

利用者均衡ネットワーク設計問題に対する 近似解法の調査

片山直登

1 はじめに

利用者均衡ネットワーク設計問題 (user equilibrium network design problem) は, 利用者均衡条件を考慮した交通ネットワークの改善を対象としたネットワーク設計問題である。利用者均衡とは, 「利用者が利用する経路の走行時間 (費用) はすべて等しく, 取りうる経路の中で最小時間 (費用) の経路であり, 利用されない経路はそれ以上の走行時間 (費用) をもつ」というものである。

利用者均衡条件を仮定したネットワーク設計問題では, 計画者はネットワーク全体の最適化を目指し, 利用者は個々の最適化を目指す 2 レベルの最適化問題として表現されるため, 制約条件に最適化問題を含む複雑な非線形計画問題となる。この問題の実行可能領域は凸領域とは限らないため, 例題程度の小規模の問題でも最適解や最適値を求めることは困難である。そのため, 多くの近似解法が提案されている。しかし, 目的関数値を求めるためには非線形計画問題を解く必要があるが, この問題自体用いる解法や収束判定基準などの計算条件によって同じ解に対する目的関数値が異なり, 基本的なベンチマーク問題でさえ最適値が特定できていないという問題点がある。

本論文では, 従来の近似解法を調査するとともに, 以前, 著者が提案したタブサーチ法を用いた近似解をリファインした解法を示す。また, いくつかのベンチマーク問題を用いて, 主な従来の解法によって得られる近似解を比較し, 片山によるタブサーチ法による結果の検証も行う。

2 利用者均衡ネットワーク設計問題の定式化

利用者均衡ネットワーク設計問題は利用者均衡条件を制約に含む最適化問題となり,

利用者均衡条件自身が最適化問題として定式化できるために、2レベル最適化問題として表される。

アーク集合を A 、品種集合を K 、品種 k のパスの集合を P^k とし、アーク (i, j) 上のフロー量を x_{ij} 、アーク (i, j) のデザイン変数を y_{ij} 、品種 k がパス p を流れるフロー量を z_p^k とする。ここで、デザイン変数は拡張する容量を表し、連続値をとるものとする。

デザイン変数 y_{ij} の下限を L_{ij} 、上限を U_{ij} 、品種 k の需要量を d^k とし、品種 k のパス p がアーク (i, j) を含むとき1、そうでないとき0である定数を δ_{ijp}^k とする。

また、 x_{ij} と y_{ij} の関数であるアーク (i, j) 上のフロー費用関数を $c_{ij}(x_{ij}, y_{ij})$ とし、 y_{ij} の関数であるアーク (i, j) のデザイン費用関数を $f_{ij}(y_{ij})$ とする。 c_{ij} は x_{ij} 、 y_{ij} で微分可能で x_{ij} に関して非減少で下に凸である関数、 f_{ij} は y_{ij} で微分可能である関数とする。

利用者均衡ネットワーク設計問題のパスフローを用いた定式化を示す。

$$\begin{array}{ll} \text{最小化} & \sum_{(i,j) \in A} c_{ij}(x_{ij}, y_{ij})x_{ij} + \sum_{(i,j) \in A} f_{ij}(y_{ij}) \\ \text{条件} & L_{ij} \leq y_{ij} \leq U_{ij} \quad (i, j) \in A \end{array} \quad (1)$$

$$\begin{array}{ll} \text{最小化} & \sum_{(i,j) \in A} \int_0^{x_{ij}} c_{ij}(t, y_{ij}) dt \end{array} \quad (2)$$

$$\begin{array}{ll} \text{条件} & \sum_{p \in P^k} z_p^k = d^k \quad k \in K \end{array} \quad (3)$$

$$x_{ij}^k = \sum_{p \in P^k} \delta_{ijp}^k z_p^k \quad (i, j) \in A, k \in K \quad (4)$$

$$z_p^k \geq 0 \quad p \in P^k, k \in K \quad (5)$$

目的関数の第一項は総フロー費用、第二項は総デザイン費用であり、システム全体の最適化を求めるものであり、この和を最小化する。

(1)式は、デザイン変数の下限と上限を表す。(2)~(5)式は、利用者均衡条件である利用者均衡フロー問題を表している。(2)式は、利用者均衡条件の目的関数である。(3)式は、品種 k のパスフロー量の合計が品種 k の需要量に一致することを表す。(4)式はアーク (i, j) 上を流れるパスフロー変数とアークフロー変数の関係、(5)式はパスフロー変数が非負であることを表す。

一方、フロー変数の端点解集合を X_e とすると、利用者均衡条件の目的関数(2)式は次のような変分不等式に置き換えることができる。

$$\sum_{(i,j) \in A} c_{ij}(x_{ij}, y_{ij})(\bar{x}_{ij} - x_{ij}) \geq 0 \quad \bar{x} \in X_e$$

いずれにしても、利用者均衡ネットワーク設計問題は制約条件に最適化問題または変分不等式を含む問題であり、その実行可能領域は凸領域とは限らないため、小規模の問題でも最適解を求めることは困難である。

3 従来の研究

(a) Steenbrink分解法

デザイン変数 \mathbf{y} を決めると、フロー変数 \mathbf{x} は利用者均衡フロー問題を解くことによって決定できるため、 \mathbf{y} の陰関数と考えることができる。このことから、利用者均衡ネットワーク設計問題は、上位の設計問題と

$$\begin{aligned} \text{最小化} \quad & \sum_{(i,j) \in A} c_{ij}(x_{ij}(\mathbf{y}), y_{ij}) x_{ij}(\mathbf{y}) + \sum_{(i,j) \in A} f_{ij}(y_{ij}) \\ \text{条件} \quad & L_{ij} \leq y_{ij} \leq U_{ij} \quad (i,j) \in A \end{aligned}$$

下位の利用者均衡フロー問題

$$\begin{aligned} \text{最小化} \quad & \sum_{(i,j) \in A} \int_0^{x_{ij}} c_{ij}(t, y_{ij}) dt \\ \text{条件} \quad & \mathbf{x} \in \text{System} \end{aligned}$$

に分解することができる。ここで、 $x_{ij}(\mathbf{y})$ は、 \mathbf{y} が与えられたときの利用者均衡フロー問題におけるアーク (i, j) 上の最適フロー解を表し、 System は (3) から (5) 式を満足する実行可能領域である。このような分解を **Steenbrink分解** とよび、これらの問題を交互に解き、収束解を求める方法を **Steenbrink分解法** とよぶ。

(b) Hooke-Jeeves法

微分を用いない探索法として、Hooke-Jeeves法 [1] がある。アーク番号を通し番号に置き換え、 $\mathbf{y} = (y_1, y_2, \dots, y_{|A|})$ のようにおくと、利用者均衡ネットワーク設計問題は次のように表すことができる。

$$\begin{aligned} \text{最小化} \quad & \phi(\mathbf{y}) = \sum_{l \in A} c_l(x_l(\mathbf{y}), y_l) x_l(\mathbf{y}) + \sum_{l \in A} f_l(y_l) \\ \text{条件} \quad & L_l \leq y_l \leq U_l \quad l \in A \end{aligned}$$

Hooke-Jeeves法の探索・移動では、1 デザイン変数毎に変化させて近傍探索し、繰返し目的関数値を評価する。ただし、近傍探索時にフロー変数は変化しないものと仮定して、現在の探索基点のフロー解をそのまま使用する。

l 番目の要素だけが1であるベクトルを $\mathbf{e}_l = (0, \dots, 1, \dots, 0)$ とおく。 \mathbf{y}_B を探索の基点、 $\bar{\mathbf{y}}$ を \mathbf{y}_B の近傍点である試行点とする。このとき、基点 \mathbf{y}_B から点 $\bar{\mathbf{y}} + \alpha \mathbf{e}_l$ への近傍探索を行う場合に、目的関数値の変化量の近似値 $\Delta \phi_l(\mathbf{y}_B, \bar{\mathbf{y}} + \alpha \mathbf{e}_l)$ は、次のように求めることができる。

$$\Delta \phi_l(\mathbf{y}_B, \bar{\mathbf{y}} + \alpha \mathbf{e}_l) = c_l(x_l(\mathbf{y}_B), \bar{y}_l + \alpha) x_l(\mathbf{y}_B) + f_l(\bar{y}_l + \alpha) - \{c_l(x_l(\mathbf{y}_B), \bar{y}_l) x_l(\mathbf{y}_B) + f_l(\bar{y}_l)\}$$

k 回目のパターン移動時における l 回目の近傍探索における試行点を \mathbf{y}_k^l 、パターン移

動のための基点を y_B^k とおく。 a を適当なステップサイズとし、 $[y]_L^U = \max(L, \min(U, y))$ とする。

Hooke-Jeeves法の探索・移動は、 $l=1, \dots, |A|$ に対して次のように行う。 $\Delta\phi_l(y_B^k, [y_l^k + ae_{lL}]^U) < 0$ であれば、 $y_{l+1}^k := [y_l^k + ae_{lL}]^U$ とする。 $\Delta\phi_l(y_B^k, [y_l^k - ae_{lL}]^U) < 0$ であれば、 $y_{l+1}^k := [y_l^k - ae_{lL}]^U$ とする。 いずれでもなければ、 $y_{l+1}^k := y_l^k$ とする。

また、パターン移動は、 $y_B^{k+1} := y_{|A|+1}^k$ とし、 $y_l^{k+1} := y_B^k + a(y_B^{k+1} - y_B^k)$ として行う。ただし、 $a (>0)$ は適当なステップサイズである。

(c) 変分不等式制約追加法

変分不等式は端点数だけ存在する。そこで、変分不等式を含まない問題から始め、逐次、必要な変分不等式を生成・追加して、収束解を求める解法が変分不等式制約追加法 [5] である。

次のような変分不等式を使った定式化を用いる。

$$\begin{aligned} \text{最小化} \quad & \sum_{(i,j) \in A} c_{ij}(x_{ij}, y_{ij})x_{ij} + \sum_{(i,j) \in A} f_{ij}(y_{ij}) \\ \text{条件} \quad & \sum_{(i,j) \in A} c_{ij}(x_{ij})(\bar{x}_{ij}^l - x_{ij}) \geq 0 \quad l = 1, \dots, L \\ & L_{ij} \leq y_{ij} \leq U_{ij} \quad (i, j) \in A \\ & x \in \text{System} \end{aligned}$$

ここで、 \bar{x}^l を端点 l の解、 L を端点数とする。

変分不等式の数を L 個未満に限定した問題は緩和問題となるため、その最適値は利用者均衡ネットワーク設計問題の下界値となる。一方、緩和問題のデザイン変数解を用いて、利用者均衡フロー問題を解き、目的関数に代入すれば上界値が得られる。

現在のデザイン変数とフロー変数による費用関数値をアークの費用とした All or Nothing 配分問題から求められたフロー解は利用者均衡ネットワーク設計問題の端点解となる [5] ので、この端点に対する変分不等式を逐次追加していけばよい。

(d) 利用者均衡の目的関数を用いた 1 レベル問題

フロー費用関数が BPR 関数である場合、システム最適化の目的関数と利用者均衡フロー問題の目的関数の形が類似していることから、利用者均衡フロー問題の目的関数を利用者均衡ネットワーク設計問題の目的関数とした 1 レベルの最適化問題 [6][15] を作成する。この 1 レベル最適化問題を解けば、利用者均衡ネットワーク設計問題の近似解を求めることができる。

フロー費用関数が BPR 関数である利用者均衡ネットワーク設計問題の定式化を示す。

$$\begin{array}{ll}
 \text{最小化} & \phi = \sum_{(i,j) \in A} t_{ij}^0 \left\{ 1 + \alpha \left(\frac{x_{ij}}{C_{ij} + y_{ij}} \right)^\beta \right\} x_{ij} \\
 \text{条件} & L_{ij} \leq y_{ij} \leq U_{ij} \quad (i, j) \in A \\
 \text{最小化} & \sum_{(i,j) \in A} \int_0^{x_{ij}} \left(t_{ij}^0 \left\{ 1 + \alpha \left(\frac{t_{ij}}{C_{ij} + y_{ij}} \right)^\beta \right\} \right) dt \\
 \text{条件} & \mathbf{x} \in \text{System}
 \end{array}$$

ここで、 t_{ij}^0 はアーク (i, j) の 0 フロー時のフロー費用、 $\alpha (\geq 0)$ と $\beta (\geq 1)$ はネットワークに特有なパラメータ、 C_{ij} は現在の容量である。

パラメータ ξ を用いて、利用者均衡フロー問題の目的関数にデザイン費用を加え、この関数を目的関数とした次のような 1 レベル問題を作成する。

$$\begin{array}{ll}
 \text{最小化} & \phi_1 = \sum_{(i,j) \in A} \int_0^{x_{ij}} \left(t_{ij}^0 \left\{ 1 + \alpha \left(\frac{t_{ij}}{C_{ij} + y_{ij}} \right)^\beta \right\} \right) dt + \xi \sum_{(i,j) \in A} f_{ij}(y_{ij}) \\
 \text{条件} & L_{ij} \leq y_{ij} \leq U_{ij} \quad (i, j) \in A \\
 & \mathbf{x} \in \text{System}
 \end{array}$$

この問題は次の設計問題と利用者均衡フロー問題の 2 つの問題に分解することができる。

$$\begin{array}{ll}
 \text{最小化} & \sum_{(i,j) \in A} \int_0^{x_{ij}^*} \left(t_{ij}^0 \left\{ 1 + \alpha \left(\frac{t_{ij}}{C_{ij} + y_{ij}} \right)^\beta \right\} \right) dt + \xi \sum_{(i,j) \in A} f_{ij}(y_{ij}) \\
 \text{条件} & L_{ij} \leq y_{ij} \leq U_{ij} \quad (i, j) \in A \\
 \text{最小化} & \sum_{(i,j) \in A} \int_0^{x_{ij}} \left(t_{ij}^0 \left\{ 1 + \alpha \left(\frac{t_{ij}}{C_{ij} + y_{ij}^*} \right)^\beta \right\} \right) dt + \xi \sum_{(i,j) \in A} f_{ij}(y_{ij}^*) \\
 \text{条件} & \mathbf{x} \in \text{System}
 \end{array}$$

ここで、 \mathbf{x}^* は利用者均衡フロー問題の最適解、 \mathbf{y}^* は設計問題の最適解である。

1 レベル問題の目的関数が総費用の最小化ではないため、この解法によって得られた解は総費用を少なくする解を算出することは保証されない。そこで、 $\xi = 1/(\beta + 1)$ と設定する。このとき、1 レベル問題の目的関数 ϕ_1 の偏微分

$$\frac{\partial \phi_1}{\partial y_{ij}} = \frac{1}{\beta + 1} \left\{ -t_{ij}^0 \alpha \beta \left(\frac{x_{ij}}{C_{ij} + y_{ij}} \right)^{\beta+1} + \frac{\partial f_{ij}}{\partial y_{ij}} \right\}$$

と、利用者均衡ネットワーク設計問題の目的関数 ϕ の偏微分

$$\frac{\partial \phi}{\partial y_{ij}} = -t_{ij}^0 \alpha \beta \left(\frac{x_{ij}}{C_{ij} + y_{ij}} \right)^{\beta+1} + \frac{\partial f_{ij}}{\partial y_{ij}}$$

を比べると、1 レベル問題の目的関数 ϕ_1 の偏微分は利用者均衡ネットワーク設計問題の $1/(\beta + 1)$ 倍となる。このため、1 レベル問題の降下方向は利用者均衡ネットワーク設計問題における総費用を減少させる方向に一致する。そこで、利用者均衡フロー問題と設計問題を交互に解けば、利用者均衡を満足し、かつ総費用を減少させる近似解が求まる

ことが期待できる。

(e) Bardのアルゴリズム

問題利用者均衡ネットワーク設計問題は2レベル最適化問題であることから、2レベル最適化問題のBardのポイントアルゴリズム[2]を適用することができる。

パラメータ λ ($0 < \lambda < 1$) を用いて、利用者均衡ネットワーク設計問題の目的関数と利用者均衡フロー問題の目的関数を結合した問題を作成する。

$$\begin{aligned} \text{最小化} \quad & \lambda \left\{ \sum_{(i,j) \in A} c_{ij}(x_{ij}, y_{ij}) x_{ij} + \sum_{(i,j) \in A} f_{ij}(y_{ij}) \right\} + (1-\lambda) \left\{ \sum_{(i,j) \in A} \int_0^{x_{ij}} c_{ij}(t, y_{ij}) dt \right\} \\ \text{条件} \quad & L_{ij} \leq y_{ij} \leq U_{ij} \quad (i, j) \in A \\ & \mathbf{x} \in \text{System} \\ & 0 \leq \lambda \leq 1 \end{aligned}$$

デザイン費用関数が凸関数である場合、 λ を固定すればこの問題は凸多品種フロー問題となり、Frank-Wolfe法などの解法を用いて解を求めることができる。

任意のパラメータ λ に対して、得られたフロー解は利用者均衡条件を満足するとは限らないが、 $\lambda = 0$ であれば利用者均衡フロー問題となるため、 λ が十分に0に近い値であれば利用者均衡条件を満足することが期待できる。そこで、パラメータ λ を1より、順次減少させて繰り返し問題を解き、フロー解が利用者均衡条件を満足していれば、その解は実行可能解となる。

しかし、 λ が十分に小さい値であっても利用者均衡条件を満足しない例[7]が存在するため、実際にはフロー解が利用者均衡条件をほぼ満足した段階で、得られたデザイン変数を用いて利用者均衡フロー問題を解き直す必要がある。

(f) 均衡分解最適化法

均衡分解最適化法[11]は、近似的な微分値を計算し、この値の正負によってデザイン変数の探索範囲を限定し、探索範囲が収束するまで繰り返す方法である。

陰関数 $x_{ij}(\mathbf{y})$ を用いた定式化を使用する。

$$\begin{aligned} \text{最小化} \quad & \phi(\mathbf{y}) = \sum_{(i,j) \in A} c_{ij}(x_{ij}(\mathbf{y}), y_{ij}) x_{ij}(\mathbf{y}) + \sum_{(i,j) \in A} f_{ij}(y_{ij}) \\ \text{条件} \quad & L_{ij} \leq y_{ij} \leq U_{ij} \quad (i, j) \in A \end{aligned}$$

目的関数 ϕ の偏微分は、

$$\frac{\partial \phi}{\partial y_{ij}} = \frac{\partial f_{ij}(y_{ij})}{\partial y_{ij}} + \frac{\partial c_{ij}(x_{ij}(\mathbf{y}), y_{ij})}{\partial y_{ij}} x_{ij}(\mathbf{y}) + c_{ij}(x_{ij}(\mathbf{y}), y_{ij}) \frac{\partial x_{ij}(\mathbf{y})}{\partial y_{ij}}$$

となる。右辺には出る $\partial x_{ij}(\mathbf{y}) / \partial y_{ij}$ が含まれているため、次のような仮定をおく。

$$\left| \frac{\partial f_{ij}(y_{ij})}{\partial y_{ij}} + \left(\frac{\partial c_{ij}(x_{ij}, y_{ij})}{\partial y_{ij}} \right) x_{ij}(\mathbf{y}) \right| \gg \left| c_{ij}(x_{ij}(\mathbf{y}), y_{ij}) \frac{\partial x_{ij}(\mathbf{y})}{\partial y_{ij}} + \delta \right|$$

ただし、 $(\partial c_{ij} / \partial y_{ij})_x$ はフロー変数が x のときの偏微分値であり、 $\delta = |(\partial c_{ij} / \partial y_{ij})_x - (\partial c_{ij} / \partial y_{ij})_{x_{ij}}|$ である。

この仮定のもとでは、現在のフロー変数が \bar{x} のとき、微分値の符号は、

$$Z'_{ij}(\mathbf{y}) = \frac{\partial f_{ij}(y_{ij})}{\partial y_{ij}} + \left(\frac{\partial c_{ij}(x_{ij}, y_{ij})}{\partial y_{ij}} \right) \bar{x}_{ij}$$

によって判断することができる。現在のデザイン変数を \bar{y} とすると、 $Z'_{ij}(\mathbf{y}) < 0$ であれば、 y_{ij} の値を増加させるほうが良いので探索範囲を \bar{y} 以上に限定することができる。また、 $Z'_{ij}(\mathbf{y}) > 0$ であれば、 y_{ij} の値を減少させるほうが良いので探索範囲を \bar{y} 以下に限定することができる。

このような限定操作を繰り返し、2分探索法によってデザイン変数の範囲を限定していけば、収束値を求めることができる。

(g) Friesz-Tobin-Cho-Mehtaの感度分析法

感度分析は、パラメータや係数の変化が解に及ぼす影響を分析する手法である。利用者均衡ネットワーク設計問題に対しては、デザイン変数が変化したときの感度分析 [10] が示されており、この感度分析にもとずいたヒューリスティック解法 [4] が示されている。

品種 k のパス p のフロー費用はフロー変数 z とデザイン変数 y の関数として表すことができるので、 $t_p^k(z, y)$ とおく。利用者均衡条件を満足しているフロー、すなわち、 $z_p^k > 0$ であるすべての品種 k のパス p に対して $t_p^k(z, y) = v^k$ かつ $s_p^k = 0$ を満足する正の均衡フローに、感度分析の対象を限定する。ここで、 v は (3) 式に対する双対変数、 s は (5) 式に対する双対変数である。

$t_p^k(z, y)$ に関する列ベクトルを $T(z, y)$ 、パス接続行列を Λ 、 d^k に関する列ベクトルを d とおく。このとき、Kuhn-Tucker条件は次のようにまとめることができる [12]。

$$\begin{aligned} T(z, y) - \Lambda^t v &= 0 \\ \Lambda z - d &= 0 \end{aligned} \tag{6}$$

この条件から、 z 、 v に関するJacobi行列 $J_{z,v}$ 、および y に関するJacobi行列 J_y を求めると、

$$J_{z,v} = \begin{bmatrix} \nabla_z T(z, y) & -\Lambda^t \\ \Lambda & 0 \end{bmatrix}, \quad J_y = \begin{bmatrix} \nabla_y T(z, y) \\ 0 \end{bmatrix}$$

となり、デザイン変数 y に対する z および v の変化量 $\nabla_y z, \nabla_y v$ は、次式で求められる [13]。

$$\begin{bmatrix} \nabla_y z \\ \nabla_y v \end{bmatrix} = -J_{z,v}^{-1} J_y \tag{7}$$

一方、利用者均衡ネットワーク設計問題の目的関数 ϕ の y_{ij} による偏微分は、

$$\frac{\partial \phi}{\partial y_{ij}} = \frac{\partial c_{ij}(x_{ij}, y_{ij})}{\partial x_{ij}} \frac{\partial x_{ij}}{\partial y_{ij}} x_{ij} + c_{ij}(x_{ij}, y_{ij}) \frac{\partial x_{ij}}{\partial y_{ij}} + \frac{\partial f_{ij}(y_{ij})}{\partial y_{ij}}$$

である。したがって、(6)式より、

$$\frac{\partial \phi}{\partial y_{ij}} = \left(\frac{\partial c_{ij}(x_{ij}, y_{ij})}{\partial x_{ij}} x_{ij} + c_{ij}(x_{ij}, y_{ij}) \right) \left(\sum_{k \in K} \sum_{p \in P^k} \delta_{ijp}^k \frac{\partial z_p^k}{\partial y_{ij}} \right) + \frac{\partial f_{ij}(y_{ij})}{\partial y_{ij}} \quad (8)$$

を得る。

(7)式の勾配を $\partial z_p^k / \partial y_{ij}$ として用いると、(8)式より $\partial \phi / \partial y_{ij}$ の値を定めることができる。降下法における k 回目の繰返しのデザイン変数を \mathbf{y}^k 、フロー変数を \mathbf{z}^k とし、

$$[\mathbf{y}]_L^U = \begin{bmatrix} \vdots \\ \max(L_{ij}, \min(U_{ij}, y_{ij})) \\ \vdots \end{bmatrix}$$

とおき、次のようにデザイン変数 \mathbf{y} を更新する。

$$\mathbf{y}^{k+1} := [\mathbf{y}^k - \alpha \nabla \mathbf{y} \phi]_L^U$$

ステップサイズ α の設定にArmijoのルール[14]を適用すれば、収束解を求めることができる。

(h) アニーリング法

一般的に、アニーリング法は離散最適化問題に適用される。しかし、利用者均衡ネットワーク設計問題の決定変数であるデザイン変数は連続値をとるため、近傍探索の方法を工夫する必要がある。そこで、乱数 \mathbf{u} とコントロール行列 Q を用いて、 m 回目の繰返しにおけるデザイン変数 \mathbf{y}^m に対して、アニーリング法の原理に従って、次のような近傍解 \mathbf{y}' を探索する [3]。

$$\mathbf{y}' = [\mathbf{y}^m + Q\mathbf{u}]_L^U$$

ここで、 $\mathbf{u} = (\dots, u_i, \dots)$ は、アークに対して独立した乱数とする。 Q は $S = QQ^T$ を満たす行列、 S は適当な共分散行列であり、 Q はコレスキー分解を用いて求めることができる。

同一温度での繰返し回数を M 、現在 l 回目の温度変更を行い、現在の温度で m 回繰返したときに、次の計算を行う。

$$a_{ij}^l = \frac{1}{M} \sum_{m=1}^M y_{ij}^{ml} \quad (i, j) \in A$$

$$r_{ij,pq}^l = \frac{1}{M} \sum_{m=1}^M (y_{ij}^{ml} - a_{ij}^l)(y_{pq}^{ml} - a_{pq}^l) \quad (i, j) \in A, (p, q) \in A$$

ここで、 y_{ij}^{ml} は、 l 回目の温度変更時の m 回目の繰返しにおける y_{ij} である。 a_{ij}^l は l 回目の

温度変更時の y_{ij} の平均値である。 $r'_{ij,pq}$ は y_{ij} の共分散であり、 l 回目の温度変更時の y_{ij} のばらつき具合を表している。 $r'_{ij,pq}$ を要素とする共分散行列 R^l を用いて、次のように $l+1$ 回目の S を設定する。

$$S^{l+1} = \frac{\chi S}{\beta M} R^l$$

ここで、 $\chi_s (>1)$ と $\beta (>0)$ は探索する範囲を決定するパラメータである。

(i) 拡張ラグランジュ法

現在の解における利用者均衡フロー問題の目的関数値と最適値の差を表す関数をギャップ関数とよぶ。利用者均衡フロー問題の最適解を \mathbf{x}^* としたとき、ギャップ関数 $h(\mathbf{y}, \mathbf{x})$ は次のように表される。

$$h(\mathbf{y}, \mathbf{x}) = \sum_{(i,j) \in A} \int_0^{x_{ij}} c_{ij}(t, y_{ij}) dt - \sum_{(i,j) \in A} \int_0^{x_{ij}^*} c_{ij}(t, y_{ij}) dt$$

このとき、利用者均衡ネットワーク設計問題は次のように定式化できる。

$$\begin{aligned} \text{最小化} \quad & \phi_1 = \sum_{(i,j) \in A} c_{ij}(x_{ij}, y_{ij}) x_{ij} + \sum_{(i,j) \in A} f_{ij}(y_{ij}) \\ \text{条件} \quad & L_{ij} \leq y_{ij} \leq U_{ij} \quad (i, j) \in A \\ & \mathbf{x} \in \text{System} \\ & h(\mathbf{y}, \mathbf{x}) = 0 \end{aligned}$$

適当なペナルティ μ 、 ρ を用いて、ギャップ関数に関する制約式をペナルティ項として目的関数に取り込み、次のような拡張ラグランジュ問題[8]を作成する。

$$\begin{aligned} \text{最小化} \quad & \phi(\mathbf{y}, \mathbf{x}, \mu, \rho) = \phi_1 + \mu h(\mathbf{y}, \mathbf{x}) + \frac{1}{2} \rho (h(\mathbf{y}, \mathbf{x}))^2 \\ \text{条件} \quad & L_{ij} \leq y_{ij} \leq U_{ij} \quad (i, j) \in A \\ & \mathbf{x} \in \text{System} \end{aligned}$$

μ 、 ρ を設定して、 $h(\mathbf{y}, \mathbf{x})=0$ を満たす解を求めることができれば、利用者均衡ネットワーク設計問題の実行可能解を求めることができる。

l 回目のイテレーションにおける (\mathbf{x}, \mathbf{y}) を $(\mathbf{x}', \mathbf{y}')$ とおく。 $(\mathbf{x}', \mathbf{y}')$ における目的関数 ϕ の x_{ij} と y_{ij} の偏微分を $\bar{t}'_{ij} = \partial \phi / \partial x_{ij}$ 、 $\bar{g}'_{ij} = \partial \phi / \partial y_{ij}$ とおく。このとき、 $(\mathbf{x}', \mathbf{y}')$ における降下方向を求める問題は、次の線形計画問題となる。

$$\begin{aligned} \text{最小化} \quad & \sum_{(i,j) \in A} \bar{t}'_{ij} x'_{ij} + \sum_{(i,j) \in A} \bar{g}'_{ij} y'_{ij} \\ \text{条件} \quad & L_{ij} \leq y'_{ij} \leq U_{ij} \quad (i, j) \in A \\ & \mathbf{x}' \in \text{System} \end{aligned}$$

この問題は、 \mathbf{x}' に関する問題と \mathbf{y}' に関する次の2つの問題に分離することができる。

$$\begin{array}{ll} \text{最小化} & \sum_{(i,j) \in A} \bar{r}_{ij}^l x'_{ij} \\ \text{条件} & \mathbf{x} \in \text{System} \end{array}$$

$$\begin{array}{ll} \text{最小化} & \sum_{(i,j) \in A} \bar{g}_{ij}^l y'_{ij} \\ \text{条件} & L_{ij} \leq y'_{ij} \leq U_{ij} \quad (i,j) \in A \end{array}$$

前者は、アーク (i, j) の長さを \bar{r}_{ij}^l とした最短路問題である。後者は 1 変数の最小化問題に分離でき、最適解は、

$$y'_{ij} = \begin{cases} L_{ij} & \text{if } \bar{g}_{ij}^l > 0 \\ U_{ij} & \text{otherwise} \end{cases} \quad (i, j) \in A$$

となる。

なお、 $h(\mathbf{y}^{l+1}, \mathbf{x}^{l+1}) \leq (\mathbf{y}^l, \mathbf{x}^l) / a$ であれば $\mu^{l+1} := \mu^l + \rho^l h(\mathbf{y}^{l+1}, \mathbf{x}^{l+1})$ とし、そうでなければ $\rho^{l+1} := \beta \rho^l$ 、 $\mu^{l+1} := \mu^l$ として、パラメータを更新して、収束解を求める。ただし、 $a (>1)$ と $\beta (<1)$ を正の定数である。

(j) Josefsson-Patrikssonの感度分析法

利用者均衡フロー問題における費用関数を $F(\mathbf{x}, \mathbf{y})$ とおく。ここで、 $F_{ij} = c_{ij}(x_{ij}, y_{ij})x_{ij} + f_{ij}(y_{ij})$ である。利用者均衡条件を満足するには、最適解 \mathbf{x}^* が次の変分不当式を満足する必要がある。

$$F(\mathbf{x}, \mathbf{y})^T (\mathbf{x} - \mathbf{x}^*) \geq 0 \quad \mathbf{x} \in \text{System}$$

現在の解 $(\bar{\mathbf{y}}, \bar{\mathbf{x}})$ において、 \mathbf{y} を \mathbf{y}' 方向に沿って摂動したときの \mathbf{x} の変化を求めるためには、

$$R(\mathbf{x}', \mathbf{y}')^T = \nabla_{\mathbf{y}} F(\bar{\mathbf{y}}, \bar{\mathbf{x}}) \mathbf{y}' + \nabla_{\mathbf{x}} F(\bar{\mathbf{y}}, \bar{\mathbf{x}}) \mathbf{x}'$$

としたとき、 \mathbf{x}^* の極錘に含まれる

$$R(\mathbf{x}', \mathbf{y}')^T (\mathbf{x} - \mathbf{x}') \geq 0 \quad \mathbf{x} \in \text{System}$$

である \mathbf{x}' を求めればよい [9]。

この変分不等式を満たす解を求める問題は、次の 2 次最適化問題と等価になる。

$$\begin{aligned}
 \text{最小化} \quad & [\nabla_{\mathbf{y}} F(\bar{\mathbf{y}}, \bar{\mathbf{x}}) \mathbf{y}'] \mathbf{x}' + \frac{1}{2} \sum_{(i,j) \in A} \frac{\partial F_{ij}(\bar{\mathbf{y}}, \bar{\mathbf{x}})}{\partial x_{ij}} (x'_{ij})^2 \\
 \text{条件} \quad & \mathbf{x}' \in \text{System} \\
 & x'_{ij} = \sum_{p \in P^k} \delta_{ijp}^k z_p^k \quad (i,j) \in A, k \in K \\
 z_p^k & \begin{cases} = \text{free} & \text{if } \bar{z}_p^k > 0 \\ \geq 0 & \text{if } \bar{z}_p^k = 0, c_p(\bar{\mathbf{y}}, \bar{\mathbf{z}}) = \bar{\pi}^k \quad p \in P, k \in K \\ = 0 & \text{if } \bar{z}_p^k = 0, c_p(\bar{\mathbf{y}}, \bar{\mathbf{z}}) > \bar{\pi}^k \quad p \in P, k \in K \end{cases}
 \end{aligned}$$

ここで、 z_p^k はパスフローの摂動、 \bar{z}_p^k は現在のパスフロー、 $\bar{\pi}_k$ は品種 k の始点・終点間の最短費用、 $c_p(\bar{\mathbf{y}}, \bar{\mathbf{z}})$ はパス p の費用である。最後の制約式は、パスフローの摂動の範囲を定めるものであり、 p が最短パスかつフローが正であれば制約なし、 p が最短パスかつフローが0であれば非負、 p が最短パスでなければ0であることを表す。

この問題は、凸2次最適化問題であるため、Frank-Wolfe法などを用いて解くことができる。また、 \mathbf{y}' が与えられたときに、この問題を解くことによって現在の解における \mathbf{y}' 方向への降下方向を求めることができる。

4 タブー探索法

タブー探索法は、過去の探索情報が含まれている短期メモリや長期メモリを用いて探索範囲を限定し、必ずしも最良解ではない解の移動を許す探索法である。本来、タブー探索法は離散変数の問題を対象としているため、連続変数の問題に適用するために若干の変更が必要である。そこで、変数の内、1つのアーク (p, q) に対するデザイン変数 y_{pq} のみについて可変のステップサイズ分だけ増減した近傍解の探索を行うことによって、連続変数に対処する。

$$y'_{ij} = \begin{cases} [y_{ij}^l \pm s^l]_{L_{ij}}^{U_{ij}} & \text{if } (i,j) = (p,q) \\ y_{ij}^l & \text{otherwise} \end{cases} \quad (i,j) \in A$$

ここで、 \mathbf{y}^l は l 回目の探索における解、 \mathbf{y}' は近傍の探索解、 s^l は l 回目の探索におけるステップサイズである。

片山のタブサーチ法[16]では、はじめにステップサイズを固定して探索を行い、改良解が得られなくなったときにステップサイズを減少させ再度探索を行うという、2種類のステップサイズを用いた探索法を提案している。本研究では、一定回数の繰返しごとに、ステップサイズを減少させることにする。

短期メモリに変化させたデザイン変数を一定回数記憶しておき、短期メモリ内に記憶されているデザイン変数の変更は禁止する。解の巡回を避けるために、短期メモリに記憶する期間は“定数+乱数”の形で設定する。

タブー探索法

- ステップ1 短期メモリを SM , 繰返し回数を $MaxIte$, 増減のステップサイズの初期値を s^l , ステップサイズの変更周期を ST , 繰返し回数を $MaxIte$, SM における記憶期間の定数部を P とする。デザイン変数の初期値を y^l , ステップサイズの減少率のパラメータを sd とし, 現在の繰返し回数を $l:=1$, 上限値を $UB:=\infty$ とする。
- ステップ2 すべてのデザイン変数 $(p, q) \in A$ について, $(p, q) \notin SM$ であれば, $y'_{pq} := [y_{pq}^l \pm s^l]_{L_{pq}^{Upq}}$, それ以外の (i, j) に対して $y'_{ij} := y_{ij}^l$ とし, 利用者均衡フロー問題を解き, 目的関数値 UB_{pq} を求める。
- ステップ3 UB_{pq} が最小となる (p^*, q^*) に対して, 最小となるのがデザイン変数増加の場合は $y_{p^*q^*}^{l+1} := \min(U_{pq^*} y_{p^*q^*}^l + s^l)$ とし, 減少の場合は $y_{p^*q^*}^{l+1} := \max(L_{pq^*} y_{p^*q^*}^l - s^l)$ とする。それ以外の (i, j) に対して $y_{ij}^{l+1} := y_{ij}^l$ とする。
- ステップ4 y^{l+1} を用いて利用者均衡フロー問題を解き, $UB_{p^*q^*}$ を求める。 $UB_{p^*q^*} < UB$ であれば, $UB := UB_{p^*q^*}$ とする。
- ステップ5 (p^*, q^*) を SM に $P+rnd$ 期間記憶し, SM を更新する。
- ステップ6 $l \text{ Mod } ST = 0$ であれば $s^{l+1} := s^l \times sd$, そうでなければ $s^{l+1} := s^l$ とする。
- ステップ7 $l = MaxIte$ であれば終了, そうでなければ $l := l+1$ としてステップ2へ戻る。

ここで, rnd は適当な正数の乱数値である。

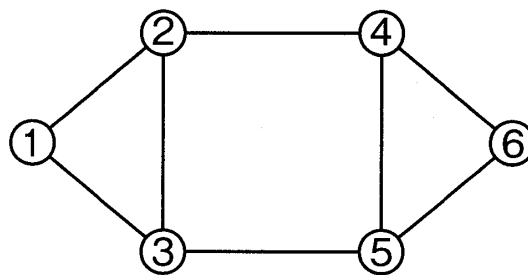


図1 : 6 ノードネットワーク

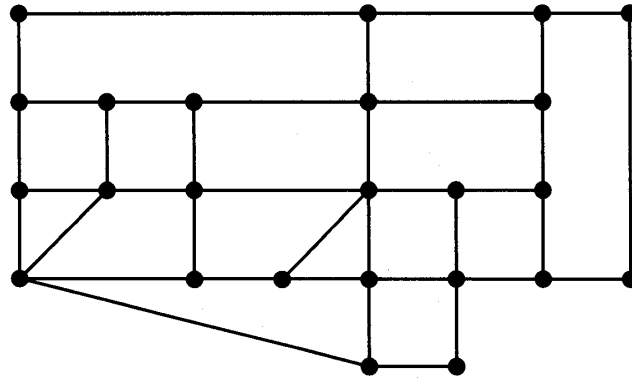


図2 : Sioux Falls ネットワーク

5 解法の比較

ベンチマーク問題として用いられている6ノードネットワークと24ノードのSioux Falls ネットワークを用いて、各解法を比較する。

6ノードネットワークは、図1に示すような6ノード、16アーク、16デザイン変数、2品種のネットワークである。フロー費用関数はBPR型、デザイン費用は線形関数であり、次式を用いる。

$$\sum_{(i,j) \in A} \left\{ a_{ij} + b_{ij} \left(\frac{x_{ij}}{C_{ij} + y_{ij}} \right)^4 x_{ij} \right\} + \sum_{(i,j) \in A} f_{ij} y_{ij}$$

なお、(1,6)の需要量5、(6,1)の10である問題と、(1,6)の需要量10、(6,1)の20である問題の2種類がある。

Sioux Fallsネットワークは、図2に示すような24ノード、76アーク、10デザイン変数、552品種のネットワークである。フロー費用関数はBPR型、デザイン費用はデザイン変数の2乗に比例し、次式を用いる。

$$\sum_{(i,j) \in A} \left\{ a_{ij} + b_{ij} \left(\frac{x_{ij}}{C_{ij} + y_{ij}} \right)^4 x_{ij} \right\} + \sum_{(i,j) \in A} f_{ij} y_{ij}^2$$

その他の詳細なデータはSuwansirikulらの研究[11]を参照のこと。

Hooke-Jeeves法 (HJ)、均衡分解最適化法 (EDO)、Friesz-Tobin-Cho-Mehtaの感度分析法 (FTCM)、Josefsson-Patrikssonの感度分析法 (JP)、拡張ラグランジュ法 (AL)、アニーリング法 (AN)、タブー検索法 (TAB1, TAB2) の結果を併記する。ここで、TAB1は片山 [16] の研究、TAB2は今回リファインしたタブサーチ法である。

TAB2で使用した条件は次の通りである。なお、順に6ノードネットワーク (需要 (5, 10)), 6ノードネットワーク (需要 (10, 20)), Sioux Fallsネットワークに対する数値である。

デザイン変数の初期値 y^1 : すべて 0, すべて 0, すべて 4

初期ステップサイズ s^1 : 1, 1, 0.5

短期メモリの記憶期間 $P+rnd$: 4~8, 5~10, 3~6

ステップサイズの変更周期 ST : 100, 100, 100

ステップサイズの減少率 sd : 0.95, 0.95, 0.85

繰返し回数 $MaxIte$: 1000000, 1000000, 100000

表1に6ノードネットワーク (需要 (5, 10)), 表2に6ノードネットワーク (需要 (10, 20)), 表3にSioux Fallsネットワークに対する近似解を示す。なお, 表内に表示していない y_{ij} はすべて0である。

表内の $\phi(\text{rep})$ は, オリジナルの論文に掲載されている上界値・目的関数値である。 $\phi(\text{JB})$ はJosefsson-Patriksson [9]が, $\phi(\text{K})$ は本論文において, 各論文で報告されたデザイン変数値を用いて目的関数値を算出したものである。JPとTAB1・2を除く $\phi(\text{rep})$ では, 最終的な利用者均衡フロー問題を射影法を用いて解き, 射影法における2次計画問題をFrank-Wolfe法で解いている。また, Josefsson-Patrikssonは非集約単純分解法, 本論文では射影法における2次計画問題を数理計画ソフトウェアであるCPLEXを用いて解いている。

6ノードネットワーク (需要 (5, 10)) において, 目的関数値が200以下 ($\phi(\text{K})$) の解を算出した解法は, Josefsson-Patrikssonの感度分析法, Friesz-Tobin-Cho-Mehtaの感度分析法と2種類のタブサーチ法である。4つの解法のデザイン変数値は比較的近い値となっている。また, Josefsson-Patrikssonによる目的関数値と本論文による目的関数値はほぼ一致している。

6ノードネットワーク (需要 (10, 20)) において, 目的関数値が523以下 ($\phi(\text{K})$) の解を算出した解法は, Josefsson-Patrikssonの感度分析法と2種類のタブサーチ法である。3つの解法のデザイン変数値は比較的近い値となっている。また, Josefsson-Patrikssonによる目的関数値と本論文による目的関数値はほぼ一致している。

Sioux Fallsノードネットワークにおいて, 目的関数値が80.3以下 ($\phi(\text{K})$) の解を算出した解法は, Josefsson-Patrikssonの感度分析法と2種類のタブサーチ法である。3つの解法のデザイン変数値は比較的近い値となっている。感度分析法の解に対するJosefsson-Patrikssonによる目的関数値 ($\phi(\text{JP})$) は79.9961と80以下となっているが, 今回の実験では80.2226となり, 残念ながら再現することはできなかった。一方, TAB2では最小値となる80.1440を算出することができた。

表1：6ノードネットワーク（需要(5, 10)）の近似解の比較

	HJ	EDO	FTCM	JP	AL	AN	TAB1	TAB2
$y_{2,1}$	1.20	0.13	0.00	0.00	0.0062	0.00	0.00	0.00
$y_{3,1}$	3.00	6.26	4.85	5.1945	5.2631	3.1639	5.48	5.1967
$y_{3,2}$	0.00	0.00	0.00	0.00	0.0032	0.00	0.00	0.00
$y_{5,3}$	0.00	0.00	0.00	0.00	0.0064	0.00	0.00	0.00
$y_{6,4}$	3.00	0.13	0.00	0.00	0.7171	0.00	0.00	0.00
$y_{6,5}$	2.80	6.26	6.48	7.5964	6.7561	6.724	7.44	7.5980
$\phi(\text{rep})$	215.08	201.84	199.94	199.6253	202.9913	198.1038	199.651	199.6253
$\phi(\text{JP})$	218.20	201.20	-	199.6253	-	201.3358	-	-
$\phi(\text{K})$	218.1952	201.1956	199.9649	199.6253	202.9962	201.3358	199.6489	199.6253

表2：6ノードネットワーク（需要10, 20)）の近似解の比較

	HJ	EDO	FTCM	JP	AL	AN	TAB1	TAB2
$y_{1,3}$	5.40	4.88	4.63	4.6144	4.6153	0.00	4.68	4.5977
$y_{2,1}$	8.18	8.59	7.80	9.9102	9.8804	10.174	9.92	9.9105
$y_{3,1}$	8.10	7.48	12.72	7.3745	7.5995	5.7769	7.28	7.3810
$y_{3,2}$	0.00	0.26	0.00	0.00	0.0016	0.00	0.00	0.00
$y_{3,5}$	0.90	0.85	0.65	0.5922	0.6001	0.00	0.64	0.5785
$y_{4,2}$	0.00	0.00	0.00	0.00	0.0010	0.00	0.00	0.00
$y_{5,3}$	0.00	0.00	1.81	0.00	0.1130	0.00	0.00	0.00
$y_{5,6}$	3.90	1.54	1.85	1.3153	1.3184	0.00	1.24	1.3067
$y_{6,4}$	8.10	0.26	16.96	0.00	2.7265	0.00	0.00	0.00
$y_{6,5}$	8.40	12.52	6.21	20.7661	17.5774	17.2786	20.80	20.7753
$\phi(\text{rep})$	557.22	540.74	563.64	522.5826	532.7100	528.497	522.593	522.5824
$\phi(\text{JP})$	564.49	540.20	-	522.5826	-	533.329	-	-
$\phi(\text{K})$	561.4879	540.1990	563.8365	522.5824	532.6895	533.3291	522.5901	522.5824

表3：Sioux Fallsネットワークの近似解の比較

	HJ	EDO	FTCM	JP	AL	AN	TAB1	TAB2
$y_{6,8}$	4.80	4.59	5.25	5.3027	5.5728	5.38	5.16	5.3663
$y_{7,8}$	1.20	1.52	1.73	2.0560	1.6343	2.26	1.93	1.7817
$y_{8,6}$	4.80	5.45	5.28	5.3430	5.6228	5.50	5.19	5.2808
$y_{8,7}$	0.80	2.33	1.60	1.9901	1.6443	2.01	1.55	2.2269
$y_{9,10}$	2.00	1.27	2.79	2.5216	3.1437	2.64	2.42	2.7234
$y_{10,9}$	2.60	2.33	2.86	2.5548	3.2837	2.47	2.85	2.5935
$y_{10,16}$	4.80	0.41	4.58	2.9883	7.6519	4.54	3.21	3.3457
$y_{13,24}$	4.40	4.59	4.40	4.8559	3.8035	4.45	4.80	4.9881
$y_{16,10}$	4.80	2.71	4.59	3.0026	7.3820	4.21	3.08	3.2898
$y_{24,13}$	4.40	2.71	4.30	4.8496	3.6935	4.67	4.91	4.8788
$\phi(\text{rep})$	80.78	83.08	82.91	79.9961	81.752	80.87	80.74	80.1440
$\phi(\text{JP})$	80.67	82.34	-	79.9961	-	-	-	-
$\phi(\text{K})$	80.7492	82.6255	80.5688	80.2226	82.6671	80.4526	80.2094	80.1440

6 おわりに

本論文では、従来の近似解法を調査するとともに、タブサーチ法をリファインした解法を示した。さらに、一般的に用いられている3つのベンチマーク問題を用いて、主な従来の解法によって得られる近似解を比較した。

片山[16]の研究で得られた解は従来の結果と比較して、非常に優れたものであったが、解が良すぎるために利用者均衡フロー問題を解く際の何らかのミスの可能性が残されていた。しかし、2005年に発表されたJosefsson-Patrikssonによる解と片山の解は非常に類似しており、算出した目的関数値も近いこと、また、今回の実験結果でも同様の値を算出したことから、片山の研究結果は妥当であると考ええる。

この研究は文部科学省科学研究費（課題番号16510119）の助成を受けた成果の一部である。

参考文献

- [1] M. Abdulaal and L. LeBlanc. Continuous equilibrium network design models. *Transportation Research B*, Vol. 13, pp. 19-32, 1979.
- [2] J. Bard. An efficient point algorithm for a linear two-stage optimization problem. *Operations Research*, Vol. 31, pp. 670-684, 1983.
- [3] T. Friesz, H. Cho, N. Mehta, R. Tobin, and G. Anandalingam. A simulated annealing approach to the network design problem with variational inequality constraints. *Transportation Science*, Vol. 26, pp. 18-26, 1992.
- [4] T. Friesz, R. Tobin, H. Cho, and N. Mehta. Sensitivity analysis based heuristic algorithms for mathematical programs with variational inequality constraints. *Mathematical Programming*, Vol. 48, pp. 265-284, 1990.
- [5] P. Marcotte. Network optimization with continues control parameters. *Transportation Science*, Vol. 17, pp. 181-197, 1983.
- [6] P. Marcotte. Network design problem with congestion effects: A case of bilevel programming. *Mathematical Programming*, Vol. 34, pp. 142-162, 1986.
- [7] P. Marcotte. A note on a bilevel programming algorithm by LeBlanc and Boyce. *Transportation Research B*, Vol. 22, pp. 233-237, 1988.
- [8] Q. Meng, H. Yang, and M. Bell. An equivalent continuously differentiable model and a locally convergent algorithm for the continuous network design problem. *Transportation Research B*, Vol. 35, pp. 83-105, 2001.
- [9] M. Josefsson and M. Patriksson. Sensitivity analysis of separable traffic equilibria, with application to bilevel optimization in network design. *Transportation Science B*, to appear.

- [10] S. Suh and T. Kim. Solving nonlinear bilevel programming models of the equilibrium network design problem: A comparative review. *Annals of Operations Research*, Vol. 34, pp. 203-218, 1992.
- [11] C. Suwansirikul, T. Friesz, and R. Tobin. Equilibrium decomposed optimization :A heuristic for the continuous equilibrium network design problem. *Transportation Science*, Vol. 21, pp. 254-263, 1987.
- [12] R. Tobin and T. Friesz. Sensitivity analysis for equilibrium network flow. *Transportation Science*, Vol. 22, pp. 242-250, 1988.
- [13] 福島雅夫. 非線形最適化の理論. 産業図書, 1980.
- [14] 今野浩, 山下浩. 非線形計画法. 日科技連, 1978.
- [15] 朝倉康夫. 利用者均衡に基づく最適道路ネットワーク形成. 交通科学, Vol. 13, pp. 21-27, 1984.
- [16] 片山直登. 連続型ネットワークデザイン問題の近似解法. 流通経済大学流通情報学部紀要, Vol. 3, pp. 1-14, 1999.