《論文》

利用者均衡条件をもつ連続型ネットワークデザイン問題の近似解法

片　山　直　登

1．はじめに

ネットワークデザイン問題は，ネットワークと品種(ODペア間)のODフロー量が与えられたときに，ルーティング費用（走行費用・時間）とアークの設定やアーク容量を変更するためのデザイン費用（建設費用）の和を最小にするネットワークの形状やアーク容量，およびネットワーク上のフローを求める問題である。交通ネットワークの計画者と利用者（車両）のように計画者と利用者が異なり，計画者が利用者を必ずしも制御できない場合がある。このような場合には，一般的に次のような前提が置かれることが多い。

(1) ネットワーク計画者は，ルーティング費用とデザイン費用の総和を最小にするネットワークの形状を選択する。

(2) 各ネットワーク利用者は，ルーティング費用が最小となる経路を選択する。

(1)の前提はシステム最適化またはWardlopの第二原理，(2)の前提は利用者均衡またはWardlopの第一原理として知られている。

(1)と(2)を同時に考慮すると，ルーティング費用がフロー量の線形関数である問題は，組合せ最適化問題となる。一方，ルーティング費用が凸非線形関数である問題は，利用者均衡条件を考慮する必要がある。利用者均衡条件自体が最適化問題として表現されることから，利用者均衡問題を制約条件にもつ2レベルの最適化問題となる。この問題は，利用者均衡条件をもつネットワークデザイン問題とよばれている。

ネットワークデザイン問題は，デザイン変数が0-1離散値を取るものと連続値を取るものに分類され，前者は離散型ネットワークデザイン問題，後者は連続型ネットワークデザイン問題とよばれている。

本研究では，デザイン変数であるアーク容量の拡張量が連続値をとり，利用者均衡条

件下のもとで，ルーティング費用とデザイン費用の総和を最小にするような利用者均衡条件をもつ連続型ネットワークデザイン問題(Continuous Equilibrium Network Design Problem: *CEND*)に対する2種類の近似解法を提案する。第一の解法は，ルーティング費用関数がBPR関数に限定された連続型ネットワークデザイン問題に対して，修正目的関数を用いて2レベル問題を1レベルの最適化問題に変換し，この問題の収束解を求める近似解法である。第二の解法は，デザイン変数に対してタブサーチ(tabu serach)法を適用した近似解法である。最後に数値実験を行ない，提案した2つの近似解法と従来の解法を比較し，提案した近似解法を有効性を示す。

2．従来の研究

問題 *CEND* は制約条件に最適化問題を含むため，実行可能領域が非凸となる。このため，一部の離散型の場合を除けば，数値例程度の小規模な問題ですら最適解を求めることが困難である。離散型ネットワークデザイン問題に対しては，LeBlancの分枝限定法[9]，Hoangの一般ベンダース分解原理[8]，Poorzahedyの近似的な分枝限定法[16]などの解法が提案されている。

一方，連続型ネットワークデザイン問題に対しても数多くの研究が行なわれている。分解・降下法タイプの解法では，Steenbrink[17]，Los[12]，Dantzig et al[3]，朝倉[21]，Suwansirikul et al[18]，佐佐木-朝倉[20]，Friesz et al[6]などの研究が行なわれている。また，Abdulaal-LeBlanc[1]やLeBlanc-Abdulaal[10]のローカルサーチ法，Marcotteの変分不等式緩和法[14]，LeBlanc-Boyceの線形結合法[11]，Friesz et alのアニーリング法[15]などの研究も行なわれている，Marcotte[15]は問題の性質の検討やワーストケース解析などを行なっている。さらに，Boyce[2]，Magnanti-Wong[13]，Friesz[4]，朝倉[22]などがサーベイを行なっている。

一方，近年，タブサーチ法，アニーリング法や遺伝的アルゴリズムなどのメタ解法とよばれる解法が開発され，複雑な組合せ最適化問題に対して適用され，数多くの成果を挙げている。特に，Gloverにより開発されたタブサーチ法[7),19]は，バリエーションも多く，メタ解法の中でも最も効果的な解法として知られている。

3．問題の定式化

アーク集合を A，品種の集合を M とする。ここで，品種とは始点と終点が同一であるODフローを表す。アーク (i,j) の拡張する容量を表すデザイン変数を y_{ij}，アーク (i,j) 上のフローの合計量を表すフロー変数を x_{ij} とする。品種 m の始点・終点間の取りうるパスの集合を P^m，品種 m のパス h 上のフロー量を表す変数を z_h^m，品種 m のパス h がアー

ク (i,j) を含むか否かを表す0-1定数を δ_{ijh}^m とおく。また，品種 m の OD フロー量を d^m とする。

ルーティング費用はアーク上のフロー変数とデザイン変数によって決定されるため，単位フロー当りのルーティング費用関数を $c_{ij}(x_{ij},y_{ij})$ とおく。デザイン費用はアーク上のデザイン変数に依存するため，デザイン費用関数を $f_{ij}(y_{ij})$ とおく。このとき，問題 *CEND* は次のように定式化できる。

$$\text{最小化}\quad \Phi_{cend}=\sum_{(i,j)\in A}c_{ij}(x_{ij},y_{ij})\,x_{ij}+\sum_{(i,j)\in A}f_{ij}(y_{ij}) \tag{1}$$

$$\text{条件}\quad y_{ij}\geq 0\quad for\ all\ (i,j)\in A \tag{2}$$

$$\text{最小化}\quad \sum_{(i,j)\in A}\int_0^{x_{ij}}c_{ij}(t,y_{ij})\,dt \tag{3}$$

$$\text{条件}\quad \sum_{h\in P^m}z_h^m=d^m\quad for\ all\ m\in M \tag{4}$$

$$x_{ij}=\sum_{m\in M}\sum_{h\in P^m}\delta_{ijh}^m z_h^m\quad for\ all\ (i,j)\in A \tag{5}$$

$$z_h^m\geq 0\quad for\ all\ h\in P^m,\ m\in M \tag{6}$$

(1)式はルーティング費用とデザイン費用の総和の最小化を表す目的関数である。(2)式はデザイン変数が非負であることを表す。(3)～(6)式は，利用者均衡条件を表す。(3)式は，ルーティング費用関数の積分値の総和の最小化であり，利用者均衡条件の目的関数を表す。(4)式は，品種 m のパスフロー量と OD フロー量の関係を表す。(5)式は，アーク (i,j) 上を流れるパスフロー変数とフロー変数の関係式である。(6)式は，パスフロー変数が非負であることを表す。

以上のように，問題 *CEND* は制約条件に最適化問題を含む問題であり，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。一方，デザイン変数を決定すれば，フロー変数を求める問題は利用者均衡問題そのものとなり，Frank-Wolfe 法や Projection 法によって収束は遅いが比較的容易にフロー解を求めることができる。このため，デザイン変数によってフロー変数は陰的に決定されるものと考えることができる。

そこで，デザイン変数が与えられたときに利用者均衡フローを与える関数を $user(\boldsymbol{y})$ とおくと，問題 *CEND* を次のような定式化で表すことができる。ここで，$\boldsymbol{x}$ と $\boldsymbol{y}$ はフロー変数とデザイン変数のベクトルである。

$$\text{最小化}\quad \Phi_{cend}(\boldsymbol{x},\boldsymbol{y})=\sum_{(i,j)\in A}c_{ij}(x_{ij},y_{ij})\,x_{ij}+\sum_{(i,j)\in A}f_{ij}(y_{ij}) \tag{7}$$

条件　$y_{ij} \geq 0 \quad for\ all\ (i,j) \in A$ (8)

$$\boldsymbol{x} = user(\boldsymbol{y}) \tag{9}$$

4．目的関数修正法

4.1　目的関数修正ネットワークデザイン問題とその性質

この節では，ルーティング費用関数がBPR関数に限定された連続型ネットワークデザイン問題に対して，修正目的関数を用いて2レベル問題を1レベルの最適化問題に変換し，この問題の収束解を求める近似解法を示す。

BPR関数は，次のように表される。

$$c_{ij}(x_{ij}, y_{ij}) = t_{ij}\left\{1+\alpha\left(\frac{x_{ij}}{C_{ij}+y_{ij}}\right)^{\beta}\right\} \tag{10}$$

ここで，t_{ij} は0フロー時のアーク (i,j) のルーティング費用，C_{ij} はアーク (i,j) の既存の容量である。この容量は上限値ではなく，費用の増加量のパラメータを表す。また，α と β は，$\alpha \geq 0$，$\beta \geq 1$ である定数である。

ここで，次のような問題 *MCND* (Modified Continuous Network Design Problem) を考える。

$$\begin{aligned} \text{最小化}\quad \Phi_{mcnd}(\boldsymbol{x},\boldsymbol{y}) &= \sum_{(i,j)\in A}\int_0^{x_{ij}} c_{ij}(t, y_{ij})\,dt + \sum_{(i,j)\in A}\frac{f_{ij}(y_{ij})}{\beta+1} \\ &= \sum_{(i,j)\in A}\left[t_{ij}\left\{1+\frac{\alpha}{\beta+1}\left(\frac{x_{ij}}{C_{ij}+y_{ij}}\right)^{\beta}\right\}x_{ij} + \frac{f_{ij}(y_{ij})}{\beta+1}\right] \end{aligned} \tag{11}$$

条件　(2)，(4)～(6)式 (12)

問題 *MCND* が問題 *CEND* と異なる点は，以下の2点である。利用者均衡条件の目的関数(3)式を取り除く。利用者均衡条件の目的関数(3)式に $1/(\beta+1)$ 倍したデザイン費用関数を加えたものを目的関数とする。以上の変換によって，問題 *MCND* は制約式が線形である非線形計画問題となる。

一般に，非線形計画問題の解法には，目的関数の1次微分や2次微分が用いられる。そこで，問題 *MCND* と問題 *CEND* の目的関数の微分を比較する。

まず，問題 *MCND* の目的関数(11)式を y_{ij} で偏微分する。

$$\frac{\partial}{\partial y_{ij}}\Phi_{mcnd}(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{\beta+1}\left\{-t_{ij}\alpha\beta\left(\frac{x_{ij}}{C_{ij}+y_{ij}}\right)^{\beta+1} + \frac{\partial}{\partial y_{ij}}f_{ij}(y_{ij})\right\} \tag{13}$$

一方，同様に問題 *CEND* の目的関数(1)式を y_{ij} で偏微分する。

$$\frac{\partial}{\partial y_{ij}}\Phi_{cend}(\boldsymbol{x},\boldsymbol{y})=-t_{ij}\alpha\beta\left(\frac{x_{ij}}{C_{ij}+y_{ij}}\right)^{\beta+1}+\frac{\partial}{\partial y_{ij}}f_{ij}(y_{ij}) \tag{14}$$

(13)式と(14)式を比較すると，$1/(\beta+1)$ 倍の定数倍だけ異なる。明らかに2つの目的関数の $\boldsymbol{y}$ による2回微分も定数倍だけ異なる。このため，1次微分や2次微分を用いた降下法を用いた場合には，2つの問題の $\boldsymbol{y}$ に対する降下方向が一致することになる。したがって，問題 *MCND* の目的関数値の $\boldsymbol{y}$ に対する降下方向は，問題 *MCND* の目的関数値を減少させるだけでなく，問題 *CEND* の目的関数値を減少させることが期待できる。

次に，問題 *MCND* の目的関数を x_{ij} で偏微分する。

$$\frac{\partial}{\partial x_{ij}}\Phi_{mcnd}(\boldsymbol{x},\boldsymbol{y})=c_{ij}(x_{ij},y_{ij}) \tag{15}$$

同様に，問題 *CEND* の利用者均衡条件の目的関数(3)式を x_{ij} で偏微分すると

$$\frac{\partial}{\partial x_{ij}}\Phi_{cend}(\boldsymbol{x},\boldsymbol{y})=c_{ij}(x_{ij},y_{ij}) \tag{16}$$

となり，(15)式に一致する。このため，問題 *MCND* の $\boldsymbol{x}$ に対する降下方向と問題 *CEND* の利用者均衡条件の目的関数の $\boldsymbol{x}$ に対する降下方向も一致することになる。したがって，問題 *MCND* の目的関数値の $\boldsymbol{x}$ に対する降下方向は，問題 *MCND* の目的関数値を減少させるだけでなく，利用者均衡条件の目的関数値を減少させる，すなわち利用者均衡フロー解が求まることが期待できる。

以上のことから，問題 *MCND* を降下法などで解くことにより，問題 *CEND* の目的関数値を減少させ，しかも利用者均衡フロー解が求まることが期待できる。

4.2 問題 *MCND* の解法

問題 *MCND* は2種類の変数をもつ最適化問題であるので，デザイン変数の最適化問題とフロー変数の最適化問題に分離し，交互に降下法的に解くことを考える。

k 回目の繰返しにおいて，フロー変数ベクトルが $\boldsymbol{x}^k$ であるとする。このとき，デザイン変数の最適化問題はアーク毎の問題に分離でき，次のように表される。

$$\text{最小化}\quad \frac{t_{ij}\alpha}{\beta+1}\left(\frac{x_{ij}^k}{c_{ij}+y_{ij}}\right)^{\beta}x_{ij}^k+\frac{f_{ij}(y_{ij})}{\beta+1} \tag{17}$$

$$\text{条件}\quad y_{ij}\geq 0 \tag{18}$$

この問題は1変数 y_{ij} の最小化問題ではあるが，目的関数は y_{ij} の凸関数とは限らな

い。しかし，微分可能であれば k 回目の繰返しにおける $(\boldsymbol{x}^k, \boldsymbol{y}^k)$ から，目的関数の降下方向に微小距離だけ移動すれば，次のデザイン変数が求まり，目的関数値を減少させることができる。

一方，フロー変数の最適化問題は利用者均衡問題となり，Frank-Wolfe 法や Projection 法を用いて利用者均衡フロー解を求めることができる。

ここで，降下方向 d_{ij}^k の例を示す。

$$d_{ij}^k = t_{ij}\alpha\beta\left(\frac{x_{ij}^k}{C_{ij}+y_{ij}^k}\right)^{\beta+1} - \frac{\partial}{\partial y_{ij}^k} f_{ij}(y_{ij}^k) \tag{19}$$

全体的な解法の手順を示しておく。

・修正目的関数法の手順

[1] 各アークのデザイン変数の初期値 $\boldsymbol{y}^1$ を設定する。k:=1 とする。

[2] デザイン変数を $\boldsymbol{y}^k$ として，利用者均衡問題を解き，利用者均衡フロー解 $\boldsymbol{x}^k$ を求める。

[3] $(\boldsymbol{x}^k, \boldsymbol{y}^k)$ において，$\boldsymbol{y}$ の降下方向ベクトル $\boldsymbol{d}^k$ を求め，適当な微小ステップサイズ λ を用いて，デザイン変数を更新する。

y_{ij}^{k+1}:=max$(0, y_{ij}^k+\lambda d_{ij}^k)$

k:=k+1

[4] 解が収束すれば終了し，そうでなければ[2]へ戻る。

ステップ[1]は，デザイン変数の初期設定である。ステップ[2]では，現在のデザイン変数おける利用者均衡条件を満足するフロー解を求めている。ステップ[3]では，k 回目の試行解から次の $\boldsymbol{y}$ の試行解 $\boldsymbol{y}^{k+1}$ を求めており，y_{ij} が負にならないように制限を加えている。ステップ[4]では，収束判定を行なっている。

問題 *MCND* の制約式は線形であるので，解空間は凸コンパクト集合となる。このアルゴリズムでは，ステップ[2]，[3]で目的関数値を減少（非増加）させる。したがって，このアルゴリズムは収束解を生成する。

次に，解法の手順によって収束解 $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ が得られたものとする。ステップ[2]において利用者均衡フロー解を求めているため，収束解 $\boldsymbol{x}^*$ は利用者均衡条件，すなわち(3)～(6)式を満足する。一方，ステップ[3]より $\boldsymbol{y}^*$ は非負であるので，(2)式を満足する。したがって，収束解 $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ は問題 *CEND* のすべての制約を満足するので，問題 *CEND* の実行可能解となる。さらに，$(\boldsymbol{x}^*, \boldsymbol{y}^*)$ を(1)式に代入すれば，問題 *CEND* の目的関数値

を求めることができる。

以上のことから，提案した解法によって得られた収束解は問題 *CEND* の近似解となり，提案した解法は問題 *CEND* の近似解法となる。

5．タブサーチ法

5.1 タブサーチ法

タブサーチ法は，組合せ最適化問題などに適用されている近似解法である。一般的なタブサーチ法は，探索解をタブリストに記憶し，タブリストによって禁止されない近傍解の中で最良の近傍解を探索していく方法である。また，変更した変数に対して一定回数の変更を禁止する短期メモリや，変更した回数に応じて目的関数にペナルティを加える長期メモリ[19]などがある。これらを組合せたタブーサーチ法を用いることによって，局所最適解から脱出でき，広範囲の実行可能領域を効率的に探索することができる。

問題 *CEND* の実行可能領域は非凸であるため，局所最適解が多数存在し，一般的な降下法では局所最適解に収束してしまう。このため，タブサーチ法を適用すれば，局所最適解から脱出し，広範囲の実行可能領域を効率的に探索できる可能性がある。タブリストによる方法，短期メモリによる方法，長期メモリによる方法などがあるが，予備実験の結果より問題 *CEND* に対して短期メモリが最も有効であったため，ここでは短期メモリを用いたタブサーチ法を中心に記述する。

5.2 近傍解の定義と目的関数値の評価

タブサーチ法では限定された領域において近傍探索を行なうため，近傍の定義と近傍解の目的関数値の評価法を明確化しなければならない。

定式化の節で示したように，デザイン変数を決定すればフロー変数を求める問題は利用者均衡問題そのものとなり，デザイン変数によってフロー変数は陰的に決定される。このため，探索禁止の対象とする変数はデザイン変数とする。デザイン変数は連続値をとるため，タブサーチ法に適用できるようにデザイン変数を適当な間隔の離散値として扱うことにする。

ここでは，１つのデザイン変数毎に独立に変数値を離散的に増加・減少した範囲を近傍と定義する。この近傍探索を行うためには，$|A|\times 2$（増加･減少）回の目的関数値の評価を行なう必要がある。デザイン変数値を変化させたときに目的関数値を評価するためには，フロー変数値を陽的に求める必要がある。フロー変数値を厳密に求めるには，１試行解に対して $2|A|$ 回の利用者均衡問題を解く必要があり，多くの計算時間を必要とする。このため，デザイン変数値が変化したときに，フロー変数値を厳密的・近似的に求める方法がいくつか考えられる。ここで，k 回目の繰返しにおけるフロー変数ベクトルを

$\boldsymbol{x}^k$，デザイン変数ベクトルを $\boldsymbol{y}^k$ とする。

(1) $\bar{\boldsymbol{x}}:=\boldsymbol{x}^k$, すなわちフロー変数は変化しないものとして，$\Phi_{cend}(\bar{\boldsymbol{x}}, \boldsymbol{y}^k)$ を目的関数値の評価値とする。

(2) 初期フロー解 $:=\boldsymbol{x}^k$ として，$\boldsymbol{y}^k$ である場合の利用者均衡フロー解 $\bar{\boldsymbol{x}}$ を求め，$\Phi_{cend}(\bar{\boldsymbol{x}}, \boldsymbol{y}^k)$ を目的関数値の評価値とする。

(3) 初期フロー解 $:=$ *All or Nothing* フローとして，$\boldsymbol{y}^k$ である場合の利用者均衡フロー解 $\bar{\boldsymbol{x}}$ を厳密に求め，$\Phi_{cend}(\bar{\boldsymbol{x}}, \boldsymbol{y}^k)$ を目的関数値の評価値とする。

(1)の方法は，利用者均衡問題を解く必要がなく，短時間で評価値を求めることができる。(2)の方法は，現在の利用者均衡フロー解を初期値として利用者均衡フロー解を求める方法である。この場合，Frank-Wolfe 法などでは数回のイテレーションで収束解を求めることができると言われている。しかし，収束解の周辺部でジグザク現象が起こるため，初期値によって利用者均衡フロー解が微妙に異なる場合が発生する。(3)の方法は，利用者均衡フロー解を厳密的に求める方法であるが，多くの計算時間を必要とする。従来の局所探索法[1]などでは，近傍探索には(1)の方法が用いられている。

近似的に目的関数値の評価値を求めた後に，試行解 $\boldsymbol{y}^k$ を用いて，利用者均衡フロー解 $\boldsymbol{x}^k$ を更新しておく必要がある。この方法には，(2)と(3)の方法を用いることができる。従来の研究では，(2)の方法でフロー解を更新する場合が多い。

従来の研究では，近傍解の評価に(1)の方法，試行解の再評価に(2)の方法が用いられていることが多いが，近年では計算機の計算能力も向上し，より厳密な評価方法を用いても十分に計算することが可能となってきている。

5.3 探索禁止変数とタブサーチ法の手順

タブーサーチ法の短期メモリでは，変数の変更を禁止することになる。問題 *CEND* のタブサーチ法では，どのように探索を禁止するかによって，いくつかの方法が考えられる。ここに，主な方法を示しておく。

(a) 変更したデザイン変数を一定回数変更を禁止する。

(b) 増加したデザイン変数を一定回数の減少を禁止し，減少したデザイン変数を一定回数の増加を禁止する。

(c) k 回目と $k+1$ 回目に変更したデザイン変数の組に対して，この組合せの変更を一定回数禁止する。

(a)は近傍の制限が強く，(c)は近傍の制限が弱く自由度が高い方法となる。一方，変更の禁止を一定回にした場合，状況によっては探索解が巡回する可能性がある。このため，一様乱数値を用いて一定の範囲内で禁止回数を変化[19]させて巡回を防ぐことにする。

ここでは，近傍解の評価に(2)の方法，試行解の再評価に(3)の方法を用い，探索禁止には(b)の方法を用いたタブサーチ法の手順を示す。ここで，アーク (i,j) の増加禁止回数を表すメモリを T^{inc}，減少禁止回数を表すメモリを T^{dec}，それぞれのベクトルを $\boldsymbol{T}^{inc}$ と $\boldsymbol{T}^{dec}$ とする。z を目的関数値の近似値とする。

・タブサーチ法の手順

[1] $\boldsymbol{T}^{inc}=\boldsymbol{0}$，$\boldsymbol{T}^{dec}=\boldsymbol{0}$ とする。繰返し回数を U，増減のステップサイズを $size$，禁止回数を P とする。デザイン変数の初期値を $\boldsymbol{y}^1$ とし，$k:=1$，$z:=\infty$ とする。

[2] すべてのデザイン変数について $\bar{y}_{ij}:=y_{ij}^k$ とし，[2-1]，[2-2]を行なう。目的関数値の評価値の最小値を求め，そのアークを (i^*,j^*) とし，増減を記憶する。

[2-1] $T_{ij}^{inc}\leq 0$ であれば，$\bar{y}_{ij}:=y_{ij}^k+size$ とし，利用者均衡解 $\bar{\boldsymbol{x}}$ と評価値 $\Phi_{cend}(\bar{\boldsymbol{x}},\bar{\boldsymbol{y}})$ を求める。$\bar{y}_{ij}:=y_{ij}^k$ とする。

[2-2] $T_{ij}^{dec}\leq 0$ であれば，$\bar{y}_{ij}:=\max(0,y_{ij}^k-size)$ とし，利用者均衡解 $\bar{\boldsymbol{x}}$ と評価値 $\Phi_{cend}(\bar{\boldsymbol{x}},\bar{\boldsymbol{y}})$ を求める。$\bar{y}_{ij}:=y_{ij}^k$ とする。

[3] (i^*,j^*) に対して，デザイン変数を増加した場合は $y_{i^*j^*}^{k+1}:=y_{i^*j^*}^k+size$，減少した場合は $y_{i^*j^*}^k:=\max(0,y_{i^*j^*}^k-size)$ とする。それ以外の (i,j) に対して，$y_{ij}^{k+1}:=y_{ij}^k$ とする。

[4] 利用者均衡解 $\boldsymbol{x}^{k+1}$ を求め，目的関数値 $\Phi_{cend}(\boldsymbol{x}^{k+1},\boldsymbol{y}^{k+1})$ を求める。
$z>\Phi_{cend}(\boldsymbol{x}^{k+1},\boldsymbol{y}^{k+1})$ であれば，$z=\Phi_{cend}(\boldsymbol{x}^{k+1},\boldsymbol{y}^{k+1})$ とする。

[5] (i^*,j^*) において，デザイン変数を増加した場合は $T_{i^*j^*}^{dec}=P$ とし，減少した場合は $T_{i^*j^*}^{inc}=P$ とする。

[6] すべてのデザイン変数について，$T_{ij}^{inc}:=T_{ij}^{inc}-1$，$T_{ij}^{dec}:=T_{ij}^{dec}-1$ とする。

[7] $k=U$ であれば終了，そうでなければ $k:=k+1$ として[2]へ戻る。

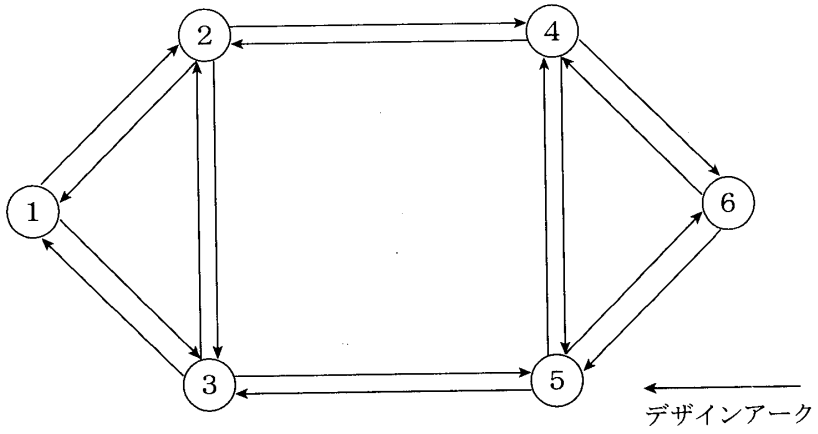

図1 ネットワーク1

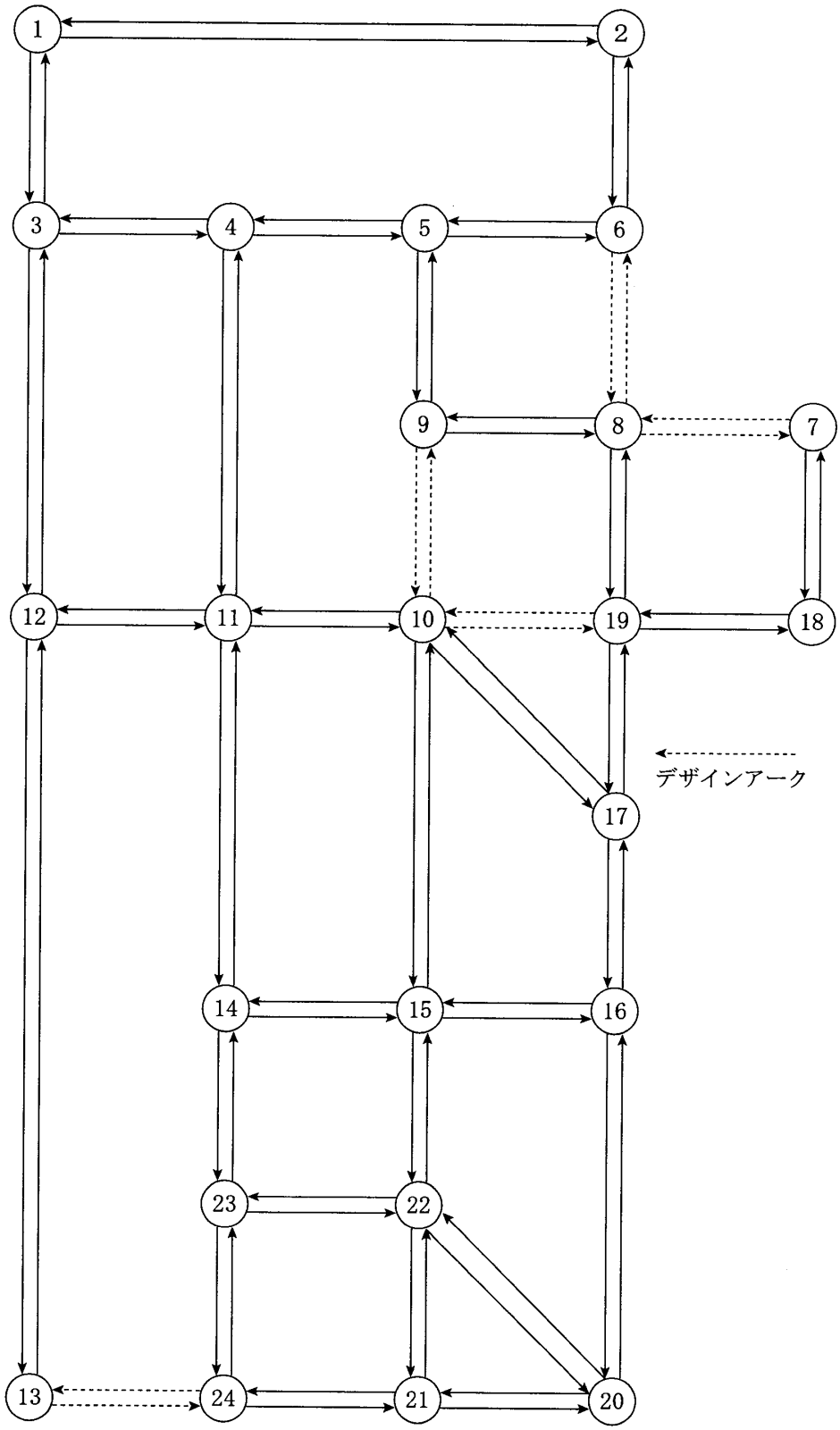

図2　ネットワーク2

6．数値実験

提案した近似解法を評価するために，2つのネットワークを用いて数値計算を行った。使用したネットワークを図1，図2に示す。

ネットワーク1は，Suwansirikul[18]，Friesz[6),5)]などが使用したものであり，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。ルーティング費用関数はBPR型，デザイン費用は線形関数である。品種(1,6)がODフロー量5.0と(6,1)が10.0であるケース1と，品種(1,6)がODフロー量10.0と(6,1)が20.0であるケース2の2通りについて計算した。

ネットワーク2は，サウスダコタ州のSioux Falls市のネットワークであり，LeBlanc[9)]，Abdulaal[1)]，Suwansirikul，Friszらが使用したものであり，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。ルーティング費用関数はBPR型，デザイン費用は定数×デザイン変数の2乗である。2つのネットワークのODフロー，BPR関数，デザイン費用関数などの詳細なデータは，Suwansirikul[18]に記述されているものを使用した。

IBM互換コンピュータ（PentiumII 266Mz）上のMicrosoft Fortran Power Station Ver.4.0コンパイラを使用して，数値計算を実施した。タブサーチ法では，禁止回数とステップサイズを変化させて計算を行い，最良のパラメータを求めた。これらのパラメータを表1に示す。禁止回数は，一様乱数を用いて一定の範囲で変化させた。計算では，ステップサイズ1で一定回数の探索を行ない，さらに最良解を初期値としてステップサイズ2でステップサイズ1の繰返し回数の10分1回の探索を行なった。

表2～4に，提案した近似解法で得られた近似解とその目的関数値Φ_{cend}を示す。

表1 タブサーチ法のパラメータ

	禁止回数	ステップサイズ1	ステップサイズ2	繰返し回数
ネットワーク1ケース1	4～5	0.4	0.04	5500
ネットワーク1ケース2	3～4	0.4	0.04	5500
ネットワーク2	20～28	0.2	0.02	55000

表2 ネットワーク1，ケース1の近似解の比較

デザイン変数	HJ法	EDO法	SDAP法	SA法	MCND法	TABU法
$y_{1,3}$	0.00	0.13	0.00	0.00	0.00	0.00
$y_{2,1}$	1.20	0.00	0.00	0.00	0.00	0.00
$y_{3,1}$	3.00	6.26	4.85	3.1639	0.00	5.48
$y_{6,4}$	3.00	0.13	0.00	0.00	0.00	0.00
$y_{6,5}$	2.80	6.26	6.48	3.724	7.30	7.44
Φ_{cend}	218.173	201.204	199.963	201.44	212.977	199.651

MCND法は目的関数修正法，TABU法はタブサーチ法の結果である。

従来の解法との比較を行うため，Hooke - Jeeves法（HJ法）[1]，Equilibrium Decomposed Optimization法（EDO法）[18]，Sensitivity Descent Algorithm Predetermined法（SDAP法）[6]，Simulated Annealing法（SA法）[5]の結果も表に併記した。これらのデザイン変数値は，Suwansirikul[18]およびFriesz[6),5)]で示された値である。これらの値を用いて利用者均衡問題を解き，(1)式に代入して目的関数値を求めた。なお，各論文に記述している目的関数値とは若干異なっている。また，表2と表3に表示していないy_{ij}はすべて0.00である。

目的関数修正法の結果をまとめておく。ネットワーク1，ケース1に対して，SDAP法，SA法とEDO法よりも解が劣るが，HJ法よりも良い解を求めることができた。ケース2に対して，SA法とEDO法よりも解が劣るが，HJ法やSDAP法より良い解を求めることができた。ネットワーク3に対して，SA法には劣るが，HJ法やEDO法より優れ，SDAP法と同程度の解を求めることができた。全体として，HJ法よりも優れ，SA法には劣るが，その他の解法に劣らない解を求めることができた。計算時間については，

表3 ネットワーク1，ケース2の近似解の比較

デザイン変数	HJ法	EDO法	SDAP法	SA法	MCND法	TABU法
$y_{1,3}$	5.40	4.88	4.63	0.00	0.00	4.68
$y_{2,1}$	8.18	8.59	7.80	10.174	13.13	9.92
$y_{3,1}$	8.10	7.48	12.72	5.7769	0.00	7.28
$y_{3,2}$	0.00	0.26	0.00	0.00	0.00	0.00
$y_{3,5}$	0.90	0.85	0.65	0.00	0.00	0.64
$y_{5,3}$	0.00	0.00	1.81	0.00	0.00	0.00
$y_{5,6}$	3.90	1.54	1.85	0.00	0.00	1.24
$y_{6,4}$	8.10	0.26	16.96	0.00	0.00	0.00
$y_{6,5}$	8.40	12.52	6.21	17.2786	20.52	20.80
Φ_{cend}	561.470	540.208	563.836	533.327	550.436	522.593

表4 ネットワーク2の近似解の比較

デザイン変数	HJ法	EDO法	SDAP法	SA法	MCND法	TABU法
初期値	2.00	12.5	9.00	6.25	0.00	4.00
$y_{6,8}$	4.80	4.59	5.25	5.38	5.241	5.16
$y_{7,8}$	1.20	1.52	1.73	2.26	1.400	1.93
$y_{8,6}$	4.80	5.45	5.28	5.50	5.266	5.19
$y_{8,7}$	0.80	2.33	1.60	2.01	1.377	1.55
$y_{9,10}$	2.00	1.27	2.79	2.64	2.743	2.42
$y_{10,9}$	2.60	2.33	2.86	2.47	2.785	2.85
$y_{10,16}$	4.80	0.41	4.58	4.54	4.661	3.21
$y_{13,24}$	4.40	4.59	4.40	4.45	4.381	4.80
$y_{16,10}$	4.80	2.71	4.59	4.21	4.701	3.08
$y_{24,13}$	4.40	2.71	4.30	4.67	4.362	4.91
Φ_{cend}	81.403	83.200	81.231	81.119	81.272	80.740

EDO法と同程度となっている。

タブサーチ法の結果をまとめておく。すべてのネットワークに対して，従来のすべての解法よりも優れた解を求めることができ，従来得られていた最良解よりも良い解を求めることができた。タブサーチ法では計算時間が繰返し回数に依存するため。繰返し回数によって計算時間を設定することができる。この数値実験では，EDO法の500～5000倍程度，HJ法の100～1000倍程度，SA法と同程度の計算時間をかけているが，コンピュータの計算能力の向上によって実用的な時間で処理が可能となっている。また，繰返し回数を限定すれば，計算時間を短縮することも可能である。

7．おわりに

本研究では，利用者均衡条件をもつ連続型ネットワークデザイン問題に対する2つの近似解法を提案した。第一の解法は修正目的関数を用いた近似解法であり，第二の解法はタブーサーチ法を適用した近似解法である。

さらに，これらの近似解法を評価するために2つのネットワークを用いて数値実験を行なった。目的関数修正法では，従来の近似解法に劣らない近似解を求めることができた。タブサーチ法を用いた近似解法では，計算時間は必要となるが，すべてのネットワークに対して従来に得られていた最良解よりも良い解を得ることができた。

参考文献

1) M. Abdulaal and L. J. LeBlanc. Continuous equilibrium network design models. *Transportation Research B*, Vol. 13, No. 1, pp. 19-32, 1979.

2) D. E. Boyce. Urban transportation network-equilibrium and design models: Recent achievements and future prospects. *Environment and Planning A*, Vol. 16, pp. 1445-1474, 1984.

3) G. D. Dantzing, R. P. Harvey, Z. F. Lansdowne, D. W. Robinson, and S. F. Maier. Formulating and solving the network design problem by decomposition. *Transportation Research B*, Vol. 13, No. 1, pp. 5-17, 1979.

4) T. L. Friesz. Transportation network equilibrium, design and aggregation: Key developments and research opportunities. *Transportation Research A*, Vol. 19, No. 4, pp. 413-427, 1985.

5) T. L. Friesz, H-J. Cho, N. J. Mehta, R. L. Tobin, G. Anandalingam. A simulated annealing approach to the network design problem with variational inequality constraints. *Transportation Science*, Vol. 26, No. 1, pp. 18-26, 1992.

6) T. L. Friesz, R. L. Tobin, H-J. Cho, and N. J. Mehta. Sensitivity analysis based heuristic

algorithms for mathematical programs with variational inequality constraints. *Mathematical Programming*, Vol. 48, pp. 265-284, 1990.

7) F. Glover. Tabu search I. *ORSA Journal on Computing*, Vol. 1, pp. 190-206, 1989.

8) H. H. Hoang. Topological optimization of networks: a nonlinear mixed integer model employing generalized benders decomposition. *IEEE Transactions on Automatic Control*, Vol. AC-27, No. 1, pp. 164-169, 1982.

9) L. J. LeBlanc. An algorithm for the discrete network design problem. *Transportation Science*, Vol. 9, pp. 183-199, 1975.

10) L. J. LeBlanc and M. Abdulaal. A comparison of user-optimum versus system-optimum traffic assignment in transportation network design. *Transportation Research B*, Vol. 18, No. 2, pp. 115-112, 1984.

11) L. J. LeBlanc and D. E. Boyce. A bilevel programming algorithm for exact solution of the network design problem with user optimal flows. *Transportation Research B*, Vol. 20, pp. 259-265, 1986.

12) M. Los. A discrete convex programming approach to the simultaneous optimization of land use and transportation. *Transportation Research B*, Vol. 13, pp. 33-48, 1979.

13) T. L. Magnanti and R. T. Wong. Network design and transportation planning: Models and algorithms. *Transportation Science*, Vol. 18, pp. 1-55, 1984.

14) P. Marcotte. Network optimization with continues control parameters. *Transportation Science*, Vol. 17, No. 4, pp. 181-197, 1983.

15) P. Marcotte. Network design problem with congestion effects: A case of bilevel programming. *Mathematical Programming*, Vol. 34, pp. 142-162, 1986.

16) H. Poorzahedy and M. A. Turnquist. Approximate algorithms for the discrete network design problem. *Transportation Research B*, Vol. 16, No. 1, pp. 45-56, 1982.

17) P. A. Steenbrink. Transport network optimization in the Dutch integral transportation study. *Transportation Research*, Vol. 8, No. 1, pp. 11-27, 1974.

18) Suwansirikul, T. L. Friesz, and R. L. Tobin. Equilibrium decomposed optimization: A heuristic for the continuous equilibrium network design problem. *Transportation Science*, Vol. 21, pp. 254-263, 1987.

19) 久保幹雄. 離散構造とアルゴリズムIV：メタヒューリスティクス，近代科学社，1995.

20) 佐佐木綱，朝倉康夫. OD需要の変動を内生化した最適道路網計画モデル，土木学会論文集，No. 383, pp. 93-102, 1987.

21) 朝倉康夫. 交通混雑を考慮した最適道路網計画モデルとその適用，土木計画学研究・論文集，Vol. 2, pp. 157-164, 1985.

22) 朝倉康夫. 利用者均衡を制約とする交通ネットワークの最適計画モデル，土木計画学研究・論文集，Vol. 6, pp. 1-17, 1988.