

$p_{im}^{(n)}$ と新しく出た選択確率 $p_{im}^{(n+1)}$ を比較し、誤差が十分小さければそこで処理を終了する.

$$\|p_{im}^{(n+1)} - p_{im}^{(n)}\| < \varepsilon \quad (3.264)$$

すなわち、閾値を ε と置いた時、上式が成り立てば十分収束したとみなし、処理を終了する.

一方誤差が閾値より大きい場合には、 $p_{im}^{(n+1)}$ を新しい選択確率とし、ステップ 1～ステップ 4 を繰り返す.

参考文献

- [1] 今村晋, 有村俊秀, 片山東: 労働政策の評価: 「構造推定アプローチ」と「実験的アプローチ」, 日本労働研究雑誌, Vol.43, pp.14-21, 2001.
- [2] Rust, J., Optimal Replacement of GMC Bus Engines: An Empirical Model of Harold Zurcher, *Econometrica*, Vol.55, pp.999-1033, 1987.
- [3] Hotz, V. J., and R. A. Miller, Conditional Choice Probabilities and the Estimation of Dynamic Models, *Review of Economic Studies*, Vol.60, pp.497-529, 1993
- [4] V, Aguirregabiria., P, Mira., Swapping the Nested Fixed Point Algorithm: A Class of Estimators for Discrete Markov Decision Models

3.3 ネットワークモデル

3.3.1 利用者均衡配分

(a) 利用者均衡配分の基礎概念

効率的な道路政策を進めるために道路投資の効果をいかに評価するかということが重要になってきている. そのためには道路整備前と後で交通量や、その旅行時間を推計することが必要となってくる. また従前は交通量の再現や予測にその焦点が当てられていたが、近年はマルチモーダル、TDM、ITS 施策といったソフト施策の評価が重要になってきている. そこで利用者均衡配分

による交通量推計が行われるようになった. 利用者均衡配分とは以下に示す *Wardrop* の第一原則を仮定したモデルである.

***Wardrop* の第一原則 (等時間原則)**

それぞれのドライバーは自分にとって最も旅行時間の短い経路を選択する. その結果として、起終点間に存在する経路のうち、利用される経路の旅行時間は皆等しく、利用されない経路の旅行時間よりも小さいか、せいぜい等しいという状態となる.

(b) 利用者均衡配分モデル

ここでは利用者均衡配分のモデルについて扱う. まずモデルを構築するにあたっての前提条件を二つ仮定する.

1. 起終点の旅行時間は、その通貨経路を構成するリンクの旅行時間の和で表わされる.
2. 各リンクのリンク旅行時間はそのリンクの容量と交通量のみによって決まり、他のリンク (対向車線や交差する道路等) の交通状況には影響を受けない.

利用者均衡配分問題は、すべての *OD* 間のフロー数が与えられたときに *Wardrop* の第一原則を満たすリンクフローを求める問題である. まず定式化に際し用いる変数の定義を表 3.4 行う.

表 3.4 変数の定義

a	リンク
r	出発地
s	目的地
k	パス (経路)
x_a	リンク a におけるフロー数
t_a	リンク a における旅行時間
f_k^{rs}	<i>OD</i> ペア rs を結ぶパス k におけるフロー数
c_k^{rs}	<i>OD</i> ペア rs を結ぶパス k における旅行時間
q_{rs}	<i>OD</i> ペア rs 間の総フロー数
$\delta_{a,k}^{rs}$	$\delta_{a,k}^{rs} = \begin{cases} 1 & \text{if リンク } a \text{ が } OD \text{ ペア } rs \text{ を結ぶパス } k \text{ 上にあるとき} \\ 0 & \text{それ以外} \end{cases}$

先の二つの前提条件を仮定したとき **Wardrop** の第一原則が成り立っている状態を数式で表現すると式 (3.65) のようになる.

$$\begin{aligned} (c_k^{rs} - u_{rs}) &\geq 0 \\ c_k^{rs} - u_{rs} &= 0 \end{aligned} \quad (3.265)$$

ここで u_{rs} は **OD** ペア rs 間の最小旅行時間である. 次に利用者均衡条件を満たす、つまり式 (3.65) を満たすような等価な数学的問題に置き換える. これはすべての **OD** 間のフロー数 q_{rs} が与えられたときに、利用者均衡条件を満たすすべてのリンクフロー x_a を求める問題である. このとき式 (3.66) のように定式化できる.

$$\min z(\mathbf{x}) = \sum_a \int_0^{x_a} t_a(\omega) d\omega \quad (3.266)$$

ここで \mathbf{x} はすべてのリンクを表わすベクトルで、 $\mathbf{x} = (\dots, x_a, \dots)$ である. 制約条件は式 (3.70)、式 (3.71) のようになる.

$$\sum_k f_k^{rs} = q_{rs} \quad (3.267)$$

$$f_k^{rs} \geq 0 \quad (3.268)$$

また当然満たされるべき関係として式 (3.72)、式 (3.73) がある.

$$x_a = \sum_r \sum_s \sum_k f_k^{rs} \delta_{a,k}^{rs} \quad (3.269)$$

$$c_k^{rs} = \sum_a t_a \delta_{a,k}^{rs} \quad (3.270)$$

式 (3.72) は **OD** ペア rs 間の経路 k の旅行時間は経路 k 上にあるすべてのリンク旅行時間の総和であることを、式 (3.73) はリンク a の総フロー数はリンク a を通るすべての経路 k 上のフロー数の総和であることを意味している.

(c) 等価性の証明

式 (3.66) に式 (3.70) を用いたラグランジアンで表わすと式 (3.73) のようになる.

$$L(\mathbf{f}, \mathbf{u}) = z[\mathbf{x}(\mathbf{f})] + \sum_{rs} u_{rs} \left(q_{rs} - \sum_k f_k^{rs} \right) \quad (3.271)$$

ただし \mathbf{f} はすべての経路上のフロー数を表わすベクトルで $\mathbf{f} = (\dots, f_k^{rs}, \dots)$ である. また u_{rs} は各 OD ペアのフロー数制約に与えられる双対係数で, $\mathbf{u} = (\dots, u_{rs}, \dots)$ である. このときフロー数の非負制約条件式 (3.71) が満たされていないといけない. ここでラグランジアン (3.73) の停留点では式 (3.272) が満たされていないといけない.

$$f_k^{rs} \frac{\partial L(\mathbf{f}, \mathbf{u})}{\partial f_k^{rs}} = 0 \quad \text{and} \quad \frac{\partial L(\mathbf{f}, \mathbf{u})}{\partial f_k^{rs}} \geq 0 \quad (3.272)$$

また双対係数については式 (3.273) が満たされていないといけない.

$$\frac{\partial L(\mathbf{f}, \mathbf{u})}{\partial u_{rs}} = 0 \quad (3.273)$$

この式 (3.273) は式 (3.70) と同じで, フロー数制約が満たされなければならないことを意味しているパスフロー f_k^{rs} に関するラグランジアンの変微分は式 (3.274) となる.

$$\frac{\partial}{\partial f_l^{mn}} L(\mathbf{f}, \mathbf{u}) = \frac{\partial}{\partial f_l^{mn}} z[\mathbf{x}(\mathbf{f})] + \frac{\partial}{\partial f_l^{mn}} \sum_{rs} u_{rs} \left(q_{rs} - \sum_k f_k^{rs} \right) \quad (3.274)$$

ここで式 (3.274) 第一項は式 (3.275) のようになる.

$$\frac{\partial}{\partial f_l^{mn}} z[\mathbf{x}(\mathbf{f})] = \sum \frac{\partial z(\mathbf{x})}{\partial x_b} \frac{\partial x_b}{\partial f_l^{mn}} \quad (3.275)$$

式 (3.275) の右辺を分解して考える. まず右辺左側については式 (3.276) のようになる.

$$\frac{\partial z(\mathbf{x})}{\partial x_b} = \frac{\partial}{\partial x_b} \sum_a \int_0^{x_a} t_a(\omega) d\omega = t_b \quad (3.276)$$

また式 (refue5) に注意すると式 (3.275) 右辺の右側は式 (3.277) のようになる.

$$\frac{\partial x_b}{\partial f_l^{mn}} = \delta_{b,l}^{mn} \quad (3.277)$$

式 (3.276)、式 (3.277) を式 (3.275) に代入し、式 (3.73) を利用すると式 (3.278) が導かれる.

$$\frac{\partial z[x(f)]}{\partial f_l^{mn}} \sum_b t_b \delta_{b,l}^{mn} = c_l^{mn} \quad (3.278)$$

これはある *OD* ペア *mn* 間のパス *l* の旅行時間を表わしている. 次に式 (3.274) の第二項について. まず式 (3.279) が成立する.

$$\frac{\partial f_k^{rs}}{\partial f_l^{mn}} = \begin{cases} 1 & \text{if } r = m, s = n \text{ and } k = l \\ 0 & \text{それ以外} \end{cases} \quad (3.279)$$

q_{rs} 、 u_{rs} は f_l^{mn} について定数なので、式 (3.274) の第二項は式 (3.280) のようになる.

$$\frac{\partial}{\partial f_l^{mn}} \sum_{rs} u_{rs} \left(q_{rs} - \sum_k f_k^{rs} \right) = -u_{mn} \quad (3.280)$$

以上式 (3.278) と式 (3.280) を式 (3.274) に代入すると式 (3.281) が導かれる.

$$\frac{\partial}{\partial f_l^{mn}} L(f, u) = c_l^{mn} - u_{mn} \quad (3.281)$$

この式 (3.281) を式 (3.272) に代入し、また式 (3.273) と非負制約条件式 (3.71) をまとめると式 (3.282) のようになる.

$$\begin{aligned} f_k^{rs} (c_k^{rs} - u_{rs}) &= 0 \\ c_k^{rs} - u_{rs} &\geq 0 \\ \sum_k f_k^{rs} &= q_{rs} \\ f_k^{rs} &\geq 0 \end{aligned} \quad (3.282)$$

式 (3.282) は利用者均衡条件に他ならないので、式 (3.66) を制約条件式 (3.70)、(3.71) のもとで解くことで利用者均衡配分問題を解くことができることが示された.

3.3.2 システム最適化配分

(a) システム最適化配分の基礎概念

前節までは *Wardrop* の第一原則が満たされるような交通量配分を考え定式化を行った. 一方で以下に示す *Wardrop* の第二原則を満たすような交通量配分をシステム最適化配分という.

Wardrop の第二原則

道路ネットワーク上の総旅行時間が最小となる.

Wardrop の第一原則と第二原則は同時に満たされる場合もあるが、一般には異なった配分結果をもたらす. システム最適配分は道路網全体としての効率化を目指した配分であり、道路事業者が車の流れを管理してドライバー全体の総走行時間を最小化しようとする場合に相当する. したがってシステム最適化配分を行ったとき各ドライバーは最小旅行時間を移動しているとは限らない. 利用者均衡配分のときと同様に式 (3.283)~式 (3.285) の等価な数学的問題に置き換えることでシステム最適化配分問題も解くことができる.

$$\min \tilde{z}(x) = \sum_a x_a t_a(x_a) \quad (3.283)$$

$$\sum_k f_k^{rs} = q_{rs} \quad (3.284)$$

$$f_k^{rs} \geq 0 \quad (3.285)$$

(b) ブラエスのパラドクス

システム最適化と利用者均衡の違いを理解していないと次のような問題が起こる. システム最適化の観点で道路ネットワーク上の総旅行時間を小さくするために新しい道路 (バイパス) をつくった結果、以前よりも総旅行時間が大きくなってしまふことがある. これをブラエスのパラドクスという. 例えば図 3.65 の上図のような道路ネットワークがあり、出発地から到着地まで 6 台の自動車移動とする. つまり $q = 6$ である. このとき各パスの旅行時間は式 (3.286) だとする.

$$\begin{aligned}
 t_1[x_1] &= 50 + x_1 \\
 t_2[x_2] &= 50 + x_2 \\
 t_3[x_3] &= 10x_3 \\
 t_4[x_4] &= 10x_4
 \end{aligned}
 \tag{3.286}$$

このとき利用者均衡条件を満たすようなフローは図 3.65 の下図のようになる。このとき各パスの旅行時間は $O \rightarrow$ リンク 1 \rightarrow リンク 4 $\rightarrow D$ 、 $O \rightarrow$ リンク 3 \rightarrow リンク 2 $\rightarrow D$ のどちらも $50 + 3 + 10 \times 3 = 83$ であり、道路ネットワーク上の総旅行時間は 498 となる。今道路ネットワーク上の総旅行時間改善のために図 3.66 の上図のようにリンク 5 をひとつ新たに設ける。リンク 5 の旅行時間は式 (3.287) である。

$$t_5 = 10 + x_5 \tag{3.287}$$

このとき利用者均衡条件を満たすようなフローは図 3.66 の下図ようになる。このとき各パスの旅行時間は $O \rightarrow$ リンク 1 \rightarrow リンク 4 $\rightarrow D$ 、 $O \rightarrow$ リンク 3 \rightarrow リンク 2 $\rightarrow D$ 、 $O \rightarrow$ リンク 3 \rightarrow リンク 5 \rightarrow リンク 4 $\rightarrow D$ のどれも 92 であり、道路ネットワーク上の総旅行時間は 552 であり、リンク 5 を設けた場合のほうが総旅行時間が大きくなっていることがわかる。この結果は利用者均衡配分の枠組みではドライバーは自分の旅行時間のみを最小とするように経路を選択し、自分の経路選択の結果ネットワーク全体の旅行時間に与える影響を考慮していないことが原因で発生する。

3.3.3 確率的利用者均衡配分

(a) 確率的利用者均衡配分の基礎概念

前節まで扱っていた利用者均衡配分では各道路利用者が各経路の旅行時間を完全に知っている状態で、最短経路を選ぶという強い仮定を置いていた。したがって最短経路以外には交通路湯が全く配分されなかった。この条件を緩和し、道路利用者が知っている各経路の旅行時間は不正確であり、旅行時間以外の要因も考慮して経路を選択しているとしたのが確率的利用者均衡配分である。この確率的均衡配分のほうがより現実的であり、すなわち道路利用者が

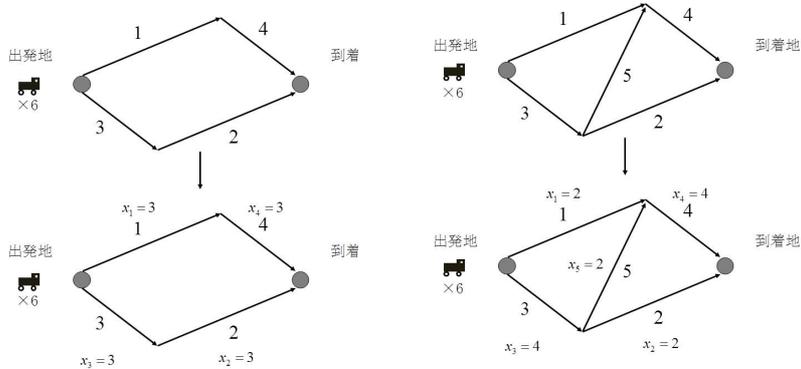


図 3.65 ブラエスのパラドクス: 初期状態 図 3.66 ブラエスのパラドクス: 新リンク追加後

認識している各経路の旅行時間は確率的に変動する誤差を含んでいるとしたモデルである. 均衡状態では **Wardrop** の考え方に対応させると以下のような状態が成り立つ.

それぞれのドライバーは自分にとって最も旅行時間の短いと思われる経路を選択する. その結果として、起終点間に存在する経路のうち、利用される経路の旅行時間は皆等しいと信じられ、利用されない経路の旅行時間よりも小さいか、せいぜい等しいと信じている状態となる.

(b) 確率的利用者均衡配分モデル

確率的利用者均衡配分において道路利用者が認識している旅行時間が誤差を含んでいるという考えかたはランダム効用理論に基づく、離散選択モデル(非集計モデル)と同等である. つまり、道路利用者が経路を選択する際に従うルールに非集計モデルを適用したのが確率的利用者均衡配分だといえる. **OD** ペア rs 間の経路 k の効用関数の確定項 V_k^{rs} を、旅行時間のみ関数とする式 (3.288) のようになる.

$$V_k^{rs} = -c_k^{rs} \tag{3.288}$$

効用関数の誤差項が正規分布に従うとした場合プロビット型など様々なモデルが考えられるが、ここでは誤差項をガンベル分布を仮定したロジットモ

デルで話を進める. このとき OD ペア rs 間の経路選択枝集合 K_{rs} から経路 k が選択される確率 P_k^{rs} は式 (3.289) のようになる.

$$P_k^{rs} = \frac{\exp(-\theta \cdot c_k^{rs})}{\sum_{k \in K_{rs}} \exp(-\theta \cdot c_k^{rs})} \quad (3.289)$$

ここで θ は分散パラメータを表わす. この θ の値によって認知旅行時間の誤差が変化する. つまり、 θ が大きいほど認知旅行時間のずれは小さくなり、確定的な利用者均衡配分に近づき、 θ が小さいほど認知旅行時間のずれは大きくなり、実際の旅行時間の大小にかかわらず、どの経路も同じように選択されるようになる. 今選択確率が決定したとすると OD ペア rs 間の経路 k のフロー数 f_k^{rs} は式 (3.291) で表わされる.

$$f_k^{rs} = q_{rs} \cdot P_k^{rs} = q_{rs} \cdot \frac{\exp(-\theta \cdot c_k^{rs})}{\sum_{k \in K_{rs}} \exp(-\theta \cdot c_k^{rs})} \quad (3.290)$$

確率的利用者均衡配分を満たす解を求めるための数理的最適化問題は式 (3.291) であり、制約式は式 (3.292) のようになる.

$$\min z(\mathbf{x}) = \sum_a \int_0^{x_a} t_a(\omega) d\omega - \sum_{rs \in \Omega} \frac{1}{\theta} \cdot q_{rs} \cdot \sum_{k \in K_{rs}} \left(-\frac{f_k^{rs}}{q_{rs}} \cdot \ln \frac{f_k^{rs}}{q_{rs}} \right) \quad (3.291)$$

$$\begin{aligned} \sum_{k \in K_{rs}} f_k^{rs} &= q_{rs} \\ f_k^{rs} &\geq 0 \end{aligned} \quad (3.292)$$

ここで Ω はすべての OD ペアの集合を表わす. また確定的利用者均衡配分のと看同様に当然満たされるべき条件として式 (3.293) がある.

$$x_a = \sum_{rs \in \Omega} \sum_{k \in K_{rs}} f_k^{rs} \delta_{a,k}^{rs} \quad (3.293)$$

3.3.4 利用者均衡配分問題解法アルゴリズム

前節までに様々な配分問題の定式化を行った. ここではその問題を解くアルゴリズムについて説明を行う. まず利用者均衡配分の解法の一つ線形計画法である *Frank – Wolfe* 法を説明する.

(a) *Frank – Wolfe* 法

Frank – Wolfe 法は計算手順が比較的簡単でプログラムを作成しやすいという長所がある一方で、均衡解に近づくに従い収束が緩慢になる. 具体的なアルゴリズムを以下に示す. また u_{rs}^n は n 回目の試行を行ったときの OD 間最小旅行時間を表わす.

ステップ 0: 初期実行可能解の設定

すべてのリンクの交通量 $x_a^0 = 0$ としリンク旅行時間 t_a^0 を計算する. この旅行時間に基づき *all – or – nothing* 配分 (交通量 0 のときのリンク旅行時間を用いた最適経路配分) により初期実行可能解 (リンク交通量の初期値) x_a^1 を設定する. このとき繰り返し計算の試行回数 $n = 1$ とする.

ステップ 1: リンク旅行時間の更新

リンク交通量 x_a^n に対するリンク旅行時間 t_a^n を計算する. このとき OD ペアごとの最小旅行時間 u_{rs}^n を計算する.

ステップ 2: 降下方向ベクトルの探索

リンク旅行時間 t_a^n の状態ですべての OD 間の交通量を最短経路に配分し、そのときの交通量を y_a^n とする. $d_a^n = y_a^n - x_a^n$ によって降下方向ベクトル d_a^n を計算する.

ステップ 3: 降下ステップサイズの探索

式 (3.294) のような x_a^{n+1} に対して式 (3.66) で表わされる目的関数の値が最も小さくなるような α^n を探索する. すなわち式 (3.295) を解くことで α^n を探索すればよい. 得られた α^n によって x_a^{n+1} を計算す

る. 新たなリンク交通量から旅行時間 t_a^{n+1} を計算し、 OD ペアごとの最小旅行時間 u_{rs}^{n+1} を計算する.

$$x_a^{n+1} = x_a^n + \alpha^n d^n = x_a^n + \alpha^n (y_a^n - x_a^n) \quad (3.294)$$

$$\min_{0 \leq \alpha \leq 1} \sum_a \int_0^{x_a^n + \alpha^n (y_a^n - x_a^n)} t_a(\omega) d\omega \quad (3.295)$$

ステップ 4: 収束判定

式 (3.296) の収束条件 (この収束判定基準は唯一のものではない) が満たされていれば、 $n = n + 1$ とステップ 1 へ戻る. 収束条件が満たされていれば、計算を終了しリンク交通量 x_a^{n+1} を解として出力する. ただし式 (3.296) 中の κ はあらかじめ決定された定数とする.

$$\sum_{rs} \frac{|u_{rs}^{n+1} - u_{rs}^n|}{u_{rs}^{n+1}} \leq \kappa \quad (3.296)$$

3.3.5 確率的利用者均衡配分解法アルゴリズム

確率的利用者均衡配分の解法アルゴリズムには確定的な利用者均衡配分で用いた **Frank – Wolfe** 法は実用上は計算が複雑になるため用いられることはない. ここでは確率的利用者均衡配分の解法アルゴリズムとして逐次平均法 (**MSA; Method of Successive Averages**)、部分線形化法、**Simplicial Decomposition** 法を紹介する.

(a) 逐次平均法

逐次平均法は計算手順がシンプルであり、記憶容量も少ない方法であるが、計算の収束に必要な繰り返し計算回数が多くなる. この逐次平均法ではあらかじめ降下ステップサイズを決定しておくことで目的関数を計算することを回避している. また最短経路にすべての配分を行った確定的利用者均衡配分のときと異なり、リンク旅行時間を固定したときの確率的配分 (**Dial** 配分法など) を用いる. 以下に具体的なアルゴリズムを示す.

ステップ 0: 初期実行可能解の設定

すべてのリンクの交通量 $x_a^0 = 0$ としリンク旅行時間 t_a^0 を計算する. この旅行時間に基づき確率的配分により初期実行可能解 (リンク交通量の初期値) x_a^1 を設定する. このとき繰り返し計算の試行回数 $n = 1$ とする.

ステップ 1: リンク旅行時間の更新

リンク交通量 x_a^n に対するリンク旅行時間 t_a^n を計算する.

ステップ 2: 降下方向ベクトルの探索

リンク旅行時間 t_a^n の状態ですべての *OD* 間の交通量をネットワーク上に確率的配分し, そのときの交通量を y_a^n とする. $d_a^n = y_a^n - x_a^n$ によって降下方向ベクトル d_a^n を計算する.

ステップ 3: 降下ステップサイズの探索

降下ステップサイズを α^n とし式 (3.297) により新しいリンク交通量を計算する.

$$\alpha^n = \frac{1}{n+1} \quad (3.297)$$

式 (3.294) により x_a^{n+1} を計算する. 新たなリンク交通量から旅行時間 t_a^{n+1} を計算する.

ステップ 4: 収束判定

式 (3.298) の収束条件 (この収束判定基準は唯一のものではない) が満たされていれば, $n = n + 1$ としステップ 1 へ戻る. 収束条件が満たされていれば, 計算を終了しリンク交通量 x_a^{n+1} を解として出力する. ただし式 (3.298) 中の κ はあらかじめ決定された定数とする.

$$\frac{\sqrt{\sum_a (\bar{x}_a^{n+1} - \bar{x}_a^n)^2}}{\sum_a \bar{x}_a^n} < \kappa \quad (3.298)$$

ただし \bar{x}_a^n は式 (3.299) で表わされるとする.

$$\bar{x}_a^n = \frac{x_a^n + x_a^{n-1} + x_a^{n-2} + \dots + x_a^{n-m+1}}{m} \quad (3.299)$$

ステップ4の収束判定で、逐次平均法では降下ステップサイズがあらかじめ決定された定数列で n が大きくなると小さくなっていくので n 回目のリンク交通量 x_a^n と $n+1$ 回目のリンク交通量 x_a^{n+1} の差は当然小さくなる。そこで、式(3.299)のように直近 m 回分の移動平均をとることで収束判定を行っている。

(b) 部分線形化法

部分線形化法ではエントロピー分解定理を用いることで起点別リンク交通量 (=各リンク交通量がどの出発地からどれだけ発生したのかを表わす) から目的関数の計算を行いステップサイズを計算しており、ステップサイズをあらかじめ決定している逐次平均法よりも効率的に解を収束させることができる。まずエントロピー分解定理について説明する。式(3.291)の第2項(エントロピー項)は式(3.300)のように分解することができる。

$$\begin{aligned} & \sum_{rs \in \Omega} q_{rs} \cdot \sum_{k \in K_{rs}} \left(-\frac{f_k^{rs}}{q_{rs}} \cdot \ln \frac{f_k^{rs}}{q_{rs}} \right) \\ &= \sum_{r \in R} \left\{ -\sum_a x_a^r \cdot \ln x_a^r + \sum_{j \in N} \left(\sum_{a \in I_j} x_a^r \right) \cdot \ln \left(\sum_{a \in I_j} x_a^r \right) \right\} \end{aligned} \quad (3.300)$$

ただし x_a^r は起点別リンク交通量(リンク a の交通量のうちノード r を出発地とするもの)、 R は出発地ノードの集合、 N はノードの集合、 I_j はノード j に流入するリンクの集合を表わす。この式(3.300)を用いることでパス交通量を用いることなく目的関数の計算が可能で降下ステップサイズ探索や収束判定に目的関数の値を利用することができる。以下に具体的なアルゴリズムを示す。

ステップ0: 初期実行可能解の設定

すべてのリンクの交通量 $x_a^0 = 0$ としリンク旅行時間 t_a^0 を計算する。この旅行時間に基づき全 OD 交通量を出発地別に確率的配分し、得られた初期実行可能解(起点別リンク交通量の初期値) $x_a^{r,1}$ を設定する。このとき繰り返し計算の試行回数 $n = 1$ とする。

ステップ1: リンク旅行時間の更新

リンク交通量 $x_a^n = \sum_{r \in R} x_a^{r,n}$ に対するリンク旅行時間 t_a^n を計算する。

ステップ 2: 降下方向ベクトルの探索

リンク旅行時間 t_a^n の状態ですべての OD 間の交通量を出発地別にネットワーク上に確率的配分し、そのときの交通量を $y_a^{r,n}$ とする。 $d_a^{r,n} = y_a^{r,n} - x_a^{r,n}$ によって降下方向ベクトル $d_a^{r,n}$ を計算する。

ステップ 3: 降下ステップサイズの探索

降下ステップサイズを α^n とし、式 (3.302) により新しい起点別交通量を計算する。

$$x_a^{r,n+1} = x_a^{r,n} + \alpha^n d_a^{r,n} \quad (3.301)$$

得られた $x_a^{r,n+1}$ を目的関数に代入した値が最も小さくなるような α を探索する。すなわち式 (3.303) を解くことで α を探索すればよい。

$$\min_{0 \leq \alpha \leq 1} \sum_a \int_0^{x_a^{n+1}} t_a(\omega) d\omega - \sum_{r \in R} \frac{1}{\theta} \left[\begin{array}{l} - \sum_a (x_a^{r,n} + \alpha^n d_a^{r,n}) \cdot \ln (x_a^{r,n} + \alpha^n d_a^{r,n}) \\ + \sum_{j \in N} \left\{ \sum_{a \in I_j} (x_a^{r,n} + \alpha^n d_a^{r,n}) \right\} \\ \cdot \ln \left\{ \sum_{a \in I_j} (x_a^{r,n} + \alpha^n d_a^{r,n}) \right\} \end{array} \right] \quad (3.302)$$

ステップ 4: 収束判定

式 (3.296) の収束条件 (この収束判定基準は唯一のものではない) が満たされていれば、 $n = n + 1$ としステップ 1 へ戻る。収束条件が満たされていれば、計算を終了しリンク交通量 x_a^{n+1} を解として出力する。

(c) *Simplicial Decomposition* 法

Simplicial Decomposition 法は逐次平均法や部分線形化法とは異なり、経路を明示的に列挙していることが特徴である。経路選択肢どのように生成するかによって得られる解が異なる。以下に具体的なアルゴリズムを示す。

ステップ 0: 初期実行可能解の設定

OD ペア別経路集合を \hat{K}_{rs} とし、初期状態を空集合とする。すべて

のリンクの交通量 $x_a^0 = 0$ としリンク旅行時間 t_a^0 を計算する. この旅行時間に基づき最短経路を探索し, \hat{K}_{rs} に加える. また最短経路へ, **all-or-nothing** によって $f_k^{rs,1}$, リンク交通量 x_a^1 を計算する. このとき繰り返し計算の試行回数 $n = 1$ とする.

ステップ 1: 端点の生成リンク交通量 x_a^n に対するリンク旅行時間 t_a^n を計算する. リンク旅行時間の下で最短経路を探索し, 得られた経路を \hat{K}_{rs} に加える.

ステップ 2: 限定親問題を解く

経路交通量ベクトル $f^{n+1} = (f_k^{rs,n+1})$ について式 (3.303) を解く.

$$\begin{aligned} \min_{f^{n+1}} \sum_{a \in A} \int_0^{x_a^{n+1}} t_a(w) dw \\ - \sum_{rs \in \Omega} \frac{1}{\theta} \cdot q_{rs} \cdot \sum_{k \in \hat{K}_{rs}} \left(-\frac{f_k^{rs,n+1}}{q_{rs}} \cdot \ln \frac{f_k^{rs,n+1}}{q_{rs}} \right) \end{aligned} \quad (3.303)$$

制約条件として式 (3.304) が与えられる.

$$\begin{aligned} f_k^{rs,n+1} &\geq 0 \\ \sum_{k \in \hat{K}_{rs}} f_k^{rs,n+1} &= q_{rs} \\ x_a^{n+1} &= \sum_{rs \in \Omega} \sum_{k \in \hat{K}_{rs}} f_k^{rs} \delta_{a,k} \end{aligned} \quad (3.304)$$

この問題の解法として部分線形化法が利用できる.

ステップ 2-0: 初期設定

限定親問題の解の初期値を $g_k^{rs,1} = f_k^{rs,1}$, $y_a^1 = x_a^1$ とする. 限定親問題を解くループの繰り返し計算回数のカウンタを $m = 1$ とする.

ステップ 2-1: 降下方向ベクトルの探索

式 (3.305) によりリンクコスト t_a^m を更新する.

$$t_a^m = t_a(y_a^m) \quad (3.305)$$

リンクコストパターン t_a^m の下での経路選択確率に従って全 OD 交通量を各経路に配分し、得られた経路交通量 $h_k^{rs,k}$ 、リンク交通量 z_a^m をそれぞれ式 (3.306)、(3.307) によって計算する。

$$h_k^{rs,m} = q_{rs} \cdot \frac{\exp\left(-\theta \cdot \sum_{a \in A} \delta_{a,k}^{rs} \cdot t_a^m\right)}{\sum_{k \in \hat{K}_{rs}} \exp\left(-\theta \cdot \sum_{a \in A} \delta_{a,k}^{rs} \cdot t_a^m\right)} \quad (3.306)$$

$$z_a^m = \sum_{rs \in \Omega} \sum_{k \in \hat{K}_{rs}} h_k^{rs,m} \delta_{a,k}^{rs} \quad (3.307)$$

また降下方向ベクトル $e_k^{rs,m}$ と d_a^m を式 (3.308)、(3.309) により計算する。

$$e_k^{rs,m} = h_k^{rs,m} - g_k^{rs,m} \quad (3.308)$$

$$d_a^{rs,m} = z_a^m - y_a^m \quad (3.309)$$

ステップ 2-2: 降下ステップサイズの算出と解の更新

降下ステップサイズ η^m を用いて交通量を式 (3.310)、(3.311) により更新する。

$$g_k^{rs,m+1} = g_k^{rs,m} + \eta^m \cdot e_k^{rs,m} \quad (3.310)$$

$$y_a^{m+1} = y_a^m + \eta^m \cdot d_a^m \quad (3.311)$$

この $g_k^{rs,m+1}$ 、 y_a^{m+1} を目的関数に代入した値が最も小さくなるような η を探索する。すなわち式 (3.312) を解くことで探索を行う。

$$\min_{0 \leq \eta^m \leq 1} \sum_{a \in A} \int_0^{y_a^m + \eta^m \cdot d_a^m} t_a(w) dw - \sum_{rs \in \Omega} \frac{1}{\theta} \cdot q_{rs} \cdot \sum_{k \in \hat{K}_{rs}} \left(-\frac{g_k^{rs,m} + \eta^m \cdot e_k^{rs,m}}{g_k^{rs,m} + \eta^m \cdot e_k^{rs,m}} \cdot \ln \frac{g_k^{rs,m} + \eta^m \cdot e_k^{rs,m}}{q_{rs}} \right) \quad (3.312)$$

ステップ 2-3: 打ち切り判定

限定親問題の解が収束すれば、すなわち $g^m \cong h^m$ ならば、 $f_k^{rs,n+1} = g_k^{rs,m+1}$ 、 $x_a^{n+1} = y_a^{m+1}$ としてステップ 3 へ進む。そうでなければ $m = m + 1$ としてステップ 2-1 に戻る。

ステップ 3: 収束判定

収束条件が満たされていないならば $n = n + 1$ としてステップ 1 に戻る。収束条件式が満たされていれば、計算を終了しリンク交通量 x_a^{n+1} と経路交通量 $f_k^{rs,n+1}$ として解を出力する。

(d) 遺伝的アルゴリズム (Genetic Algorithm)

遺伝的アルゴリズム (以下 GA とする) は、データの集合を遺伝子 DNA の進化プロセスになぞらえて、最適解を計算する最適化アルゴリズムの 1 つである。ここまで述べた他の最適化アルゴリズムは目的関数の微分可能性や変数の探索空間の構造 (実数値) を前提においていた。また、ネットワークの規模が大きくなるほど計算に膨大な時間がかかり、現実的な時間内に計算が終了しない場合がある。GA は整数問題など広範な問題に対応でき、また必ずしも厳密解が得られるわけではないものの、ある程度それに近い解を比較的短時間で得ることができるヒューリスティックなアルゴリズムである。

以下の図 3.67 に遺伝的アルゴリズムによる計算プロセスを示す。個体群 (この場合は解の候補の集合) が適応度 (目的関数の値) が高くなるように自ら「進化」していくことが GA の大きな特徴である。

各プロセスにおける操作は以下の 3 つのステップに分けられる。

ステップ 0 : 初期集団の生成

初めに解の候補の初期値を与える。この時、解の表現の仕方を工夫する必要がある。例えば合計 10 台の車が 3 つの経路に流れるフロー数を考える時、遺伝子型を **(2, 3, 5)** のように直接経路の台数で表現することもできるが、図 3.68 のように車両 10 台と仕切り 2 枚を並べたときの仕切りの位置と考えると、**(3, 7)** のように表すこともできる。実は、後者の方が STEP2 の遺伝的操作を行う際に適した形である (理由は後述する)。このように直接的な表現型から、遺伝子型と呼ばれる遺伝的操

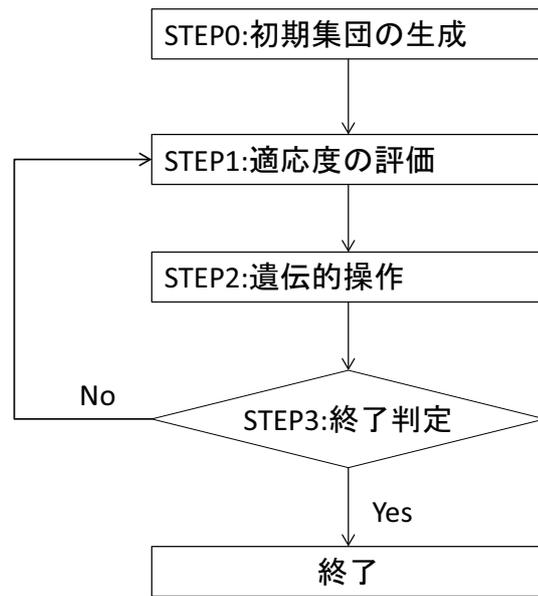


図 3.67 GA の計算プロセス

作に適した形に変換することを**エンコード**と呼ぶ。逆に遺伝子型から表現型に戻す操作を**デコード**と呼ぶ。

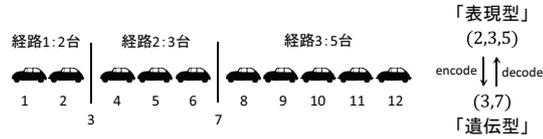


図 3.68 エンコード/デコード

また、個体の個数の設定も重要である。個体数が少ないと収束までに時間がかかり、逆に個体数が多いと1世代あたりの計算時間が長くなる。一般的に、遺伝子の長さが長いほど個体数を多くとる必要がある。

ステップ1: 適応度の評価

現在 (t 期とする) の個体群に含まれる全ての個体の適応度を評価する。適応度は最大化問題の場合は目的関数をそのまま用いる。一方で最小化問題の場合は目的関数を -1 倍する、逆数をとるなどの工夫が必要となる。

STEP2: 遺伝的操作

遺伝的操作は、選択・交叉・突然変異の3つの操作からなる。

選択

選択では次の交叉で親となる遺伝子を選出する。適応度の高いものを優先的に選び、優れた個体ほど子孫を残しやすくする。選択の方法としては、各個体の選択確率を適応度に応じた比例配分によって求めるルーレット選択、予め順位ごとに選ばれる確率を設定するランキング選択、ランダムに一定数個体を取り出しその中で適応度が最大な個体を選ぶ操作を繰り返すトーナメント選択などが存在する。また、上位の個体をそのまま次の世代に残し(エリート選択という)、残りの個体から選択を行うことで、より適応度の高い集団にする方法も存在する。

交叉

選択で選んだ2つの個体(親)のデータを交叉し、新しい2つの個体をつくる。交叉の方法には、1箇所を選んでその前後のデータを入れ替える1点交叉、2箇所を選んでその間のデータを入れ替える2点交叉、入れ替える場所をランダムに選んだ一様交叉などが用いられる。一般的には一様交叉が用いられることが多い。図3.69に一様交叉の例を示す。一様交叉では、はじめに1と0からなるランダムなビット列を作成し、その値が1の地点を入れ替えることで新しい個体を作る。

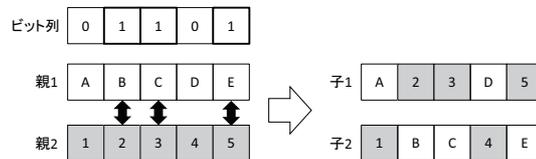


図 3.69 一様交叉

ここで遺伝子の表現方法の議論が重要になる。先程の例でいえば、各経路のフロー数を遺伝子型とした場合、交叉によって生じる個体の要素の合計が総車両数を超過してしまう、あるいは総車両数に達しないと、制約条件を満たさない個体ができる恐れがある。このように制約条件を満たさない個体の遺伝子を**致死遺伝子**と呼び、なるべく交叉によって致死遺伝子が発生しないようなエンコードが求められる。

突然変異

交叉で生成した新しい個体は、親の個体の性質を引き継いでいるため、初期収束のように個体の特性が偏る恐れがある。突然変異は交叉によって生じた個体を一定確率で変化させることによって個体の多様性を保つ操作である。一般的な突然変異では、要素の一部を乱数で置き換えることによって突然変異を行う。他にもランダムに選んだ2点の遺伝子を入れ替える転座や、2点間の遺伝子の順序を逆にする逆位といった手法がとられることもある。

ステップ3:終了判定

遺伝子操作によって第 t 期の遺伝子群から新しい第 $t + 1$ 期の遺伝子群ができる。ここで終了条件の判定を行う。終了条件としては、一定回数だけ計算する、閾値を超える適応度の個体が現れる、世代の平均値が閾値を超える、前の世代からの平均値の増加率が閾値以下になる状態が一定期間続く、など様々である。計算環境や問題の性質に応じて使い分ける必要がある。

以上の操作を行うことによって、GA は比較的短時間でおよそ最適解を導くことができる。しかし、実際にアルゴリズムを実行する際、以下のような問題が起こりうることに注意が必要である。

初期収束

比較的初期の世代の集団の中で、適応度が他の個体に比べ著しく高い個体が存在する時、交叉によってその個体の遺伝子が集団に急速に広がり、局所最適解に収束してしまう恐れがある。一般的には初期収束を避けるために突然変異の頻度を調節したり、個体の集中度合いによって適応度に重みづけするなどの対策がとられる。

ヒッチハイキング

最適解にきわめて近い2つの個体を交叉させても、最適解を得られないことがある。この場合、遺伝子長が長いほど最適解が得られる確率は極端に低くなっていく。一般的に、この問題は一様交叉によって簡単に回避することが可能である。

GA 困難

GA は積み木仮説 (部分解の組み合わせによって最適解が生成される) に基づいて解を探索している。しかし、その仮説が成立しないような問題も存在する。このような問題に対して GA を適用した場合、局所最適解に陥りやすく最適解を発見することが難しくなる。この問題を回避するためには、これまで述べたような単純な形の GA の枠を超えた拡張が必要となる。

(e) 計算例

ここでは利用者均衡配分を図 3.70 のネットワークを使って実際に計算を行う。ただし各パスごとの旅行時間は式 (3.313) とする。

$$\begin{aligned} t_1 &= 10 \left[1 + 0.15 \left(\frac{x_1}{2} \right)^4 \right] \\ t_2 &= 20 \left[1 + 0.15 \left(\frac{x_2}{4} \right)^4 \right] \\ t_3 &= 25 \left[1 + 0.15 \left(\frac{x_3}{3} \right)^4 \right] \end{aligned} \quad (3.313)$$

Frank - Wolfe 法の計算例を図 3.71~図 3.73 に示す。最終的に計算を繰り返していくと $x_1 = 3.58$ 、 $x_2 = 4.62$ 、 $x_3 = 1.81$ を得る。ただし収束判定のための定数 $\kappa = 0.001$ とした。

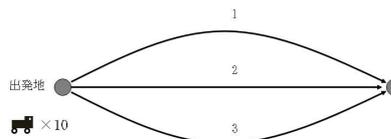


図 3.70 利用ネットワーク

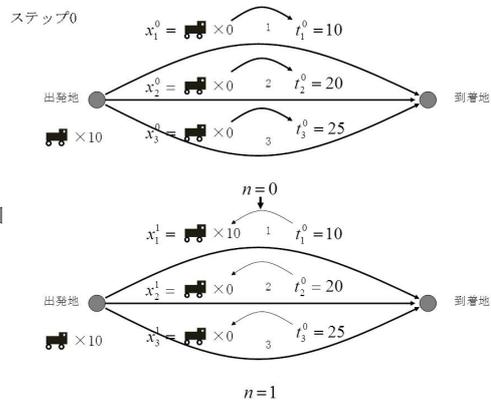


図 3.71 *Frank - Wolfe* 法: ステップ 0

(f) ゲーム理論を適用したリスク回避利用者均衡配分

Bell and Cassir(2002)[5] では、経路コストに不確実性が伴う状況、すなわち脆弱性を伴うネットワーク上での利用者の確率的な経路選択を定式化するため、ゲーム理論を導入した。はじめに、ネットワーク利用者が n 人であるとき、交通量配分はナッシュ均衡状態の n 人参加混合戦略非協力ゲームと等価であることを示す。

決定論的利用者均衡と確率論的利用者均衡の等価性の証明

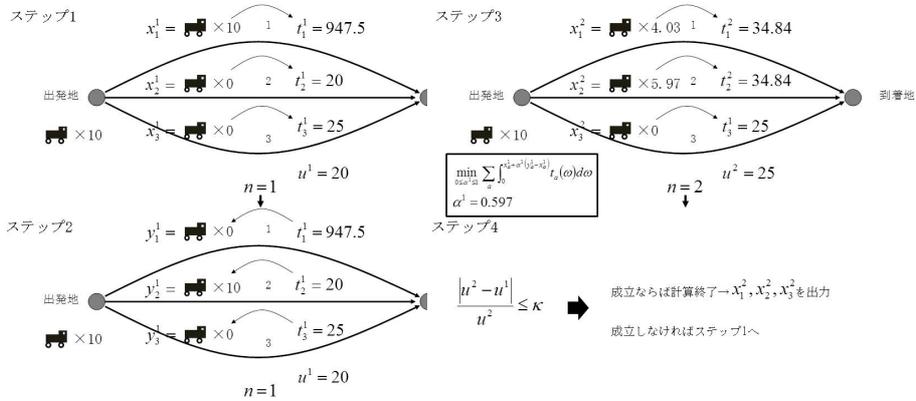


図 3.72 Frank – Wolfe 法: ステップ 1、ステップ 2 図 3.73 Frank – Wolfe 法: ステップ 3、ステップ 4

前提として、1つの OD ペア、複数の経路選択肢が存在するネットワークを、一様な n 人が利用する状況を想定する。

(i) 決定論的交通量均衡配分の必要十分条件

Wardrop の第一原則は、以下のように定式化される。

$$h_j = 0 \Leftrightarrow g_j(\mathbf{h}) > \min_k g_k(\mathbf{h}) \text{ for all paths } j \quad (3.314)$$

$$h_j > 0 \Rightarrow g_j(\mathbf{h}) = \min_k g_k(\mathbf{h}) = g_{OD}(\mathbf{h}) \quad (3.315)$$

ここで、 \mathbf{h} は経路交通量ベクトル、 \mathbf{g} は経路旅行時間ベクトルである。 n が十分大きいと仮定すると、大数の法則により、 $\mathbf{h}_j \cong \mathbf{p}_j n$ あるいは $\mathbf{h}_j = \mathbf{0}$ のとき $\mathbf{p}_j = \mathbf{0}$ が成り立つ。ただし \mathbf{p}_j は経路 j がどの利用者にも選択される確率である。それ故、もし $\mathbf{g}_j(\mathbf{h}^*)$ が最小 OD コスト $g_{OD}(\mathbf{h})$ より大きいとき、均衡点 \mathbf{h}^* 、 $\mathbf{p}_j = \mathbf{0}$ となる。

(ii) n 人非協力混合戦略ゲームが混合戦略ナッシュ均衡であるための必要十分条件

どの利用者 a も複数の経路選択肢があり、経路選択は他の全ての利用者に対するゲームであると仮定し、経路選択混合戦略である状況を想定する。利用者 a に対する混合戦略 \mathbf{s}_a は、経路選択純戦略 π_{aj} の凸結合である。経路選択純戦略 π_{aj} は、利用者 a が経路 j を選択する場合に用いられる。故に、

$$s_a = \sum_j \pi_{aj} p_{aj} \quad (3.316)$$

ここで、 p_{aj} は利用者 a が経路 j を選択する確率である。利用者 a の経路は、全ての利用者による混合戦略の n 次元ベクトル s によって次のように定義される。

$$c_a = \sum_j p_{aj} c_a(s_{-a}, \pi_{aj}) \quad (3.317)$$

ここで $c_a(s_{-a}, \pi_{aj})$ は利用者 a が経路 j を選択し、他の利用者が $n-1$ 次元ベクトル s_{-a} で表される混合戦略に従った場合のコストである。利用者は一様とする仮定から、コストは全ての利用者間で同一である。全ての利用者が同じ選択肢 (同じ経路) を持ち、全員がコストを最小化するため、プレーヤーは入れ替え可能である。それ故どの経路 j に対しても、以下の等式が成り立つ。

$$p_{1j} = p_{2j} = \dots = p_{nj} = p_j \quad (3.318)$$

$$c_1(s_{-1}, \pi_{1j}) = c_2(s_{-2}, \pi_{2j}) = \dots = c_n(s_{-n}, \pi_{nj}) \quad (3.319)$$

h はおよそ p_n と等しいため、利用者 a にとっての経路 j のコストはおよそ h に基づく経路 j のコストと同等である。

$$c_a(s_{-a}, \pi_{aj}) \cong g_j(h) \quad (3.320)$$

大数の法則により、利用者 a にとっての経路 j のコストは n の増加に従い、 h に基づく経路 j のコストに近づく。 n が十分大きい場合、以下の式より、決定論的利用者均衡配分はナッシュ均衡と同等であるといえる。

$$p_{aj} \Leftarrow c_a(s_{-a}, \pi_{aj}) > \min_k c_a(s_{-a}, \pi_{ak}) = 0 \quad (3.321)$$

$$p_{aj} > 0 \Rightarrow c_a(s_{-a}, \pi_{aj}) = \min_k c_a(s_{-a}, \pi_{ak}) \quad (3.322)$$

混合戦略ナッシュ均衡 (Nash, 1951) であるための必要十分条件はそれ故、 n が十分大きいときの決定論的利用者均衡の必要十分条件と同等であることが示された。

Network demon を導入した n+1 人ゲームの定式化

次に、新たなゲームのプレーヤーとなる demon の存在を導入する． demon は 1 つのリンクにダメージを与えることで、ネットワーク利用者のコストを最大化する． 純粋ナッシュ均衡でない限り、demon はネットワーク内のリンクの数と同数の純粋戦略を実行しなくてはならない． n+1 人ゲーム G_1 は以下のように定式化される．

G_1 : solve *simultaneously*

$$\begin{aligned} c_a(s, q) &= \sum_j p_{aj} \sum_k q_k c_{ak}(s_{-a}, \pi_{aj}) \text{ for each network user } a, a \in (1, \dots, n) \\ c_{n+1}(s, q) &= \sum_k q_k c_{n+1,k}(s_a) \text{ for the demon player } n+1 \end{aligned} \quad (3.323)$$

ここで、 q はリンクダメージシナリオ k の発生確率ベクトル、 $c_{ak}(s_{-a}, \pi_{aj})$ は利用者 a がシナリオ k において経路 j を選択し、経路戦略 n-1 次元ベクトル s_{-a} が適用されるときのコスト、 $c_{n+1,k}(s)$ は demon がシナリオ k を選択し、n 次元ベクトルである利用者戦略 s を適用するときの効用を表す． このプレーヤーの狙いは全プレーヤーのコストを増加させることである． シナリオ k におけるネットワーク利用者の期待コストの合計 $c_{n+1,k}(s)$ は、以下のように定式化される．

$$c_{n+1,k}(s) = \sum_a \sum_j p_{aj} c_{ak}(s_{-a}, \pi_{aj}) = \sum_a c_{ak}(s) \quad (3.324)$$

ここで、少なくともひとつの混合戦略均衡解が存在する (Nash(1951))． しかし、特に n が大きいとき、均衡点を見つけるのは困難である． そこで、以下では二段階最適化問題によって均衡解を求めることで、ゲームのマクロ的な解法とおよそ同等の解を得る．

二段階最適化問題 B_1 の定式化

B_1 : solve simultaneously

$$U : \max_q c_a(s, q) = \sum_j \sum_k q_k g_{jk}(h) \text{ subject to } \sum_k q_k = 1, q \geq 0$$

$$L : \min_h c_{n+1}(s, q) = \sum_u \sum_k q_k \int_0^{v_u(h)} t_{uk}(x) dx \quad (3.325)$$

$$\text{subject to } v_u = \sum_j a_{uj} h_j, \sum_j h_j = n, h \geq 0$$

ここで $g_{jk}(h)$ はシナリオ k , 交通量ベクトル h 下での経路 j のコスト, $t_{uk}(v_u)$ はシナリオ k におけるリンク u の交通量に依存するコスト, a_{uj} は経路 j にリンク u が含まれるとき 1, そうでない場合は 0 となる指示関数である. 以下に, n が十分大きいとき, 二段階最適化問題 B_1 は G_1 のマクロ的な解を与えることを示す.

B_1 の上位問題は demon の視点からゲームを解いており, ネットワーク利用者に課される期待コストの和を最大化させる. 下位問題は, 期待コストに基づく標準的な決定論的利用者均衡配分問題である. 相互に一定の点 (ナッシュ均衡であり, シュタッケルベルグ均衡ではない) において, 以下の条件が適用される.

For $U : \forall k$

$$q_k = 0 \Leftrightarrow \sum_j g_{jk}(h) h_j < \max_r \sum_j g_{jr}(h) h_j \quad (3.326)$$

$$q_k > 0 \Rightarrow \sum_j g_{jk}(h) h_j = \max_r \sum_j g_{jr}(h) h_j \quad (3.327)$$

For $L : \forall k$

$$h_j = 0 \Leftrightarrow \sum_k g_{jk}(h) q_k < \min_r \sum_k g_{rk}(h) q_k \quad (3.328)$$

$$h_j > 0 \Rightarrow \sum_k g_{jk}(h) q_k = \min_r \sum_k g_{rk}(h) q_k \quad (3.329)$$

大数の法則により, 集団レベルでの合計期待ネットワークコストは, n が大きくなるにつれて個々の期待コストの和に近づく.

$$\sum_j g_{jk}(h) h_j \cong c_{n+1,k}(s) \quad (3.330)$$

この差は n が大きくなるにつれて小さくなる。従って、ミクロ的なレベルでの上式は、以下の様に示される。

$$q_k = 0 \Leftarrow c_{n+1,k}(s) < \max_r c_{n+1,r}(s) \quad (3.331)$$

$$q_k > 0 \Rightarrow c_{n+1,k}(s) = \max_r c_{n+1,r}(s) \quad (3.332)$$

n が大きいとき、任意の経路 j と大数 n に対し、

$$\sum_k g_{jk}(h) q_k \cong \sum_k c_{ak}(s_{-a}, \pi_{aj}) q_k = c_a(s_{-a}, \pi_{aj}) \quad (3.333)$$

また、 $h_j \cong p_j n \cong p_{aj} n$ であるために、上式はミクロレベルにおいて、

$$p_{aj} = 0 \Leftarrow c_a(s_{-a}, \pi_{aj}) < \max_r c_a(s_{-a}, \pi_{ar}) \quad (3.334)$$

$$p_{aj} > 0 \Rightarrow c_a(s_{-a}, \pi_{aj}) = \max_r c_a(s_{-a}, \pi_{ar}) \quad (3.335)$$

n が大きいとき、これらの条件は非協力混合戦略 $n+1$ 人ナッシュ均衡問題における必要十分条件である (Nash, 1951)[6]。従って、 n の増加に伴って、 B_1 を解くことでこれまでに定義した $n+1$ 人ゲームである G_1 に関するマクロ的な近似解を得ることが示された。

参考文献

- [1] 土木計画学研究委員会交通需要予測技術検討小委員会: 道路交通需要予測の理論と適用 第1編 利用者均衡配分の適用に向けて, 土木学会, 2003
- [2] 土木計画学研究委員会交通需要予測技術検討小委員会: 道路交通需要予測の理論と適用 (第2編) 利用者均衡配分モデルの展開, 土木学会, 2006
- [3] Yosef Sheffi: Urban Transportation Networks: Equilibrium Analysis With Mathematical Programming Methods, Prentice Hall, 1985
- [4] 有村幹治, 田村享, 井田直人: 土木計画分野における遺伝的アルゴリズム: 最適化と適応学習, 土木学会論文集 D, Vol.62 No.4, pp505-518, 2006.10
- [5] Michael G. H. Bell, Chris Cassir: Transportation Research Part B, Vol.36, pp.671-681, 2002.
- [6] Nash, J.: Non-cooperative games: Annals of Mathematics, Vol.54, pp.286-295, 1951.

