

アセットバランスを考慮したサービスネットワーク設計問題のMIP近傍探索法

A MIP Neighborhood Search Algorithm for Service Network Design Problem with Asset-Balance Requirements

片山 直登 流通経済大学 流通情報学部

平成30年2月7日

1 はじめに

サービスネットワーク設計問題は、ネットワーク上のサービスやスケジューリングに関連した計画問題であり、ネットワークの設計と共にサービス資源であるアセットの割当を求める問題である。アセットは配送車や乗務員などであり、これらのアセットの時間的・空間的なバランスを図ることが必要となる。サービスネットワーク設計問題では、アセットをネットワーク上のアークに対応させ、アセットのバランスを取る、すなわち、ノードに出入りするアークの次数が一致するというアセットバランス条件を付加することで、アセットの時間的・空間的なバランスを図ることが行われる。

このようなアセットのバランスに焦点をおいたサービスネットワーク設計問題をアセットバランスを考慮したサービスネットワーク設計問題 (Service Network Design with Asset-Balance Requirements : *SNDAB*) とよぶ。この設計問題は、アーク上の資源であるアセットの容量をもつネットワーク上で、かつノードにおけるアセットの入出次数のバランスを保ちながら、全体の費用が最小となるようなモノが移動するパス、およびアセットの配置を求める問題である。考慮する費用はアセットの有無に応じて生じる固定費用であるアセット費用と品種のフロー量に関係して生じる変動費用であるフロー費用であり、これらの和を最小化する。サービスネットワーク設計問題に関しては、Crainic (2000, 2003) が詳しい解説を行っている。

本研究では、Pedersen et al. (2009) が提示したアセットバランスを考慮したサービスネットワーク設計問題を対象とする。この設計問題は、容量制約をもつネットワーク設計問題にアセットバランス条件を付加した問題である。この問題に対して、従来から多くの研究が行われている。Pedersen et al. (2009) はタブー探索法による近似解法、Chouman and Crainic (2011) は数理計画ソルバーとタブー探索法を組み合わせた近似解法、さらに?は切除平面法を用いたメタヒューリスティクスを示

している． Bai et al. (2010) はガイド付き局所探索法， Bai et al. (2012) はタブー補助ガイド付き局所探索法， さらに Bai et al. (2018) は k ノード隣接を用いたタブー探索法およびガイド付き局所探索法を示している． また， Vu et al. (2013) は3段階のメタヒューリスティクス解法， Katayama (2015) および片山直登 (2012) は容量スケールリングと局所分枝法を組み合わせた解法を開発している． 一方， 容量制約をもつ設計問題に対する高速な近似解法として， 片山直登 (2017) は容量スケールリング法と MIP 近傍探索法を組み合わせた高速解法を提案しており， 精度の高い近似解を短時間で生成することに成功している．

本研究では， アセットバランスを考慮したサービスネットワーク設計問題 *SN DAB* に対して容量スケールリング法と MIP 近傍探索法を組合せた高速な近似解法を提案する．

2 問題の定式化

ノード集合 N ， 向きをもつアーク集合 A ， ネットワーク上で移動する品種集合 K ， アーク (i, j) 上においてアセットを配置したときに発生する非負のアセット費用 f_{ij} ， アーク (i, j) 上を移動する品種 k に関する非負の単位当たりのフロー費用 c_{ij}^k ， アーク (i, j) 上のアセット容量 C_{ij} ， 品種 k の始点 O^k と終点 D^k 間を流れる品種 k の需要 d^k が与えられるものとする．

アーク (i, j) 上を移動する品種 k のフロー量を表す非負の連続変数であるアークフロー変数を x_{ij}^k とし， アーク (i, j) 上にアセットを配置するとき 1， そうでないとき 0 である 0-1 離散変数であるアセットデザイン変数を y_{ij} とする． このとき， *SN DAB* のアークフローによる定式化 *SN DABA* は次のように表すことができる．

(*SN DABA*)

$$\Phi = \text{最小化} \quad \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k x_{ij}^k + \sum_{(i,j) \in A} f_{ij} y_{ij} \quad (1)$$

$$\text{条件} \quad \sum_{i \in N_n^+} x_{in}^k - \sum_{j \in N_n^-} x_{nj}^k = \begin{cases} -d^k & \text{if } n = O^k \\ d^k & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N, k \in K \quad (2)$$

$$\sum_{i \in N_n^+} y_{in} - \sum_{j \in N_n^-} y_{nj} = 0 \quad \forall n \in N \quad (3)$$

$$\sum_{k \in K} x_{ij}^k \leq C_{ij} y_{ij} \quad \forall (i, j) \in A \quad (4)$$

$$x_{ij}^k \leq d^k y_{ij} \quad \forall k \in K, (i, j) \in A \quad (5)$$

$$x_{ij}^k \geq 0 \quad \forall k \in K, (i, j) \in A \quad (6)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (7)$$

(1) 式は目的関数であり， フロー費用とアセット費用の総和を最小化する． なお， Φ は目的関数の最適値である． (2) 式はアークフロー保存式である． この式は， ノー

ドに流入するフローと流出するフローの差が、品種 k の始点であれば $-d^k$ 、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、各品種について、必ず始点から終点まで需要が移動することを保証する。(3) 式はアセットバランス式であり、ノードに流入するアーク上に配置されるアセットの回数と流出するアセットの回数が一致することを表す。(4) 式は、アセット容量制約式である。この式は、アーク (i, j) 上にアセットが配置されるときはアーク上を移動するフロー量の合計はアセット容量以下であり、アセットが配置されないときは 0 であることを表す。(5) 式は、アーク上の品種とその需要に関する強制制約式である。この式は、アーク (i, j) 上のアセットが配置されるときはアーク上を移動する品種 k のフロー量の合計は品種 k の需要以下であり、アセットが配置されないときは 0 であることを表す。(6) 式はアークフロー変数の非負条件、(7) 式はアセットデザイン変数の 0-1 条件である。

P^k を品種 k の取りうるパス集合、パス p 上を移動する品種 k のフロー量を表す非負の連続変数であるパスフロー変数を z_p^k 、パス p にアーク (i, j) が含まれるとき 1、そうでないとき 0 を表す定数を δ_{ij}^p とする。このとき、アーク集合 A 、パス集合 P 、アーク容量 C に対する $SNDAB$ のパスフローによる定式化 $SNDABP(A, P, C)$ は次のように表すことができる。

($SNDABP(A, P, C)$)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k \sum_{p \in P^k} \delta_{ij}^p z_p^k + \sum_{(i,j) \in A} f_{ij} y_{ij} \quad (8)$$

$$\text{条件} \quad \sum_{p \in P^k} z_p^k = d^k \quad \forall k \in K \quad (9)$$

$$\sum_{i \in N_n^+} y_{in} - \sum_{j \in N_n^-} y_{nj} = 0 \quad \forall n \in N \quad (10)$$

$$\sum_{k \in K} \sum_{p \in P^k} \delta_{ij}^p z_p^k \leq C_{ij} y_{ij} \quad \forall (i, j) \in A \quad (11)$$

$$\sum_{p \in P^k} \delta_{ij}^p z_p^k \leq d^k y_{ij} \quad \forall k \in K, (i, j) \in A \quad (12)$$

$$z_p^k \geq 0 \quad \forall p \in P^k, k \in K \quad (13)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (14)$$

(8) 式は目的関数であり、フロー費用とアセット費用の総和を最小化する。(9) 式は、品種 k のパスフローの合計が品種 k の需要 d^k に一致することを表すパスフロー保存式である。(10) 式はアセットバランス式であり、ノードに流入するアーク上に配置されるアセットの回数と流出するアセットの回数が一致することを表す。(11) 式はアセット容量制約式であり、(12) 式は強制制約式である。(13) 式はパスフロー変数の非負制約であり、(14) 式はアセットデザイン変数の 0-1 条件である。

ここで、 $SNDABP(A, P, C)$ の 0-1 条件を線形緩和した問題を $SNDABPL(A, P, C)$ とおく。

アセットバランス式である (3) 式や (10) 式を考慮しない場合，この問題は容量制約をもつネットワーク設計問題に帰着される．アーク集合 A ，パス集合 P ，アーク容量 C に対する容量制約をもつネットワーク設計問題の定式化 $CND(A, P, C)$ は次のように表される．

($CND(A, P, C)$)

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k \sum_{p \in P^k} \delta_{ij}^p z_p^k + \sum_{(i,j) \in A} f_{ij} y_{ij} \quad (15)$$

$$\text{条件} \quad \sum_{p \in P^k} z_p^k = d^k \quad \forall k \in K \quad (16)$$

$$\sum_{k \in K} \sum_{p \in P^k} \delta_{ij}^p z_p^k \leq C_{ij} y_{ij} \quad \forall (i, j) \in A \quad (17)$$

$$\sum_{p \in P^k} \delta_{ij}^p z_p^k \leq d^k y_{ij} \quad \forall k \in K, (i, j) \in A \quad (18)$$

$$z_p^k \geq 0 \quad \forall p \in P^k, k \in K \quad (19)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (20)$$

3 近似解法

$SNDABA$ や $SNDABP(A, P, C)$ は，アセットバランス式以外は容量制約をもつネットワーク設計問題に一致する．問題が類似していることから，容量制約をもつネットワーク設計問題に対する近似解法である容量スケールリング法とMIP近傍探索法(片山直登 2017)を適用する．

3.1 容量スケールリング法と初期実行可能解

$SNDABA$ や $SNDABP(A, P, C)$ はアークに対する0-1変数を含む最適化問題であり，大規模な問題では最適解または適切な近似解を算出することが困難である．そこで，容量スケールリング法を用いて，対象となるアークから近似解に含まれるであろうアークを適切に絞り込む．

容量スケールリング法は，デザイン変数に対する線形緩和問題を解き，そのデザイン変数解の値とスケールリングパラメータに従ってアーク容量を変化させ，0 または1のデザイン変数解を導出するものである．容量スケールリングでは，少ない繰り返し回数で多くのデザイン変数が0に収束することが知られている．そこで，アーク集合 A に対して，0に収束しないデザイン変数の数が終了判定数 α 以下になるまで容量スケールリング法を適用し，0に収束しないデザイン変数のみを選定する．この処理により，わずかな計算量で多くのデザイン変数を除外することができ，効率的に問題規模を縮小することが可能となる． $SNDABP(A, P, C)$ に対する容量スケールリング法の詳細は，片山直登 (2012) を参照のこと．

容量スケールリング法により得られたアセットデザイン変数は、すべて0または1に収束する訳ではない。また、収束したアセットデザイン変数や切り上げたアセットデザイン変数を用いた場合、アセットバランス式を満足することは保証されない。そこで、容量スケールリング法により得られた小数を含むアセットデザイン解より、 $SNDABP(A, P, C)$ の実行可能解を導出する。

容量スケールリング法により0に収束していないアセットデザイン解に対応するアーク集合を \bar{A} とする。また、列生成法にて容量スケールリング法を解いた場合に生成されたパス集合を \bar{P} とする。アセットバランス式である(10)式を緩和し、 \bar{A} 、 \bar{P} と C に対する $CNDP(\bar{A}, \bar{P}, C)$ をMIPソルバーを用いて解く。 $CNDP(\bar{A}, \bar{P}, C)$ は相対的に小規模な問題になるため、MIPソルバーの分枝限定法などを用いて適当な近似解を算出することができる。得られた $CNDP(\bar{A}, \bar{P}, C)$ のアセットデザイン変数解を \hat{y} とおく。

アセットバランス式を緩和しているため、解 \hat{y} は $SNDABP(A, P, C)$ の実行可能解とは限らない。そこで、 $\hat{y}_{ij} = 1$ であるアセットに対応するアーク集合を \hat{A} としたとき、次の \hat{A} に関するアセットバランス問題 $ABL(\hat{A})$ を解くことにより、アセットバランス式を満足する解を求める。

$(ABL(\hat{A}))$

$$\text{最小化} \quad \sum_{(i,j) \in A \setminus \hat{A}} f_{ij} y_{ij} \quad (21)$$

$$\text{条件} \quad \sum_{i \in N_n^+} y_{in} - \sum_{j \in N_n^-} y_{nj} = 0 \quad \forall n \in N \quad (22)$$

$$y_{ij} = 1 \quad \forall (i, j) \in \hat{A} \quad (23)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \setminus \hat{A} \quad (24)$$

(21)式は、 \hat{A} に含まれるアークとアセットバランス式を満足するために新たに付加されたアークに対するアセット費用の合計である。(23)式は、 \hat{A} に含まれるアークのアセットデザイン変数を1に固定する式である。

$ABL(\hat{A})$ を解くことによって、 $CNDP(\bar{A}, \bar{P}, C)$ の解を含み、アセットバランス式を満足し、アセット費用が最小となるアセットデザイン解を求めることができる。 $ABL(\hat{A})$ の最適解が求められれば、この解を \tilde{y} とおく。 \tilde{y} は $SNDABP(A, P, C)$ や $SNDABA$ の実行可能解となる。 $\tilde{y}_{ij} = 1$ であるアセットデザイン変数に対応するアーク集合を \tilde{A} とおく。続いて、 $SNDABP(\tilde{A}, \bar{P}, C)$ をMIPソルバーを用いて解き、得られた解を \bar{y} とおく。なお、 $ABL(\hat{A})$ が実行不可能であれば、 \hat{A} を \tilde{A} 、 \hat{y} を \bar{y} とする

\bar{y} は $SNDABP(A, P, C)$ の実行可能解となり、得られた目的関数値は上界値となる。なお、 $ABL(\hat{A})$ が実行不可能であれば $SNDABP(A, \bar{P}, C)$ を解き、これが実行不可能であれば \hat{y} を \bar{y} とする。

容量スケールリング法と初期実行可能解探索のアルゴリズムをAlgorithm2に示す。

3.2 MIP 近傍探索法

前節で求めた近似解 \bar{y} を初期の暫定解とし、*SNDABA* において解 \bar{y} の近傍を MIP ソルバーを用いて探索していく。

適当な初期解から探索範囲を限定する条件を付加した問題を MIP ソルバーを用いて解く解法として局所分枝法 (Fischetti and Lodi 2003) が知られており、多くのネットワーク設計問題に適用されている。*SNDAB* に対しては容量スケール法と組み合わせた解法 (Katayama 2015) が提案され、計算時間は必要となるものの高精度な解を提供している。

本研究では、別のタイプの探索範囲を限定する条件を付加する MIP 近傍探索法を適用する。始めに、*SNDABA* に次の制約式を追加する。

$$\sum_{(i,j) \in A | \bar{y}_{ij}=1} y_{ij} \leq L - 1 \quad (25)$$

ここで、 L は暫定解 \bar{y} において $\bar{y}_{ij} = 1$ であるアセットデザイン変数の数である。(25) 式は、暫定解で $\bar{y}_{ij} = 1$ に対するアセットから少なくとも 1 つのアセットはネットワークから取り除くことを表しており、実行可能領域から暫定解を排除することができる。(25) 式が等式である場合はデリート型の近似解法になる。しかし、1 つのアセットのみを削除するとアセットバランス式が成立しないため、不等号としている。

また、現在までの最良の上界値を UB とおき、次の式も追加する。

$$\Phi < UB \quad (26)$$

(25) 式と (26) 式により探索済みの解を排除し、解の循環を防ぐことができる。なお、現在までの最良値よりも良い上界値が存在しなければ、問題は実行不可能となる。

計算に制限時間 T を設け、MIP ソルバーを用いて *SNDABA* に (25) 式と (26) 式を付加した問題を解く。実行可能解が得られた場合は、改善された解が探索されたことになる。その場合、得られた解を新たな暫定解 \bar{y} とする。そうでない場合は、暫定解は更新しない。

暫定解が更新された場合、(25) 式と (26) 式を取り除き、更新された暫定解 \bar{y} に対する (25) 式と (26) 式を加え、さらに次の制約式を追加する。

$$\sum_{(i,j) \in A | \bar{y}_{ij}=1} y_{ij} \geq L - M \quad (27)$$

(27) 式は、暫定解で選択されたアセットを高々 M 個のアセットをネットワークから取り除くことを表す。なお、 M は正の整数で、近傍の範囲である。 M が大きければ、条件を付加した *SNDABA* の実行可能領域は広くなる。一方、 M が小さければ実行可能領域は狭くなるため、相対的に短時間で実行可能解を探索できる可能性が高まることになる。

続いて，MIP ソルバーを用いて，一定時間内で解を求める．暫定解より良い解が求められれば，暫定解を更新する．続いて，追加した3本の制約式を削除して，更新された暫定解に対応した3本の制約式を追加し，近傍探索を繰り返す．

M 近傍において，実行不可能または暫定解よりも良い解が無いと判断できた場合は，探索を終了する．一方，一定時間内に暫定解より良い解を算出することができず暫定解が更新されない場合は，計算時間内で実行不可能とは判断できないが実行可能解が算出できていない状態である．そこで， $M := \lfloor M/\beta \rfloor$ として M を減少させ，探索範囲を縮小する． $M > 0$ である間，同様の探索を繰り返す．ここで， $\beta (> 1)$ は M の変更基準である．

このような探索法を MIP 近傍探索法とよぶ．従来の局所分枝法の制約と比べると，削除するアーク数には制約があるが，追加するアークには制限が無いのが特徴である．また，局所分枝法では領域を二分割しながら分枝・限定操作を行っていくが，この近傍探索法は単に近傍探索と解の移動を繰り返すのみであり，得られる上界値が短調減少となる貪欲的な探索法となる．

4 数値実験

容量制約をもつネットワーク設計問題で用いられるベンチマーク問題である C 問題の内，*SNDAB* で用いられる 24 問 (Pedersen et al. 2009) に対して，数値実験を行った．

数値実験で使用した設定した機器等は以下の通りである．

- 使用 OS および言語 : UBUNTU 17.04, C++
- 最適化ソルバー : Gurobi 7.5
- CPU AMD Ryzen7 1800X 3.6GHz 8Cores, RAM 16GByte
- 使用コア数 : 容量スケールリング 1 コア, MIP 近傍探索 8 コア

また，数値実験で使用した設定したパラメータは以下の通りである．

- スケールリングパラメータ λ : 0~0.1
- スケールリング法の終了判定数 α : 200
- 近傍 M : 5
- 近傍 M の変更基準 β : 2
- MIP 近傍探索における MIP ソルバー計算時間の制限時間 T : 60 秒

近似解の誤差を算出するために，MIP ソルバーである Gurobi により，定式化 *SNDABA* を 30 時間解き，下界値を算出した．同時に上界値も算出した．

C 問題に対しては多くの研究が行われ，その結果が公開されている．ここでは，タブー探索法 (TS) (Pedersen et al. 2009)，並列タブー探索法 (PTS) (Pedersen et al. 2009)，ガイド付き局所探索法 (GLS) (Bai et al. 2010)，MIP タブー探索法 (MTS) (Chouman and Crainic 2011)，タブー補助ガイド付き局所探索法 (TAGLS) (Bai et al. 2012)，3 段階メタヒューリスティクス (TSM) (Vu et al. 2013)，切除平面メタ

ヒューリスティクス(CP)(Chouman and Crainic 2016), 容量スケールリング・限定分枝限定法(CSRBB)(Katayama 2015), 容量スケールリング・局所分枝法(CSLB)(Katayama 2015), k ノード近傍ハイブリッド法(kNH)(Bai et al. 2018)の結果を示す。これらに加え, 本研究である容量スケールリング・MIP近傍探索法(CSMNS), およびパラメータチューニングをした容量スケールリング・MIP近傍探索法(CSMNST)の結果を示す。前者では, Algorithm1内のスケールリングパラメータ λ を0.048と設定した。後者は, 問題ごとにスケールリングパラメータ λ を変化させた中の最良値である。なお, 誤差(Gaps)は「(各解法の上界値-下界値)/下界値」とし, 平均誤差はこれらの平均値である。

表1にC問題に対する上界値の平均誤差を示す。従来の研究では, タブー探索法は平均誤差5.65%, 並列タブー探索法は4.29%, ガイド付き局所探索法は4.74%と大きく, 誤差が4%を超えている。また, MIPタブー探索法は平均誤差1.48%, タブー補助ガイド付き局所探索法は2.29%, 切除平面メタヒューリスティクスは2.04%, k ノード近傍ハイブリッド法は1.05%であり, 1~2%の平均誤差となっている。一方, 3段階メタヒューリスティクスは平均誤差0.92%, 容量スケールリング・限定分枝限定法は0.66%, 容量スケールリング・局所分枝法は0.39%と誤差が1%未満となり, 良好な解を算出している。容量スケールリング・局所分枝法は, 近似解法の中では最も誤差が小さい解を算出している。なお, MIPソルバーであるGurobiは平均誤差0.36%であり, 最も誤差が小さいが, 後述のように膨大な計算時間を必要とする。

一方, 本研究である容量スケールリング・MIP近傍探索法は平均誤差0.69%であり, パラメータをチューニングした容量スケールリング・MIP近傍探索法は0.52%であった。容量スケールリング・MIP近傍探索法は, 容量スケールリング・限定分枝限定法および容量スケールリング・局所分枝法よりそれぞれ0.03%, 0.30%劣っているが, 3段階メタヒューリスティクスよりも0.23%優れている。また, チューニングした容量スケールリング・MIP近傍探索法は, 容量スケールリング・局所分枝法より0.13%劣っているが, 容量スケールリング・限定分枝限定法よりも0.14%優れている。

表2に, 個別問題の上界値を示す。なお, LBは下界値または最適値であり, 太字は最適値, 斜体文字は最良値を表している。容量スケールリング・局所分枝法では8問の最適値, 最適値を除く1問の最良値を算出している。 k ノード近傍ハイブリッド法では6問の最良値, 1問の最良値である。また, パラメータをチューニングした容量スケールリング・MIP近傍探索法では, 6問の最適値, 2問の最良値を算出している。なお, Gurobiでは15問の最適値, 6問の最良値, 24問の内の21問の最良値を算出している。

表3にC問題に対する個々の計算時間と平均計算時間を示す。従来の研究の計算時間は, 各論文に掲載しているものであり, 使用しているコンピュータが異なっているため, 計算時間を直接比較することはできない。使用しているコンピュータは以下の通りである。

- LB, Giurobi : AMD Ryzen 1800x (8-Core, 3.6GHz)
- タブー探索法 : Intel Pentium 4(2.26GHz)
- 並列タブー探索法 : Intel Pentium 4(2.26GHz)

- ガイド付き局所探索法 : Intel Core 2 (1.8GHz)
- タブー補助ガイド付き局所探索法 : Intel Core 2 (1.8GHz)
- MIP タブー探索法 : AMD Opteron(Dual-Core)
- 3段階メタヒューリスティクス : AMD Opteron(Dual-Core, 2.4GHz)
- 切除平面メタヒューリスティクス : AMD Opteron(Dual-Core)
- 容量スケールリング・限定分枝限定法, 容量スケールリング・局所分枝法 : Intel Pentium i7(4-Core, 3.4GHz)
- k ノード近傍ハイブリッド法 : Intel Core 2 (2.0GHz)

平均誤差の最も小さい Gurobi では 30 時間の制限を付けているため, 最適解を求められない場合は 108000 秒となり, 平均計算時間は 52740.4 秒で 14 時間を越えている. タブー探索法, 並列タブー探索法, ガイド付き局所探索法および MIP タブー探索法は 3600 秒, タブー補助ガイド付き局所探索法は 2400 秒, k ノード近傍ハイブリッド法は 7200 秒の計算時間を設定しており, 良好な解を算出するためには大きな計算時間が必要としている. 3段階メタヒューリスティクス, 切除平面メタヒューリスティクスおよび容量スケールリング・局所分枝法はそれぞれ 7785 秒, 3581.4 秒, 8637.1 秒であり, 1 時間から 3 時間程度となり, これらも大きな計算時間を必要としている. また, 容量スケールリング・限定分枝限定法の平均計算時間は 524.1 秒と他の解法に比べて短い.

一方, 本研究である容量スケールリング・MIP 近傍探索法の平均計算時間は 277.3 秒, パラメータをチューニングした容量スケールリング・MIP 近傍探索法の平均計算時間は 253.1 秒であり, 他の解法に比べて, 大幅に計算時間を短縮することができている. なお, 後者は, 実際にはパラメータ選定のための事前の計算時間が必要である. 使用しているコンピュータが異っているにしても, 従来の研究に比べ, 精度を保ちながら大幅に計算時間が短縮できることが分かる.

5 おわりに

本研究では, アセットバランスを考慮したサービスネットワーク設計問題に対して, 容量スケールリング法と MIP ソルバーによる近傍探索を組み合わせた高速で精度の高い MIP 近傍探索法を提案した. また, ベンチマーク問題である C 問題に対して, 数値実験を行い, 従来の研究との比較を行った.

本研究では, 非分割フローを考慮した容量制約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と MIP ソルバーによる近傍探索を組み合わせた高速で精度の高い MIP 近傍探索法を提案した. また, ベンチマーク問題である C 問題に対して, 数値実験を行い, 従来の研究との比較を行った.

本研究で提案した解法は, 従来の容量スケールリング・局所分枝法と比較すると誤差はわずかながら大きい, 容量スケールリング・局所分枝法よりも 1/30 程度と大幅に短い計算時間で解を算出することができた. また, 容量スケールリング・局所探索法以外のいずれの近似解法よりも, 短い計算時間で精度の高い近似解を算出することができた.

本研究は科学研究費基盤研究C(課題番号17K01268)による成果の一部である。

参考文献

- R. Bai, G. Kendal, and J. Li. An efficient guided local search approach for service network design problem with asset balancing. *ICLSIM*, 1:110–115, 2010.
- R. Bai, G. Kendal, R. Qu, and J. Atkin. Tabu assisted guided local search approaches for freight service network design. *Information Science*, 189(15):266–281, 2012.
- R. Bai, J.R. Woodward, N. Subramanian, and J. Carlidge. Optimisation of transportation service network using k -node large neighbourhood search. *Computers & Operations Research*, 89:193–205, 2018.
- M. Chouman and T. G. Crainic. MIP-based tabu search for service network design with design-balanced requirements. Technical Report CIRRELT-2011-68, Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation, Université de Montréal, 2011.
- M. Chouman and T. G. Crainic. Cutting-plane matheuristic for service network design with design-balanced requirements. *Transportation Science*, 49(1):99–113, 2016.
- T. G. Crainic. Service network design in freight transportation. *European Journal of Operational Research*, 122(2):272–288, 2000.
- T. G. Crainic. Long-haul freight transportation. In R. W. Hall, editor, *Handbook of Transportation Science*, pages 451–516. Kluwer Academic Publishers, 2003.
- M. Matteo Fischetti and A. Lodi. Local branching. *Mathematical Programming*, 98(1-3):23–47, 2003.
- N. Katayama. A combined matheuristics for service network design problem. *The International Federation of Logistics and SCM Systems*, 8:11–20, 2015.
- M. B. Pedersen, T. G. Crainic, and O. B. G. Madsen. Models and tabu search metaheuristics for service network design with asset-balance requirements. *Transportation Science*, 43:158–177, 2009.
- D. M. Vu, T. G. Crainic, and M. Toulous. A three-stage matheuristic for the capacitated multi-commodity fixed-cost network design with design-balance constraints. *Journall of Heuristics*, 19:757–795, 2013.
- 片山直登. アセットバランスを考慮したサービスネットワーク設計問題. *流通経済大学流通情報学部紀要*, 17(1):29–50, 2012.
- 片山直登. 容量制約をもつネットワーク設計問題に対するMIP近傍探索法. *流通経済大学流通情報学部紀要*, 22(1):1–18, 2017.

Algorithm 1: Capacity Scaling

Set \bar{P} , α , λ , ITE_{min} , ITE_{max} , T ;
 $l \leftarrow 1$, $C^1 \leftarrow C$;
repeat
 Solve $SNDABPL(A, \bar{P}, C^l)$;
 Get the solution \tilde{y} of $SNDABPL(A, \bar{P}, C^l)$;
 Add paths to \bar{P} by Column Generation;
 $n \leftarrow 0$;
 for $(i, j) \in A$ **do**
 $C_{ij}^{l+1} \leftarrow \lambda C_{ij}^l \tilde{y}_{ij} + (1 - \lambda) C_{ij}^l$;
 if $\tilde{y}_{ij} > 0$ **then** $n \leftarrow n + 1$;
 end
until $l \geq ITE_{min}$ and $n \leq \alpha$, or $l \geq ITE_{max}$;
 $\bar{A} \leftarrow \emptyset$;
for $(i, j) \in A$ **do**
 if $\tilde{y}_{ij} > 0$ **then** $\bar{A} \leftarrow \bar{A} \cup \{(i, j)\}$;;
end
Solve $CNDP(\bar{A}, \bar{P}, C)$ within time T and get the solution \hat{y} ;
 $\hat{A} \leftarrow \emptyset$;
for $(i, j) \in A$ **do**
 if $\hat{y}_{ij} > 0$ **then** $\hat{A} \leftarrow \hat{A} \cup \{(i, j)\}$;
end
Solve $ABL(\hat{A})$;
if the feasible solution \tilde{y} of $ABL(\hat{A})$ is found **then**
 $\tilde{A} \leftarrow \emptyset$;
 for $(i, j) \in A$ **do**
 if $\tilde{y}_{ij} = 1$ **then** $\tilde{A} \leftarrow \tilde{A} \cup \{(i, j)\}$;
 end
else
 $\tilde{A} = A$; $\tilde{y} \leftarrow \tilde{y}$;
end
Solve $SNDABP(\tilde{A}, \bar{P}, C)$ within time T ;
if the feasible solution \bar{y} of $SNDABP(\tilde{A}, \bar{P}, C)$ is found **then**
 Get the upper bound UB of $SNDABP(\tilde{A}, \bar{P}, C)$;
else
 Solve $SNDABP(A, \bar{P}, C)$ within time T ;
 if the feasible solution \bar{y} of $SNDABP(A, \bar{P}, C)$ is found **then**
 Get the upper bound UB of $SNDABP(A, \bar{P}, C)$;
 else
 $\bar{y} \leftarrow \tilde{y}$, $UB \leftarrow \infty$;
 end
end
Return \bar{y}, UB ;

Algorithm 2: MIP Neighborhood Search(\bar{y}, UB)

Set β, M, T ;
Add equations (25) and (26) to *SNDAB* associated with the solution \bar{y} ;
Solve *SNDAB* within time T ;
if the solution \tilde{y} of *CNDA*(A) is found **then**
 Get the upper bound UB_{neigh} of *SNDAB*;
 $\bar{y} \leftarrow \tilde{y}$;
 $UB \leftarrow UB_{neigh}$;
end
Delete equations (25) and (26) from *SNDAB*;
repeat
 Add equations (25), (26) and (27) to *SNDAB* associated with the current
 solution \bar{y} ;
 Solve *SNDAB* within time T ;
 if *SNDAB* has no feasible solution **then**
 break;
 else
 if the solution \tilde{y} of *SNDAB* is found **then**
 Get the upper bound UB_{neigh} of *SNDAB*;
 $\bar{y} \leftarrow \tilde{y}$;
 $UB \leftarrow UB_{neigh}$;
 else
 $M \leftarrow \lfloor M/\beta \rfloor$;
 end
end
Delete equations (25), (26) and (27) from *SNDAB*;
until $M = 0$;
Return \bar{y}, UB ;

表 1: Average Gaps for C-Category Problems(%)

Gurobi	TS	PTS	GLS	MTS	TAGLS	TSM
0.36	5.65	4.29	4.74	1.48	2.29	0.92
CP	CSRBB	CSLB	kNH	CSMNS	CSMNST	
2.04	0.66	0.39	1.05	0.69	0.52	

表 2: Results for C-Category Problems

Problem	LB	Gurobi	TS	PTS	GLS	MTS	TAGLS	TSM	CP	CSRBB	CSLB	kNH	CSMNS	CSMNST
20/230/200/VL	97273.5	97273.5	102919	101345	101267	98421	98760	97274	98699	97597	97273.5	97274	97273.5	97273.5
20/230/200/FL	139395	139395	150764	148384	143017	141744	142213	139395	141744	139831	139395	139395	139395	139395
20/230/200/VT	100221	100221	103371	103371	103428	101244	102137.3	100720	103103	100530	100221	100478	100558	100478
20/230/200/FT	138962	138962	149942	144766	143446	141130	141802	138962	142638	139253	138994	138994	139414	138962
20/300/200/VL	77436	77436	82533	80269	80183	78576	78787	77584	79953	77584	77502	77463	77584	77502
20/300/200/FL	118800.1	119390.5	128757	126258	126097	121106	121773	119987	120979	119324.3	119259	119259	119554	119554
20/300/200/VT	76207.5	76207.5	78571	78444	77839	76545	77066	76450	76545	76207.5	76207.5	76207.5	76207.5	76207.5
20/300/200/FT	111333.2	111333.2	116338	116338	116712	113412	114465	111776	113412	111462	111462	111475	112176.5	112176.5
30/520/100/VL	54683	54683	55981	55786	55437	55159	55422	54783	55733	54733	54683	54683	54683	54683
30/520/100/FL	97026.3	98084	104533	101612	100999	101129	100290	100098	104235	99193	98170.8	98595	99734	98381
30/520/100/VT	53023	53023	54493	54092	53644	53224	53744	53035	53224	53246	53023	53030	53029	53029
30/520/100/FT	99967.2	101160	105167	104702	106096	104426	103996	101412	106251	102043	101131	101576	101360	101130
30/520/400/VL	114546.8	114546.8	119735	118071	119344	115477	116196	115528	115220	114688.2	114565.1	114891	114870.3	114688.2
30/520/400/FL	151270.8	152675.9	162360	160979	163182	153943	154941	153409	153737	152893.4	152776.3	154336	153017.4	153017.4
30/520/400/VT	116508.7	116508.7	120421	120421	122877	116959	118335.7	117226	117056	116670.9	116616.4	117141	116681.2	116681.2
30/520/400/FT	153681	154747.7	161978	161987	166488	155863	157939.6	155906	155942	155121	154820.2	157655	155766	154984.8
30/700/100/VL	48693	48693	49429	49429	49465	49139	49221	48807	49268	48708	48693	48693	48693	48693
30/700/100/FL	61263	61263	63889	63292	62936	62000	62055	61408	62267	61528	61298	61433	61948	61667
30/700/100/VT	46750	46750	48202	47487	47518	46865	47518	46812	46928	47137	46750	46750	47062	47033
30/700/100/FT	56131	56131	58204	57187	58559	56599	57571	56237	57701	56609	56169	56207	56235	56180
30/700/400/VL	98923.7	99252.3	103932	103932	104534	99588	101609.5	100589	99458	99332.5	99314	101316	99796	99351
30/700/400/FL	134825.8	137711.9	157043	148114	152580	139088	142563.2	141037	139607	137889	137724.2	145185	138026.4	137989.6
30/700/400/VT	96662.5	97200.3	103085	103085	103581	97901	98656.8	97875	97737	97425.5	97278	99133	97352.6	97349.5
30/700/400/FT	130839.5	132297	141917	138609	142575	132999	135777.5	133686	132855	132486.7	132486.7	134122	132370.2	132200.5

表 3: Computation Time for C-Category Problems(Seconds)

Problem	LB/Gurobi	TS	PTS	GLS	MTS	TAGLS	TSM	CP	CSRBB	CSLB	kNH	CSMNS	CSMNST
20/230/200/VL	5988.5	3600	3600	3600	3600	2400	5460	635	91.5	5579.7	7200	440.2	297.9
20/230/200/FL	10323.2	3600	3600	3600	3600	2400	5520	10260	219.3	8836.8	7200	328.5	324.1
20/230/200/VT	1158.6	3600	3600	3600	3600	2400	4680	544	46.3	4223.3	7200	309.4	308.3
20/230/200/FT	47822.6	3600	3600	3600	3600	2400	5040	634	556.0	9577.5	7200	259.3	256.3
20/300/200/VL	23597.1	3600	3600	3600	3600	2400	9720	432	78.8	8596.3	7200	171.8	170.3
20/300/200/FL	108000.0	3600	3600	3600	3600	2400	6600	1027	421.1	9056.2	7200	385.6	318.7
20/300/200/VT	375.0	3600	3600	3600	3600	2400	5400	283	36.5	1976.6	7200	211.5	160.1
20/300/200/FT	90749.8	3600	3600	3600	3600	2400	8460	958	605.5	7863.3	7200	129.3	127.9
30/520/100/VL	174.7	3600	3600	3600	3600	2400	5100	225	14.9	1606.7	7200	150.0	92.2
30/520/100/FL	108000.0	3600	3600	3600	3600	2400	7080	1630	412.7	10337.4	7200	218.2	209.7
30/520/100/VT	664.7	3600	3600	3600	3600	2400	6180	82	10.2	5332.9	7200	238.8	167.7
30/520/100/FT	108000.0	3600	3600	3600	3600	2400	9900	870	422.3	21799.0	7200	359.1	346.0
30/520/400/VL	31637.7	3600	3600	3600	3600	2400	13260	6830	193.7	9283.5	7200	291.2	279.7
30/520/400/FL	108000.0	3600	3600	3600	3600	2400	8760	10529	806.3	5213.6	7200	401.6	362.0
30/520/400/VT	21570.5	3600	3600	3600	3600	2400	12600	5413	82.3	12285.8	7200	271.4	267.4
30/520/400/FT	108000.0	3600	3600	3600	3600	2400	8160	10696	1292.9	11929.2	7200	353.3	354.3
30/700/100/VL	65.8	3600	3600	3600	3600	2400	4440	101	8.1	381.2	7200	66.6	61.0
30/700/100/FL	53565.2	3600	3600	3600	3600	2400	9180	631	68.7	9823.1	7200	92.3	83.2
30/700/100/VT	1463.5	3600	3600	3600	3600	2400	8100	125	16.7	8587.2	7200	72.4	72.0
30/700/100/FT	4612.0	3600	3600	3600	3600	2400	9120	309	51.4	8211.8	7200	399.3	331.3
30/700/400/VL	108000.0	3600	3600	3600	3600	2400	8160	7893	497.5	11660.6	7200	301.0	288.0
30/700/400/FL	108000.0	3600	3600	3600	3600	2400	4800	15723	1317.7	7745.6	7200	418.7	422.5
30/700/400/VT	108000.0	3600	3600	3600	3600	2400	13320	8405	1825.4	14481.3	7200	384.9	367.4
30/700/400/FT	108000.0	3600	3600	3600	3600	2400	7800	10953	3503.5	12900.6	7200	400.6	406.1
Average	52740.4	3600	3600	3600	3600	2400	7785	3581.4	524.1	8637.1	7200	277.3	253.1