

動的利用者均衡配分の数値解法の比較実験*

Numerical Experiments of Some Algorithms for Dynamic Equilibrium Assignment
with One-to-Many / Many-to-One OD Pattern

赤松 隆**・大石直史***・高松 望***

by Takashi AKAMATSU, Naoshi OISHI and Nozomu TAKAMATSU

1. はじめに

動的な利用者均衡(DUE: *Dynamic User Equilibrium*)配分は、最近の研究¹⁾²⁾により、理論的には多くの進展をみた。しかし、現実的な大規模ネットワークでも均衡解を計算可能な信頼性の高いアルゴリズムについては、未だ確立しているとは言えない。

この問題に対して、著者等³⁾は、1起点・多終点(あるいは多起点・1終点)OD構造のDUE配分と等価な変分不等式問題(VIP)/相補性問題(NCP)を導出し、それらを活用した2種類の計算法(部分線形化法と大域収束的Newton法)を提案している。本稿は、その研究結果を踏まえ、提案アルゴリズムの収束性、効率性、大規模問題への適用可能性等を数値実験によって検討することを目的とする。

本論文の構成は以下の通りである。第2節では、記号および解くべきDUE配分問題の定義を行なう。続く第3節では、DUE配分問題に対する2種類の提案アルゴリズムを整理する。第4節では、数値実験の各種条件を述べた後、各種条件下での実験結果およびその考察を示す。

2. 動的な利用者均衡配分

2.1 ネットワーク・OD需要・時間系

N個の要素を持つノード集合NおよびL個の要素を持つ有向リンク集合Lからなるネットワーク $G(N,L)$ を考える。各ノードは整数の連番で区別され、ノードiからjへのリンクは (i,j) と表される。本稿では、1起点・多終点のネットワークを考える(多起点・1終点の場合については紙面の制約により省略)。起点がoで起点oを時刻sまでに出発する累積交通量 Q_{od}^s は

所与であるとし、(起点出発時刻についての)その流量を $q_{od}^s \equiv dQ_{od}^s / ds$ と書く。起点oの出発時刻sは、適当な単位で測った区間 $[0,T]$ の実数である。ただし、以下では数学的煩雑さを避けるため、時間は離散系で近似されている(i.e. 起点出発時刻を自然数で識別)とし、その離散的な出発時刻の集合をUと書く。

2.2 起点出発時刻別分解

起点が唯一の場合のDUE配分は起点出発到着時刻sごとに分解し、sの昇順に逐次計算することができる¹⁾²⁾³⁾。その分解された定式化では、出発時刻sごとに以下のように定義された2種類の変数(y_{ij}^s, τ_i^s)が中心的な役割をはたす。まず、均衡状態において起点を時刻sに出発する車がノードiに最も早く到着する時刻を $\hat{\tau}_i^s$ 、起点からノードiまでの所要時間を $\tau_i^s \equiv \hat{\tau}_i^s - s$ と書く。このとき、起点出発時刻がsの車のリンク (i,j) への(sについての)流入率は、 $y_{ij}^s \equiv dA_{ij}(\hat{\tau}_i^s) / ds$ と定義される。ここで、 $A_{ij}(t)$ は時刻tまでにリンク (i,j) へ流入した累積台数。

本稿では、リンクでの待ち行列を point queue model で表現する。この場合、分解されたDUE問題では、起点出発時刻sの車のリンク旅行時間 c_{ij}^s は以下の表現に帰着する：

$$c_{ij}^s = \max. [\alpha_{ij} y_{ij}^s + \beta_{ij} + \tau_i^s, m_{ij}] \quad (1)$$

ここで、 $\alpha_{ij} \equiv ds / \mu_{ij}$ 、 $\beta_{ij} \equiv c_{ij}^{s-ds} - \tau_i^{s-ds}$ 、 m_{ij} と μ_{ij} は、各々、リンク (i,j) の自由走行時間、最大流出率(ともに所与の定数)、dsは離散近似された時刻の微小な時間間隔である。出発時刻sの均衡フローを計算する際には、出発時刻s-dsに関する変数は全て計算を終えているので、 β_{ij} は全て与件とみなせる。

2.2 DUE配分と等価な変分不等式問題

起点出発時刻sに対応するDUE配分は、以下のような非線型相補性問題NCP(F)/変分不等式問題VIP(K_1, F)と等価である²⁾³⁾：

* Keywords: 利用者均衡, 動的な交通量配分, 交通流, 変分不等式

** 正会員 工博 豊橋技術科学大学 知識情報工学系
(〒441 豊橋市天伯町雲雀ヶ丘1-1
E-mail: akamatsu@tutkie.tut.ac.jp)

*** 学生会員 豊橋技術科学大学 知識情報工学専攻

a) NCP(F): Find $\mathbf{x}^*$ in $K_1 = R_+^L \times R_+^N$ such that

$$\mathbf{x}^* \cdot \mathbf{F}(\mathbf{x}^*) = 0, \mathbf{x}^* \geq 0, \mathbf{F}(\mathbf{x}^*) \geq 0 \quad (2)$$

b) VIP($K_S, \mathbf{F}$): Find $\mathbf{x}^*$ in K_1 such that

$$\mathbf{F}(\mathbf{x}^*) \cdot (\mathbf{x} - \mathbf{x}^*) \geq 0 \quad \forall \mathbf{x} \in K_1 \quad (3)$$

ここで $\mathbf{x} = [\mathbf{y}, \boldsymbol{\tau}]^T$, $\mathbf{y} = (\dots, y_{ij}^s, \dots)^T$, $\boldsymbol{\tau} = (\dots, \tau_k^s, \dots)^T$,

$$\mathbf{F}(\mathbf{x}) = \begin{bmatrix} \mathbf{c}(\mathbf{y}, \boldsymbol{\tau}) \\ -\mathbf{q} \end{bmatrix} + \begin{bmatrix} 0 & -\mathbf{A}^T \\ \mathbf{A} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \boldsymbol{\tau} \end{bmatrix}, \quad (4)$$

$\mathbf{A}$ は node-link接続行列, $\mathbf{c} = (\dots, c_{ij}^s, \dots)^T$, $\mathbf{q} = (\dots, q_{kl}^s, \dots)^T$.

3. アルゴリズム

出発時刻別のDUE配分問題を解く方法として、本稿では、2種類のアプローチ：(a)大域収束的Newton法、(b)部分線形化法を検討する。いずれの手法もNCP/VIPに対する merit関数を用いたステップサイズ決定により大域的収束性を強化している。本研究では、そのmerit関数の具体的な関数形として、Fisher(1992) 関数：

$$\Psi(\mathbf{x}) = \sum_{i=1}^{L+N} \phi(x_i, F_i(\mathbf{x}))^2, \phi(x, y) = \sqrt{x^2 + y^2} - (x + y)$$

を、解法(a),(b) を通じて採用する。以下に、起点出発時刻 s に対応する問題を解く場合の解法(a), (b) の具体的な手順をまとめておく。

3.1 NCPに対する大域収束的 Newton法

NCPに対する大域収束的Newton法に基づいたDUE配分の解法は、以下のようにまとめられる：

Step 0: (Initialization)

$n := 0$, (ε, ρ, p) の値を設定 ($\varepsilon > 0$, $\rho > 0$, $p > 2$).

初期解 $\mathbf{x}^{(n)}$ を設定.

Step 1: (Stopping Test)

収束判定条件が満たされれば停止.

Step 2: (Direction Finding)

a) 変数分割集合 α, β を設定:

$$\alpha = \{i \mid |x_i^{(n)}| \leq \varepsilon F_i(x_i^{(n)})\}, \beta = \{i \mid |x_i^{(n)}| > \varepsilon F_i(x_i^{(n)})\}$$

b) 線形連立方程式:

$$[\nabla_{\beta} \mathbf{F}_{\beta}(\mathbf{x}^{(n)})] \mathbf{d}_{\beta}^{(n)} = -\mathbf{F}_{\beta}(\mathbf{x}^{(n)}) \quad (5)$$

を解き, $\mathbf{d}_{\beta}^{(n)}$ を設定. $\mathbf{d}_{\alpha}^{(n)} := -\mathbf{x}_{\alpha}^{(n)}$.

c) もし, 連立方程式(5) が解けないか, または $\mathbf{d}^{(n)}$ が

$$\nabla \Psi(\mathbf{x}^{(n)}) \cdot \mathbf{d}^{(n)} \leq -\rho \|\mathbf{d}^{(n)}\|^p$$

を満たさないなら, $\mathbf{d}^{(n)} := -\nabla \Psi(\mathbf{x}^{(n)})$.

Step 3: (Line Search and Move)

a) 一次元探索問題:

$$\min. \{ \Psi(\mathbf{x}^{(n)} + \alpha \mathbf{d}) \mid \mathbf{x}^{(n)} + \alpha \mathbf{d} \geq 0, \alpha \geq 0 \}$$

を解き, 解を改訂: $\mathbf{x}^{(n+1)} := \mathbf{x}^{(n)} + \alpha \mathbf{d}^{(n)}$,

b) $n := n+1$ とし Step 1へ.

連立方程式(5)は、具体的には、

$$\begin{cases} \mathbf{D} \Delta \mathbf{y} + \mathbf{A}_-^T \Delta \boldsymbol{\tau} = -\mathbf{g}(\mathbf{y}^{(n)}, \boldsymbol{\tau}^{(n)}) \equiv \bar{\mathbf{g}} \\ \mathbf{A} \Delta \mathbf{y} = -\mathbf{h}(\mathbf{y}^{(n)}, \boldsymbol{\tau}^{(n)}) \equiv \bar{\mathbf{h}} \end{cases} \quad (6)$$

である。ここで $\mathbf{d}^{(n)} = (\Delta \mathbf{y}, \Delta \boldsymbol{\tau})$; $(\mathbf{g}, \mathbf{h})$ は、各々、式(4)のベクトル関数 $\mathbf{F}(\mathbf{x})$ の最初の L 個の要素、 N 個の要素に対応するベクトル関数; $\mathbf{A}_- = \mathbf{A}$ の+1要素を0とした行列; $\mathbf{D} = \mathbf{c}(\mathbf{y}, \boldsymbol{\tau})$ の $\mathbf{y}$ に関するJacobian (対角要素 α_{ij} を持つ対角行列) である。方程式(6) は、“渋滞ネットワーク”における均衡解を求める問題³⁾と全く同構造の問題であるので、ネットワーク情報を用いると極めて効率的に解ける(ただし、多起点・1終点問題の場合、 $\mathbf{A} \mathbf{D}^{-1} \mathbf{A}_-^T$ のランク落ちによる問題が生じるが、文献4)に示された方法で解決できる)。

3.2 VIPに対する部分線形化法

VIPに対する部分線形化法に基づいたDUE配分の解法は、以下のようにまとめられる：

Step 0: (Initialization)

$n := 0$.

初期解 $\mathbf{x}^{(n)} = [\mathbf{y}^{s(n)}, \boldsymbol{\tau}^{s(n)}]$ を設定.

Step 1: (Stopping Test)

収束判定条件が満たされれば停止.

Step 2: (Direction Finding)

$$\text{a) 最適化問題: } \min_{\hat{\mathbf{y}}} \sum_{ij} \int_0^{\hat{y}_{ij}} c_{ij}^s(\omega, \tau_i^{s(n)}) d\omega \quad (7)$$

subject to $\mathbf{A} \hat{\mathbf{y}} = \mathbf{q}$, $\hat{\mathbf{y}} \geq 0$ を Frank-Wolfe 法等のアルゴリズムで解き, その最適解 $\hat{\mathbf{y}}$ における制約条件 $\mathbf{A} \hat{\mathbf{y}} = \mathbf{q}$ の Lagrange 乗数を $\hat{\boldsymbol{\tau}}$ とする.

$$\text{b) 方向ベクトル計算: } \mathbf{d} := (\hat{\mathbf{y}} - \mathbf{y}^{s(n)}, \hat{\boldsymbol{\tau}} - \boldsymbol{\tau}^{s(n)})$$

Step 3: (Line Search and Move)

a) 一次元探索問題:

$$\min. \{ \Psi(\mathbf{x}^{(n)} + \alpha \mathbf{d}) \mid \mathbf{x}^{(n)} + \alpha \mathbf{d} \geq 0, \alpha \geq 0 \}$$

を解き, 解を改訂: $\mathbf{x}^{(n+1)} := \mathbf{x}^{(n)} + \alpha \mathbf{d}^{(n)}$,

b) $n := n+1$ とし Step 1へ.

4. 数値実験

4.1 実験条件

(1) ネットワーク

本稿では、図1(a)～(c)に示す3種類のネットワークでの数値実験結果を示す。(a)は予備的考察・解析のためのもので、 $L=6$, $N=5$, 起点 $o=1$, 終点 $d=(4, 5)$ である。(b)は、 $L=48$, $N=16$, 起点 $o=1$, 終点 $d=$

(3,7,10,13, 16)である。(c) は交通配分のテスト問題として有名な Sioux Fall City Network (LeBranc et al. (1975)) で、 $L=76$, $N=24$, 起点 $o=1$, 終点 d は ノード 1 を除く23個のノードである。各ネットワークの最大リンク流出率は、各図のリンク脇に記した数値である(自由走行時間 $\{m_{ij}\}$ については、紙面の制約により記載を省略)。なお、多起点・1終点のOD構造の場合についても同様の実験を行なったが、紙面に制約があるため、その記述は省略する。

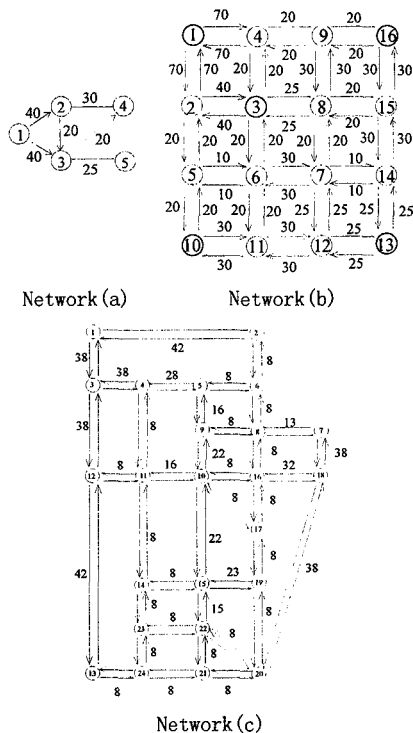


図1 数値実験に用いたネットワーク

(2) OD需要

本稿では、実験結果の本質的な結果の比較を容易にするために、ある一つの出発時刻 s に対応したDUE配分のみを解く実験を行う(出発時刻 s - ds に対応したDUE配分状態のリンクコストあるいはリンク存在台数は初期条件として適宜与える)。従って、実験で与える出発時刻別OD需要 q は、一つの出発時刻 s のみに対応したOD交通流率ベクトルである。ネットワーク (a)では $q = (50,50)$, (b)では $q = (50,100,100,100,100)$, (c)では $q = (1,1,5,2,3,5,8,5,13,5,2,5,3,5,5,4,1,3,3,1,4,3,1)$ と設定した。

(3) その他の諸条件

収束判定は、大域収束的Newton法・部分線形化法

ともに、メリット関数の値を利用した基準で行なった。より具体的には、 $\Psi \leq \varepsilon$ なら繰り返し計算停止とし、 ε の値については $(10^{-1}, 10^{-4}, 10^{-6})$ の3種類の精度で比較実験を行なった。

大域収束的Newton法の方方向ベクトル決定のための補助問題(線形連立方程式)を解く際には、ネットワーク・データ構造を活用したSOR (Successive Over Relaxation)法を採用した。その際の収束判定条件は、 $\max_i \{(\tau_i^{(m+1)} - \tau_i^{(m)}) / \tau_i^{(m)}\} \leq \varepsilon_2$ なら停止とし、 ε_2 の値については ε の値と同様の3種類の精度で比較実験を行なった。ここで、 $\tau_i^{(m)}$ はSOR法の m 回目繰り返しにおける未知変数 τ_i の値である。

部分線形化法の補助問題を解く際には、Frank-Wolfe法を採用した。その際の収束判定条件は、 $\max_{ij} \{(\hat{y}_{ij}^{(m+1)} - \hat{y}_{ij}^{(m)}) / \hat{y}_{ij}^{(m)}\} \leq \varepsilon_3$ なら停止とし、 ε_3 の値については 0.005と設定した。

なお、実験に使用した計算機は Intel Pentium-Pro 200型CPU搭載のIBM互換PC, OS はWindows-NT 4.0, プログラム言語はC++である。

4.2 実験結果

(1) 予備的解析

一般的な構造のネットワーク(b), (c) での数値実験に先立って、解析的に均衡解の求められるネットワーク(a) で正しい均衡解が得られるか否かをチェックした。その結果、大域収束的Newton法・部分線形化法ともに、図2に示す正しい均衡解が得られた。

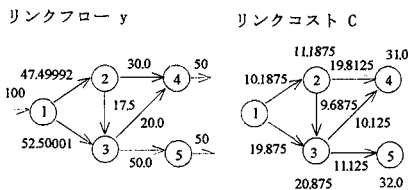


図2 ネットワーク(a) における均衡解

また、この実験過程で、多起点・1終点問題では、(1起点・多終点問題とは異なり)線形連立方程式(5)を解く際にランク落ちが発生することが判明した。この解決法については、文献4) に述べられている。

ネットワーク(b), (c) ではDUE状態で利用されないリンクが存在するため、解析的に均衡解を求めることはできない。これらについては、収束計算結果のフロー y , コスト c , 最早到着時刻 τ が各リンク・ノードでのDUE条件を満たしているか否かを、(収束判定に用いたmerit関数が 0に収束していれば、

当然、DUE条件を満たしているはずであるが）念のため確認した。

(2) 計算精度と計算時間

解の計算精度とそれを確保するために必要な計算時間の関係を、大域収束的Newton法・部分線形化法の両手法について調べた。図3は、横軸を繰返し計算回数、縦軸をmerit関数値（対数スケール）として、ネットワーク(b)での実験結果を示したものである。

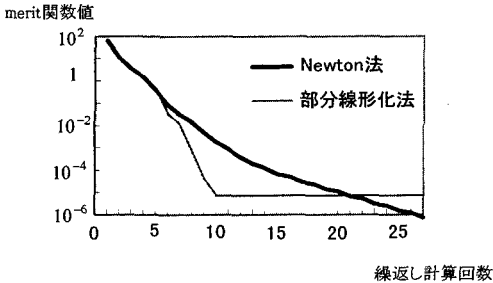


図4 計算精度と繰返し回数の関係

この図から、大域収束的Newton法では、非常に高精度な解が、極めて速やかに求められることが判る。一方、部分線形化法は、ある程度の精度までは比較的効率的であるが、高精度な解は得ることができなかった（収束判定条件をある程度以上厳しくすると、停止しなくなる。これは、補助問題の解の精度の影響を受けている可能性もあると考えられ、今後、追加的な実験・解析を行なう予定である）。ネットワーク(c)でも同様の結果が得られたが、その結果の記述は、紙面の制約により省略する。

(3) 計算時間の内訳

表1は、ネットワーク(b)において収束判定定数を $\varepsilon_1=10^{-3}$ 、 $\varepsilon_2=10^{-3}$ とした場合について、各手法の計算時間の内訳をまとめたものである。

表1 計算時間の内訳と比較

	Newton法	部分線形化法
収束までの繰返し回数	11	8
総計算時間 (sec)	0.32	45.0
(1)補助問題の計算時間	0.003	43.2
(2)1次元探索の計算時間	0.28	0.08

大域収束的Newton法は、極めて短い時間で問題を解いている。ステップサイズ決定のための一次元探索のシェアが大きいのは、ネットワーク・サイズがあまり大きくないためと考えられる。一方、部分線形化法では、補助問題を精度良く解くため（1000回

以上の繰返し）の計算に非常に多くの時間を費やしている。その結果、全体の繰返し計算回数は少ないにもかかわらず、かなり長い計算時間を要している。

(4) 混雑度と計算時間

静的な交通均衡配分では、一般に、ネットワークの混雑度が増加するに従って、均衡解を求めるために必要な収束計算回数が増加することが知られている。DUE配分においても、そのような傾向があることが予想される。その影響の度合いを調べるために、全てのODペアのOD交通流率を基準OD交通流率パターン q の k 倍($k=1,2,3,\dots$)とした場合の繰返し計算回数を計測した。図5は、ネットワーク(c)に大域的Newton法を適用した場合について、横軸を k 、縦軸を収束までの繰返し計算回数として、その関係を示したものである。収束判定定数 $\varepsilon=0.1$ あるいは $\varepsilon=0.0001$ の2つの場合が描かれている。

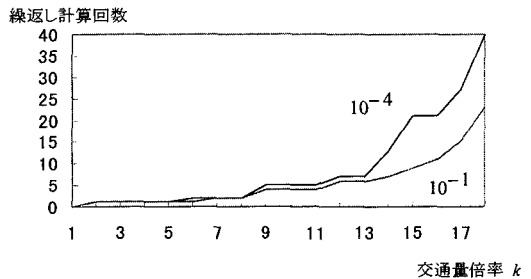


図5 混雑度と繰返し計算回数

この図より、混雑度の増加に伴って、必要な計算の手間は増加することがわかる。しかし、 $k=10$ 程度のかかなり混雑した状態においても、10回未満の繰返し計算で収束しており、提案アルゴリズムが十分実用的な性能を持っていることが判る。

参考文献

- 1) 赤松隆・桑原雅夫，“渋滞ネットワークにおける動的利用者均衡配分”，土木学会論文集488, pp.21-30, 1994.
- 2) 赤松隆，“交通流の予測・誘導・制御と動的なネットワーク配分理論”，土木計画学研究・論文集No.13, pp.23-48, 1996.
- 3) 赤松隆・高松望，“動的な利用者均衡配分の効率的解法”，土木計画学研究・講演集 No.19, pp.549-552, 1996.
- 4) 赤松隆・高松望，“動的均衡配分におけるパラドクス”，土木計画学研究・講演集 No.20, 1997.
- 5) F.Facchinei and J.Soares, “Testing a New Class of Algorithms for Nonlinear Complementarity Problems”, in *Variational Inequalities and Network Equilibrium Problems* (Eds. F.Giannessi and A.Maugeri), Plenum Press, 1995.
- 6) A.Fisher, “A Special Newton-type Optimization Method”, *Optimization* 24, pp.269-284, 1992.
- 7) M.Patricksson, “Partial Linearization Methods in Nonlinear Programming”, *Journal of Optimization Theory and Applications* 78, pp.227-246, 1993.