《論　文》

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

片　山　直　登

1　はじめに

容量制約をもつネットワーク設計問題（capacitated network design problem, *CND*）は，容量をもつアークとノードからなるネットワークと，ネットワーク上を流れる多品種の需要量が与えられたときに，アークに関するデザイン費用と品種に関するフロー費用の合計が最小化となるアークを選択し，各品種のフローの経路を求める問題である．これは，通信ネットワークの設計や輸送・配送ネットワークの設計など，幅広い分野で応用されている問題（Magnanti et al. 1986）である．

近年，*CND* に対して，メタヒューリスティクスと最適化ソルバーを組合せた MIP アプローチが盛んに研究されている．Rodríguez-Martín and Salazar-Gonzáleza（2010）は最適化ソルバーを用いた局所分枝法，Hewitt et al.（2010）は限定された広範囲の近傍を探索する厳密解法とヒューリスティクスを組合せた IP 探索法，Chouman and Crainic（2010）はアークバランスサイクルによる MIP 探索とタブー探索法を用いた解法，Paraskevopoulos et al.（2016）はサイクルに基づく進化的アルゴリズムを示している．また，Yaghini and Foroughi（2014）は蟻コロニー法，Katayama（2015）は容量スケーリング法と局所分枝法を合わせたコンバインド法，Yaghini et al.（2016）は切除平面隣接構造と局所分枝法を組み合わせた解法，Momeni and Sarmadi（2016）は遺伝的アルゴリズム，緩和誘導近傍探索および局所分枝法を組み合わせた解法，Gendron et al.（2016）は線形緩和と擬似切除平面を用いた反復線形計画法を提案している．さらに，Munguí et al.（2017）はマルチ CPU を用いた並列局所探索法を提案し，96コアの CPU をもつ計算機群を用いて，高速に近似解を算出している．一方，片山（2015）は，容量スケーリング法と貪欲法を組み合わせた高速解法を開発している．

本研究では，*CND* に対して，新しい近傍を提案し，容量スケーリング法と MIP ソル

バーによる制約付きの近傍探索法を組み合わせた近似解法である MIP 近傍探索法を提案する．提案した MIP 近傍探索法が従来のベンチマーク問題に対して高速で有用な近似解を算出できることを示す．

2 *CND* の定式化

ノード集合を N，アーク候補集合を A，品種の集合を K とし，アーク（i, j）を選択するか否かを表すデザイン変数を y_{ij}，アーク（i, j）上の品種 k のフロー量を表すアークフロー変数を x_{ij}^{k} とする．アーク（i, j）の容量を C_{ij}，デザイン費用を f_{ij} とし，アーク（i, j）上の品種 k の単位当たりのフロー費用を c_{ij}^{k} とし，品種 k の需要量を d^{k} とする．品種 k の始点を O^{k}，終点を D^{k} とする．アーク A からなるネットワーク上でノード n から出るアーク集合を $N_n^{+}(A)$，アーク A からなるネットワーク上でノード n に入るアーク集合を $N_n^{-}(A)$ とする．また，目的関数値である総費用の最小値を表す変数を Φ とおく．

このとき，アーク集合 A からなるネットワークにおいて，アークフロー変数を用いた *CND* の定式化 *CNDA*(A)は，次のように表される．

CNDA(A)

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

subject to

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

$$\sum_{k\in K} x_{ij}^{k} \le C_{ij} y_{ij} \quad \forall\ (i,j) \in A \tag{3}$$

$$x_{ij}^{k} \le d^{k} y_{ij} \quad \forall\ k \in K,\ (i,j) \in A \tag{4}$$

$$y_{ij} \in \{0,1\} \quad \forall (i,j) \in A \tag{5}$$

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

(1)式は，フロー費用とデザイン費用の和であり，これを最小化することを表す．(2)式は，ノード n における流入量と流出量の差が，ノード n が品種 k の始点 O^{k} であれば $-d^{k}$，終点 D^{k} であれば d^{k}，その他のノードであれば 0 となることを表すフロー保存式

である．(3)式の左辺はアーク（i, j）上のフロー量の合計であり，右辺はアーク（i, j）が選択されるときに C_{ij}，選択されないとき 0 となる容量制約式である．(4)式の左辺はアーク（i, j）上の品種 k のパスフロー量の合計であり，右辺はアーク（i, j）が選択されるときに d^k，選択されないとき 0 となる強制制約式である．(5)式はデザイン変数の 0－1条件，(6)式はアークフローの非負制約である．

品種 k の取り得るパス集合を P^k，品種 k のパス p のフロー量であるパスフロー変数を x_p^k とし，パス p がアーク（i, j）を含むとき 1，そうでないとき 0 である定数を δ_{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 に対して，パスフロー変数を用いた *CND* の定式化 *CNDP*（A, P, C）は，次のように表される．

CNDP（A, P, C）

$$\min \sum_{(i,j)\in A} \sum_{k\in K} c_{ij}^k \sum_{p\in P^k} \delta_{ij}^p x_p^k + \sum_{(i,j)\in A} f_{ij} y_{ij} \tag{7}$$

subject to

$$\sum_{p\in P^k} x_p^k = d^k \quad \forall k \in K \tag{8}$$

$$\sum_{k\in K} \sum_{p\in P^k} \delta_{ij}^p x_p^k \leq C_{ij} y_{ij} \quad \forall (i,j) \in A \tag{9}$$

$$\sum_{p\in P^k} \delta_{ij}^p x_p^k \leq d^k y_{ij} \quad \forall (i,j) \in A, k \in K \tag{10}$$

$$y_{ij} \in \{0,1\} \quad \forall (i,j) \in A \tag{11}$$

$$x_p^k \geq 0 \quad \forall p \in P^k, k \in K \tag{12}$$

(7)式は，フロー費用とデザイン費用の和であり，これを最小化することを表す．(8)式は，品種 k のパスフロー量の和は需要量になることを表す需要保存式である．(9)式の左辺はアーク（i, j）上のフロー量の合計であり，右辺はアーク（i, j）が選択されるときに C_{ij}，選択されないとき 0 となる容量制約式である．(10)式の左辺はアーク（i, j）上の品種 k のパスフロー量の合計であり，右辺はアーク（i, j）が選択されるときに d^k，選択されないとき 0 となる強制制約式である．(11)式はデザイン変数の0－1条件，(12)式はパスフローの非負制約である．

3　近似解法

3．1　容量スケーリング法

アークに対するデザイン変数は0－1の離散変数であるため，*CND* は離散変数を含む

最適化問題となる．このため，多くのアークを含む大規模な問題では多くの0－1変数が含まれるため，最適解または適切な近似解を算出することが困難になる．そこで，容量スケーリング法を用いて，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．容量スケーリング法の詳細は，Katayama et al.（2009）を参照のこと．

容量スケーリング法は，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1のデザイン変数解を導出するものである．容量スケーリングでは，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（Katayama et al. 2009）．そこで，アーク集合 A に対して，収束しないアーク数が終了判定アーク数に達するまで容量スケーリング法を適用し，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アーク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．

容量スケーリング法により，選定されたアーク集合を $\bar{A}$ とする．また，列生成法にて容量スケー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．このとき，$CNDP$（$\bar{A}$, $\bar{P}$, C）は，相対的に小規模な問題になるため，計算時間の上限を設けて，MIP ソルバーの分枝限定法などを用いると，適当な近似解を算出することができる．この近似解を $\bar{y}$ とする．この $\bar{y}$ を次に示す近傍探索法の初期解とする。

3．2　MIP を用いた近傍探索

$CNDA$（A）において，容量スケーリング法で求めた近似解 $\bar{y}$ を初期解として，その近傍を MIP ソルバー用いて探索し，近傍解を算出していく．

関連した従来の方法として局所分枝法（Fischetti and Lodi 2003）が知られており，局所分枝法単体（Rodríguez-Martín and Salazar-Gonzáleza 2010），または他の解法と組み合わせた解法（Katayama 2015, Yaghini et al. 2016, Momeni and Sarmadi 2016）など，ネットワーク設計問題にも数多く適用され，高精度な解を提供している．

局所分枝法において追加する制約式は次式のいずれかである．

$$\sum_{(i,j)\in A|\bar{y}_{ij}=1}(1-y_{ij})+\sum_{(i,j)\in A|\bar{y}_{ij}=0}y_{ij}\leq M \tag{13}$$

$$\sum_{(i,j)\in A|\bar{y}_{ij}=1}(1-y_{ij})+\sum_{(i,j)\in A|\bar{y}_{ij}=0}y_{ij}\geq M+1 \tag{14}$$

⒀式は，現在の解 $\bar{y}$ から距離 M 以内の M 近傍に含まれる領域を表す．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．

$$\Phi < UB \tag{15}$$

この式により，現在までの最良値よりも良い上界値が存在しなければ $CNDA$（A）は実行不可能となる．

CNDA（*A*）に⒀式と⒂式を追加し，MIP ソルバーを用いて一定時間内で解を求める．これにより，最良解が求められれば，現在の解をこの解に更新する．*M* 近傍において，更新する解がないと判断できた場合，⒀式の代わりに⒁を追加することにより，探索範囲を変更して解探索を継続する．一定時間内で更新解を求めることができない場合は，適宜，*M* を減少させて，探索を繰り返す．

容量スケーリング法と局所分枝法を組み合わせた解法のアルゴリズムを Algoritm1に示す．

これに対して，本研究では次の２本の制約式を追加して，MIP ソルバーを用いて近傍を探索する．

$$\sum_{(i,j)\in A|\bar{y}_{ij}=1} y_{ij} \leq L-1 \tag{16}$$

$$\sum_{(i,j)\in A|\bar{y}_{ij}=1} y_{ij} \geq L-M \tag{17}$$

ここで，*L* は現在の解で $y_{ij}=1$である選択されたアーク数，*M* は近傍の範囲である．なお，⒂式も追加する．⒂式により，探索済みの解を排除することができる。

⒃式は，現在の解で選択されたアークから，少なくとも１本のアークはネットワークから取り除くことを表す．⒄式は，現在の解で選択されたアークを高々 *M* 本のアークをネットワークから取り除くことを表す．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制約付きの近傍探索法である．

CNDA（*A*）に，⒃式，⒄式および⒂式を追加し，MIP ソルバーを用いて，一定時間内で解を求める．現在の解より良い解が求められれば，現在の解を更新する．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，実行不可能または現在の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．また，そうでなく，一定時間内に現在の解より良い解を算出することができない場合は，$M:=\lfloor M/\alpha \rfloor$（$\alpha>1$）として *M* を減少させて探索を繰り返す．ここで，α は *M* の変更基準である．この方法を MIP 近傍探索法とよぶことにする．容量スケーリング法と MIP 近傍探索法を組み合わせた解法のアルゴリズムを Algoritm2に示す．

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37問およびＲ問題の81問（Crainic et al. 2001）に対して，数値実験を行った．なお，Ｒ

問題は容易な問題 r01から r09を除く，問題 r10から r18を対象とする．

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 7.02

・CPU AMD Ryzen7 1800X 3.6GHz 8Cores，RAM 32GByte

・使用コア数：容量スケーリング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．

・スケーリングパラメータ：０～１

・スケーリング法の終了判定アーク数 ArcNum：100

・近傍 M：５

・近傍 M の変更基準 α：2

・１回の局所分枝・近傍探索における MIP ソルバー計算時間の上限 T：40秒

近似解の誤差を算出するために，Katayama（2017）が示した下界値または最適値を使用した．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，近年の平均誤差が２％以下の結果をもつ研究を中心に，局所分枝法（LOBR）（Rodríguez-Martín and Salazar-Gonzáleza 2010），IP 探索法（IPSE）（Hewitt et al. 2010），MIP タブー探索法（MIPT）（Chouman and Crainic 2010），サイクル進化法（CEVO）（Paraskevopoulos et al. 2013），蟻コロニー法（ANCO）（Yaghini and Foroughi 2014），遺伝的・局所分枝法（GELO）（Momeni and Sarmadi 2016），反復線形計画法（ILPH）（Gendron et al. 2016），コンバインド法（COMB）（Katayama 2015），容量スケーリング法・貪欲法（CSGR）（片山 2015），切除平面・局所分枝法（CUTP）（Yaghini et al. 2016），並列局所探索法（PALS）（Mungu et al. 2017）の結果を示す．これらに加え，容量スケーリング法による解を初期値とした容量スケーリング・局所分枝法（CSBR），および容量スケーリング・MIP 近傍探索法（CMIP）の結果を示す．さらに，容量スケーリング・MIP 近傍探索法においてスケーリングパラメータを変化させた中の最良値を BEST として示した．なお，容量スケーリング・局所分枝法におけるスケーリングパラメータは0.81，容量スケーリング・MIP 近傍探索法では0.27と設定した．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，局所分枝法は平均誤差3.10％と大きく，IP 探索法は1.69％，MIP タブー探索法は0.87％，サイクル進化法は0.93％と，誤差が0.8％を超えている．蟻コロニー法は平均誤差0.79％，遺伝的・局所分枝法は0.59％，反復線形計画法は0.74％，コンバインド法は0.27％，容量スケーリング・貪欲法は0.65％，切除平面・局所分枝法は0.33％，並列局所探索法は0.45％となっている．コンバインド法は，現在までの解法の中で最も誤差が小さい．一方，容量スケーリング・局所分枝法は誤差0.50％であり，容量スケーリング・MIP 近傍探索法は

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

LOBR	IPSE	MIPT	CEVO	ANCO	GERO	ILPH
3.10	1.69	0.87	0.93	0.79	0.59	0.74
COMB	CSGR	CUTP	PALS	CSLB	CMIP	BEST
0.27	0.65	0.33	0.45	0.50	0.40	0.30

0.40%，容量スケーリング・MIP 近傍探索法の最良値は0.30%であった．容量スケーリング・局所分枝法は，コンバインド法，並列局所探索法と切除平面・局所分枝法にはそれぞれには0.23%，0.17%，0.05%劣っている．容量スケーリング・MIP 近傍探索法は．コンバインド法と切除平面・局所分枝法にはそれぞれ0.13%，0.07%劣っているが，並列局所探索法よりも0.05%優れている．一方，容量スケーリング・MIP 近傍探索法の最良値は，コンバインド法には0.03%劣っているが，並列局所探索法と切除平面・局所分枝法よりもそれぞれ0.03%，0.15%優れており，コンバインド法以外の解法よりも良い解を算出しており，誤差が0.4%以内となる解法はコンバインド法，切除平面・局所分枝法と容量スケーリング・MIP 近傍探索法のみである．

表２に，局所分枝法に加えて，C 問題に対して精度の高い蟻コロニー法，遺伝的・局所分枝法，反復線形計画法，コンバインド法，容量スケーリング・貪欲法，切除平面・局所分枝法，並列局所探索法，容量スケーリング・局所分枝法，容量スケーリング・MIP 近傍探索法，および最良値における個別問題の上界値を示す．なお，LB/OPT 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．

コンバインド法は，最適値が25問，最適値を除く最良値が６問であり，37問の内の31問の最良値を算出している．切除平面・局所分枝法は，最適値が20問，最適値を除く最良値は１問，合計21問の最良値を算出している．また，並列局所探索法は，最適値が17問，最良値が１問の合計18問の最良値を算出している．一方，容量スケーリング・局所分枝法は最適値が９問，最適値を除く最良値が１問の合計10問，容量スケーリング・MIP 近傍探索法は最適値が13問，最適値を除く最良値が１問の合計14問の最良値を算出している．また，容量スケーリング・MIP 近傍探索法の最良値では，最適値が20問，最適値を除く最良値が４問の合計24問の最良値を算出しており，これはコンバインド法に次ぐ良い結果となっている．

表３にC 問題に対する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．使用している主なコンピュータは以下の通りである．

・局所分枝法：Intel Core 2 Duo-2Cores，2.4GHz，RAM 2GB

・IP 探索法：Intel Xeon×8CPUs，2.66GHz，RAM8GByte

・MIP タブー探索法：Intel Xeon×8CPUs，2.66GHz，RAM8GByte

・サイクル進化法：Intel Xeon E5507-4Cores，2.26GHz

表 2 ：Results for C-Category Problems

N/A/K/FC	LB/OPT	LOBR	ANCO	GELO	ILPH	COMB	CSGR	CUTP	PALS	CSLB	CMIP	BEST
100/400/10/F/L	**23949.0**	24690	**23949**	**23949**	**23949**	**23949**	**23949**	**23949**	24022	24022	**23949**	**23949**
100/400/10/F/T	**63753.0**	67357	65187	64126	65247	**63753**	68448	64034	64207	65112	64207	64150
100/400/10/V/L	**28423.0**	**28423**	**28423**	**28423**	**28423**	**28423**	28599	**28423**	28486	**28423**	**28423**	**28423**
100/400/30/F/L	**49018.0**	49872	50012	49058	49018	**49018**	49500	**49018**	**49018**	49115	49115	**49018**
100/400/30/F/T	132769.4	141633	139380	137845	138784	136803	139181	*136621*	136861	138346	137122	137017
100/400/30/V/T	**384802.0**	384809	**384802**	**384802**	**384802**	**384802**	384809	**384802**	**384802**	**384802**	**384802**	**384802**
20/230/040/V/L	**423848.0**	**423848**	**423848**	**423848**	**423848**	**423848**	423933	**423848**	424075	**423848**	**423848**	**423848**
20/230/040/V/T	**371475.0**	**371475**	**371475**	**371475**	**371475**	**371475**	**371475**	**371475**	371573	**371475**	**371475**	**371475**
20/230/040/F/T	**643036.0**	**643036**	**643036**	**643036**	**643036**	**643036**	644132	**643036**	**643036**	643187	**643036**	**643036**
20/230/200/V/L	**94213.0**	95295	**94213**	**94213**	94218	**94213**	94228	**94213**	**94213**	94247	94247	94218
20/230/200/F/L	**1 137642.3**	143446	138109	137854	138348	**137642.3**	138123	**137642**	**137642**	138031.7	137763.7	137763.7
20/230/200/V/T	**97914.0**	98039	**97914**	**97914**	**97914**	**97914**	97968	**97914**	**97914**	98130	97968	**97914**
20/230/200/F/T	**135863.1**	141128	137162.5	137449	136102	**135863.1**	136115	135991	135867	136204.3	136290	136080
20/300/040/V/L	**429398.0**	**429398**	**429398**	**429398**	**429398**	**429398**	**429398**	**429398**	**429398**	**429398**	**429398**	**429398**
20/300/040/F/L	**586077.0**	**586077**	**586077**	**586077**	**586077**	**586077**	588229	**586077**	**586077**	588229	**586077**	**586077**
20/300/040/V/T	**464509.0**	**464509**	**464509**	**464509**	**464509**	**464509**	**464509**	**464509**	**464509**	**464509**	**464509**	**464509**
20/300/040/F/T	**604198.0**	**604198**	**604198**	**604198**	**604198**	**604198**	**604198**	**604198**	**604198**	**604198**	**604198**	**604198**
20/300/200/V/L	**74811.0**	76375	75084	75366	75003	**74811**	74840	74946	**74811**	75055.3	74840	74840
20/300/200/F/L	114422.9	119143	116861	115963	116759	*115526*	115676.5	115574	115580	116074	115753.7	115731
20/300/200/V/T	**74991.0**	76168	**74991**	**74991**	**74991**	**74991**	74995	**74991**	**74991**	75038.2	74995	74995
20/300/200/F/T	**107102.0**	109808	107169	**107102**	**107102**	107167	107420.7	107284	**107102**	107666.8	107420.7	**107102**
30/520/100/V/L	**53958.0**	54026	53966	53974	**53958**	**53958**	53974	**53958**	53978	54004.5	53970	**53958**
30/520/100/F/L	**93922.3**	96255	94653	94094	93991	*93967*	94066	94043	*93967*	94201	94147	*93967*
30/520/100/V/T	**52046.0**	52129	52079	**52046**	**52046**	**52046**	**52046**	**52046**	**52046**	52058	**52046**	**52046**
30/520/100/F/T	96518.6	101102	97581.3	98333	97711	*97107*	97913.5	97361	97862	97707	97865.5	*97107*
30/520/400/V/L	**112774.4**	114367	113018.6	112871	112957	**112774.4**	**112774.4**	112786	112787	112785.7	**112774.4**	**112774.4**
30/520/400/F/L	**148257.9**	157726	149944	150624	151134	149151	*149093.2*	149328	149677	*149093.2*	*149093.2*	*149093.2*
30/520/400/V/T	**114640.5**	115240	115038	114884	114757	**114640.5**	**114640.5**	**114640**	**114641**	**114640.5**	114680	**114640.5**
30/520/400/F/T	151085.6	168561	159919.5	154044	154859	*152476.7*	152556.3	152745	154137	152924.5	153068.8	152513.3
30/700/100/V/L	**47603.0**	**47603**	**47603**	**47603**	**47603**	**47603**	47612	**47603**	**47603**	**47603**	**47603**	**47603**
30/700/100/F/L	**59958.0**	60272	60184	60184	60011	**59958**	60058	59987	60058	60066	60122	**59958**
30/700/100/V/T	**45871.5**	45905	45875	45905	45908	**45871.5**	45907	45875	45879	45898	45873	45873
30/700/100/F/T	**54904.0**	55104	**54904**	54925	**54904**	54912	55024	**54904**	**54904**	54938	55096	**54904**
30/700/400/V/L	97543.9	103787	*97808.6*	98747	98534	97853.4	97924	97960	98090	98024.5	97889.5	97889.5
30/700/400/F/L	132027.9	169760	137539.7	136748	141170	*134553.7*	1348707	135128	136257	134885.7	134920.5	134609.6
30/700/400/V/T	94785.4	96680	95271	95704	95863	*95249.6*	95268.9	95321	95651	95283.4	95268.4	95268.4
30/700/400/F/T	128682.1	144926	130106.9	130842	131814	129990	130115.7	130197	131104	130166	130262.4	*129982.3*

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

LOBR	IPSE*	MIPT	CEVO	ANCO	GELO*	ILPH
574.6	1768.8	6957.0	7200.0	3135.4	291.6	3600
COMB	**CSGR**	**CUTP**	**PALS***	**CSLB**	**CMIP**	**BEST**
6679.4	23.6	1849.0	152.7	87.7	85.5	82.5

*: Time to Best Solutions

・コンバインド法：Intel i7-4Cores，3.4GHz，RAM24GB
・蟻コロニー法：未記載
・遺伝的・局所分枝法：Core 2 CPU，2.66GHz，RAM4GByte
・反復線形計画法：Intel i7-4900MQ-4Cores，2.8GHz，RAM16GB
・容量スケーリング・貪欲法：Intel i7-4Cores，3.5GHz，RAM32GB
・切除平面・局所分枝法：Intel Core 2 Duo-2Cores，2.53GHz，RAM4GB
・並列局所探索法：Intel Xeon X5650-6Cores×2CPUs8Nodes，2.66GHz, RAM24GB

局所分枝法と切除平面・局所分枝法では古い CPU を用いており，使用コンピュータの計算速度は相対的に遅い．一方，並列局所探索法は2CPU（６コア）のノードを８台をつなげたクラスタコンピュータであり，単純計算では16台分の CPU（96コア）に相当する．なお，本研究では8Core の CPU を使用している．

従来の研究では，局所分枝法は574.6秒，IP 探索法は1768.8秒，MIP タブー探索法は6957.0秒，サイクル進化法は7200秒，蟻コロニー法は3135.4秒，遺伝的・局所分枝法は291.6秒，反復線形計画法は3600秒である．なお，サイクル進化法は２時間，反復線形計画法は１時間の計算時間を設定している．また，IP 探索法，遺伝的・局所分枝法および並列局所探索法は最良解が求められた時間である．平均誤差の最も小さいコンバインド法は範囲の広い局所分枝法を行っているため，大きな計算時間が必要であり，6679.4秒となっている．同様に，切除平面・局所分枝法は1849.0秒と大きな計算時間となっている．一方，並列局所探索法は152.7秒と比較短い計算時間ではあるが，96コアの高性能なクラスタコンピュータ上で並列計算を行っており，計算機の能力が高いため，直接，他の解法を比較することはできない．また，最良解が求められた時間であることに注意する．容量スケーリング・貪欲法は23.6秒と最も計算時間が短い．それに対して，提案した容量スケーリング・局所分枝法は87.7秒，容量スケーリング・MIP 近傍探索法は85.5秒または82.5秒と100秒以内であり，容量スケーリング・貪欲法よりは計算時間が必要であるが，それ以外の解法よりも大幅に計算時間が短縮されている．なお，容量スケーリング・MIP 近傍探索法の最良値では，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，従来に研究に比べ，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．

表４に個別のＣ問題の計算時間を示す．容量スケーリング・局所分枝法は最大で260秒

程度，容量スケーリング・MIP 近傍探索法も最大300秒程度であり，最良値（BEST）では最大400秒程度である．容量スケーリング・MIP 近傍探索法では，最良値との間で同じ問題であっても計算時間大きく異なる場合がある．これは，MIP 近傍探索法において，スケーリングパラメータによって初期解が異なるため，近傍探索における解の更新の回数が異なることに関係している．

表５にＲ問題に対する上界値の平均誤差を示す．容量スケーリング・局所分枝法におけるスケーリングパラメータは0.38，容量スケーリング・MIP 近傍探索法では0.44とした．

Ｒ問題に対して公開されている結果はそれほど多くない．ここでは，コンバインド法（COMB）と容量スケーリング法・貪欲法（CSGR）の結果を併記する．従来の研究では，コンバインド法が誤差0.11％，容量スケーリング・貪欲法が誤差0.42％となっている．一方，容量スケーリング・局所分枝法は誤差0.34％であり，容量スケーリング・MIP 近傍探索法は0.28％，容量スケーリング・MIP 近傍探索法の最良値は0.17％であった．容量スケーリング・局所分枝法は，コンバインド法に0.23％劣っている．また，容

表４：Computation Time for C-Category Problems（Seconds）

N/A/K/FC	LOBR	ANCO	GELO*	ILPH	COMB	CSGR	CUTP	PALS*	CSLB	CMIP	BEST
100/400/10/F/L	547	2833.7	1	3600	78.0	0.2	786	1	1.8	0.3	0.3
100/400/10/F/T	600	10800.0	1	3600	10423.1	0.6	822	53	43.8	172.9	225.8
100/400/10/V/L	600	7.1	33	3600	2.0	0.0	5	3	1.7	0.7	0.7
100/400/30/F/L	600	699.3	386	3600	2165.0	3.1	367	29	109.3	70.3	22.7
100/400/30/F/T	600	10800.0	239	3600	18731.2	8.2	377	79	260.1	138.8	144.8
100/400/30/V/T	600	691.4	362	3600	68.9	0.4	31	42	17.2	20.7	19.8
20/230/040/V/L	328	6.3	507	3600	0.9	0.1	11	2	0.9	1.0	0.8
20/230/040/V/T	440	8.9	1	3600	3.2	0.1	5	1	2.4	0.5	0.4
20/230/040/F/T	600	23.3	8	3600	9.8	0.1	38	10	7.9	5.8	5.2
20/230/200/V/L	600	2306.0	2	3600	6061.1	9.5	1523	123	30.3	25.2	61.4
20/230/200/F/L	600	2668.0	2	3600	6722.5	10.9	2085	144	89.7	45.3	34.7
20/230/200/V/T	600	1413.8	379	3600	2705.9	8.3	2483	52	25.0	16.9	56.1
20/230/200/F/T	600	5216.0	432	3600	9707.3	28.5	1297	240	68.1	63.1	216.1
20/300/040/V/L	146	6.6	391	3600	0.8	0.0	4	1	0.2	0.1	0.1
20/300/040/F/L	600	16.2	461	3600	8.4	0.1	56	16	0.6	7.3	6.3
20/300/040/V/T	600	10.8	304	3600	3.8	0.1	3	23	0.8	1.9	0.5
20/300/040/F/T	600	11.8	319	3600	3.5	0.1	3	3	0.3	0.2	0.2
20/300/200/V/L	600	2646.0	273	3600	10622.0	18.2	4392	361	65.1	60.5	59.4
20/300/200/F/L	600	2564.0	369	3600	14044.3	23.8	3279	250	108.8	184.2	72.3
20/300/200/V/T	600	2750.0	583	3600	2549.6	13.3	1837	58	72.8	62.1	60.3
20/300/200/F/T	600	1988.2	452	3600	8720.0	25.4	6083	122	51.4	57.0	162.4
30/520/100/V/L	600	672.9	296	3600	1680.4	2.1	329	20	34.8	105.6	36.8
30/520/100/F/L	600	1983.0	571	3600	7500.0	13.3	2004	83	49.6	253.0	227.6
30/520/100/V/T	600	751.1	25	3600	4048.5	1.2	1850	32	30.3	70.2	4.8
30/520/100/F/T	600	964.0	198	3600	15791.7	15.4	978	158	186.7	167.9	165.3
30/520/400/V/L	600	3303.0	204	3600	9501.6	64.1	3009	542	136.2	85.6	67.0
30/520/400/F/L	600	3723.0	278	3600	11131.2	70.6	4231	463	180.3	116.6	112.1
30/520/400/V/T	600	7801.7	441	3600	8262.8	29.6	4103	461	127.2	62.1	52.8
30/520/400/F/T	600	13257.6	335	3600	19593.7	122.0	5238	288	294.5	150.4	111.5
30/700/100/V/L	600	114.3	536	3600	64.4	0.8	84	179	5.6	37.6	3.7
30/700/100/F/L	600	700.4	678	3600	7631.5	17.0	1029	111	95.7	155.5	125.4
30/700/100/V/T	600	1922.0	381	3600	8806.6	6.0	976	258	94.7	235.1	128.2
30/700/100/F/T	600	801.1	191	3600	7639.7	12.1	2109	173	146.9	138.1	179.1
30/700/400/V/L	600	3477.0	105	3600	13311.9	45.6	1938	243	176.3	76.9	73.1
30/700/400/F/L	600	12961.0	524	3600	16946.6	206.6	2419	223	265.7	231.8	414.2
30/700/400/V/T	600	5311.2	331	3600	10743.0	46.2	8327	374	190.2	84.5	79.4
30/700/400/F/T	600	10800.0	191	3600	13656.0	70.9	4297	428	271.0	256.8	122.2

*: Time to Best Solutions

表５：Average Gap for R-Category Problems（%）

COMB	CSGR	CSLB	CMIP	BEST
0.11	0.42	0.34	0.28	0.17

量スケーリング・MIP 近傍探索法はコンバインド法に0.17％劣っているが，その最良値は0.06％の差に留まっている．

表６および７に，Ｒ問題に対する個別の目的関数値を示す．コンバインド法では，81問の内，最適値が70問，最適値を除く最良値が８問の合計78問の最良値を算出しており，３問以外は最も良い解が求められている．一方，容量スケーリング・局所分枝法では最適値が37問，最適値を除く最良値はない．また，容量スケーリング・MIP 近傍探索法

表６：Results for R-Category Problems

Group	C	F	LB/OPT	COMB	CSGR	CSLB	CMIP	BEST
r10	C1	F01	**200087**	**200087**	200407	**200087**	**200087**	**200087**
		F05	**346813.5**	**346813.5**	350735	**346813.5**	**346813.5**	**346813.5**
		F10	**488015**	**488015**	490875	**488015**	**488015**	**488015**
	C2	F01	**229196**	**229196**	**229196**	**229196**	**229196**	**229196**
		F05	**411664**	**411664**	414876	**411664**	**411664**	**411664**
		F10	**609104**	**609104**	616201	611702	610293	**609104**
	C8	F01	**486895**	**486895**	486951	486951	**486895**	**486895**
		F05	**951056**	**951056**	956414	**951056**	**951056**	**951056**
		F10	**1421740**	**1421746**	1421862	1421862	**1421746**	**1421746**
r11	C1	F01	**714431**	**714431**	**714431**	**714431**	**714431**	**714431**
		F05	**1263713**	**1263713**	1267626	**1263713**	1263981	**1263713**
		F10	**1843611**	**1843611**	1844097	1855297	1846817	1844978
	C2	F01	**870451**	**870451**	870784	870784	**870451**	**870451**
		F05	**1623640**	**1623640**	1625191	**1623640**	**1623640**	**1623640**
		F10	**2414060**	**2414060**	2419709	2418126	**2414060**	**2414060**
	C8	F01	**2294912**	**2294912**	**2294912**	**2294912**	**2294912**	**2294912**
		F05	**3507100**	**3507100**	**3507100**	**3507100**	**3507100**	**3507100**
		F10	**4579353**	**4579353**	**4579353**	**4579353**	**4579353**	**4579353**
r12	C1	F01	**1639443**	**1639443**	**1639443**	**1639443**	**1639443**	**1639443**
		F05	**3396050**	**3396050**	3417612	3411515	3403049.5	3403049.5
		F10	**5228711**	**5228711**	5273253.3	5277903.8	5274357	5270417
	C2	F01	**2303557**	**2303557**	2306043.5	**2303557**	**2303557**	**2303557**
		F05	**4669799**	**4669799**	**4669799**	**4669799**	**4669799**	**4669799**
		F10	**7100019**	**7100019**	**7100019**	**7100019**	**7100019**	**7100019**
	C8	F01	**7635270**	**7635270**	**7635270**	**7635270**	**7635270**	**7635270**
		F05	**10067742**	**10067742**	**10067742**	**10067742**	**10067742**	**10067742**
		F10	**11967768**	**11967768**	**11967768**	**11967768**	**11967768**	**11967768**
r13	C1	F01	**142947**	**142947**	**142947**	**142947**	**142947**	**142947**
		F05	**263800**	**263800**	266453	**263800**	**263800**	**263800**
		F10	**365836**	**365836**	370400	**365836**	367454	**365836**
	C2	F01	**150977**	**150977**	151269	**150977**	**150977**	**150977**
		F05	**282682**	**282682**	285415	283080	**282682**	**282682**
		F10	**406790**	**406790**	409626	409626	409626	409626
	C8	F01	**208088**	**208088**	208656.3	**208088**	**208088**	**208088**
		F05	**444826**	**444826**	450795	449636	**444826**	**444826**
		F10	**697967**	**697967**	720788	711699	698000	**697967**
r14	C1	F01	**403414**	**403414**	403529	**403414**	**403414**	**403414**
		F05	**749503**	**749503**	754757	751520	750944	**749503**
		F10	**1063098**	**1063098**	1076340	1069357	**1063098**	**1063098**
	C2	F01	**437607**	**437607**	**437607**	**437607**	437638	**437607**
		F05	**849163.0**	**849163**	853202	851570	**849163**	**849163**
		F10	**1214609.0**	**1214609**	1216473	1215904	1215904	1215904
	C8	F01	**668216.3**	**668216.3**	670267	670635.7	**668216.3**	**668216.3**
		F05	**1613428.8**	**1613428.8**	1627952	1623886	1625143	1615053
		F10	**2602690**	**2602690**	2650638	2640929	2624207.2	2614995.5

表７：Results for R-Category Problems

Group	C	F	LB/OPT	COMB	CSGR	CSLB	CMIP	BEST
		F01	**1000787**	**1000787**	1001491	**1000787**	**1000787**	**1000787**
	C1	F05	**1966206**	**1966206**	1968199	1969727	1973359	1966584
		F10	2841252.2	*2884076.5*	2902001	2889066.5	2896832.5	2889749
		F01	**1148604**	**1148604**	1149188.7	1149192	**1148604**	**1148604**
r15	C2	F05	2460618.5	*2476246*	2477502	2485023.7	2478747	*2476246*
		F10	3792777.6	*3831870.4*	3856335.5	3843106	3843106	3842799
		F01	**2297919**	**2297919**	2298492.5	2299115.4	2299115.4	2298081.3
	C8	F05	**5573412.8**	**5573412.8**	5577946.5	5574776	5577946.5	**5573412.8**
		F10	**8696932**	**8696932**	8699691.5	**8696932**	**8696932**	**8696932**
		F01	**136161**	**136161**	136722	**136161**	**136161**	**136161**
	C1	F05	**239500**	**239500**	240356	240242	239810	**239500**
		F10	**325671**	**325671**	326458	**325671**	**325671**	**325671**
		F01	**138532**	**138532**	**138532**	**138532**	**138532**	**138532**
r16	C2	F05	**241801**	**241801**	243456	**241801**	**241801**	**241801**
		F10	**337762**	**337762**	337908	**337762**	**337762**	**337762**
		F01	**169233**	**169233**	171179	169488	169336	**169233**
	C8	F05	**348167**	**348167**	355611	349822	350208	348186
		F10	**529988**	**529988**	537096	534626	534626	531347
		F01	**354138**	**354138**	354223	**354138**	**354138**	**354138**
	C1	F05	**645488**	**645488**	653017	**645488**	**645488**	**645488**
		F10	**910518**	**910518**	920776	917683	915958	911515
		F01	**370590**	**370590**	371269	**370590**	**370590**	**370590**
r17	C2	F05	**706746.5**	**706746.5**	709316	708993	708993	708368
		F10	**1019646**	*1019917*	1033068	1027434	1024738	1021097
		F01	**501634.5**	**501634.5**	503770	502748	502385	501751
	C8	F05	**1100679.3**	*1105083*	1108224.3	1107677	1107677	*1105083*
		F10	**1763131.8**	*1781436*	1790088	1785819	1793141	1785819
		F01	**828117**	**828117**	830366	**828117**	**828117**	**828117**
	C1	F05	**1533675**	**1533675**	1544783	1535457	1536306	1535457
		F10	**2174031**	*2174276*	2208324	2200547	2188346	2183614
		F01	**919325**	**919325**	921565	920505	921565	920408
r18	C2	F05	1802828.4	1826245	1833652	1831418	1832967	*1826039*
		F10	2645545.6	2703852	2730882	2725746	2745858	*2701714*
		F01	1470580.5	*1477395*	1481353.3	1479760.4	1480756.4	1478466
	C8	F05	**3887238**	3896893.2	3914562.6	3905815	3900861	*3895908.5*
		F10	**6361906**	**6361906**	6376447	6378087	6376278	6370685

では最適値が44問，最適値を除く最良値はない．しかし，容量スケーリング・MIP 近傍探索法の最良値では，最適値が53問，最適値を除く最良値が５問の合計58問の最良値を算出している．

表８にＲ問題に対する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．平均誤差の最も小さいコンバインド法は，範囲の広い局所分枝法を行っているため，大きな計算時間が必要であり，3310.5秒となっている．容量スケーリング・貪欲法は10.1秒と短い計算時間となっている．提案した容量スケーリング・局所分枝法は36.1秒，容量スケーリング・MIP 近傍探索法は53.9秒または56.4秒であり，容量スケーリング・貪欲法よりは計算時間が必要であるが，誤差の小さなコンバインド法と比べて大幅に計算時間が短縮されている．提案した解法では，従来に研究に比

表８：Average Computation Time for R-Category Problems（Seconds）

COMB	CSGR	CSLB	CMIP	BEST
3310.5	10.1	36.1	53.9	56.4

表９：Computation Time for R-Category Problems（Seconds）

Group	C	F	COMB	CSGR	CSLB	CMIP	BEST
r10	C1	F01	0.6	0.0	1.4	1.0	0.8
		F05	8.5	0.3	5.9	3.9	3.2
		F10	17.8	0.3	10.3	5.9	5.5
	C2	F01	5.5	0.1	0.4	0.4	0.3
		F05	59.6	0.2	6.0	10.1	9.5
		F10	140.7	0.6	23.4	14.7	22.3
	C8	F01	11.0	0.6	0.8	3.8	3.2
		F05	22.0	0.5	10.3	9.5	7.2
		F10	23.3	0.5	5.5	10.1	6.2
r11	C1	F01	10.2	0.4	0.8	0.5	0.4
		F05	573.8	2.2	30.8	87.5	39.6
		F10	2334.4	2.5	11.5	77.0	120.6
	C2	F01	83.5	1.1	2.5	14.2	2.6
		F05	303.4	2.5	27.1	43.2	42.6
		F10	875.9	6.3	47.5	87.0	5.3
	C8	F01	5.1	0.8	1.0	0.4	0.4
		F05	8.3	0.7	1.6	0.6	0.5
		F10	5.4	0.5	1.9	0.9	0.9
r12	C1	F01	83.5	1.8	6.0	2.6	2.1
		F05	5311.2	23.8	53.7	204.0	199.8
		F10	7333.6	19.9	29.4	146.5	39.6
	C2	F01	77.3	3.6	26.9	23.4	22.4
		F05	114.7	3.6	8.4	6.1	5.7
		F10	132.1	2.8	9.5	3.7	3.6
	C8	F01	7.3	1.6	2.1	0.7	0.7
		F05	4.3	1.0	2.9	1.4	1.3
		F10	13.6	0.6	2.7	1.3	1.2
r13	C1	F01	1.1	0.1	0.6	0.2	0.2
		F05	40.7	0.7	9.8	7.9	4.3
		F10	61.1	0.7	10.0	17.6	9.5
	C2	F01	3.8	0.1	0.6	0.6	0.4
		F05	74.0	0.8	8.6	14.6	7.3
		F10	356.2	0.9	4.7	3.4	3.1
	C8	F01	225.0	0.4	40.7	26.8	1.0
		F05	4011.8	0.6	29.4	145.7	123.2
		F10	5376.8	3.1	32.0	146.0	128.0
r14	C1	F01	14.6	0.5	6.9	7.2	6.0
		F05	1834.9	2.9	83.3	111.4	114.6
		F10	3223.1	3.0	27.5	104.1	62.1 1
	C2	F01	25.9	0.6	10.0	11.6	0.8
		F05	4905.9	3.8	102.8	182.2	132.7
		F10	4769.8	3.8	58.0	53.3	53.3
	C8	F01	3995.8	5.4	10.6	98.1	45.3
		F05	8440.1	22.5	37.2	200.0	200.3
		F10	6806.1	13.8	21.7	216.2	252.5

表10：Computation Time for R-Category Problems（Seconds）

Group	C	F	COMB	CSGR	CSLB	CMIP	BEST
r15	C1	F01	208.6	1.8	6.9	3.3	3.1
		F05	13158.1	10.6	110.2	176.2	100.2
		F10	11905.3	12.5	128.7	78.0	74.2
	C2	F01	3049.4	5.7	14.3	8.8	7.6
		F05	9029.4	33.3	81.9	74.7	57.4
		F10	21993.4	31.6	69.8	57.7	59.0
	C8	F01	2465.6	15.8	17.4	11.1	13.4
		F05	316.4	5.6	19.0	12.4	14.3
		F10	103.8	2.3	39.5	37.7	34.3
r16	C1	F01	0.6	0.1	0.7	0.3	0.3
		F05	59.3	1.3	4.3	14.8	9.3
		F10	114.3	1.4	16.2	18.1	11.5
	C2	F01	1.9	0.1	0.3	0.2	0.2
		F05	33.5	1.4	10.3	9.7	2.3
		F10	212.6	1.5	37.0	30.9	26.8
	C8	F01	2973.7	0.3	20.9	31.3	146.3
		F05	6730.1	1.1	111.0	81.7	41.9
		F10	7734.3	3.1	44.3	45.2	321.2
r17	C1	F01	15.2	0.8	6.7	6.3	1.6
		F05	1145.0	4.5	105.0	150.5	53.3
		F10	7501.1	5.0	112.0	140.6	300.0
	C2	F01	69.8	1.1	33.8	26.0	22.9
		F05	5679.2	5.7	21.3	17.4	95.3
		F10	7823.3	6.4	149.2	105.2	342.6
	C8	F01	6135.1	8.7	20.7	178.1	38.5
		F05	8979.1	18.1	46.6	42.6	33.2
		F10	12489.6	25.6	91.3	47.5	85.8
r18	C1	F01	2455.5	10.8	50.2	107.4	9.4
		F05	6817.3	13.1	84.6	70.7	65.1
		F10	12505.6	13.1	170.0	197.2	233.1
	C2	F01	9410.5	13.5	59.8	49.3	207.9
		F05	14321.7	19.2	91.7	73.9	75.0
		F10	12970.2	20.9	94.3	161.0	161.2
	C8	F01	14508.3	40.0	75.7	81.3	100.0
		F05	13431.7	54.5	77.2	65.9	63.8
		F10	25608.2	37.2	76.3	48.0	68.4

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．表９に個別問題の計算時間を示す．

5　おわりに

本研究では，アークに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問題およびＲ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．

従来の研究の最良の解法の一つであるであるコンバインド法と比較すると誤差はわずかながら大きいが，コンバインド法は膨大な計算時間を必要としている．近年に提案されたコンバインド法以外のいずれの解法よりも，誤差の小さな良い解を求めることがで

きた．また，高速で精度の劣る容量スケーリング・貪欲解法よりも計算時間を必要とするが，それ以外の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

M. Chouman and T. G. Crainic. A MIP-tabu search hybrid framework for multicommodity capacitated fixed-charge network design. Technical Report CIRRELT-2010-31, Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation, Université de Montréal, 2010.

T. G. Crainic, A. Frangioni, and B. Gendron. Bundle-based relaxation methods for multicom modity capacitated fixed charge network design problems. *Discrete Applied Mathematics*, 112: 73-99, 2001.

M. Matteo Fischetti and A. Lodi. Local branching. *Mathematical Programming*, 98（1-3）: 23-47, 2003.

B. Gendron, S. Hanif, and R. Todosijević. An efficient matheuristic for the multicommodity fixed-charge network design problem. *IFAC-PapersOnLine*, 49（12）: 117-120, 2016.

M. Hewitt, G. L. Nemhauser, and M. Savelsbergh. Combining exact and heuristics approaches for the capacitated fixed charge network flow problem. *Journal on Computing*, 22: 314-325, 2010.

N. Katayama. A combined capacity scaling and local branching approach for capacitated multi-commodity network design problem. *Far East Journal of Applied Mathematics*, 92: 1-30, 2015.

N. Katayama. A combined fast greedy heuristic for the capacitated multicommodity network design problem. *Working Paper*, 2017.

N. Katayama, M. Z. Chen, and M. Kubo. A capacity scaling procedure for the multi-commodity capacitated network design problem. *Journal of Computational and Applied Mathematics*, 232（2）: 90-101, 2009.

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

M. Momeni and M. Sarmadi. A genetic algorithm based on relaxation induced neighborhood search in a local branching framework for capacitated multicommodity network design. *Networks and Spatial Economic*, 16（2）: 447468, 2016.

L. Munguí, S. Ahmed, D. A. Bader, G. L. Nemhauser, V. Goel, and Y. Shao. A parallel local search frame work for the fixed-charge multicommodity network flow problem. *Computers & Operations Research*, 77: 44-57, 2017.

D. C. Paraskevopoulos, T. Bektas, T. Crainic, and C. N. Potts. A cycle-based evolutionary algorithm for the fixed-charge capacitated multi-commodity network design problem. *European Journal of Operatinal Research*, 253 (2), 265-279, 2016.

I. Rodríguez-Martín and J. J. Salazar-Gonzáleza. A local branching heuristics for the capacitated fixed-charge network design problem. *Computers & Operations Research*, 37: 575-581, 2010.

M. Yaghini and A. Foroughi. ACO-based neighborhoods for fixed-charge capacitated multicommodity network design problem. *International Journal of Transportation Engineering*, 1: 311-334, 2014.

M. Yaghini, M. Karimi, M. Rahbar, and M. H. Sharitabaro. A cutting-plane neighborhood structure for fixed-charge capacitated multicommodity network design problem. *INFORMS Journal on Computing*, 27 (1): 45-58, 2016.

片山直登．容量制約をもつネットワークデザイン問題の高速な貪欲解法．流通経済大学流通情報学部紀要，20 (1): 1-24, 2015.

Algorithm 1: Capacity Scaling and Local Branch

Set A, $\bar{P}$;
Set λ, ϵ, ITE_{min}, ITE_{max}, $ArcNum$, M, α, T;
Solve the linear relaxaion problem $CNDPL(A, \bar{P}, C)$ of $CNDP(A, \bar{P}, C)$;
$C^0 \leftarrow C$; $l \leftarrow 1$;
repeat
  Solve $CNDPL(A, \bar{P}, C^l)$;
  Get the solution $\tilde{y}$ of $CNDPL(A, \bar{P}, C^l)$;
  Add paths to $\bar{P}$ by Column Genaration;
  $n \leftarrow 0$;
  for $(i, j) \in A$ **do**
    $C^l_{ij} \leftarrow \lambda C^{l-1}_{ij} \tilde{y}_{ij} + (1 - \lambda) C^{l-1}_{ij}$;
    if $\tilde{y}_{ij} > \epsilon$ **then**
      $n \leftarrow n + 1$;
    end
  end
until $l \geq ITE_{min}$ *and* $n \leq ArcNum$*, or* $l \geq ITE_{max}$;
$\bar{A} \leftarrow \varnothing$;
for $(i, j) \in A$ **do**
  if $\tilde{y}_{ij} > \epsilon$ **then**
    $\bar{A} \leftarrow \bar{A} \cup \{(i, j)\}$;
  end
end
Solve $CNDP(\bar{A}, \bar{P}, C)$, and get the solution $\bar{y}$ and the upper bound UB;
repeat
  Add equations (13) and (17) to $CNDA(A)$ for the current solution $\hat{y}$;
  Solve $CNDA(A)$ within time T;
  if $CNDA(A)$ *has no feasible solution* **then**
    Delete equations (13) and (15) from $CNDA(A)$;
    Add equations (14) and (15) to $CNDA(A)$ for the current solution $\hat{y}$;
    Solve $CNDA(A)$ within time T;
  end
  if *the solution* $\tilde{y}$ *of* $CNDA(A)$ *is found* **then**
    Get the upper bound UB_{local} of $CNDA(A)$;
    $\hat{y} \leftarrow \tilde{y}$;
    $UB \leftarrow UB_{local}$
  else
    $M \leftarrow \lfloor M/\alpha \rfloor$;
  end
until $M = 0$;
Return $\hat{y}$,UB;

Algorithm 2: Capacity Scaling and MIP Neighborhood Search

Set A, $\bar{P}$;
Set λ, ϵ, ITE_{min}, ITE_{max}, $ArcNum$, M, α, T;
Solve the linear relaxaion problem $CNDPL(A, \bar{P}, C)$ of $CNDP(A, \bar{P}, C)$;
$C^0 \leftarrow C$; $l \leftarrow 1$;
repeat
    Solve $CNDPL(A, \bar{P}, C^l)$;
    Get the solution $\tilde{y}$ of $CNDPL(A, \bar{P}, C^l)$;
    Add paths to $\bar{P}$ by Column Genaration;
    for $(i, j) \in A$ **do**
        $C^l_{ij} \leftarrow \lambda C^{l-1}_{ij} \tilde{y}_{ij} + (1 - \lambda) C^{l-1}_{ij}$;
    end
    $n \leftarrow 0$;
    for $(i, j) \in A$ **do**
        if $\tilde{y}_{ij} > \epsilon$ **then**
            $n \leftarrow n + 1$;
        end
    end
until $l \geq ITE_{min}$ *and* $n \leq ArcNum$, *or* $l \geq ITE_{max}$;
$\bar{A} \leftarrow \varnothing$;
for $(i, j) \in A$ **do**
    if $\tilde{y}_{ij} > \epsilon$ **then**
        $\bar{A} \leftarrow \bar{A} \cup \{(i, j)\}$;
    end
end
Solve $CNDP(\bar{A}, \bar{P}, C)$, and get the solution $\bar{y}$ and the upper bound UB;
repeat
    Add equations (16), (17) and (15) to $CNDA(A)$ for the current solution $\hat{y}$;
    Solve $CNDA(A)$ within time T;
    Delete equations (16), (17) and (15) from $CNDA(A)$;
    if $CNDA(A)$ *has no feasible solution* **then**
        break;
    else
        if *the solution* $\tilde{y}$ *of* $CNDA(A)$ *is found* **then**
        Get the upper bound UB_{neigh} of CNDA（A）
            $\hat{y} \leftarrow \tilde{y}$;
            $UB \leftarrow UB_{neigh}$
        else
            $M \leftarrow \lfloor M/\alpha \rfloor$;
        end
    end
until $M = 0$;
Return $\hat{y}, UB$;