ホップ数を考慮した容量制約をもつネットワーク設計問題

Hop-Constrained Capacitated Network Design Problem

片山直登：流通経済大学 流通情報学部 教授

略　歴

1982年早稲田大学理工学部卒業。1988年同大学院博士後期課程単位取得退学。博士（物流情報学）。早稲田大学理工学部助手、金沢工業大学工学部講師、1996年流通経済大学流通情報学部助教授を経て、2002年から現職。

［要約］　本研究では、ホップ数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、ホップ数変数による定式化と限定されたパスによる定式化を示し、限定されたパスによる定式化に対して容量スケーリング法を用いた近似解法を提案する。続いて、最適化ソルバーと提案した容量スケーリング法による近似解法を用いた数値実験を行ない、定式化と解法の有効性を検討する。ホップ数変数による定式化と最適化シルバーを組合せた場合では誤差の小さな解を算出することができ、限定されたパスによる定式化と容量スケーリング法の組合せでは短時間で比較的良い近似解を算出することができ、さらに局所分枝法を組み合せることによって計算時間を抑えながら誤差の小さな解を算出できることを示す。

キーワード　ネットワーク設計問題、ホップ数、容量スケーリング、最適化問題

1.はじめに

今日の物流・ロジスティクスは、輸送ネットワークやロジスティクスネットワークといったネットワーク上で展開されている。このような物流・ロジスティクスを始めとする様々なネットワークの形態を決める数理的な問題がネットワーク設計問題である。ここでは、ネットワーク設計問題の中の基本的な問題、すなわちネットワーク上の輸送路線などのアークにかかる固定的なデザイン費用と輸送であるフローにかかる変動的なフロー費用を考慮して、アークを適切に選択することによりネットワークを形成し、かつ異なる始点と終点もつ複数種類の荷物である品種の経路を決める問題を対象とする。

一般的なネットワーク設計問題はNP－困難な問題であることが知られている（Magnanti and Wong 1984）。また、ネットワーク設計問題の現実問題への応用に関しては、Magnanti et al.（1986）　やPowell and Sheffi（1989）に詳しい記述がなされ、Balakrish-nan et al.（1997）、Costa（2005）、Crainic（2003）、Gendron et al.（1997）、Magnanti and Wong(1984)、Minoux（1989）、Wong（1984, 1985）、Yaghini and Rahbar（2012）などがサーベイを示している。

通信ネットワークにおいては、サーバー等の経由数であるホップ数の大小に通信のフローは大きな影響を受けない。一方、物流・ロジスティクスでは、配送センターを経由し、荷物の積替えを行うことによって輸送・配送を実現している。このような場合、配送センターでは積降し、積替え等の作業により多くの時間と費用が発生するため、経由する配送センターの数は限られたものとなる。本研究では、荷物の輸送経路上で経由する配送センター等の数に上限をもつような問題を対象とし、輸送経路上の輸送能力を考慮した問題を対象とする。このような問題を経由回数を考慮した容量制約をもつネットワーク設計問題またはホップ数を考慮した容量制約をもつネットワーク設計問題とよぶ。

このようなホップ数を考慮したネットワーク設計問題に関連した研究が数多くなされている。Gouveia and Requejo (2001)、Gouveia et al. (2011)、Gouveia (1995, 1996) およびDahl et al. (2006) はホップ数を考慮した最小木問題、Costa et al. (2009) やVoβ (1999) はホップ数を考慮したスタイナー木問題、Balakrishnan and Altinkemer (1992) はホップ数を考慮した通信ネットワーク問題を取り扱っている。また、Botton et al. (2013) およびGouveia et al. (2006) は、ホップ数を考慮したサバイバルネットワーク設計問題を対象としている。また、Dahl et al. (1999) やDahl and Gouveia (2004) は、ホップ数制約をもつ最短路問題に対して解析を行っている。

一方、ホップ数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は、Thiongane et al. (2015) が3種類の非分割フローをもつ問題の定式化およびラグランジュ緩和を含むいくつかの緩和問題を示し、列勾配法を用いた解法を示している。

本研究では、ホップ数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、ホップ数変数による定式化と限定されたパスによる定式化を示す。続いて、ホップ数変数による定式化に対しては最適化ソルバーを用いて解を求めることとし、限定されたパスによる定式化に対して容量スケーリング法を用いた近似解法を提案する。

2 問題の定式化

この節では、ホップ数を考慮した容量制約をもつネットワーク設計問題の定義および前提条件、使用する記号を示す。続いて、ホップ数変数によるホップアークフローを用いた定式化、および限定されたパスフローによる定式化を示す。

2.1 問題の定義

はじめに、対象とする問題の定義を示す。

定義：（ホップ数を考慮した容量制約をもつネットワーク設計問題）

デザイン費用f、フロー費用c、アーク容量Cをもつ向きをもつアーク集合Aが与えられ、ノード集合Nおよび品種の需要dをもつ品種集合K、および各品種に対するホップ数の上限Hが与えられている。このとき、品種の需要を満足し、フロー費用とデザイン費用の合計を最小にするアーク集合A' $(\subseteq A)$、およびアーク容量とホップ数制約を満足するパス

フローまたはアークフローxを求めよ。

つづいて、前提条件を示す。

- ノード集合が与えられている。
- 向きをもつアーク集合が与えられている。
- アークには、非負のデザイン費用が与えられている。
- 複数の品種からなる品種集合が与えられている。
- アークには、品種ごとの単位当たりの非負のフロー費用が与えられている。
- アークには、単位期間当たりの処理量の上限であるアーク容量が与えられている。
- 各品種ごとの需要が与えられている。
- 各品種ごとに、パスに含まれるノードの経由数に1を加えたホップ数の上限値が与えられている。
- 各品種の需要は、始点から終点までのいくつかのパス上を移動する。

ここで、ホップ数は"ノードの経由数+1"と定義し､これはパス上のアーク数と一致する。

ホップ数を考慮した容量制約をもつネットワーク設計問題の定式化で使用する記号の定義を示す。

- N：ノード集合
- A：アーク集合
- K：品種集合
- P：ホップ数制約を満足するパス集合
- P^k：品種kのホップ数制約を満足するパス集合
- $\bar{P}$：パス集合Pの部分集合で、生成されたパスの集合
- $\bar{P}^k$：品種kパス集合P^kの部分集合で、生成されたパスの集合
- $A_{\bar{P}^k}$：パス集合$\bar{P}^k$のパスに含まれるアーク集合
- c^k_{ij}：アーク（i, j）上における品種kの単位当たりの非負のフロー費用
- f_{ij}：アーク（i, j）の非負のデザイン費用
- C_{ij}：アーク（i, j）のアーク容量
- d^k：品種kの需要量
- H^k：品種kのホップ数
- δ^p_{ij}: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
- x^{kh}_{ij}：アーク（i, j）上を移動する品種kのフロー量を表すホップ数hのホップアークフロー変数；連続変数
- x^k_p：品種kのパスpのパスフロー量を表すパスフロー変数；連続変数
- y_{ij}：アーク（i, j）を選択するとき1、そうでないとき0であるデザイン変数；0-1変数

2.2 ホップ数変数による定式化

ホップ数を考慮した容量制約をもつネットワーク設計問題のホップアークフローによる定式化$HNDA$を示す。この定式化は、ホップ数変数であるホップアークフローを用いたものであり、Thiongane et al.（2015）の非分割フローに対する定式化を分割フローに対応させたものである。

ホップ数はパス上の始点からのアーク数であり、始点から続くアークをホップ数1、次のアークをホップ数2などとする。なお、品種kのホップ数の最大値はH^kであり、ホップアークフロー変数によりH^kを超えるフローは存在しないため、すべてのフローはホップ数制約を満足することができる。

（$HNDA$）

$$\min \sum_{(i,j)\in A}\sum_{k\in K}\sum_{h=1}^{H^k} c_{ij}^k x_{ij}^{kh} + \sum_{(i,j)\in A} f_{ij}y_{ij} \quad (1)$$

subject to

$$\sum_{i\in N_n^+}\sum_{h=1}^{H^k} x_{in}^{kh} - \sum_{j\in N_n^-}\sum_{h=1}^{H^k} x_{nj}^{kh} = \begin{cases} -d^k & if\ n = O^k \\ d^k & if\ n = D^k \end{cases} \quad k\in K \quad (2)$$

$$\sum_{i\in N_n^+} x_{in}^{kh} - \sum_{j\in N_n^-} x_{nj}^{k,h-1} = 0 \quad k\in K, \quad n\in N\backslash\{O^k, D^k\}, h = 2,\cdots,H^k \quad (3)$$

$$\sum_{i\in N_n^+} x_{in}^{k1} = 0 \quad k\in K, n\in N\backslash O^k \quad (4)$$

$$\sum_{j\in N_n^-} x_{nj}^{k,H^k} = 0 \quad k\in K, n\in N\backslash D^k \quad (5)$$

$$\sum_{k\in K}\sum_{h=1}^{H^k} x_{ij}^{kh} \le C_{ij}y_{ij} \quad \forall(i,j)\in A \quad (6)$$

$$\sum_{h=1}^{H^k} x_{ij}^{kh} \le d^k y_{ij} \quad \forall(i,j)\in A, k\in K \quad (7)$$

$$x_{ij}^{kh} \ge 0 \quad \forall(i,j)\in A, k\in K, h = 1,\cdots,H^k \quad (8)$$

$$y_{ij}\in\{0,1\} \quad \forall(i,j)\in A \quad (9)$$

(1) 式は目的関数であり，第一項はフロー費用，第二項はデザイン費用であり，それらの総和を最小化する．フロー費用は，アークごと、品種ごとおよびホップ数ごとのフロー費用の合計となる。(2) 式は、流入するホップアークフローの合計と流出するホップアークフローの合計の差が品種kの始点では$-d^k$、終点ではd^kであることを表すフロー保存式である。(3) 式は、品種kにおいて、始点と終点以外のノードでは、流入するホップ数$h-1$のホップアークフローと流出するホップ数hのホップアークフローが一致することを表すフロー保存式である。この式により、ホップ数の整合性を取ることができる。(4) 式は始点以外のノードから出るホップ数1のホップアークフローの合計は0であり、(5) 式は終点以外のノードに入るホップ数H^kのホップアークフローの合計は0であることを表す。(6) 式は、アーク (i, j) が選択されるときはアーク上を移動するホップアークフロー量の合計がアーク容量以下であり、アークが選択されないときは0であることを表す容量制約式である。(7) 式は、アーク (i, j) における品種kに関する強制制約式であり、アーク (i, j) が選択されるときはアーク上を移動する品種kのホップアークフローの合計は品種kの需要量以下であり、アークが選択されないときは0であることを表す。(8) 式はホップパスフロー変数の非負条件、(9) 式はデザイン変数の0－1条件である。

2.3 ホップ数制約を満足するパスフローによる定式化

定式化に含むパスおよびパスフロー変数の集合をホップ数制約を満足するものに限定する。これにより、ホップ数を考慮した容量制約をもつネットワーク設計問題の定式化は、パス集合の定義以外は一般の容量制約をもつネットワーク設計問題の定式化と同一となる。この定式化は、Thiongane et al. (2015) の非分割フローに対する定式化を分割フローに対応させたものである。パスフローによる定式化$HNDP$を示す。

($HNDP$)

$$\min \quad \sum_{(i,j)\in A}\sum_{k\in K} c_{ij}^k \sum_{p\in P^k} \delta_{ij}^p x_p^k + \sum_{(i,j)\in A} f_{ij}y_{ij} \tag{10}$$

subject to

$$\sum_{p\in P^k} x_p^k = d^k \quad \forall k \in K \tag{11}$$

$$\sum_{k\in K}\sum_{p\in P^k} \delta_{ij}^p x_p^k \le C_{ij}y_{ij} \quad \forall (i,j)\in A \tag{12}$$

$$\sum_{p\in P^k} \delta_{ij}^p x_p^k \le d^k y_{ij} \quad \forall k\in K, (i,j)\in A \tag{13}$$

$$x_p^k \ge 0 \quad \forall p\in P^k, k\in K \tag{14}$$

$$y_{ij} \in \{0,1\} \quad \forall (i,j)\in A \tag{15}$$

ここで、P^kは品種kのホップ数制約を満足するパス集合であり、定式化に現れるパスフロー変数はホップ数制約を満足していることに注意する。(10)式は目的関数であり、フロー費用とデザイン費用の総和を最小化する。(11) 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需要保存式である。(12) 式は、アーク (i, j) が選択されるときはアーク上を移動するフロー量の合計がアーク容量以下であり、アークが選択されないときは0であることを表す容量制約式である。(13) 式は、アーク (i, j) における品種kの需要d^kに関する強制制約式である。(14) 式はパスフロー変数の非負制約であり、(15) 式はデザイン変数の0−1条件である。

*HNDP*は、$\sum_{k\in K}|P^k|$個である指数個のパスフロー変数、$|A|$個のデザイン変数と$|K|$+$|A|$+$|A||K|$本の制約式をもつ問題となる。変数が指数乗個存在するので、小規模な問題であってもこの定式化を直接的に解くことは困難である。実際には、逐次、必要なホップ数制約を満足するパスフロー変数を生成して問題を解く列生成法が用いられる。この列生成法をうまく適用すれば、ホップアークフローによる定式化の場合よりも、陽的に使用する変数の数を抑えることができる。

3 定式化 *HNDP* に対する近似解法

3.1 容量スケーリング法・列生成・行生成・局所分枝法

ホップ数変数による定式化*HNDA*は、直接的に最適化ソルバーで解くことにする。ここでは、ホップ数制約を満足するパスフローによる定式化*HNDP*に対する近似解法を提案する。

*HNDP*の取り得るパスは集合で与えられており、ホップ数制約を満足するパスは指数乗個存在する。このような定式化をもつ線形計画問題に対しては、変数である列を適時生成する列生成法が有効である。一方、強制制約の数も多いため、強制制約に対しては、制約式である行を適時生成する行生成法が有効である。

本研究では、線形緩和問題を用いて近似解を求める手法として、容量スケーリング法、列生成および行生成を組み合せた解法を使用する。得られた線形緩和解をもとに、適時、限定された分枝限定法を行い、元の問題の近似解を生成する。さらに、これらの解法により得られた近似解に対して、局所分枝法を適用する。この一連の解法は、一般の容量制約をもつネットワーク設計計画問題に対して提案され、有効であることが示されている。この容量スケーリング法、列生成、行生成、限定分枝限定法、および局所分枝法を組み合せ

た解法の詳細はKatayama（2014）に詳しい解説がなされている。

*HNDP*に対しては、列生成の際に、被約費用が負となるパスを生成するのではなく、ホップ数制約を満足し、かつ被約費用が負となるパスを生成することに注意する。次の節では、容量制約をもつネットワーク設計問題の解法とは大きく異なる列生成法の部分を解説する。

3.2 限定主問題

容量スケーリング法では、*HNDP*における0－1条件を線形緩和した問題*HNDPL*を対象とする。ホップ数があるため取り得るパスの数は限定されているとはいえ、*HNDPL*は非常に多くのパスフロー変数を含む。そのため、*HNDPL*を直接解くことは困難である。そこで、あらかじめ対象となるすべてのパスフロー変数を含む問題を対象とするのではなく、逐次、必要なパスフロー変数を生成し、問題に追加していく。生成するパスフロー変数が単体法の列に相当することから、このような方法を列生成法とよぶ。

一方、*HNDPL*には、非常に多くの強制制約式が含まれている。しかし、列生成により生成されたパスフロー変数が含まれる強制制約式はそれほど多くなく、生成されたパスフロー変数が左辺に含まれていない強制制約式は不要なものとなる。そこで、生成したパスフロー変数が初めて左辺に含まれる強制制約式を逐次生成し、問題に追加する。生成する制約式が単体法の行に相当することから、このような方法を行生成法とよぶ。

品種kのホップ数の限定されたパスの適当な部分集合$\bar{P}^k$が求められているものとし、$\bar{P}$＝（$\bar{P}^k$）とする。このとき、パス集合が$\bar{P}$に限定されている次のような限定主問題*HNDPL*（$\bar{P}$）を考える。

（*HNDPL*（$\bar{P}$））

$$\min \sum_{(i,j)\in A}\sum_{k\in K} c_{ij}^k \sum_{p\in\bar{P}^k} \delta_{ij}^p x_p^k + \sum_{(i,j)\in A} f_{ij} y_{ij} \tag{16}$$

subject to

$$\sum_{p\in\bar{P}^k} x_p^k = d^k \quad \forall k\in K \tag{17}$$

$$\sum_{k\in K}\sum_{p\in\bar{P}^k} \delta_{ij}^p x_p^k \le C_{ij} y_{ij} \quad \forall (i,j)\in A \tag{18}$$

$$\sum_{p\in\bar{P}^k} \delta_{ij}^p x_p^k \le d^k y_{ij} \quad \forall (i,j)\in A_{\bar{P}^k}, k\in K \tag{19}$$

$$x_p^k \ge 0 \quad p\in\bar{P}^k, k\in K \tag{20}$$

$$0\le y_{ij}\le 1 \quad \forall (i,j)\in A \tag{21}$$

ここで、$A_{\bar{P}^k}$は品種kのパス集合$\bar{P}^k$に含まれるアーク集合であり、(19）式はアーク（i, j）を通る品種kのパスフロー変数が生成されているときのみ存在する強制制約式となる。

この問題は線形計画問題であるため、パスの部分集合の要素数が少なければ、汎用の最適化ソルバーを用いて解くことができる。

3.3 列行生成法と行生成法

HNDPL（$\bar{P}$）は変数が限定された問題である。そのため、*HNDPL*（$\bar{P}$）の最適解を求めるためには、逐次、基底に入るであろうホップ数を満足するパスフロー変数を生成しなければならない。そのために、価格付け問

題を解き、被約費用が負であり、かつホップ数を満足するパスフロー変数を求める必要がある。このようなパスフロー変数を問題に加え、この変数に対応するパスを$\bar{P}^k$に加えて、再度$HNDPL$（$\bar{P}$）を解き直す。この操作を被約費用が負であるパスフロー変数がなくなるまで繰り返す。被約費用が負であるパスフロー変数がなければ、$HNDPL$の最適解が得られたことになる。

（17）式に対する双対変数をπ、(18)、(19)式に対する非負の双対変数をu（≥ 0)、w（≥ 0）とする。これらの値は、最適化ソルバーを用いて$HNDPL$（$\bar{P}$）を最適に解くことにより求めることができる。なお、(19）式が生成されていない場合、対応するwは0と考える。

このとき、パスフロー変数xに関する被約費用は、

$$\sum_{(i,j)\in A} \delta_{ij}^{p}(c_{ij}^{k} + u_{ij} + w_{ij}^{k}) - \pi^{k} \qquad \forall p \in P^{k}, k \in K \quad (22)$$

となる。$\sum_{(i,j)\in A} \delta_{ij}^{p}$（$c_{ij}^{k}+u_{ij}+w_{ij}^{k}$）は、アーク（$i, j$）の長さを$c_{ij}^{k}+u_{ij}+w_{ij}^{k}$としたとき、パス$p$の長さに相当する。また、$\pi^{k}$は現在のパス集合$\bar{P}^k$における品種$k$の最短距離である。被約費用は「パス$p$の長さ－現在の最短距離」であるので、被約費用が負である変数を見つけることは現在の最短距離よりも短いパスを見つけることになる。

π^{k}は定数項として扱えるので、ホップ数を満足する負のパスフロー変数を見つけるには、品種kに対して、ホップ数制約の下で（22）式の第一項を最小化するパスpを見つければ良い。したがって、$HNDPL$（$\bar{P}$）における品種kに関する価格付け問題は、次のような問題PRP^kに帰着される。

（PRP^k）

$$\min \sum_{p\in P^{k}} \left\{ \sum_{(i,j)\in A} \delta_{ij}^{p}(c_{ij}^{k} + u_{ij} + w_{ij}^{k}) - \pi^{k} \right\} x_{p}^{k} \qquad (23)$$

subject to

$$\sum_{p\in P^{k}} x_{p}^{k} = 1 \qquad (24)$$

$$x_{p}^{k} \geq 0 \quad \forall p \in P^{k} \qquad (25)$$

ここで、品種kの需要に関する強制制約式（19）が生成されていない、すなわち$A_{\bar{P}k}$に含まれない制約式に対するw_{ij}^{k}は0であること、またP^kはホップ数制約を満足するパス集合であることに注意する。

PRP^kは次のようなアーク（i, j）の長さを$c_{ij}^{k}+u_{ij}+w_{ij}^{k}$とした品種$k$に対する始点・終点間のホップ数制約をもつ最短路問題SPP^kに帰着される。

（SPP^k）

$$\psi^{k} = \min \sum_{(i,j)\in A} (c_{ij}^{k} + u_{ij} + w_{ij}^{k})\nu_{ij}^{k} \qquad (26)$$

subject to

$$\sum_{i\in N_n^+} \nu_{in}^{k} - \sum_{j\in N_n^-} \nu_{nj}^{k} = \begin{cases} -1 & if\ n = O^{k} \\ 1 & if\ n = D^{k} \\ 0 & otherwise \end{cases} \qquad \forall n \in N \quad (27)$$

$$\sum_{(i,j)\in A} \nu_{ij}^{k} \leq H^{k} \qquad (28)$$

$$\nu_{ij}^{k} \in \{0,1\} \quad \forall (i,j) \in A \qquad (29)$$

ここで、ψ^{k}はSPP^kの最適値である。

このSPP^kはホップ数制約件である（28）式をもつ最短路問題であるため、一般的には容易に最適解を求めることはできない。そこで、ラグランジュ乗数λ（≥ 0）を用いてホッ

プ数制約である（28）式をラグランジュ緩和した問題$SPPL^k$（λ）を考える。

（$SPPL^k$（λ））

$$\phi^k = \min \sum_{(i,j)\in A} (c^k_{ij} + u_{ij} + w^k_{ij} + \lambda)\nu^k_{ij} - H^k\lambda \quad (30)$$

subject to

$$\sum_{i\in N_n^+} \nu^k_{in} - \sum_{j\in N_n^-} \nu^k_{nj} = \begin{cases} -1 & if\ n = O^k \\ 1 & if\ n = D^k \\ 0 & otherwise \end{cases} \quad \forall n \in N \quad (31)$$

$$\nu^k_{ij} \in \{0,1\} \quad \forall (i,j) \in A \quad (32)$$

ここで、ϕ^kは$SPPL^k$（λ）の最適値である。

この問題は最短路問題となり、係数が非負であるのでDijkstra法を用いて容易に解くことができる。

$SPPL^k$（λ）の最適値であるϕ^kは、λに対して上に凸の微分不可能な関数となる。しかし、ラグランジュ乗数は1変数のみであるので、λの値を二分探索法によって設定し、その都度、$SPPL^k$（λ）を解くことにし、それらの最適値の最大値を求め、ϕ^kまたはその近似値とすることにする。

$SPPL^k$（λ）はSPP^kのラグランジュ緩和問題であるので、その最適値ϕ^kはSPP^kの下界値となり$\phi^k<\psi^k$となる。$\phi^k-\pi^k<0$であれば、$SPPL^k$（λ）の解に対応する$\psi^k-\pi^k$が負になる可能性はあるが、$\phi^k-\pi^k<\psi^k-\pi^k$であるため、$\psi^k-\pi^k$は必ずしも負になるとは限らない。しかし、$SPPL^k$（$\lambda$）の最適解に対して$\psi^k-\pi^k<0$で、かつ（28）式を満足するパスが得られれば、ホップ数制約を満足する負の被約費用をもつパスが得られたことになる。

得られた解において、$v^k_{ij}=1$または$v^k_{ij}=1$であるアークの集合からなるパスをp^kとすると、p^kが生成すべきパス、パスに対応するパスフロー変数$x^k_{p^k}$が生成すべき変数となる。生成したp^k上のアークに関して、新たに$A_{\overline{P}k}$の要素となったアーク集合に対して、品種kの需要に関する強制制約式を問題に追加する。

一方、$SPPL^k$（λ）の最適解において$\phi^k-\pi^k\geq0$であれば、$\psi^k-\pi^k\geq\phi^k-\pi^k\geq0$となり、被約費用が負となるパスが存在しないことになる。また、すべての品種kについて$\phi^k-\pi^k\geq0$であれば、被約費用が負である変数が存在しないため、$HNDPL$が最適に解けたことになる。

ここでは、緩和問題$SPPL^k$（λ）の解からパスを導出しているため、ホップ数制約である（28）式を満し、被約費用が負となるパスをすべて生成できる保証はない。しかし、解法自体が近似解法であるため、被約費用が負となるパスをすべて生成できなくとも、近似解を求めることは十分に可能である。

4 数値実験

ホップ数変数による定式化$HNDA$およびホップ数制約を満足するパスフローによる定式化$HNDP$に対して、数値実験を行なった。使用した問題は、容量制約をもつネットワーク設計問題に対するCrainicらのベンチマーク問題であるC問題とR問題（Crainic et al. 2000）である。それぞれの問題に対して、各品種のホップ数の上限値は、“すべてのアー

クを含むネットワーク上での最短経路のホップ数+1”とした。このため、ホップ数の上限がきついため、いくつかの問題では実行不可能となっている。

ホップ数変数による定式化*HNDA*に対して、これを最適化ソルバーCPLEXにより解き、上界値または最適値を求めた。なお、CPLEXの計算時間が30時間を超えた場合は、その時点における最良の上界値および下界値を求めた。

計算環境および*HNDP*に対する近似解法で設定した主なパラメータおよび条件は以下の通りである。

- 使用OS および言語：UBUNTU 12.4, GNU C++ compiler
- CPU INTEL i7 3440K 3.4GHz 4Core, RAM 24GByte
- 最適化ソルバー：CPLEX 12.6（Parallel Version）
- 容量スケーリングパラメータ：0.025 ～ 0.200
- 局所分枝の計算時間の上限：1000 秒
- 局所分枝の近傍：10

なお、その他の容量スケーリング法・局所分枝法におけるパラメータはKatayama（2015）と同一であり、パラメータの内容については同文献を参照のこと。

C問題に対する上界値と誤差を表1に示す。表内のCPLEXは最適化ソルバーCPLEXによる上界値、CAPは局所分枝法を行わない容量スケーリング法による上界値、LOBは局所分枝法を行った容量スケーリング法による上界値である。これらは実行可能であった33問題の結果であり、太字は最適値、斜体文字は3つの方法の中の最良値である。また、CPLEX(%)はCPLEXによる上界値とCPLEXによる最適値または下界値との誤差であり、CAP（%）とLOB（%）も同様である。

CPLEXでは33問中28問、局所分枝法を行った容量スケーリング法では33問中26問の最適解を求めることができている。また、局所分枝法を行わない容量スケーリング法は問題100/400/010/VLの実行可能解を求めることができていない。局所分枝法を行わない容量スケーリング法の平均誤差は0.83%であった。また、局所分枝法を行った容量スケーリング法の平均誤差は0.20%であり、CPLEXの平均誤差0.18%と同程度となった。

C問題に対する計算時間を表2に示す。CPLEXの平均計算時間は24692.7秒と非常に大きく、最適解を求められなかった5問では上限の30時間となっている。一方、局所分枝法を行わない容量スケーリング法の平均計算時間は91.1秒と短かく、短時間で多くの近似解を求めることができている。また、局所分枝法を行った容量スケーリング法の平均計算時間は1836.5秒であり、局所分枝法を行わない方法よりも長いが、CLPEXに比べると計算時間は1/13程度であり、大幅に計算時間を削減することができている。

R問題に対する上界値と誤差を表3および表4に示す。R問題はr01.1からr18.9までであるが、ここでは、容易に最適解を求めることができるか実行不可能となった問題r01.1からr12.9までを除く問題 r13.1からr18.9までの54問の結果を示す。なお、表5の平均は54問の

平均誤差である。

CPLEXでは54問中46問、局所分枝法を行った容量スケーリング法では54問中38問の最適解を求めることができている。局所分枝法を行わない容量スケーリング法の平均誤差は1.99%と比較的大きいが、局所分枝法を行った容量スケーリング法の平均誤差は0.43%であった。一方、CPLEXの平均誤差は0.21%と小さなものとなっている。

R問題に対する計算時間を表5に示す。CPLEXでは20464.1秒を要した。局所分枝法を行わない容量スケーリング法の計算時間は58.9秒であり、非常に短時間で近似解を算出できている。また、局所分枝法を行った容量スケーリング法の計算時間は1739.5秒であり、CLPEXに比べると計算時間は1/12程度となり、大幅に計算時間を削減することができている。

5 おわりに

本研究では、ホップ数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、ホップ数変数による定式化と限定されたパスによる定式化を示し、限定されたパスによる定式化に対して容量スケーリング法および局所分枝法を用いた近似解法を提案した。続いて、ホップ数変数による定式化に対しては最適化ソルバーを用いた数値実験、限定されたパスによる定式化に対しては提案した近似解法を用いた数値実験を行ない、定式化と解法の有効性を検討した。

ホップ数変数による定式化とCPLEXを組合せた場合、誤差の小さな解を算出することができたが、大きな計算時間が必要となった。このため、数値実験で用いた問題よりもさらに大規模な問題対しては、このまま適用することは困難である。一方、限定されたパスによる定式化と容量スケーリング法の組合せは、短時間で比較的良い近似解を算出することができた。さらに局所分枝法を組み合せることによって、計算時間を抑えながら、誤差の小さな解を算出することがきた。

本研究は科学研究費基盤研究C（課題番号25350454）による成果の一部である。

表1：C問題の上界値と誤差

N/A/K/FC	CPLEX	CAP	LOB	CPLEX(%)	CAP(%)	LOB(%)
100/400/010/VL	**29016.0**	*	**29016.0**	0.00	*	0.00
100/400/030/FL	**56260.0**	60939.0	**56260.0**	0.00	8.32	0.00
20/230/040/VL	**424075.0**	**424075.0**	**424075.0**	0.00	0.00	0.00
20/230/040/VT	**374013.0**	375326.0	**374013.0**	0.00	0.35	0.00
20/230/040/FT	**644361.0**	645821.0	**644361.0**	0.00	0.23	0.00
20/230/200/VL	**104072.0**	105125.0	**104072.0**	0.00	1.01	0.00
20/230/200/FL	**160969.0**	163378.5	**160969.0**	0.00	1.50	0.00
20/230/200/VT	**105729.0**	106183.0	**105729.0**	0.00	0.43	0.00
20/230/200/FT	**144587.0**	144803.0	**144587.0**	0.00	0.15	0.00
20/300/040/VL	**429700.0**	**429700.0**	**429700.0**	0.00	0.00	0.00
20/300/040/FL	**588325.0**	588502.0	**588325.0**	0.00	0.03	0.00
20/300/040/VT	**464550.0**	**464569.0**	**464550.0**	0.00	0.00	0.00
20/300/040/FT	**604237.0**	604997.0	**604237.0**	0.00	0.13	0.00
20/300/200/VL	**78649.0**	79071.0	**78649.0**	0.00	0.54	0.00
20/300/200/FL	**122151.0**	122695.0	**122151.0**	0.00	0.45	0.00
20/300/200/VT	**78242.0**	78330.0	**78242.0**	0.00	0.11	0.00
20/300/200/FT	**111729.7**	112053.0	**111729.7**	0.00	0.29	0.00
30/520/100/VL	**54586.0**	54650.0	**54586.0**	0.00	0.12	0.00
30/520/100/FL	**99745.0**	100925.0	99919.0	0.00	1.18	0.17
30/520/100/VT	**52439.0**	52711.0	**52439.0**	0.00	0.52	0.00
30/520/100/FT	*98087.0*	98848.0	*98087.0*	0.42	1.20	0.42
30/520/400/VL	**115313.1**	115699.0	**115313.1**	0.00	0.33	0.00
30/520/400/FL	*155191.7*	156664.0	156196.5	0.59	0.95	0.65
30/520/400/VT	**116650.3**	116789.4	116659.8	0.00	0.12	0.01
30/520/400/FT	*156311.2*	156933.5	156819.0	0.79	1.19	1.12
30/700/100/VL	**48693.0**	48770.0	**48693.0**	0.00	0.16	0.00
30/700/100/FL	**61144.0**	61365.0	**61144.0**	0.00	0.36	0.00
30/700/100/VT	**46341.0**	46495.5	**46341.0**	0.00	0.33	0.00
30/700/100/FT	**55960.0**	56646.0	**55960.0**	0.00	1.23	0.00
30/700/400/VL	**100875.0**	101292.5	**100875.0**	0.00	0.41	0.00
30/700/400/FL	*139239.7*	143680.0	143037.8	2.68	3.19	2.73
30/700/400/VT	**96500.5**	96571.7	**96500.5**	0.00	0.07	0.00
30/700/400/FT	134329.5	134694.0	*134266.6*	1.39	1.66	1.34
平均	-	-		0.18	0.83	0.20

*:実行可能解を算出せず.

表2：C問題の計算時間（秒）

N/A/K/FC	CPLEX	CAP	LOB	N/A/K/FC	CPLEX	CAP	LOB
100/400/010/VL	0.3	0.4	1.2	30/520/100/VL	114.2	1.1	147.4
100/400/030/FL	51.9	1.7	144.1	30/520/100/FL	21711.2	44.8	3459.8
20/230/040/VL	0.1	0.2	0.8	30/520/100/VT	1313.2	1.3	2353.2
20/230/040/VT	0.9	0.4	2.9	30/520/100/FT	108000.0	49.4	6302.7
20/230/040/FT	2.0	0.3	5.4	30/520/400/VL	9600.3	24.2	2809.1
20/230/200/VL	1074.6	23.2	553.2	30/520/400/FL	108000.0	249.6	4476.0
20/230/200/FL	55726.7	67.5	1786.9	30/520/400/VT	29797.4	73.1	3087.0
20/230/200/VT	329.2	13.3	499.5	30/520/400/FT	108000.0	392.1	4602.5
20/230/200/FT	2059.9	88.2	2544.9	30/700/100/VL	12.4	0.7	36.6
20/300/040/VL	0.2	0.2	1.1	30/700/100/FL	943.8	7.4	213.0
20/300/040/FL	2.4	0.3	6.0	30/700/100/VT	9138.3	3.7	3174.9
20/300/040/VT	0.8	0.2	2.9	30/700/100/FT	51337.0	8.7	5388.6
20/300/040/FT	0.8	0.2	2.6	30/700/400/VL	41416.7	75.1	3595.3
20/300/200/VL	4636.3	169.1	1853.0	30/700/400/FL	108000.0	524.8	4596.8
20/300/200/FL	38500.8	371.7	3396.8	30/700/400/VT	65574.8	54.4	2809.0
20/300/200/VT	2448.6	32.2	1072.6	30/700/400/FT	108000.0	1000.1	6728.8
20/300/200/FT	13141.7	90.0	2291.0	平均	24692.7	91.1	1836.5

表3：R問題の上界値と誤差（1/2）

Problem	CPLEX	CAP	LOB	CPLEX(%)	CAP(%)	LOB(%)
r13.1	**143541.0**	143674.0	**143541.0**	0.00	0.09	0.00
r13.2	**281252.0**	287864.0	**281252.0**	0.00	2.35	0.00
r13.3	**411372.0**	429319.0	**411372.0**	0.00	4.36	0.00
r13.4	**151641.0**	152517.0	**151641.0**	0.00	0.58	0.00
r13.5	**299410.0**	304001.0	**299410.0**	0.00	1.53	0.00
r13.6	**440367.0**	450587.0	**440367.0**	0.00	2.32	0.00
r13.7	**212638.5**	214378.0	212821.0	0.00	0.82	0.09
r13.8	**450918.0**	467427.0	**450918.0**	0.00	3.66	0.00
r13.9	**715480.0**	742683.0	**715480.0**	0.00	3.80	0.00
r14.1	**414252.0**	**414252.0**	**414252.0**	0.00	0.00	0.00
r14.2	**844465.0**	857546.0	**844465.0**	0.00	1.55	0.00
r14.3	**1289818.0**	1303938.0	**1289818.0**	0.00	1.09	0.00
r14.4	**442306.0**	443159.0	**442306.0**	0.00	0.19	0.00
r14.5	**912191.0**	916065.0	**912191.0**	0.00	0.42	0.00
r14.6	**1373849.0**	1386597.0	1378534.0	0.00	0.93	0.34
r14.7	**668216.3**	**668216.3**	**668216.3**	0.00	0.00	0.00
r14.8	**1624706.7**	1638949.0	1625832.0	0.00	0.88	0.07
r14.9	**2661294.0**	2687144.0	**2661294.0**	0.00	0.97	0.00
r15.1	**1031834.0**	1033277.0	**1031834.0**	0.00	0.14	0.00
r15.2	**2231916.0**	12331456.0	2235527.0	0.00	4.46	0.16
r15.3	**3455061.0**	3619800.0	**3455061.0**	0.00	4.77	0.00
r15.4	**1158115.0**	**1158137.0**	**1158115.0**	0.00	0.00	0.00
r15.5	**2585256.0**	2606421.5	**2585256.0**	0.00	0.82	0.00
r15.6	*4132383.0*	4198413.0	4133941.0	1.13	2.75	1.17
r15.7	**2300838.7**	2305518.5	**2300838.7**	0.00	0.20	0.00
r15.8	**5869005.0**	5951961.5	**5869005.0**	0.00	1.41	0.00
r15.9	**9978303.7**	10174604.5	**9978303.7**	0.00	1.97	0.00

表4：R問題の上界値と誤差（4/4）

Problem	CPLEX	CAP	LOB	CPLEX(%)	CAP(%)	LOB(%)
r16.1	**138735.0**	140156.0	**138735.0**	0.00	1.02	0.00
r16.2	**247883.0**	253099.0	**247883.0**	0.00	2.10	0.00
r16.3	**348778.0**	371599.0	**348778.0**	0.00	6.54	0.00
r16.4	**140619.0**	141310.0	**140619.0**	0.00	0.49	0.00
r16.5	**254948.0**	263359.0	**254948.0**	0.00	3.30	0.00
r16.6	**361568.0**	382101.0	**361568.0**	0.00	5.68	0.00
r16.7	**170091.0**	171344.0	**170091.0**	0.00	0.74	0.00
r16.8	**352792.0**	362471.0	354443.0	0.00	2.74	0.47
r16.9	**535386.0**	546723.0	537597.0	0.01	2.13	0.42
r17.1	**368471.0**	370834.0	**368471.0**	0.00	0.64	0.00
r17.2	**668216.3**	732285.0	725760.0	0.00	9.59	8.61
r17.3	**1092098.0**	1120213.0	**1092098.0**	0.00	2.57	0.00
r17.4	**377809.0**	378669.0	**377809.0**	0.00	0.23	0.00
r17.5	**755019.0**	762624.0	**755019.0**	0.00	1.01	0.00
r17.6	**1143868.0**	1144883.0	**1143868.0**	0.00	0.09	0.00
r17.7	**502058.0**	505281.0	**502058.0**	0.00	0.64	0.00
r17.8	*1115633.3*	1120564.3	1115633.3	0.70	1.14	0.70
r17.9	*1808907.0*	1831656.0	1809268.0	1.70	2.98	1.72
r18.1	**855561.5**	857729.0	**855561.5**	0.00	0.25	0.00
r18.2	**1804839.0**	1863766.0	**1804839.0**	0.00	3.26	0.00
r18.3	*2785908.0*	2906025.0	2785908.0	4.31	8.81	4.31
r18.4	**935866.0**	938272.0	**935866.0**	0.00	0.26	0.00
r18.5	**1963823.0**	1994580.0	1964142.0	0.00	1.57	0.02
r18.6	*3063289.0*	3102872.0	3080864.0	2.84	4.17	3.43
r18.7	*1479097.6*	1488695.0	1479640.5	0.51	1.17	0.55
r18.8	*3926968.0*	3947124.8	3933131.0	0.25	0.76	0.40
r18.9	**6570230.8**	6680270.7	6606947.0	0.00	1.67	0.56
平均	-	-	-	0.21	1.99	0.43

表5：R問題の計算時間（秒）

Problem	CPLEX	CAP	LOB	Problem	CPLEX	CAP	LOB
r13.1	0.6	0.2	1.1	r16.1	0.2	0.0	1.3
r13.2	26.7	0.4	4.9	r16.2	6.4	1.3	7.6
r13.3	45.6	1.8	14.1	r16.3	14.8	3.6	20.1
r13.4	4.4	0.3	2.7	r16.4	0.6	0.2	2.0
r13.5	92.0	0.9	11.0	r16.5	9.9	1.6	23.1
r13.6	568.3	5.7	32.0	r16.6	33.3	4.4	53.3
r13.7	393.5	1.1	312.9	r16.7	207.9	1.1	995.0
r13.8	2094.6	3.3	1218.0	r16.8	13836.3	10.6	2614.5
r13.9	2371.9	9.2	3263.0	r16.9	29735.9	25.8	2179.5
r14.1	0.9	0.2	2.4	r17.1	3.7	0.6	13.4
r14.2	44.7	5.7	60.3	r17.2	302.9	9.6	144.4
r14.3	627.3	15.0	135.9	r17.3	6622.3	44.5	1591.6
r14.4	2.6	0.7	9.5	r17.4	4.8	0.7	13.4
r14.5	869.7	42.3	554.3	r17.5	569.6	41.0	307.1
r14.6	2186.8	29.5	1533.2	r17.6	11929.2	60.3	1683.4
r14.7	311.9	5.3	1686.5	r17.7	8537.4	12.8	2250.1
r14.8	50274.8	31.3	4915.5	r17.8	108000.0	231.1	5555.7
r14.9	7965.9	43.3	2381.8	r17.9	108000.0	308.6	4222.5
r15.1	18.0	1.10	36.4	r18.1	232.0	4.4	157.3
r15.2	8573.8	37.5	2329.0	r18.2	65556.3	163.0	2276.0
r15.3	38726.2	200.8	3269.5	r18.3	108000.0	312.5	4903.0
r15.4	99.3	3.40	202.7	r18.4	9771.7	14.6	3682.2
r15.5	26476.3	158.9	4581.0	r18.5	54733.4	134.5	4555.1
r15.6	108000.0	432.4	7059.0	r18.6	108000.0	445.5	5712.2
r15.7	303.6	11.5	818.7	r18.7	108000.0	56.8	4138.5
r15.8	137.8	9.8	374.6	r18.8	108000.0	130.0	5333.3
r15.9	106.4	12.5	392.2	r18.9	4630.8	98.8	6295.5
平均	-	-	-	-	20464.1	58.9	1739.5

参考文献

Balakrishnan, A., K. Altinkemer. 1992. Using a hop-constraintd model to generate alternative communication network design. *ORSA journal on Computing* **4** 192-205.

Balakrishnan, A., T. L. Magnanti, P. Mirchandani. 1997. Network design. M. Dell'Amico, F. Maffioli, S. Martello, eds., *Annotated Bibliographies in Combinatorial Optimization*. John Wiley & Sons, New York, 311-334.

Botton, Q., B. Fortz, L. Gouveia, M. Poss. 2013. Benders decomposition for the hop-constrained survivable network design problem. *INFORMS Journal on Computing* **25** 13-26.

Costa, A., J. Cordeau, G. Laporte. 2009. Models and branch-and-cut algorithms for the steiner tree problem with revenues, budget and hop constraints. *Networks* **53** 141-159.

Costa, A. M. 2005. A survey on benders decomposition applied to fixed-charge network design problems. *Computers and Operations Research* **32** 1429-1450.

Crainic, T. G. 2003. Long-haul freight transportation. R. W. Hall, ed., *Handbook of Transportation Science*. Kluwer Academic Publishers, 451-516.

Crainic, T. G., M. Gendreau, J. M. Farvolden. 2000. A simplex-based tabu search for capacitated network design. *INFORMS journal on Computing* **12** 223-236.

Dahl, G., A.Martin, M. Stoer. 1999. Routing through virtual paths in layered telecommunication networks. *Operations Research* **47** 693-702.

Dahl, G., L. Gouveia. 2004. On the directed hop-constrained shortest path problem. *Operations Research Letters* **32** 15-22.

Dahl, G., L. Gouveia, C.Requejo. 2006. On formulations and methods for the hopconstrained minimum spanning tree problem. M. Resende, P. Pardalos, eds., *Handbook of Optimization in Telecommunications*. Springer, 493-515.

Gendron, B., T. G. Crainic, A. Frangioni. 1997. Multicommodity capacitated network design. Tech. Rep. CIRRELT-98-14, Centre de recherche sur les transports,Université de Montréal.

Gouveia, L. 1995. Using the miller-tucker-zemlin constraints to formulate a minimal spanning tree problem with hop constraints. *Computers & Operations Research* **22** 959-970.

Gouveia, L. 1996. Multicommodity ow models for spanning trees with hop constraints. *European Journal of Operational Research* **95** 178-190.

Gouveia, L., P. Patricio, A.Sousa. 2006. Compact models for hop-constrained node survivable network design: An application to MPLS. S. Raghavan, G. Anandalingam, eds., *Telecommunications Planning: Innovations in Pricing, Network Design and Management. Springer*, 167-180.

Gouveia, L., C. Requejo. 2001. A new lagrangian relaxation approach for the hopconstrained minimum spanning tree problem. *European Journal of Operational Research* **132** 539-552.

Gouveia, L., L. Simonetti, E.A. Uchoa. 2011. Modeling hop-constrained and diameterconstrained minimum spanning tree problems as steiner tree problems over layered graphs. *Mathematical Programming* **128** 123-148.

Katayama, N. 2015. Combined capacity scaling and local branching approach for capacitated multicommodity network design problem. Working Paper.

Katayama, N. 2014. A combined matheuristic for service network design problem. *The International Journal of Logistics and SCM Systems* **8**. Submitted for publication.

Magnanti, T. L., P. Mireault, R. T. Wong. 1986. Tailoring benders decomposition for uncapacitated network design. *Mathematical Programming Study* **26** 112-155.

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

Minoux, M. 1989. Network synthesis and optimum network design problems: Models, solution methods and applications. *Networks* **19** 313-360.

Powell, W. B., Y. Sheffi. 1989. Design and implementation of an interactive optimization system for network design in the motor carrier industry. *Operations Research* **37** 12-29.

Thiongane, B., J. Cordeau, B.Gendron. 2015. Formulations for the nonbifurcated hopconstrained multicommodity capacitated fixed-charge network design problem. *Computers & Operations Research* **53** 1-8.

Vo β S. 1999. The steiner tree problem with hop constraints. *Annals of Operations Research* **86** 321-345.

Wong, R. T. 1984. Introduction and recent advances in network design: Models and algorithms. M. Florian, ed., *Transportation Planning Models*. Elsevier Science, North Holland, Amsterdam, 187-225.

Wong, R. T. 1985. Location and network design. M. O'heEigeartaigh, J. Lenstra, A. RinnooyKan, eds., *Combinatorial Optimization Annotated Bibliographies*. John Wiley & Sons, New York, 129-147.

Yaghini, M., M. Rahbar. 2012. Multicommodity network design problem in rail freight transportation planning. *Procedia - Social and Behavioral Sciences* **43** 728-739.