

区分的線形費用をもつネットワーク設計問題に対する 容量スケールリング法

Capacity Scaling Procedures for the Network Design Problem with Piecewise Linear Costs

片山 直登* 陳明哲** 久保幹雄**

Naoto KATAYAMA* Mingzhe CHEN** Mikio KUBO**

*流通経済大学流通情報学部

**東京海洋大学 海洋工学部

*Ryutsu Keizai University

**Tokyo University of Marine

*Email:katayama@rku.ac.jp

Science and Technology

Key words: Piecewise linear costs, Capacity scaling, Network Design, Optimization

要旨:区分的線形費用をもつネットワーク設計問題は, 輸送・ロジスティクスや通信の領域で頻繁に発生するネットワークを設計する問題であり, 現実的な費用を考慮してネットワークの形状とモノの移動経路を同時に決定する問題である. 本研究では, この問題に対する基本的, 強い, および拡張した強制制約式を用いた3つのモデルに対して, 容量スケールリング法を用いた近似解法を提案する. この容量スケールリング法は, 容量を変化させながら線形緩和問題を解き直す方法である. さらに, 数値実験によって, 従来の解法と比較し, 提案した解法の有効性を検討する.

Abstract : The network design problem with piecewise linear costs is an applied problem which is used for transport, logistic and communication design, and is determined the design of a network configuration and multicommodity flows considering real costs. We present capacity scaling procedures for the network design problem with piecewise linear costs . Our procedures consist in solving linear approximate problems with adjusted arc capacities at each iteration . Three type models of formulations involving normal , strong and extended forcing constraints are analyzed by our procedures . Numerical experiments are illustrated that our approach can obtain better solutions than the previous method.

1. はじめに

ネットワーク設計問題は, 輸送・ロジスティクスや通信などの領域におけるネットワークを設計する問題であり, ネットワークの形状とモノの移動経路を同時に決定する問題である. これらの分野では, 複数の始点・終点の組合せをもつモノの移動を対象とすることが多く, これらは多品種問題とよばれる. また, 現実的な問題では, 宅配便, 混載, 貸切といった

ような複数種類の輸送手段や, 複数の容量の通信回線を選択することができ, 処理量によって費用が最小となる輸送手段や回線容量を利用することになる. そのため, 複数種類の輸送手段や回線容量を考慮した場合, 費用は処理量に対して区分的線形である凹関数として表現できることが多い[1]. このような問題は区分的線形費用をもつネットワーク設計問題とよばれる. 本研究では, 多品種の需要をもち, 区分的線形である費用をもつネットワーク設計問題を対象とする.

ネットワーク設計問題の基本的な問題は, アー

クの処理能力である容量を考慮しない問題であり、数多くの研究 [2][3][4] が行われている。一方、アークの処理能力を考慮した問題は、容量制約をもつネットワーク設計問題とよばれる。この問題に対して、Gendron-Crainic[5][6]、Gendron-Crainic-Frangioni[7] および Holmberg-Yuan[8] は、Lagrange 緩和法をもとにした解法を示している。また、近似解法として、Gendron-Crainicのリソース分解法[9]、Crainic-Gendreau-Farvoldenの基底変換タブーサーチ法 [10]、Ghamlouche-Crainic-Gendreauのサイクル隣接探索法 [11] およびパス再リンク探索法 [12] が提案されている。

近年、ネットワーク設計問題に対して、スロープスケール法や容量スケール法といったスケール法が適用され成果を挙げている。ネットワーク設計問題に対するスケール法は、変動費用やアーク容量を変化させながら線形緩和問題を繰り返し解き、デザイン変数の収束解を求める近似解法である。Crainic-Gendron-Hernu[13] は、弱い定式化に基づいたスロープスケール法を示している。久保 [14] はメタ解法の中で容量スケール法を解説している。また、片山 [15] は弱い定式化と強い定式化に対する容量スケール法を提案し、陳-片山-久保 [16] は列生成法を加えた解法を提案している。

一方、区分的線形費用をもつ問題に関する従来の研究は比較的少ない。Kim-Pardals[1][17] は、単一品種の区分的線形費用をもつ問題に対するスロープスケール法を示している。多品種を対象とした研究として、Croxtton-Gendron-Magnanti[18] は基本モデル、強いモデルおよび拡張モデルを示し、この問題の性質を解析し、線形緩和による変数固定と分枝限定法を組み合わせた解法を提案している。

本研究では、区分的線形費用と多品種の需要をもつネットワーク設計問題に対して容量スケール法を適用した近似解法を提案し、数値実験によって提案した解法の有効性を検討する。

2. 問題の定式化

発地・着地間を移動すべきモノを需要とよび、需要の発地を始点、着地を終点とよぶ。本研究では、

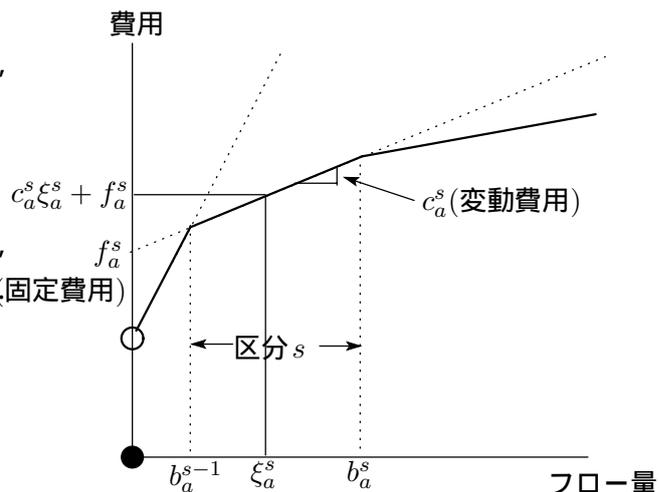


図1. 区分的線形関数

一般的な表現とは異なり、始点・終点の異なるものを異なる品種と定義する。そのため、多品種は複数の始点と終点の組合せをもつ需要が存在することを意味する。ノードは発地・配送センター・着地やルータ・サーバ・コンピュータなど需要の始点・終点や中継点を表し、アークは輸送路線や通信回線などのノード間をつなぐものを表す。また、フローは、荷物やデータなどのノード間を移動するものやその移動を表す。

アーク上には、フロー量に応じて、複数種類の輸送手段・通信回線から費用が最小となるものを選択して設置できるものとする。このため、費用は図1に示すようなフロー量に関して区分的線形となる凹関数とする。線形の区間を区分とよぶ。対象とする関数は、必ずしも区分の前後で連続である必要はないが、下半連続であり、各区分に関する変動費用と固定費用は非負とする。

本研究では、区分的線形費用をもつネットワーク設計問題に対して、基本モデル、強いモデルおよび拡張モデルによる3種類の定式化 [18] を用いる。基本モデルは制約式が少ないため、本研究で提案する解法では計算時間が短い、得られる解は相対的に悪いことが予想される。一方、拡張モデルは制約式が最も多いため、計算時間が長い、得られる解は相対的に良いことが予想される。

ノード集合を N 、アーク集合を A とする。ノード n に入るアークの集合を A_n^+ 、 n から出るアークの集合を A_n^- とする。品種集合を K とし、品種 k の始点

を O^k , 終点を D^k , 需要を d^k とする . アーク a における区分集合を S_a とする . また , 図1に示すように区分 s の上限値を b_a^s , 下限値を b_a^{s-1} とし , $b_a^0=0$ とする . アーク a の区分 s における単位フロー量当たりの変動費用を c_a^s , 固定費用を f_a^s とする . また , フロー量が0であるとき , これらの費用は発生しない .

アーク a において , アーク上のフロー量を表す総フロー変数を X_a , 品種 k のフロー量を表す品種フロー変数を x_a^k とする . また , X_a が区分 s の範囲にあるとき X_a , それ以外は0である区分フロー変数を ξ_a^s とする . さらに , X_a が区分 s の範囲にあるとき1 , それ以外は0を表す0-1変数である区分変数を y_a^s とする .

$$\xi_a^s = \begin{cases} X_a & \text{if } b_a^{s-1} < X_a \leq b_a^s \\ 0 & \text{otherwise} \end{cases} \quad s \in S_a, a \in A$$

$$y_a^s = \begin{cases} 1 & \text{if } b_a^{s-1} < X_a \leq b_a^s \\ 0 & \text{otherwise} \end{cases} \quad s \in S_a, a \in A$$

$y_a^s = 1$ であるときには , アーク a における区分 s の区分フローが存在し , この区分フローに対する費用関数が発生する . 区分フロー ξ_a^s に対する費用は , $b_a^{s-1} < \xi_a^s = X_a \leq b_a^s$ のとき $c_a^s \xi_a^s + f_a^s$, そうでないとき0となるため , $c_a^s \xi_a^s + f_a^s y_a^s$ と表すことができる .

区分的線形費用をもつネットワーク設計問題に対する基本的な定式化によるモデルを基本モデル *PLCB* とよぶ . *PLCB* は , 次のように表わされる .

PLCB

$$\min \sum_{a \in A} \sum_{s \in S_a} (c_a^s \xi_a^s + f_a^s y_a^s) \quad (1)$$

$$\text{st } \sum_{a \in A_n^+} x_a^k - \sum_{a \in A_n^-} x_a^k = \begin{cases} -d^k & \text{if } n = O^k \\ d^k & \text{if } n = D^k \\ 0 & \text{その他} \end{cases}$$

$$n \in N \quad k \in K \quad (2)$$

$$X_a = \sum_{k \in K} x_a^k \quad a \in A \quad (3)$$

$$X_a = \sum_{s \in S_a} \xi_a^s \quad a \in A \quad (4)$$

$$b_a^{s-1} y_a^s \leq \xi_a^s \leq b_a^s y_a^s \quad s \in S_a, a \in A \quad (5)$$

$$\sum_{s \in S_a} y_a^s \leq 1 \quad a \in A \quad (6)$$

$$0 \leq x_a^k \leq d^k \quad k \in K, a \in A \quad (7)$$

$$y_a^s \in \{0, 1\} \quad s \in S_a, a \in A \quad (8)$$

(1) 式は , 変動費用と固定費用の和である総費用を最小化する目的関数である . 第一項は変動費用 , 第二項は固定費用である . (2) 式は , 品種 k について , ノード n に入るフロー量と出るフロー量の差が , ノード n が始点であれば $-d^k$, 終点であれば d^k , その他のノードであれば0であることを表す . この式はフロー保存式とよばれ , 必ず需要が始点から終点に流れることを保証する . (3) 式は , アーク上の品種フロー量の合計が総フロー量に一致することを表す . (4) 式は , 区分フロー量の合計が総フロー量に一致することを表す . (5) 式は , 区分変数が1である場合 , 区分フロー量は区分の上限値と下限値の間の範囲に限定され , そうでない場合は0となることを表す強制制約式である . (5) 式には区分フローの下限に等号が含まれているが , これは $y_a^s = 0$ のとき , $\xi_a^s = 0$ とするためである . 区分フローの定義から $b_a^{s-1} < \xi_a^s \leq b_a^s$ (または0) であるため , 区分 s の下端のフロー量 ($\xi_a^s = b_a^{s-1}$) は定義に反する . しかし , 費用関数は下半連続であり , かつ最小化問題であるため , 最適解において区分 $s-1$ の上端のフロー量である $\xi_a^{s-1} = b_a^{s-1}$ を採用すればよい . 妥当な問題表現となる . (6) 式は , 各アークに対して区分変数の合計が0または1であることを表し , 1となる区分変数は高々1つであることを表す . (7) 式は , 品種フロー変数の範囲を表す . フローの変動費用は非負であることから , 最適解において x_a^k が d^k 以下であることは , (2) 式により保証されるため , $x_a^k \leq d^k$ の条件は省くことができる . (8) 式は , 区分変数の0-1条件を表す .

最適値とその連続緩和問題の最適値との差が大きな定式化を弱い定式化よぶ . 基本モデルの定式化は強制制約式とよばれる制約式を含まないため , 弱い定式化である [2][18] ことが知られている . そこで , 次のような制約式を追加する .

$$x_a^k \leq d^k \sum_{s \in S_a} y_a^s \quad a \in A, k \in K \quad (9)$$

(9) 式は , アーク a における品種 k の品種フローの上限値を表す強制制約式である . これは , アーク a において , S_a に含まれるいずれかの区分変数が1であ

れば品種 k の品種フローの上限値は d^k であり、すべての区分変数が0であれば品種 k のフローが存在しない、すなわち、 $x_a^k = 0$ であることを表す。

この強制制約式を加えたモデルを強いモデル $PLCS$ とよぶ。 $PLCS$ は、次のように表される。

$PLCS$

$$\begin{aligned} \min \quad & \sum_{a \in A} \sum_{s \in S_a} (c_a^s \zeta_a^s + f_a^s y_a^s) \\ \text{st} \quad & x_a^k \leq d^k \sum_{s \in S_a} y_a^s \quad k \in K, a \in A \end{aligned}$$

(2) ~ (8) 式

アーク a における品種 k 、区分 s に対する区分品種フロー変数を ζ_a^{ks} とおく。 ζ_a^{ks} は、アーク a における総フロー X_a が区分 s の範囲にあるときに品種 k の品種フロー x_a^k となり、それ以外は0となる変数である。

$$\zeta_a^{ks} = \begin{cases} x_a^k & \text{if } b_a^{s-1} < X_a \leq b_a^s \\ 0 & \text{otherwise} \end{cases}$$

$s \in S_a, a \in A, k \in K$

区分品種フロー変数と、区分フロー変数、品種フロー変数の関係は次式となる。

$$\begin{aligned} \zeta_a^s &= \sum_{k \in K} \zeta_a^{ks} \quad s \in S_a, a \in A \\ x_a^k &= \sum_{s \in S_a} \zeta_a^{ks} \quad k \in K, a \in A \end{aligned}$$

区分品種フロー変数を用いると、次のように(9)式を区分について非集約化することができる。

$$\zeta_a^{ks} \leq d^k y_a^s \quad k \in K, s \in S_a, a \in A \quad (10)$$

この制約は、(9)式よりも強い強制制約式となる。

区分品種フロー変数を用いたモデルを拡張モデル $PLCE$ とよぶ。 $PLCE$ は、次のように表される。
 $PLCE$

$$\begin{aligned} \min \quad & \sum_{a \in A} \sum_{s \in S_a} (c_a^s \zeta_a^s + f_a^s y_a^s) \\ \text{st} \quad & \zeta_a^s = \sum_{k \in K} \zeta_a^{ks} \quad s \in S_a, a \in A \\ & x_a^k = \sum_{s \in S_a} \zeta_a^{ks} \quad k \in K, a \in A \end{aligned}$$

$$0 \leq \zeta_a^{ks} \leq d^k y_a^s \quad k \in K, s \in S_a, a \in A$$

(2) ~ (8) 式

フロー変数は非負の実数であり、区分変数は0-1変数であるため、3つの定式化とも混合整数計画問題となる。 $|S|$ を区分数の上限値とすると、0-1変数は $O(|A||S|)$ 個であり、連続変数の数は $PLCB$ と $PLCS$ では $O(|A||K| + |A||S|)$ 、 $PLCE$ では $O(|A||K||S|)$ 個となる。また、制約式の数も、 $PLCB$ では $O(|N||K| + |A||S|)$ 、 $PLCS$ では $O(|A||K| + |A||S|)$ 、 $PLCE$ では $O(|A||K||S|)$ となる。このため、実用規模の問題に対して、これらの定式化は大規模な混合整数計画問題となるため、汎用の数理計画ソフトウェアを用いて直接的に最適解を求めることは困難となる。そこで、近似解法が必要となる。

3. 容量スケールリング法

容量スケールリング法は、容量制約をもつネットワーク設計問題に適用され、有効性が示されている近似解法である[14][16]。

アークの容量制約をもつ問題において、なんらかの手段を用いて問題の最適フローを求めることができれば、容量をこのフロー量に変更しても最適値は変わらない。しかし、最適フローを求めること自体が問題の目的であるため、直接、最適フローを求めることはできない。

そこで、線形緩和問題において、最適フローに近づくように容量を変更しながら問題を解き直す。容量スケールリング法は、このように線形緩和問題のフロー解をもとに容量を変更して繰り返し線形緩和問題を解き、0-1変数解を導く近似解法である。

3.1 容量制約と多重ダミーアーク

区分数が1である区分的線形費用をもつ問題を考えると、この問題は区分の上限値が容量である容量制約をもつネットワーク設計問題と見なすことができる。そこで、区分的線形費用をもつ問題に対して、アークにおける各区分を1本のダミーアークと見なし、図2に示すように、1本のアークに対して多重のダミーアークを設定する。ダミーアークの容量を対応する区分の上限値とし、固定費用と変動費用は対応する区分のものと同一とする。

$PLCB$ において多重アークに置き換えた問題を

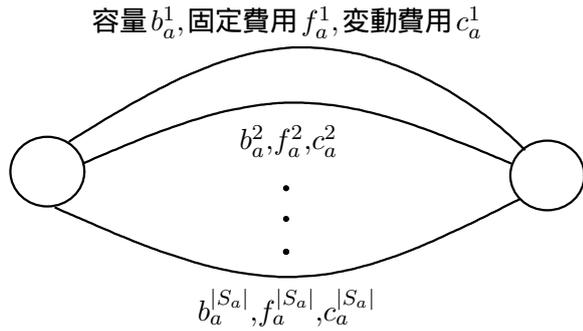


図2. ダミーアーク

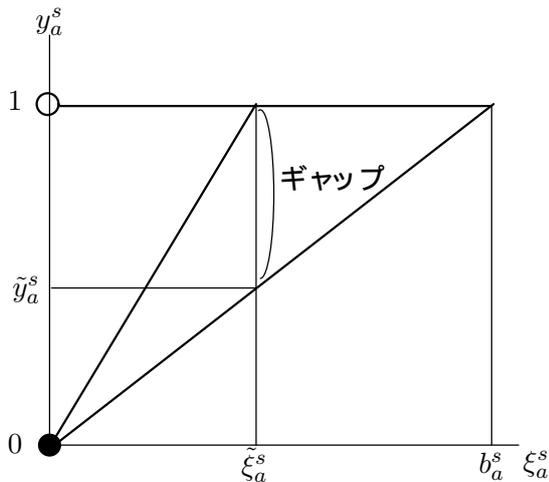


図3. 線形緩和

$PLCBD$ とみると、 $PLCBD$ は(5)式を次式で置き換えた問題となる。

$$0 \leq \xi_a^s \leq b_a^s y_a^s \quad a \in A, \quad s \in S_a \quad (11)$$

区分変数 y_a^s は、アーク a のダミーアーク s に対する 0-1 変数となる。

$PLCBD$ は最小化問題であることから、この問題の最適解では、各ノード間の多重アークの中で、フロー量に対して費用が最小となるダミーアークにフローが流れることになる。一方、 $PLCB$ においても、最適解において、フロー量に対して費用が最小となる区分上にフローが存在する。これは、 $PLCBD$ が $PLCB$ と等価な問題となることを意味する。

$PLCBD$ はアークが容量をもつ問題となるので、容量制約をもつネットワーク設計問題に対する容量スケールリング法を適用することができる。

3.2 $PLCB$ に対する容量スケールリング法

はじめに、基本モデル $PLCB$ に対する容量スケールリング法を示す。 $PLCBD$ において、0-1 変数でダミーアークに対応する区分変数 $y_a^s (s \in S_a, a \in A)$ を線形緩和し、 $0 \leq y_a^s \leq 1$ とした線形緩和問題を考える。この緩和問題は最小化問題であり、 y_a^s の下限に関する制約式は(11)式のみである。このため、区分フロー変数と区分変数の関係は、(11)式から $y_a^s = \xi_a^s / b_a^s$ となり、図3に示すように線形関数で表わすことができる。線形緩和問題における区分フローの解を $\tilde{\xi}_a^s$ としたとき、対応する区分変数の線形緩和解は $\tilde{y}_a^s = \tilde{\xi}_a^s / b_a^s$ となる。 y_a^s は本来0または1であるため、 \tilde{y}_a^s が小数值を取った場合、これらの中にギャップが存在する。そこで、区分フローが変化しないものと想定すると、ギャップが存在する場合、アーク容量 b_a^s を $\tilde{\xi}_a^s$ に変更すれば、 $\tilde{y}_a^s = 1$ となり、このギャップを埋めることが期待できる。

実際には、容量を変更させた問題を解き直すと、区分フロー自体が変化してしまい、ギャップが埋まる保証はない。そこで、パラメータ λ を用いて、徐々に容量 b_a^s を変更した問題を解き直し、新たな解における区分フローが大きく変化しないように制御する。

アーク a 、区分 s に対応するダミーアークの変更した容量を \hat{b}_a^s とおき、 \hat{b}_a^s の初期値を b_a^s とする。容量を \hat{b}_a^s とした線形緩和問題を解き直し、新たな区分フロー $\hat{\xi}_a^s$ を求める。続いて、パラメータ $\lambda (0 < \lambda \leq 1)$ を用いて、 \hat{b}_a^s を次のように更新する。

$$\hat{b}_a^s := \lambda \hat{\xi}_a^s + (1 - \lambda) \hat{b}_a^s \quad a \in A, s \in S_a$$

容量の変更を行っているが、 $\hat{\xi}_a^s$ は本来の容量 b_a^s 以下であればよい。そこで、 y_a^s の上限値である1を変更し、 $y_a^s \leq \hat{b}_a^s / \hat{\xi}_a^s$ に緩和する。このとき、 y_a^s は1以上の値を取ることができが、区分フロー $\hat{\xi}_a^s$ は本来の容量 b_a^s 以下であることが保証される。なお、この変更を(6)式にも反映し、(6)式における y_a^s の係数を $\hat{b}_a^s / \hat{\xi}_a^s$ に変更する。

容量 $\hat{b} = (\hat{b}_a^s)$ とした $PLCBD$ の線形緩和問題を $PLCBL(\hat{b})$ とする。

$PLCBL(\hat{b})$

$$\min \sum_{a \in A} \sum_{s \in S_a} (c_a^s \xi_a^s + f_a^s y_a^s)$$

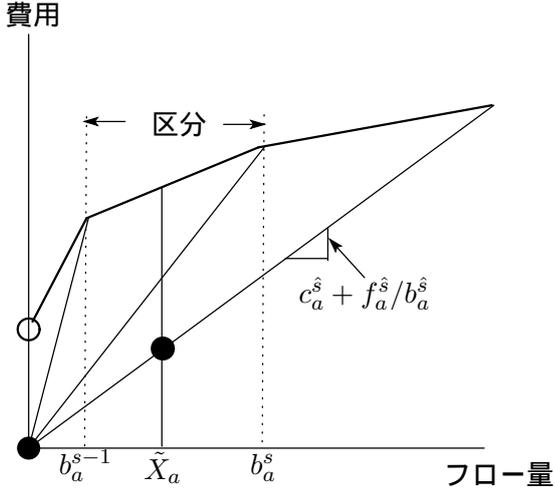


図4. 線形緩和問題

$$\text{st } 0 \leq \xi_a^s \leq \hat{b}_a^s y_a^s \quad a \in A, s \in S_a \quad (12)$$

$$\sum_{s \in S_a} (\hat{b}_a^s / b_a^s) y_a^s \leq 1 \quad a \in A \quad (13)$$

$$0 \leq y_a^s \leq b_a^s / \hat{b}_a^s \quad a \in A, s \in S_a \quad (14)$$

(2) ~ (4) and (7) 式

この問題は線形計画問題であるため、汎用の数理計画ソフトウェアを用いて解くことができる。

3.3 近似解法

容量スケール法では、容量 \hat{b} を変更して、 $PLCBL(\hat{b})$ を繰り返し解き直す。 $PLCBL(\hat{b})$ は最小化問題であることから、最適解において $y_a^s = \xi_a^s / \hat{b}_a^s$ が成り立ち、目的関数値は $\sum_{a \in A} \sum_{s \in S_a} (c_a^s + f_a^s / \hat{b}_a^s) \xi_a^s$ と等価になる。このため、最適解では多重アークの中で $c_a^s + f_a^s / \hat{b}_a^s$ (図4参照) の値が最小であるダミーアーク \hat{s} にフローが集中する。単純に線形緩和が正である区分変数を1としてしまうと、 $PLCB$ において総フロー量と区分変数の範囲が対応しない可能性が生じる。そこで、総フローの線形緩和を \hat{X}_a とし、 \hat{X}_a がどの区分の範囲にあるかによって、区分変数の値を設定する。図4に示すように、 \hat{X}_a が上限 b_a^s と下限 b_a^{s-1} の範囲にある区分に対応する区分変数を1、それ以外を0とした区分変数を \bar{y}_a^s とする。

$$\bar{y}_a^s = \begin{cases} 1 & \text{if } b_a^{s-1} < \hat{X}_a \leq b_a^s \\ 0 & \text{otherwise} \end{cases} \quad s \in S_a, a \in A$$

このとき、 \bar{y} は $PLCB$ の制約式である (6) 式と (8) 式を満足する。

次に、区分変数を $\bar{y} = (\bar{y}_a^s)$ としたネットワーク上で、次のような多品種フロー問題 $MCF(\bar{y})$ を作成する。

$MCF(\bar{y})$

$$\min \sum_{a \in A} \sum_{s \in S_a} (c_a^s \xi_a^s + f_a^s \bar{y}_a^s)$$

$$\text{st } b_a^{s-1} < \xi_a^s \leq b_a^s \quad \text{if } \bar{y}_a^s = 1, a \in A, s \in S_a$$

$$\xi_a^s = 0 \quad \text{if } \bar{y}_a^s = 0, a \in A, s \in S_a$$

(2) ~ (4) and (7) 式

$MCF(\bar{y})$ は線形計画問題であるため、汎用の数理計画ソフトウェアを用いて解くことができる。 $MCF(\bar{y})$ の実行可能解を求めることができれば、 \bar{y} および $MCF(\bar{y})$ のフロー解は $PLCB$ の実行可能解となり、その目的関数値は $PLCB$ の上界値となる。

一方、容量スケール法では、繰り返し回数とともに、多くの0-1変数である区分変数が0または1に収束する傾向がある。汎用の数理計画ソフトウェアの分枝限定法を用いると、少数の0-1変数をもつ計画問題に対しては比較的短時間で最適解を求めることができる。そこで、0または1に収束していない区分変数の合計を B とし、 B がある決められた数以下となった繰り返し回数において、各アークに関して、次のように $PLCB$ における区分変数の範囲を制限した問題 $PLCBF$ を作成する。

$$\begin{cases} y_a^s = 0 & s \in S \quad \text{if } \sum_{s \in S_a} \hat{y}_a^s < \epsilon_1 \\ \sum_{s \in S_a} y_a^s = 1 & \text{if } \hat{y}_a^s \geq \epsilon_2 \quad \exists s \in S \end{cases} \quad a \in A \quad (15)$$

ここで、 \hat{y}_a^s は $PLCBL(\hat{b})$ の区分変数解、 ϵ_1 と ϵ_2 は $0 < \epsilon_1 < \epsilon_2 < 1$ である定数である。 $PLCBF$ に対して、分枝限定法を適用すれば、 $PLCB$ の近似解と上界値を求めることができる。

しかし、このように0-1変数を限定した問題であっても、問題の規模や性質によって、分枝限定法により最適解を求めることが困難な場合がある。そのため、分枝限定法は一定時間で打ち切ることとし、その時点で得られた最良の解を近似解とする。また、 B を一定値とした場合、連続した繰り返し回数にお

いて分枝限定法を繰り返し、同一の解を生成することが多くなる。そこで、分枝限定法を適用することに B を α ($0 < \alpha < 1$) 倍して減少させ、解の多様性を確保する。

近似解法を含めた $PLCB$ に対する容量スケールリング法のアルゴリズムを示す。

- [ステップ1] パラメータ $\lambda, \epsilon_1, \epsilon_2, \alpha$ および B を決め、 $\hat{b}_a^s := b_a^s (s \in S_a, a \in A)$, $ub_{min} := \infty$, $t := 0$ とする。
- [ステップ2] $t := t + 1$ とする。 $PLCBL(\hat{b})$ を解き、解 $\hat{y}_a^s, \hat{\xi}_a^s (s \in S_a, a \in A)$ と $\hat{X}_a (a \in A)$ を求める。
- [ステップ3] $\hat{b}_a^s := \lambda \hat{\xi}_a^s + (1 - \lambda) \hat{b}_a^s (s \in S_a, a \in A)$ とする。
- [ステップ4] \bar{y} を求め、 $MCF(\bar{y})$ を解き、上界値 ub_t を求める。 $ub_{min} > ub_t$ であれば、 $ub_{min} := ub_t$ とする。
- [ステップ5] $\epsilon_1 < \hat{y}_{at}^s < \epsilon_2$ である区分変数が B 個以下であれば、 $PLCBF$ を作成し、分枝限定法を適用し、上界値 ub_f を求める。 $ub_{min} > ub_f$ であれば、 $ub_{min} := ub_f$ とする。 $B := \alpha B$ とする。
- [ステップ6] $t = t_{max}$ であれば終了、そうでなければステップ2へ戻る。

ここで、 t は現在の繰り返し回数、 t_{max} は最大繰り返し回数であり、 ub_{min} は最良の上界値、 ub_f は $PLCBF$ の目的関数値である。

3.4 $PLCS$ と $PLCE$ に対する容量スケールリング法

$PLCB$ と同様な手順で、 $PLCS$ と $PLCE$ に対して容量スケールリング法を適用することができる。

$PLCE$ に対して、(5) 式の代わりに (11) 式を用いて多重アーク化し、容量を \hat{b} とした問題の線形緩和問題を $PLCSL(\hat{b})$ とする。同様に、 $PLCE$ に対して、(11) 式を用いて多重アーク化し、かつ容量を \hat{b} とした問題の線形緩和問題を $PLCEL(\hat{b})$ とする。 $PLCSL(\hat{b})$ と $PLCEL(\hat{b})$ は線形計画問題であるため、汎用の数理計画ソフトウェアを用いて解くことができる。

$PLCS$ と $PLCE$ に対する容量スケールリング法は、 $PLCBL(\hat{b})$ のかわりに $PLCSL(\hat{b})$ と $PLCEL(\hat{b})$ 、

$PLCBF$ のかわりに $PLCS$ と $PLCE$ に (15) 式を付加した問題を用いる以外は、 $PLCB$ のものと同様である。ただし、区分変数に関する強制制約式である (9) 式と (10) 式を追加しているため、緩和解において、多重アークの中で複数のダミーアークのフロー量が正となり、複数の区分変数が正になる可能性がある。しかし、 $PLCB$ の場合と同様に、総フロー変数の線形緩和解を用いて区分変数を設定すれば、制約式を満足する区分変数を一意に設定することができる。

4. 数値実験

提案した解法の有効性を検証するために、次のような条件のもとで数値実験を行った。

- 1) データ：Crainicら [10] のデータをもとに、次のようなデータを作成した。
 - a) ノード数、アーク数、品種数：(25,100,10), (25,100,30), (100,40,10), (100,40,30) の組合せに対して、各3問
 - b) 区分数：すべてのアークに対して、3
 - c) 区分の上限值： $b_a^s = (2^s - 1)b_a/2$, b_a は元のデータのアーク容量
 - d) 変動費用： $c_a \beta^{s-1}$ とする。 c_a は元のデータの変動費用、 $\beta = 0.7$
 - e) 固定費用：第1区分の固定費用を元のデータの固定費用とし、他の区分の固定費用は費用関数が b_a^s で連続になるように設定
- 2) λ : 0.1, 0.15, 0.2, 0.25, 0.30
- 3) ϵ_1, ϵ_2 : 0.01
- 4) α : 0.75
- 5) 分枝限定法の実施基準の初期値 B : 50
- 6) 分枝限定法の計算時間：最大30分
- 7) 最大繰り返し回数 t_{max} : 250回、ただし、計算時間が10時間を超える場合は、繰り返し回数100回または12時間で打ち切り
- 8) 言語：Visual Studio 2005 C++
- 9) 数理計画ソフトウェア：CPLEX10.0
- 10) 使用コンピュータ：CPU Pentium Dual-Core 3.0GHz, メモリ4GB

提案した容量スケールリング法による結果を表1に示す。表の数値は、ノード数 ($|N|$)、アーク数 ($|A|$)、

表1. 容量スケールリング法の計算結果

問題	N	A	K	問題の タイプ	PLCB		PLCS		PLCE	
					誤差	時間	誤差	時間	誤差	時間
1	25	100	10	VL	0.9	1.2	0.2	4.2	0.0	0.1
2	25	100	10	FT	5.6	2.3	0.0	3.0	0.0	13.6
3	25	100	10	FL	5.2	2.0	0.0	3.0	0.0	37.5
4	25	100	30	VT	0.9	28.6	2.5	45.1	0.1	82.9
5	25	100	30	FT	3.2	22.5	1.6	38.0	0.0	232.8
6	25	100	30	FL	0.2	16.1	1.1	1.4	0.6	293.8
7	100	400	10	VL	0.0	14.0	0.0	288.1	0.0	8.4
8	100	400	10	FT	16.3	80.2	4.2	34.6	0.0	1390.3
9	100	400	10	FL	10.3	31.4	0.0	25.2	0.0	16.5
10	100	400	30	VT	4.1	313.3	2.5	8440.6	0.7	19196.9
11	100	400	30	FT	25.3	938.8	15.9	9082.7	10.8	36266.4
12	100	400	30	FL	21.7	440.7	12.5	58.7	8.9	35096.1

誤差：(%) 計算時間：(秒)

図2. 従来法の計算結果

問題	PLCB		PLCS		PLCE	
	誤差	時間	誤差	時間	誤差	時間
1	×	×	×	×	0.0	0.3
2	×	×	5.3	0.1	7.4	0.6
3	×	×	×	×	×	×
4	×	×	×	×	1.8	1.2
5	×	×	6.8	0.7	1.3	2.7
6	×	×	7.9	45.1	×	×
7	×	×	×	×	×	×
8	×	×	×	×	×	×
9	×	×	×	×	5.4	5.8
10	×	×	×	×	×	×
11	×	×	×	×	14.0	36145.6
12	×	×	×	×	9.1	22919.7

誤差：(%) 計算時間：(秒)

品種数(|K|), 問題のタイプと, 各モデルによる誤差と計算時間である。タイプ欄の記号は, 相対的にFは固定費, Vは変動費が高い, Tは区分の上限值が小さい, Lは区分の上限值が大きいデータであることを示す。誤差は, (上界値 - 下界値)/下界値 × 100(%) である。上界値は, パラメータを変化させた中の最良の上界値である。下界値は, PLCEを直接CPLEXを用いて最大48時間解いたときの最良の下界値または最適値である。

問題のタイプで見ると, 基本モデルPLCBでは, VL, VTに対して誤差は0.0~4.1%と小さいが, FTに対しては3.2~25.3%, FLに対しては0.2~21.7%となった。強いモデルPLCSでは, VL, VTに対して0.0~2.5%, FTに対して0.0~15.9%, FLに対して0.0~12.5%となった。拡張モデルPLCEでは, VL, VTに対して0.0~0.7%, FTに対して0.0~10.8%, FLに対して0.0~8.9%となった。

一方, 問題の規模で見ると, 規模の小さな問題1

~9に対して、*PLCB*による誤差は0.0~16.3%と比較的大きいが、*PLCS*では0.0~4.2%に減少した。さらに、*PLCE*では0.0~0.6%となり、多くの問題で最適解を求めることができた。規模の大きな問題10~12に対して、*PLCB*による誤差は4.1~25.3%、*PLCS*では2.5~15.9%と比較的誤差が大きいが、拡張モデル*PLCE*では0.7~10.8%に減少した。しかし、規模の大きな問題では、変数、制約式ともに大幅に増加するため、計算時間は大幅に増加している。

問題4と6では、*PLCS*よりも*PLCB*の方が誤差の小さな解が求められている。提案した容量スケールリング法では、未収束のデザイン変数に対する分枝限定法を行っている。このため、デザイン変数の収束状況により、*PLCB*が良い解を算出する可能性があることが分かる。

Croxtton-Gendron-Magnanti[18]は本研究と同様の3つのモデルに対して、閾値と分枝限定法を組み合わせた近似解法を提案している。この先行研究では、*PLCB*、*PLCS*および*PLCE*の線形緩和問題を解き、求めた区分変数の線形緩和が0であれば区分変数を0に固定し、閾値(0.7)以上であれば区分変数を1に固定し、それ以外の区分変数を0-1変数に設定した問題に対して、分枝限定法を適用する方法である。

先行研究の解法による結果を表2に示す。×は実行不可能であったため、実行可能解が求められなかったことを表す。*PLCB*を用いた場合、先行研究の解法では、すべての問題に対して実行不可能となった。*PLCS*では、問題2, 5, 6のみで実行可能解を求めることができたが、その他の問題では実行可能解を求めることができなかった。また、*PLCE*では、12問中、5問で実行可能解を求めることができなかった。このように、先行研究では多くの問題で実行可能解を求めることができなかった。Croxttonらの論文でも、その可能性が指摘されている。彼らの計算結果では実行可能解が求められているが、これは区分の最大の上限值を全需要量またはその半分という大きな値に設定しており、実行可能解が求め易い問題を用いていることが要因と考えられる。

先行研究の解法で解が求められた問題で比べると、*PLCS*では、容量スケールリング法では先行研究の解

法よりも誤差が5.3~6.8%小さい解を求めることができた。また、*PLCE*では、先行研究の解法よりも誤差が0.0~7.4%小さい解を求めることができた。先行研究の解法の計算時間は容量スケールリング法より短いものが多いが、規模の大きな問題11, 12では容量スケールリング法と同様に計算時間が長くなっている。

5. おわりに

本研究では、区分的線形費用をもつネットワーク設計問題に対して、容量スケールリング法を用いた近似解法を提案した。これは、区分をダミーアークに置き換えた問題に容量スケールリング法を適用し、区分変数の範囲を制限した問題に対する分枝限定法を組み込んだ解法である。また、数値実験を行い、提案した解法の有効性を示した。

基本モデルと強いモデルを用いた場合には相対的に大きな誤差をもつ解が得られたが、拡張モデルを用いた場合には比較的誤差の小さい解を求めることができた。また、先行研究の解法と異なり、すべての問題で実行可能解を求めることができ、かつ先行研究の解法よりも誤差の小さい解を求めることができた。

ノード数100、アーク数400、品種数30の規模の問題では、拡張モデルにおける定式化が大規模なものとなるため、計算時間が10時間を超える場合があった。今後の課題として、パスフローによる定式化と列生成法を組み合わせることなどによって、問題の規模の縮小と計算時間の短縮化を行い、より大規模な問題に適用できる解法を開発することが挙げられる。

参考文献

- [1] D. Kim and P. Pardalos: "A dynamic domain contraction algorithm for nonconvex piecewise linear network flow problems", *Journal of Global Optimization*, Vol.17, pp.225-234,2000
- [2] 片山直登:「ネットワーク設計」, 朝倉出版,2008
- [3] T. L. Magnanti and R. T. Wong: "Network design and transportation planning: Models and algorithms", *Transportation Science*,

- Vol.18, pp.1–55, 1984
- [4] M. Minoux: “Network synthesis and optimum network design problems: Models, solution methods and applications”, *Networks*, Vol.19, pp.313–360, 1989
- [5] B. Gendron and T. Crainic: “Parallel implementations of bounding procedures for multicommodity capacitated network design”, Technical Report CRT-94-45, Centre de recherche sur les transports, Université de Montreal, 1994
- [6] B. Gendron and T. Crainic: “Relaxations for multicommodity capacitated network design problems”, Technical Report CRT-965, Centre de recherche sur les transports, Université de Montreal, 1994
- [7] B. Gendron, T. Crainic and A. Frangioni: “Multicommodity capacitated network design”, Technical Report CRT-98-14, Centre de recherche sur les transports, Université de Montreal, 1997
- [8] K. Holmberg and D. Yuan: “A lagrangian heuristic based branch-and-bound approach for the capacitated network design problem”, *Operations Reserach*, Vol.48, pp.461–481, 2000
- [9] B. Gendron and T. Crainic: “Bounding procedures for multicommodity capacitated fixed charge network design problems”, Technical Report CRT-96-06, Centre de recherche sur les transports, Université de Montreal, 1996
- [10] T. Crainic, M. Gendreau and J. Farvolden: “A simplex-based tabu search for capacitated network design”, *INFORMS Journal on Computing*, Vol.12, pp. 223–236, 2000
- [11] I. Ghamlouche, T. Crainic and M. Gendreau: “Cycle based neighborhood structures for fixed-charge capacitated multicommodity network design”, *Operations Research*, Vol.51, pp.655–667, 2003
- [12] I. Ghamlouche, T. G. Crainic and M. Gendreau: “Path relinking, cycle-based neighborhoods and capacitated multicommodity network design”, *Annals of Operations Research*, Vol.131, pp.109–134, 2003
- [13] T. G. Crainic, B. Gendron and G. Hernu: “A slope scaling/Lagrangian perturbation heuristic with long-term memory for multicommodity capacitated fixed-charge network design”, *Journal of Heuristics*, Vol.10, pp.525–545, 2004
- [14] 久保: “数理計画ソルバーを用いたメタ解法”, *システム/制御/情報*, Vol.50, pp. 357–362, 2006
- [15] 片山直登: “容量スケールリング法を用いた容量制約をもつ多品種フローネットワーク設計問題の近似解法”, *流通経済大学流通情報学部紀要*, Vol.9, No.2, pp.1–12, 2005
- [16] 陳明哲, 片山直登, 久保幹雄: “容量制約をもつ多品種フロー輸送ネットワーク設計問題に対する容量スケールリング法”, *日本物流学会誌*, No.14, pp.85–92, 2006
- [17] D. Kim and P. Pardalos: “Dynamic slope scaling and trust interval techniques for solving concave piecewise linear network flow problems”, *Networks*, Vol.35, pp.216–222, 2000
- [18] K. L. Croxton, B. Gendron and T. L. Magnanti: “Variable disaggregation in network flow problems with piecewise linear costs”, *Operations Research*, Vol.55, pp.146–157, 2007