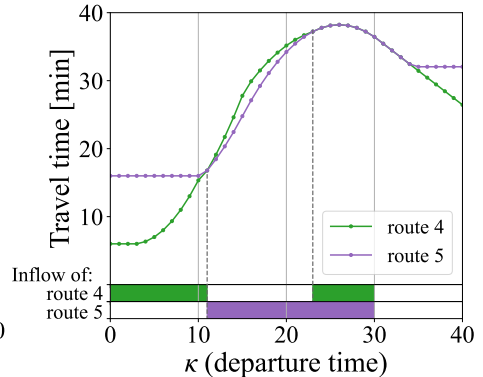


図-4: 1-5 OD ペアでの経路選択均衡

よる数値計算が、ネットワーク規模やその他実験条件を様々に変更させた場合にも頑健に実行できることを、各種の数値実験により提示する。

はじめに、本研究の数値実験を通しての計算機環境を示す：

- OS: windows 11
- CPU: AMD Ryzen 9 3960X 24-core 48-thread
- RAM: 128 GB
- Language: Python 3.9

なお、アルゴリズム中での線形計画問題の求解は全て Gurobi Optimizer¹⁴⁾ を使用した。

テスト計算には図-1 に示す 5 node 6 link テストネットワークを用い、起点ノードを 1, 終点ノードを 4, 5 と設

定した。各リンクの自由走行時間 (f.f.t.t.), 容量は表-1 の通りである。配分対象時間は 60 分間 ($S = [0, 60]$), 微小時間幅 ds は 1 分とし、このうち交通需要は前半の 30 分間 ($s \in [0, 30]$) に、各終点へそれぞれ計 320 の利用者を山なりに与えた。なお、収束判定式 (18) においては $\varepsilon = 10^{-6}$ とした。

本ケースにおいて、全時刻を通しての配分計算は約 0.4 秒で終了した。得られた交通状態から、各リンクにおけるボトルネックでの累積交通量グラフが図-2 のように作成できた。これらのグラフから、配分計算により得た交通量が各リンク間で整合していること (e.g., リンク (2, 3) とリンク (2, 4) の交通量の合計はリンク (1, 2) の交通量に一致する) が確認できる。

また図-3, 4 に、予備実験の結果から得られる、各出発時刻に対応する経路旅行時間と、経路流入交通流の

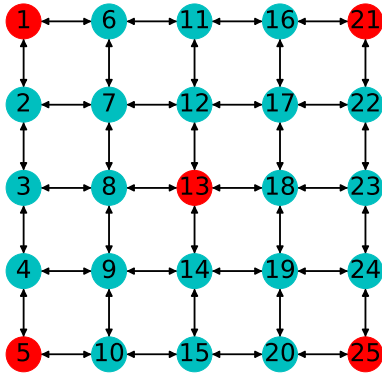


図-5: 単位ネットワーク ($m = 1$)

表-2: ネットワークデータ

m	1	2	3	4
nodes	25	81	169	289
links	80	288	624	1088
zones	5	13	25	41

表-3: ネットワークデータ

m	8	14	17	20	22	25
nodes	1089	3249	4761	6561	7921	10201
links	4224	12768	18768	25920	31328	40400
zones	145	421	613	841	1013	1301

有無とを図示する。図下部の帯は、着色されている時間帯には対応する経路への流入交通流が存在することを表す。ここで経路番号について、経路1: $1 \rightarrow 2 \rightarrow 4$, 経路2: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$, 経路3: $1 \rightarrow 3 \rightarrow 4$, 経路4: $1 \rightarrow 2 \rightarrow 3 \rightarrow 5$, 経路5: $1 \rightarrow 3 \rightarrow 5$, と定義した。両図から、2つの起終点ペア ($1 \rightarrow 4, 1 \rightarrow 5$) のいずれも、終点までの旅行時間が最小でない経路は選択されていないことが読み取れる^{注3)}。このことから、計算して得られた解はDUEの定義を真に満たしていることが明らかになった。

ノード1, 5, 13, 21, 25の位置に設定する(図中赤ノード)。全ての格子ネットワークにおいて、起点ノードはネットワークの中心ゾーン(そのノード番号は m を用いて $8m^2 + 4m + 1$ と表せる)とし、その他の全てのゾーンに対応するノードを終点ノードとした。以上のルールに基づき、実験に使用したネットワークのノード数・リンク数・ゾーン数を表-2, 3に示す。なお、格子ネットワーク中、全てのリンクの自由旅行時間は2とし、その容量は付録IIIに示すルールを用いて適宜渋滞が発生するよう調整した。

6. 数値実験

本章では様々な実験ケースを対象とした数値実験により、提案手法の、ネットワーク規模の変化に対する計算パフォーマンスの変化を確認する。

本章の構成は次に示す通りである。まず(1)にて、本章での数値実験の概要と条件を示す。続く(2)以降の各節では、数値実験の結果を通して提案手法の性能や特徴を紹介する。(2)では、素朴な解法と比較して手案手法が効率的であること、(3)では提案手法がこれまでに適用例のない大規模ネットワークへも適用できることを、それぞれ示す。(4)では微小時間幅 ds を変化させた場合の、提案手法の性質について確認する。(5)では提案手法の計算収束過程の特徴について述べ、最後に(6)にて提案手法により得た均衡解の正確性を示す。

(1) 実験準備

はじめに、本章で示す数値実験の条件や前提について説明する。

本章で示す数値実験は格子ネットワークを用いて行った。格子ネットワークは、図-5に示す単位ネットワークを縦横にそれぞれ m 個並べて作成する。単位ネットワークにおいて、起終点ノードに対応する“ゾーン”を

本章の数値実験では特に断らない限り、配分対象時間 S は60分間、微小時間幅 ds は1分とした。このうち交通需要は前半の30分間に、各終点へそれぞれ山なりに与えた。交通需要の規模は、小(demand-S)・中(demand-M)・大(demand-L)の3段階用意した(需要決定に際するルールの詳細は付録IIIに記す)。これらの需要規模はそれぞれ、ネットワーク中での渋滞が発生するリンクの割合が、小需要では約5%、中需要では約35%、大需要では50%となるように設定されている。なお渋滞リンクの割合が50%であることは、本実験におけるネットワークが一起点多終点、かつ全てのリンクがその逆向きのリンクとペアになっていることを踏まえると、利用されうる全てのリンクが渋滞していることを意味する。また収束判定式(18)の基準は前章同様に $\varepsilon = 10^{-6}$ とした。

本研究では3.において、DUE配分を相補性問題として定式化している。これに伴い本章では、提案手法と効率性を比較する対象として、相補性問題からメリット関数を構築し素朴に解く手法を導入する。具体的には、[DUE-QP(κ)]を解く代わりに次に示すFisher¹⁵⁾の

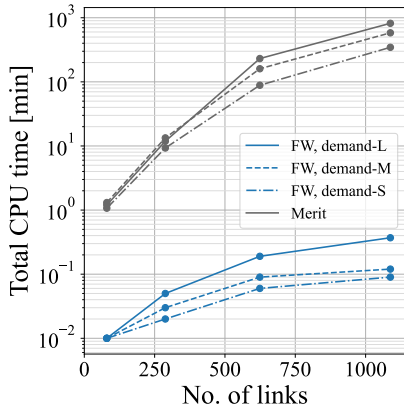


図-6: 提案手法の効率性

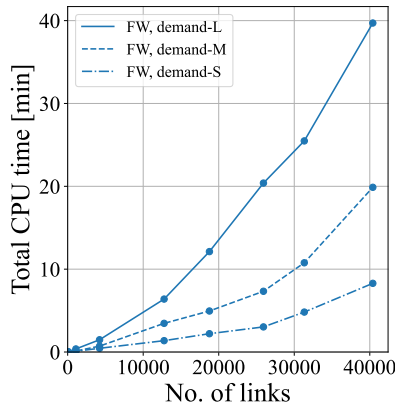


図-7: 大規模問題への適用結果

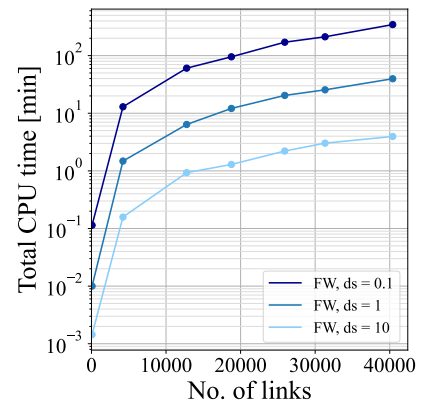


図-8: 時間刻み変更結果

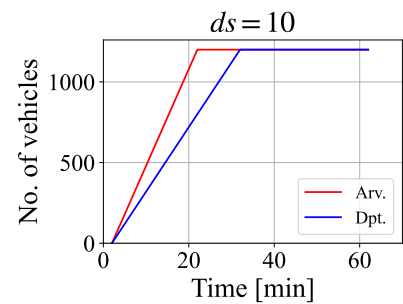
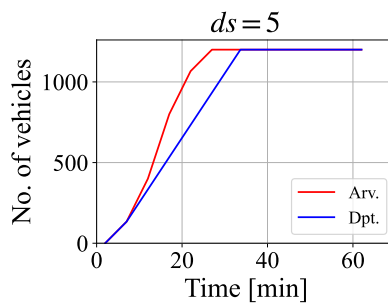
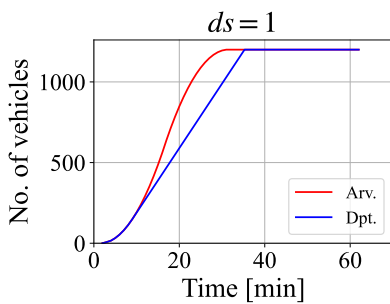


図-9: 微小時間幅とリンク (13, 12) の累積交通量グラフ

メリット関数を最小化する問題：

$$\min_{\mathbf{x}^k} \sum_{l=1}^{2L+N-1} \phi(x_l^k, [\mathbf{M}\mathbf{x}^k + \mathbf{b}^k]_l)^2,$$

$$\text{s.t.} \quad \pi^k \geq \max [\pi^{k-1} - \mathbf{1}ds, \pi^0],$$

$$\text{where} \quad \phi(p, q) \equiv \sqrt{p^2 + q^2} - (p + q).$$

を解く。なお、この問題の求解には、Python の科学計算ライブラリ SciPy¹⁶⁾ に実装されている Broyden-Fletcher-Goldfarb-Shanno (BFGS) 法最適化パッケージを用いた。BFGS 法の収束判定については [DUE-QP(κ)] の目的関数値で行い、その判定基準は Frank-Wolfe 法と同様の $z(\mathbf{x}^k) < 10^{-6}$ とした。ただし十分な繰り返し計算を経てこの判定基準が満たされない場合、 $z(\mathbf{x}^k) < 10^{-4}$ を満たしていれば \mathbf{x}^k を解とみなし、計算を次の時刻に進めることとする。

(2) 比較実験結果

本節では、6. (1) にて導入したメリット関数最適化問題の手法と比べて、提案手法の求解が効率的であることを示す。表-2 に示す小規模格子ネットワークについて、提案手法とメリット関数最適化問題の手法との比較実験を行った。実験は用意した三種類の需要全てに対して行った。

全ての配分対象時間を解き終えるのに要した CPU time の結果を図-6 に示す。グラフの横軸はネットワークのリンク数、縦軸は要した CPU time (対数軸) である。なお図中の“FW”は提案手法を、“Merit”はメリット関数手法をそれぞれ表す。

図-6 より、提案手法とメリット関数手法との間で、CPU time に関しておよそ 100 倍から 1000 倍ほどの差があり、かつその差はネットワークが大きくなるにつれて開いていく傾向があることが読み取れる。このことから提案手法はメリット関数手法に比べて、計算効率の観点から大幅に有利であることが示された。

両手法間では、リンク数が 1000 程度のネットワークにおいても図-6 に示した通りの卓越した計算効率の差があることから、以降のさらに巨大なネットワークに関する数値実験では提案手法のみをその対象とする。

(3) 大規模ネットワークにおける実験結果

本節では、問題規模が大きくなったとしても提案手法での求解が効率的であることを示す。表-3 に示す中・大規模格子ネットワークでの DUE 配分について、提案手法を適用する実験を行った。実験は用意した三種類の需要全てに対して行った。

全ての配分対象時間を解き終える計算に要した CPU

time をそれぞれ図-7 に示す。グラフの横軸はネットワークのリンク数、縦軸は要した CPU time (線形軸) である。

DUE 配分の解法に関する既存の研究で適用例のあるネットワーク規模は、リンク数にして多くとも 10000 弱であった²⁾。一方で本研究の提案手法では、この適用限界を大きく上回る、リンク数にして 40000 の規模のネットワークに対しても問題なく適用可能である。ネットワーク中での渋滞が多く発生するほど計算には時間を要するものの、最大限に渋滞した状況下での巨大ネットワークに対しても要する CPU time は 40 分ほどである。

(4) 微小時間幅 ds を変更した実験結果

提案手法での計算においては、外生的なパラメータである微小時間幅 ds を自由に変更することができる。本節ではこのことを数値実験を通して紹介する。実験は表-2, 3 から抜粋したネットワークに関してそれぞれ $ds = 0.1, 1, 10$ として行った。なお、配分対象時間はいずれの場合も 60 分間である。需要は大需要 (demand-L) のみを扱い実験した。

実験の結果を図-8 に示す。グラフの軸についてはこれまで同様に、横軸はネットワークのリンク数、縦軸は全ての配分対象時間を解き終えるのに要した CPU time (対数軸) である。

S を 60 分、 ds を 0.1, 1, 10 分としたとき、離散時刻 κ の個数 (解くべき均衡問題の個数を意味する) はそれぞれ 600, 60, 6 個である。このことから、各場合の計算に要する CPU time はそれぞれ 10 倍程度異なることが理論的に予想され、実際に図-8 に示す結果もこのことを示唆している。

微小時間幅 ds を変更すると、得られる解やそこから導出される交通状態の“解像度”が変化する。図-9 には、単位ネットワーク ($m = 1$) のリンク (13, 12) における累積交通量グラフを、 $ds = 1, 5, 10$ としてそれぞれ求めた結果を示す。図からも、 ds が大きくなるほど交通状態の解像度が下がり、累積曲線に角が残ることが読み取れる。以上のことより、解の解像度と計算時間との間にはトレードオフの関係がある。

(5) 収束効率

本節では提案手法による数値計算における、Frank-Wolfe 法の収束過程について確認する。ここで示す実験の条件は、 $m = 25$ のネットワークを対象として、大需要 (demand-L) $\cdot ds = 1$ とした。

図-10 に、ある離散時刻 κ について、計算に要した CPU time と、その時点の反復での解における最短旅行時間 π^κ の平均相対誤差 Err との関係を示す。図には、 $m = 25$ ネットワークの計算で最も計算に時間を要した離散時刻 ($\kappa = 7$) と、平均的な計算時間を要した離散

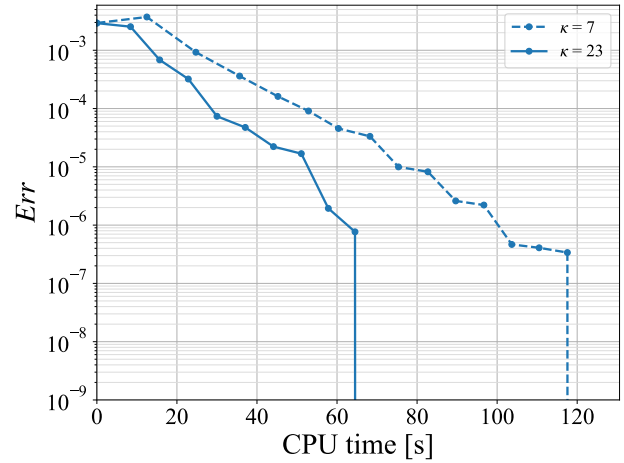


図-10: 提案手法の収束過程 ($m = 25$)

時刻 ($\kappa = 23$) の結果を示した。なお、図中プロット上の点は各反復計算を表している。ここで反復計算 n 回時点での平均相対誤差 $Err(n)$ は次の式で計算した：

$$Err(n) \equiv \frac{1}{N-1} \sum_{k \in N \setminus o} \frac{|\pi_{(n)k}^\kappa - \pi_k^{\kappa*}|}{\pi_k^{\kappa*}}. \quad (28)$$

本来 $\pi_k^{\kappa*}$ は厳密解を表すが、図-10 では $\pi_k^{\kappa*}$ を、十分に大きな反復計算回数 (99 回) を経た数値解 $\pi_{(99)k}^\kappa$ としている ($\pi_{(99)k}^\kappa$ 自体の正確性は次節で述べる)。

図-10 の結果から、 $\kappa = 7$ では 15 回ほどの計算で、 $\kappa = 23$ では 10 回ほどの計算で、それぞれ 99 回計算を行った場合の解と全く同じものが得られていることがわかる。加えて、提示したどちらの場合においても、最終的な収束値に収束する直前までは緩やかな収束速度であるものの、収束値に向かう最後の一回の繰り返し計算で急激に収束値を得ている様子が確認できる。

このような収束挙動は、Frank-Wolfe 法で解いている問題の最適解が、問題の実行可能領域の端点に相当することと関連している。実行可能領域の端点とは、問題の制約条件のうちいくつか等式で成立するような解状態である。また Frank-Wolfe 法は、まず線形計画問題を解くことにより実行可能領域の端点を見つけ出し、続いて次元探索により最適な次点解を求めるアルゴリズムであった。今回対象としている経路選択 DUE 配分では、定式化した均衡条件が相補性問題であったことを踏まえると、均衡状態が成り立つような状態 (均衡解) においては、[DUE-QP(κ)] の制約条件の約半分は等式にて成立することが予想できる。すなわち、経路選択 DUE 配分の均衡解は、[DUE-QP(κ)] の端点解である。以上の観点から、経路選択 DUE 配分と Frank-Wolfe 法とは親和性がきわめて高いと結論付けられる。

(6) 解の正確性

本章の最後に、提案手法により計算した数値解（収束値）の正確性について示す。具体的には $\mathbf{x}_{(99)}^\kappa$ が、均衡条件 (11)-(15) を精度よく満足することを示す。

精度評価には、以下に定義する $E_w^\kappa(n)$, $E_y^\kappa(n)$, $E_\pi^\kappa(n)$ を用いる：

$$\begin{aligned} E_w^\kappa(n) &\equiv \left| \mathbf{w}_{(n)}^{\kappa T} \left\{ \alpha(\mathbf{w}_{(n)}^\kappa - \mathbf{w}_{(99)}^{\kappa-1}) - \mathbf{y}_{(n)}^\kappa \right. \right. \\ &\quad \left. \left. + \alpha \mathbf{A}_+^T (\boldsymbol{\pi}_{(n)}^\kappa - \boldsymbol{\pi}_{(99)}^{\kappa-1}) + \boldsymbol{\mu} \right\} \right|, \\ E_y^\kappa(n) &\equiv \left| \mathbf{y}_{(n)}^{\kappa T} \{ \mathbf{c}^0 + \mathbf{w}_{(n)}^\kappa + \mathbf{A}^T \boldsymbol{\pi}_{(n)}^\kappa \} \right|, \\ E_\pi^\kappa(n) &\equiv \left| \boldsymbol{\pi}_{(n)}^{\kappa T} \{ -\mathbf{A} \mathbf{y}_{(n)}^\kappa - \mathbf{q}^\kappa \} \right|. \end{aligned}$$

まず次に示す二項目：(i) 均衡条件 (11)-(13) について、相補性条件を成す両要素の双方が非負であること、(ii) 均衡条件の不等式 (14), (15) が満たされていること、を確認したうえで、 $E_w^\kappa(n)$, $E_y^\kappa(n)$, $E_\pi^\kappa(n)$ の値がそれぞれ 0 であれば、 $\mathbf{x}_{(n)}^\kappa$ は均衡解である。

実際、 $m = 25$ のネットワークにおいて、 $n = 99$ として以上を計算すると、 $E_w^\kappa(99)$, $E_y^\kappa(99)$, $E_\pi^\kappa(99)$ の値は全ての κ において全て 10^{-8} 未満となり、十分に精度の良い解であることが確認できた。

7. おわりに

本研究では、一起点多終点ネットワークにおける経路選択 DUE 配分の効率的な計算方法を提案した。具体的には、DUE 配分を混合線形相補性問題として定式化した。そしてこの問題を起点出発時刻別に分解し、また最適化問題に変換することで、DUE 配分の均衡解を求めるアルゴリズムを提案した。その結果、提案手法では定式化した問題に対してきわめて精度の良い均衡解を求めることができ、その効率性も、小規模な問題の時点で、素朴なメリット関数手法の 100 倍から 1000 倍以上であることが示された。さらに、提案手法はこれまでに例のない大規模問題に適用可能であることも確認できた。

本研究の今後の発展性として、本研究の成果を用いた経路・出発時刻同時選択 DUE 配分の効率的解法の開発がある。経路・出発時刻同時選択 DUE 配分の解法に関して、**1. (3)** にも述べたように既存の手法はいずれも、問題が経路選択 DUE 配分を包含していることを明示的に用いてはいない。この点から、既存手法よりもさらに効率的な解法を提示できる可能性は未だ残されている。同時選択 DUE 配分に対し、経路選択 DUE 配分がその部分問題となるような分解を施すことができれば、本研究の成果を活用した新たな効率的解法の構築が期待できる。

謝辞： 本研究は、日本学術振興会・科学研究費補助金 (JP20J21744, JP21H01448) の助成を受けた研究の一部です。ここに記し、感謝を表します。

付録 I [DUE-FW-LP] の制約条件

本章では、[DUE-FW-LP] の制約条件に関する以下の命題を証明する。

命題 4 [DUE-FW-LP] の変数 $\hat{\mathbf{x}} = \{\hat{\mathbf{w}}, \hat{\mathbf{y}}, \hat{\boldsymbol{\pi}}\}^T$ が式 (19), (20) を満たすならば、制約条件式 (21) は常に成立する。

証明. 条件 (19) が成り立っていることから、[DUE-FW-LP] の各変数について次の式も成り立つ：

$$\alpha \hat{\mathbf{w}} - \hat{\mathbf{y}} + \alpha \mathbf{A}_+^T \hat{\boldsymbol{\pi}} - \alpha \boldsymbol{\beta}^\kappa \geq \mathbf{0}. \quad (\text{I.1})$$

式 (I.1), (21), (20) をそれぞれ要素表示に書き下す：

$$(\hat{w}_{ij} - w_{ij}^{\kappa-1}) - \frac{ds}{\mu_{ij}} \hat{y}_{ij} + (\hat{\pi}_i - \pi_i^{\kappa-1}) + ds \geq 0, \quad (\text{I.2})$$

$$(\hat{\pi}_i - \pi_i^{\kappa-1}) + ds \geq 0, \quad (\text{I.3})$$

$$(\hat{w}_{ij} - w_{ij}^{\kappa-1}) - \frac{ds}{\mu_{ij}} \hat{y}_{ij} \leq 0. \quad (\text{I.4})$$

ここで式 (I.2) の左辺は、式 (I.3) の左辺と式 (I.4) の左辺との和であることに注意すると、式 (I.2) が成立しているもとでは、式 (I.4) が成立しているなら式 (I.3) は必ず成立することが確認できる。したがって式 (I.2) のもとで式 (I.4) は式 (I.3) の十分条件であり、同様に条件 (19) のもとで式 (20) は式 (21) の十分条件である。以上より命題は示された。□

命題 4 より、[DUE-FW-LP] の制約条件として式 (20) を採用した場合の解は式 (21) を満たす。ゆえに、元の制約条件式 (21) を式 (20) で置換することは妥当である。

付録 II 渋滞進展アルゴリズムでの解の不整合とその解消

4. (1) にも述べたように、渋滞進展アルゴリズムにおいて単に Frank-Wolfe 法で [DUE-QP(κ)] を解くのみでは、均衡条件を全て満たしていながらネットワーク上で物理的に不整合な解が得られる場合がある。本章ではこの不整合の発生要因、およびその解消のために渋滞進展アルゴリズムの **Step 5** が有効であることを示す。

ネットワーク上のリンク $(i, j) \in \mathcal{L}$ は全て、出発時刻 κ において次の三種類のいずれかに分類できる：

1. \mathcal{L}_1^κ : 出発時刻 κ において、あるノードへの最短経路上にあり、流入交通流がある。
2. \mathcal{L}_2^κ : 出発時刻 κ において、あるノードへの最短経路上にあるが、流入交通流がない。

3. \mathcal{L}_3^k : 出発時刻 k において、どのノードへの最短経路上にもない。

これらを用いて、ネットワーク上のノード $k \in \mathcal{N}$ も全て、出発時刻 k において次の二種類に分類できる：

1. \mathcal{N}_1^k : 出発時刻 k において、 \mathcal{L}_1^k のリンクのみを経由して起点からたどり着ける。
2. \mathcal{N}_2^k : 出発時刻 k において、 \mathcal{L}_2^k や \mathcal{L}_3^k のリンクを経由しなければ起点からたどり着けない。

以上の分類のうち、 \mathcal{L}_2^k に該当するリンク (i, j) や \mathcal{N}_2^k に該当するノード k に対応する変数 w_{ij}^k, π_k^k は、均衡条件のみからは一意に決定されない。このため、**[DUE-QP(k)]** を Frank-Wolfe 法により解いて得られた π_k^k が、起点から最短経路を辿って k に至る旅行時間と異なる場合がある。

以上の状況を具体的に数式を用いて説明する。本研究で定式化した均衡条件は式 (11) - (15) であった。このうち式 (12) には、起点からその他全てのノードへ向かう最短経路を、リンクへの流入交通量 y^k の有無により判断し、その経路上で w^k, π^k を一意に定める役割がある。一方で、時刻 k において \mathcal{L}_2^k に含まれるリンク (i, j) では、分類条件から $y_{ij}^k = 0$ である。このとき式 (12) より、リンク (i, j) について、任意のノードへの最短経路上にあるリンクが満たすべき条件：

$$c_{ij}^0 + w_{ij}^k + \pi_i^k - \pi_j^k = 0 \quad (\text{II.1})$$

が無効となる。これによりリンク (i, j) に関する $w_{ij}^k, \pi_i^k, \pi_j^k$ を規定する条件は次の二式：

$$\mathbf{0} \leq w^k \perp \{ (w^k - w^{k-1}) + \mathbf{A}_+^\top (\pi^k - \pi^{k-1}) + \mathbf{1}ds \} \geq \mathbf{0}, \quad (\text{II.2})$$

$$\pi^k \geq \max [\pi^{k-1} - \mathbf{1}ds, \pi^0] \quad (\text{II.3})$$

のみとなり、 π_j^k は一意に定まらない。さらに、ノード j に続くリンク (j, k) が \mathcal{L}_2^k に含まれる場合、 π_j^k が非一意であることにより式 (II.2) から w_{jk}^k も非一意となる。そして、 w_{jk}^k により規定される π_k^k も非一意となり、という具合に解の非一意性は下流側に伝播する。

このような不整合を正しく修正するために **Step 5** が必要である。**Step 5** は、リンク旅行時間も逐次更新するダイクストラ法である。具体的には、起点に近いノードから順に π を最短経路旅行時間に、そのノードから延びるリンクについて w を式 (II.2) を満たすように、それぞれ書き換えながら、全てのノードへ最短経路を探索する。この操作を経ると式 (II.2) は成立し、式 (II.3) についても、式 (II.2) から $\pi^k \geq \pi^{k-1} - \mathbf{1}ds$ が、最短経路探索から $\pi^k \geq \pi^0$ がそれぞれ満たされる。また、最短経路探索から物理的整合性も満たされる。すなわ

ち、本操作を経た解 x^k はネットワーク上で物理的に整合的で、かつ全ての均衡条件を満たす。

付録 III 格子ネットワークの容量・需要

本章では、本研究での格子ネットワークを用いた数値実験において入力データとなる、ネットワーク上のリンク容量と、ネットワーク上での OD 需要の決定方法を述べる。

本研究での実験では、交通需要が一起点多終点であることを考慮し、ネットワークのリンク容量を適宜調整して行った。具体的には、ネットワーク上で起点に近いリンクの容量は相対的に大きく、起点から遠いネットワークの容量は相対的に小さく設定した。これは、一起点多終点の交通需要に対しても、実験の結果として意味のある程度の渋滞発生を確保する（起点の周辺のみでしか渋滞が発生しない、といった状況を避ける）ためである。具体的には、以下に示す、Dial¹⁷⁾ のアルゴリズムを応用したアルゴリズムによりリンク容量を決定した：

Step 0: リンク (i, j) のリンク尤度を次式で計算する：

$$L(i, j) = \begin{cases} e^{[\pi_j^0 - \pi_i^0 - c_{ij}^0]} & \text{if } \pi_i^0 < \pi_j^0, \\ 0 & \text{otherwise.} \end{cases} \quad (\text{III.1})$$

Step 1: π_i^0 が小さいノード i から順に、そのノードから延びるリンク (i, j) の重みを次式で計算する：

$$w(i, j) = \begin{cases} L(i, j) & \text{if } i = o, \\ L(i, j) \sum_{m \in \mathcal{I}_i} w(m, i) & \text{otherwise.} \end{cases} \quad (\text{III.2})$$

Step 2: 起点からノード k への仮想需要を \hat{q}_k と定義する。 π_j^0 が大きいノード j から順に、そのノードへ延びるリンク (i, j) の交通量を次式で計算する：

$$x(i, j) = \left[\sum_{l \in \mathcal{O}_j} x(j, l) + \hat{q}_j \right] \frac{w(i, j)}{\sum_{l \in \mathcal{I}_j} w(l, j)}. \quad (\text{III.3})$$

Step 3: $\pi_{\max}^0 \equiv \max_k \pi_k^0$ とする。リンク (i, j) の容量を次の式で計算する：

$$\mu_{ij} = x(i, j) \times \frac{1}{1 + (2\pi_i^0 / \pi_{\max}^0)}. \quad (\text{III.4})$$

Step 4: $\mu_{ij} = 0$ と計算された全てのリンクについて、 $\mu_{ij} = 1$ と改めて設定する。

仮想需要 \hat{q}_k については、終点ノード以外のノードには 0 を、終点ノード k には次の式により計算される値を、それぞれ設定した：

$$\hat{q}_k = \bar{q} N_d \frac{\exp(-v\pi_k^0)}{\sum_{l \in \mathcal{N}_d} \exp(-v\pi_l^0)}. \quad (\text{III.5})$$

ここで N_d は終点ノード集合（その要素数は N_d ）であり、 \bar{q}, ν はパラメータである。本稿で掲載した数値実験では、 $\bar{q} = 40$ veh/min, $\nu = 0.01$ と設定した。

各終点に与える時々刻々の交通需要については、需要が存在する時間帯内の平均値が \bar{q} となるような三角形形状の需要を“大需要 (demand-L)”とし、絶対量にしてその 0.3 倍のものを“中需要 (demand-M)”, 0.2 倍のものを“小需要 (demand-S)”とそれぞれ定義した。

付録 IV 実都市ネットワークへの適用

本研究での提案手法は格子ネットワークに限らず一般の都市交通ネットワークへ適用可能である。その一例として、本章では Transportation Networks for Research¹⁸⁾ が提供している GoldCoast, Australia ネットワーク (4807 nodes, 11140 links, 1068 zones) への適用例を示す。計算条件は 6. での条件と等しく設定し、需要は大需要 (demand-L) のみを扱い実験した。

実験の結果、全ての計算に要した CPU time は約 19.7 分であり、ネットワーク全体中の渋滞発生リンクの割合は約 13%であった。格子ネットワークのときと同様に計算は正確に行われているものの、同規模の格子ネットワーク ($m = 14$) での同条件の計算 CPU time が約 6.4 分であったことを考慮すると、実都市交通ネットワークでの計算は格子ネットワークのときよりも計算コストが大きいと考えられる。この原因は、ノード・リンクの接続構造が複雑であることや、均衡解において各変数のオーダーが大きく異なることなど、複数の可能性が考えられる。これに関する詳細な検討は今後の課題としたい。

NOTES

- 注1) 1. (3) に挙げた既存研究は全て、交通流を連続体として扱う“流体モデル”である。一方でこれらとは別のアプローチとして、交通流中の各利用者を離散粒子として扱う“粒子モデル”が存在する。この粒子モデルについても、経路選択 DUE 配分の解法 (井料 (2011)¹⁹⁾) が提案されている。しかしながらこのモデルは元来、比較的狭小な範囲における交通均衡状態を精緻に再現し、その性質を解析することを想定したものである。粒子モデル解法の計算コストは、モデルの性質上、対象とする交通ネットワークの規模と、これと一般に比例関係にある交通需要の総量との双方から直接影響を受けるため、大規模ネットワークへ適用するという本研究の目的には適わない。
- 注2) 本稿に登場する単位行列 I , 零行列 (ベクトル) $\mathbf{0}$, 1 行列 (ベクトル) $\mathbf{1}$, その他定数行列 (ベクトル) は原則として、その演算に対して適切な次元であるとする。ただし、特に注意が必要な場合には、行列 (ベクトル) の次元を下付き添え字で表す。
- 注3) 提案手法において、均衡状態での交通流は全てリンクの流入率率として得られるため、図-3, 4 の $23 \leq \kappa \leq 30$ の時刻帯において実際には全ての経路が使用されていない場合も考えられる。しかしながら、このような場合を全て勘案しても、「どの利用者也自分一人が経路を変更して

も、各自の総旅行費用をそれ以上改善できない」DUE 状態であることに変わりはない。

REFERENCES

- Ban, X. J., Liu, H. X., Ferris, M. C. and Ran, B.: A link-node complementarity model and solution algorithm for dynamic user equilibria with exact flow propagations, *Transportation Research Part B: Methodological*, Vol.42, No.9, pp.823–842, 2008.
- Gentile, G.: Solving a dynamic user equilibrium model based on splitting rates with gradient projection algorithms, *Transportation Research Part B: Methodological*, Vol.92, pp.120–147, 2016.
- Friesz, T. L. and Mookherjee, R.: Solving the dynamic network user equilibrium problem with state-dependent time shifts, *Transportation Research Part B: Methodological*, Vol.40, No.3, pp.207–229, 2006.
- Akamatsu, T.: An efficient algorithm for dynamic traffic equilibrium assignment with queues, *Transportation Science*, Vol.35, No.4, pp.389–404, 2001.
- Lo, H. K. and Szeto, W. Y.: A cell-based variational inequality formulation of the dynamic user optimal assignment problem, *Transportation Research Part B: Methodological*, Vol.36, No.5, pp.421–443, 2002.
- Long, J., Huang, H.-J., Gao, Z. and Szeto, W. Y.: An intersection-movement-based dynamic user optimal route choice problem, *Operations Research*, Vol.61, No.5, pp.1134–1147, 2013.
- Long, J., Szeto, W. Y., Gao, Z., Huang, H.-J. and Shi, Q.: The nonlinear equation system approach to solving dynamic user optimal simultaneous route and departure time choice problems, *Transportation Research Part B: Methodological*, Vol.83, pp.179–206, 2016.
- Han, K., Eve, G. and Friesz, T. L.: Computing dynamic user equilibria on large-scale networks with software implementation, *Networks and Spatial Economics*, Vol.19, No.3, pp.869–902, 2019.
- 長江剛志, 赤松隆, 清水廉, 符皓然: 経路・出発時刻同時選択型の動的利用者均衡配分の求解法: 二次計画問題アプローチ, 土木学会論文集 D3 (土木計画学), Vol.76, No.3, pp.264–281, 2020. [Nagae, T., Akamatsu, T., Shimizu, R. and Fu, H.: A quadratic programming approach for solving a dynamic user equilibrium with simultaneous departure time and route choice, *Transaction of the Japan Society of Civil Engineers*, Vol.76, No.3, pp.264–281, 2020.]
- Kuwahara, M. and Akamatsu, T.: Dynamic equilibrium assignment with queues for a one-to-many od pattern, *Transportation and Traffic Theory*, Vol.12, pp.185–204, 1993.
- 土木学会: 交通ネットワークの均衡分析-最新の理論と解法, pp. 280–281, 丸善, 1998. [the Japan Society of Civil Engineers: *koutuu-network no kinnkoubunnseki saishinn no rironn to kaihou*, pp. 280–281, Maruzenn, 1998.]
- Frank, M. and Wolfe, P.: An algorithm for quadratic programming, *Naval research logistics quarterly*, Vol.3, No.1-2, pp.95–110, 1956.
- Dijkstra, E. W.: A note on two problems in connexion with graphs, *Numerische mathematik*, Vol.1, No.1, pp.269–271, 1959.
- Gurobi Optimization, LLC: *Gurobi optimizer reference manual*, 2022, Accessed 19 Oct., 2022.
- Fischer, A.: A special newton-type optimization method, *Optimization*, Vol.24, No.3-4, pp.269–284, 1992.
- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M.,

- Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P. and SciPy 1.0 Contributors: SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, *Nature Methods*, Vol.17, pp.261–272, 2020.
- 17) Dial, R. B.: A probabilistic multipath traffic assignment model which obviates path enumeration, *Transportation research*, Vol.5, No.2, pp.83–111, 1971.
- 18) Transportation Networks for Research Core Team: *Transportation Networks for Research*, 2022, Accessed 19 Oct., 2022.
- 19) 井料隆雅: 車両を離散化した動的交通量配分問題の nash 均衡解の解法, 土木学会論文集 D3 (土木計画学), Vol.67, No.1, pp.70–83, 2011. [Iryo, T.: Solution algorithm of nash equilibrium in dynamic traffic assignment with discretised vehicles, *Transaction of the Japan Society of Civil Engineers*, Vol.67, No.1, pp.70–83, 2011.]

(Received ?? ??, 2022)

(Accepted ?? ??, 2022)

EFFICIENT ALGORITHM FOR SOLVING DYNAMIC USER EQUILIBRIUM TRAFFIC ASSIGNMENT

Masanao WAKUI, Takara SAKAI and Takashi AKAMATSU

This paper presents an efficient algorithm for solving dynamic user equilibrium (DUE) traffic assignment problems with one-to-many origin-destination demand. We first formulate the DUE problem, with the point queue model, as a linear complementarity problem (LCP). Then, we transform the LCP into a quadratic programming problem (QP). The QP enables us to build an efficient algorithm based on the convex combination method (Frank–Wolfe algorithm). Numerical experiments reveal that the proposed algorithm can (i) solve extremely large-scale problems to which no conventional methods can be applied, (ii) calculate the solution more than 1000 times faster than a general-purpose solver for optimization problems, and (iii) yield an extremely accurate equilibrium solution.