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の高速解法
Fast algorithms for Unsplittable Capacitated Network Design Problem

片山 直登 流通経済大学 流通情報学部

平成 29 年 2 月 5 日

1 はじめに

ネットワーク設計問題は，ネットワーク上のアークにかかる固定的な費用とものの移動にかかる変動的な費用を考慮して，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，始点と終点をもつ複数の需要の移動経路を決める問題である．この問題は，輸送，ロジスティクス，通信や生産システムなどに幅広い応用分野をもつネットワークの構造を設計する問題である．ネットワーク設計問題とその関連問題に関するサーベイとして，Magnanti and Wong (1984)，Wong (1984, 1985)，Minoux (1989)，Balakrishnan et al. (1997)，Gendron et al. (1997)，Crainic (2003)，Costa (2005)，片山直登 (2008) および Yaghini and Rahbar (2012) などがある．

ネットワーク設計問題には，アークに容量制約をもたない問題と容量制約をもつ問題がある．後者の容量制約をもつネットワーク設計問題は，ネットワーク設計問題の中でも最適解や良い近似解を求めることが困難な問題である．ロジスティクスやサプライチェーンのネットワーク上では，同一の発地と着地をもつ荷物は，複数に分割されて別々の経路で輸送されることはなく，単一の経路上で輸送されるのが一般的である．このような分割されないフローを非分割フローとよび，この非分割フローを考慮した問題を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 ($UCND$: Unsplittable Capacitated Network Design Problem) とよび，従来の設計問題を分割フローを許すネットワーク設計問題 ($SCND$: Splittable Capacitated Network Design Problem) とよぶ．$UCND$ では，アークのデザイン変数のみならず，パスやアークフロー変数も 0–1 離散変数となり，すべての変数が 0–1 離散変数である困難な組合せ最適化問題となる．

$SCND$ において，0–1 離散変数であるデザイン変数を 0 または 1 に固定した問題は多品種フロー問題となる．この問題は線形計画問題となるため，汎用の最適化ソルバー用いることにより比較的容易に最適解を求めることができる．一方，

$UCND$ではデザイン変数を固定した問題は0–1離散変数であるフロー変数をもつ多品種離散フロー問題となるため，この大規模な組合せ最適化問題を最適に解くことは困難である．

$UCND$に対する研究として，Yaghini and Kazemzadeh (2012) のシミュレーテッドアニーリング法を用いた研究，Hewitt et al. (2013) のIP探索法および分枝価格法とガイドつき探索法を用いた研究，片山直登 (2013) の容量スケーリング法と局所探索法およびパス再結合法を用いた研究がある．これらの研究は高い精度の近似解を算出することに主眼を置いているため，大規模な問題では多くの計算時間を要する．

一方，$SCND$に対して，片山直登 (2015) は高速な貪欲法を提案し，従来の有力な解法の1/100程度の計算時間で，従来の解法と同程度の精度の解の算出に成功している．

本研究では，$UCND$に対する高速な近似解法を提案する．この解法は，容量スケーリング法によりアーク集合，パス集合を限定し，これらに限定した問題を最適化ソルバーを用いて解く近似解法と，高速な貪欲法を適用した近似解法である．

2 問題の定式化

はじめに，$UCND$の前提条件，使用する記号および$UCND$の定義を示す．続いて，アークフローによる定式化，およびパスフローによる定式化を示す．

2.1 前提条件，記号および問題の定義

$UCND$の前提条件は次のようなものである．ノード集合，向きをもつアーク集合と複数の品種からなる品種集合が与えられている．また，アークには非負のデザイン費用，品種ごとの全需要に対する非負のフロー費用，および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．加えて，各品種ごとの需要が与えられており，各品種の需要は始点から終点までの1つのパス上を移動する．

続いて，$UCND$の定式化で使用する記号の定義を示す．ノード集合をN，アーク集合をA，品種集合をKとし，品種kの取り得るパス集合をP^kとする．また，アーク(i,j)上における品種kの全需要に対する非負のフロー費用をc_{ij}^k，アーク(i,j)の非負のデザイン費用をf_{ij}，アーク(i,j)のアーク容量をC_{ij}，品種kの需要をd^kとする．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とし，品種kの始点をO^k，品種kの終点をD^kとする．一方，アーク(i,j)を選択するとき1，そうでないとき0である0–1変数であるデザイン変数をy_{ij}とする．品種kのフローがアーク(i,j)上に含まれるとき1，そうでないとき0である0–1変数であるアークフロー変数をx_{ij}^kとし，品種kのフローがパスp上を移動するとき1，そうでないとき0である0–1変数であるパスフロー変数をz_p^kとする．

次に，$UCND$ の定義を示す．

定義 2.1 ($\boldsymbol{UCND}$) 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 が与えられている．このとき，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 $A'(\subseteq A)$，およびアーク容量を満足する各品種の始点・終点間の非分割アークフロー x または非分割パスフロー z を求めよ．

2.2　パスフローによる定式化

アーク集合 A，アーク容量 C とパス集合 P が与えられたときに，$UCND$ のパスフローによる定式化 $UCNDP(A, C, P)$ を示す．

$UCNDP(A, C, P)$:

$$\text{minimize} \quad \sum_{(i,j)\in A} \sum_{k\in K} c_{ij}^k \sum_{p\in P^k} \delta_{ij}^p z_p^k + \sum_{(i,j)\in A} f_{ij} y_{ij} \tag{1}$$

subject to

$$\sum_{p\in P^k} z_p^k = 1 \quad \forall k \in K \tag{2}$$

$$\sum_{k\in K} \sum_{p\in P^k} d^k \delta_{ij}^p z_p^k \leq C_{ij} y_{ij} \quad \forall (i,j) \in A \tag{3}$$

$$\sum_{p\in P^k} \delta_{ij}^p z_p^k \leq y_{ij} \quad \forall k \in K, (i,j) \in A \tag{4}$$

$$z_p^k \in \{0, 1\} \quad \forall p \in P^k, k \in K \tag{5}$$

$$y_{ij} \in \{0, 1\} \quad \forall (i,j) \in A \tag{6}$$

(1) 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．ここで，$\sum_{p\in P^k} \delta_{ij}^p z_p^k$ はアーク (i,j) 上の品種 k のフロー量である．(2) 式は，品種 k のパスフロー変数値の合計が 1，すなわちすべてのパスの中で単一のパス上のみにフローが流れることを表す．(3) 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．これは，アーク (i,j) が存在するときにのみフローの存在を許し，かつそのフロー量は容量以下であることを表している．(4) 式は，品種 k のアーク (i,j) 上のフロー量がアーク (i,j) 存在するときに最大 1 で最小 0 であり，アークが存在しないときに 0 であることを表す強制制約式である．(5) 式はパスフロー変数の 0–1 条件，(6) 式はデザイン変数の 0–1 条件である．

2.3　アークフローによる定式化

$UCND$ には，2 種類の変数，デザイン変数とフロー変数が含まれている．ここで，デザイン変数 y を $\bar{y}$ に固定し，$\bar{y}_{ij} = 1$ からなるアーク集合 A の部分集合を $\bar{A}$ とする．このとき，y を $\bar{y}$ に固定した $UCND$ は，アークフロー変数 x からなる $\bar{A}$ 上の多品種離散フロー問題 $MCFI$ として表すことができる．そこで，部分集合 $\bar{A}$ に対する $MCFI(\bar{A})$ の最適目的関数値とデザイン費用の和を $\phi(\bar{A})$ とおく．アーク集合 A が与えられたときに，$UCND$ は $\phi(\bar{A})$ を最小化するように，アーク集合 A から部分集合 $\bar{A}$ を選択する 2 段階の問題 $UCNDA(A)$ として定式化できる．

$UCNDA(A)$:

$$\text{minimize}_{\forall \bar{A} \subseteq A} \quad \phi(\bar{A}) \tag{7}$$

subject to

$MCFI(\bar{A})$:

$$\phi(\bar{A}) = \text{minimize} \sum_{(i,j)\in \bar{A}} \sum_{k\in K} c_{ij}^k x_{ij}^k + \sum_{(i,j)\in \bar{A}} f_{ij} \tag{8}$$

subject to

$$\sum_{i\in \bar{N}^-(n)} x_{in}^k - \sum_{j\in \bar{N}^+(n)} x_{nj}^k = \begin{cases} -1 & if\ n = O^k \\ \ \ 1 & if\ n = D^k \\ \ \ 0 & otherwise \end{cases} \quad \forall n \in N,\ k \in K \tag{9}$$

$$\sum_{k\in K} d^k x_{ij}^k \le C_{ij} \quad \forall (i,j) \in \bar{A} \tag{10}$$

$$x_{ij}^k \in \{0,1\} \quad \forall k \in K,\ (i,j) \in \bar{A} \tag{11}$$

ここで，$\bar{N}^-(n) = \{j \in N | (j,n) \in \bar{A}\}$，$\bar{N}^+(n) = \{j \in N | (n,j) \in \bar{A}\}$ である．

(8) 式は，フロー費用とデザイン費用の総和の最小化するアークを選択することを表す．(9) 式はフロー保存式であり，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 −1，終点 D^k であれば 1，その他の中継ノードであれば 0 となることを表している．これは，ネットワーク上のアークを経由して，与えられたすべての品種のフローが始点から終点に到着することを表している．(10) 式の左辺はアーク (i,j) 上のフロー量の合計であり，これがアーク容量以下となることを表す容量制約式である．(11) 式はアークフロー変数の 0–1 条件である．

3　容量スケーリング法と限定された分枝限定法

容量スケーリング法は容量制約をもつ問題に対する近似解法であり，$SCND$ に対して適用されている解法である．この解法では，線形緩和問題の解をもとに，アーク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．ネットワーク設計問題では，問題のサイズを縮約するために列生成法と行生成法が用い

られている．列生成法は限定主問題を解き，適時，被約費用が負となる変数を生成する方法である．また，行生成法は，列生成法により生成された変数に対する有効な制約式を逐次追加する方法である．これらの方法については，Katayama et al. (2009) に詳細が記述されている．

容量スケーリング法では，比較的少ない繰り返し回数で，大半のデザイン変数が 0 または 1 に収束することが知られている．容量スケーリング法は線形緩和問題の解をもとに 0–1 変数解を導く近似解法であるが，この 0–1 変数は容量をもつアークに対するデザイン変数である．このため，パスフロー変数に対して 0–1 変数解は導出することはできない．そこで，$UCND$ に対しては，パスフロー変数も連続緩和することにより，容量スケーリング法を適用する．

容量スケーリング法はデザイン変数の 0–1 解を導出するというよりも，解の候補となる可能性の高いデザイン変数を選定するために活用する．このため，適当な繰り返し回数で容量スケーリング法を打ち切り，その時点で 0 に収束しているデザイン変数を解の候補から外すことにする．

すべてのデザイン解が 0 または 1 に収束した場合，$SCND$ ではこれらのデザイン解を用いて，実行可能解を容易に導出することができる．しかし，$UCND$ ではフロー解は小数値を取りうるため，フロー解を 0 または 1 の離散値に修正した場合は多くのアークでアーク容量を超過する可能性がある．このため，容量スケーリング法から導出されるデザイン変数値が 1 であるアークのみを対象とした問題では，$UCND$ の実行可能性を保証することができない．

また，$SCND$ であれば，0 または 1 に収束したデザイン変数をそれぞれ 0 または 1 に固定し，それ以外を 0–1 変数に限定した問題で，多品種フロー問題を解けば，比較的に短い計算時間で適当な実行可能なフロー解を求めることができる．しかしながら，$UCND$ の場合では，このようにデザイン変数を固定しても多品種離散フロー問題となるため，最適解を求ることは困難である．

そこで，絞り込んだデザイン変数集合を対象とし，容量スケーリングにより生成されたパスフロー変数のみを変数とした限定された問題を作成し，最適化ソルバーの分枝限定法を適用する．デザイン変数およびフロー変数が限定された問題であるため，比較的容易に解くことができる．しかし，必ずしも短時間で最適解を求めることができる保証はないため，計算時間の上限を設定する．

容量スケーリング法をアルゴリズム 1 に示す．容量スケーリング法のアルゴリズムにおいて，$\hat{P}$ は現在までに生成されているパスの集合，ϵ はデザイン変数の収束判定基準，λ は容量スケーリングのスケーリングパラメータ，Ite_{min} と Ite_{max} は容量スケーリングの繰り返し回数の下限と上限である．また，α は容量スケーリング法の終了基準であり，0 または 1 に未収束のデザイン変数の数が α 以下となったら容量スケーリング法を終了する.アルゴリズムの出力は，0 に収束していないアーク集合 $\hat{A}$，限定されたパス集合 $\hat{P}$，およびアーク容量が $\hat{C}$ のときの線形緩和解 $\hat{y}$ である．

限定された分枝限定法のアルゴリズムをアルゴリズム 2 に示す．このアルゴリズムにおいて，T は最適化ソルバーの分枝限定法の計算時間の上限である．アル

ゴリズムの出力は，限定された分枝限定法により選択されたアーク集合 $\bar{A}$ と上界値 UB_{BB} である．

4 貪欲法

分割フローを許す容量制約のないネットワークデザインに対する基本的な解法として，アーク集合からアークを順々に取り除いていくデリートタイプの貪欲解法がある．現在のネットワークに含まれるすべてのアーク対して，あるアークを取り除いたネットワークにおいて最小費用（最短）経路問題を解き，目的関数値を求め，目的関数値の減少量の最大のアークを削除する．アークを取り除いたときの目的関数値の最大減少量が正である限り，この操作を繰り返す方法である．この解法では，削除アークを求めるたびにすべてのアークに対して目的関数値の減少量を定める必要があるため，計算量は膨大なものとなる．

Minoux (1976) は，目的関数値の減少量を繰り返しのたびに厳密に求めるのではなく，通常の場合は前回の目的関数値の減少量の近似値をそのまま利用し，アークが実際の削除の対象となった場合に限り，この減少量の近似値を再計算し，この値が依然アークの中で最大であればこのアークを削除し，そうでなければこのアークの評価値の更新だけを行う貪欲法を提案している．この解法は目的関数値の減少量の計算回数を大幅に減少することができるため，容量制約をもたないネットワークデザイン問題では効率的な解法となっている．片山直登 (2010) は，この Minoux の解法に厳密性を加えた改良貪欲法を開発している．一方，片山直登 (2015) は，改良貪欲法を $SCND$ に適用した高速な解法を開発している．ここでは，この高速な改良貪欲法の概念を $UCND$ に適用する．

$UCND$ では目的関数値の減少量を求めるために多品種離散フロー問題を解く必要があることから，すべてのアークを貪欲法の対象とした場合，非常に大きな計算時間を必要となる．そこで，あらかじめ容量スケーリング法によって実行可能なアーク集合に絞り込んでおく．続いて，アーク集合に含まれるアークの内，対応するデザイン変数が 1 に収束したものを 1 に固定する．また，0 または 1 に収束していない小数解をもつデザイン変数については，デザイン変数値の降順にソートし，アークをこの順に適当な数のアーク群に分割しておく．順次，降順でアーク群に含まれるアークのデザイン変数値を 1 に固定し,そうでない場合は 0 に固定し，汎用の最適化ソルバーの分枝限定法を用いて，多品種離散フロー問題を解く．この問題が実行不可能であれば，次に大きな小数解をもつアーク群に含まれるアークのデザイン変数を 1 に固定して，実行可能となる問題を解き直す．実行可能であれば，1 に固定したデザイン変数に対応するアークを貪欲法の対象とするアーク集合に加え，実行可能となるアーク集合を選定する．なお，選択されたアーク集合は必要最低限なものとなるので，貪欲法におけるアークの削除回数を抑えることができる．多品種離散フロー問題を解く際には，計算時間の上限を設定しておくが，この問題では実行可能性のみを判断するので，比較的計算時間は短くてすむ．

Algorithm 1: Capacity Scaling(A)

Set $\hat{P}$, α, λ, ϵ, Ite_{min}, Ite_{max};
Solve a linear relaxation problem $UCNDPL(A, C, \hat{P})$ of $UCNDP(A, C, \hat{P})$;
Get the solution $\tilde{y}$ of $UCNDPL(A, C, \hat{P})$;
Add generated paths to $\hat{P}$ by Column Generation;
Update $A_{\hat{P}k}, k \in K$, and add forcing constraints associated with $A_{\hat{P}k}$ by Row Generation;
$\hat{C}^0 \leftarrow C$;
$l \leftarrow 1$;
$\hat{A} \leftarrow \emptyset$;;
repeat
    for $(i, j) \in A$ **do**
        $\hat{C}^l_{ij} \leftarrow \lambda \hat{C}^{l-1}_{ij} \hat{y}_{ij} + (1 - \lambda)\hat{C}^{l-1}_{ij}$;
    end
    Solve $UCNDPL(A, \hat{C}^l, \hat{P})$;
    Get the solution $\hat{y}$ of $UCNDPL(A, C^l, \hat{P})$;
    $n \leftarrow 0$;
    for $(i, j) \in A$ **do**
        if $\hat{y}_{ij} > \epsilon$ *and* $\hat{y}_{ij} < 1 - \epsilon$ **then**
            $n \leftarrow n + 1$;
        end
        if $\hat{y}_{ij} > \epsilon$ **then**
            $\hat{A} \leftarrow \hat{A} \cup \{(i, j)\}$;
        end
    end
    Add generated paths to $\hat{P}$ by Column Generation;
    Update $A_{\hat{P}k}, k \in K$, and add forcing constraints associated with $A_{\hat{P}k}$ by Row Generation;
    $l \leftarrow l + 1$;
until $l > Ite_{min}$ *and* $n \leq \alpha$*, or* $l > Ite_{max}$;
Return $\hat{A}$,$\hat{P}$,$\hat{y}$;

Algorithm 2: Restricted Branch and Bound($\hat{P}, \hat{y}$)

Set ϵ, T;
for $(i,j) \in A$ **do**
  if $\tilde{y}_{ij} < \epsilon$ **then** add constraint $y_{ij} = 0$ to $UCNDP(A, \hat{C}, \hat{P})$;
  else if $\tilde{y}_{ij} > 1 - \epsilon$ **then**
    add constraint $y_{ij} = 1$ to $UCNDP(A, \hat{C}, \hat{P})$;
  end
end
Solve $UCNDP(A, \hat{C}, \hat{P})$ by an MIP solver within time T;
Get the objective function value UB_{BB} and $\bar{A}$ with arcs such as solution $y = 1$ of $UCNDP(A, \hat{C}, \hat{P})$;
Return $\bar{A}$, UB_{BB};

$SCND$ では，すべてのデザイン変数を 0 または 1 に固定し，ネットワークから特定のアークを削除，デザイン変数を 0 に固定した場合の目的関数値の減少量は多品種フロー問題となり，汎用の最適化ソルバーを用いれば比較的容易に最適解を求めることができる．一方，$UCND$ では特定のアーク (i,j) を削除したときの目的関数値 $\phi(\bar{A}\backslash\{(i,j)\})$ を求める問題は多品種離散フロー問題となることから，計算時間に制限を付けて最適化ソルバーにより直接解く方法と，タブサーチ法を用いて実行可能なフローを求める方法の 2 つの解法を提案する．

改良貪欲解法のアルゴリズムをアルゴリズム 3 に示す．改良貪欲解法において，β は，実行可能アーク集合を求めるために付加するアーク群の要素数である．なるべく少ない数で実行可能なアーク集合を求めるための初めの repeat–until 部分以外は，$SCND$ に対する改良貪欲解法 (片山直登 (2015)) と同一である．

貪欲解法終了後に，貪欲解法によって選択されたアーク集合に対して，アークフローによる定式化 $UCND(\bar{A})$ に計算時間の上限を設定して最適化ソルバーの分枝限定法で解き，実行可能解と上界値 UB_{MBB} を求め，貪欲解法による上界値 UB_{MB} と小さい方を UB_{MB} とする．アルゴリズムの出力は，実行可能なアーク集合 $\bar{A}$ と上界値 UB_{MG} である．

タブサーチ法のアルゴリズムをアルゴリズム 4 に示す．γ は容量超過に対するフロー費用のペナルティ係数，ζ はペナルティ係数の増加量，η はタブサーチ法における禁止回数である．タブサーチ法を用いて，特定のアークを削除したときの目的関数の近似値を求める．始めに，特定のアークを削除し，アーク容量を緩和したネットワーク上で，各品種の最小費用パスを求め，これらのパスに各フローを流す．続いて，アーク容量を超過しているアーク上を流れている品種の需要に対して，超過分に比例したペナルティーを当該アークのフロー費用に加えて増加させたネットワーク上で最小費用パスを求め，現在のパスフローを最小費用パス上に流し換える．パスを変更した品種にはタブ値 η を与え，この回数だけ品種のパスの変更を禁止する．この操作をすべてのアークフローがアーク容量を満たすま

で繰り返す．なお，ペナルティー係数γは，繰り返しとともにζにより増加させる．なお，このアルゴリズムは，必ずしも実行可能な解を算出できる訳ではない．アルゴリズムの出力は，特定のアークを削除したときの目的関数の近似値である．

5 数値実験

$SCND$ で用いられているベンチマーク問題である C 問題と 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う．使用した C 問題は 31 問，R 問題は R10 から R18 までの 81 問である．これらは，フロー変数が 0–1 である以外は $SCND$ で用いられるベンチマーク問題と同じインスタンスである．

限定分枝限定法 (RBB)，貪欲法を最適化ソルバーで解く方法 (MIP)，および貪欲法をタブサーチ法で解く方法 (TAB) による結果を示す．比較する研究は，C 問題に対する Yaghini and Kazemzadeh (2012) のシミュレーテッドアニーリング法 (SA) と Hewitt et al. (2013) の IP 探索法 (IPS)，分枝価格ガイドつき探索法 (BPG)，および片山直登 (2013) のパス再リンク法 (PRL) である．なお，記載した上界値と計算時間は，各論文に記載されているものである．また，上界値の誤差を算出するために，アークフローによる定式を最適化ソルバー Gurobi(最大 30 時間) により解き，下界値または最適値 (LB/OPT) を求めている．なお，同時に Gurobi により上界値 (GUR) も求めている．

本研究の解法で設定した主な条件およびパラメータなどは以下の通りである．

- 使用 OS および言語：UBUNTU 15.1，GNU C++ compiler
- 最適化ソルバー：Gurobi 6.51
- スケーリングパラメータ λ : 0.250
- 容量スケーリング法の最小繰り返し回数 ITE_{min} : 2
- 容量スケーリング法の最大繰り返し回数 ITE_{max} : 100
- 限定分枝限定法における閾値 ϵ : 0.001
- 計算時間の上限 T : 60 秒
- 容量スケーリング法の終了基準 α : 10
- アーク群のアーク数 β : 10
- タブサーチ法における容量オーバーペナルティγ : 10
- タブサーチ法における容量オーバーペナルティの増加量 ζ : 1.1
- タブサーチ法における禁止回数 η : min(品種数/5, 7) × (1 + 0〜1 の一様乱数)

なお，各論文内で使用しているコンピュータなどは以下の通りである．

Algorithm 3: Modified Greedy Algorithm ($\hat{A}$)

Set β;
Solve a linear relaxation problem $UCNDAL(\hat{A})$ of $UCNDA(\hat{A})$;
Get the arc solution $\tilde{y}$ of $UCNDAL(\hat{A})$ and sort the solutions in descending order of the value of $\tilde{y}$;
$\bar{A} \leftarrow \{(i,j)|\tilde{y}_{ij} = 1, \forall(i,j)\}$;
Set the group A_i, which is i_{th} group of arcs with size β in descending order of the value of $\tilde{y}$ in $\hat{A}\backslash\bar{A}$;
$i \leftarrow 1$;
repeat
    $\bar{A} \leftarrow \bar{A} \cap A_i$;
    Solve $MCFI(\bar{A})$ by an MIP solver or Tabu Search;
    $i \leftarrow i+1$;
until $UCNDA(\bar{A})$ *is feasible*;
Get $\phi(\bar{A})$;
$L \leftarrow \emptyset$;
for $(i,j) \in \bar{A}$ **do**
    Solve $MCFI(\bar{A}\backslash\{(i,j)\})$ by an MIP solver or Tabu Search;
    if $MCFI(\bar{A}\backslash\{(i,j)\})$ *is feasible* **then**
        Get $\phi(\bar{A}\backslash\{(i,j)\})$;
        $\psi_{ij} \leftarrow \phi(\bar{A}) - \phi(\bar{A}\backslash\{(i,j)\})$;
        if $\psi_{ij} > 0$ **then** $L \leftarrow L \cup \{(i,j)\}$;
    end
end
repeat
    $\pi \leftarrow \max_{(i,j)\in L} \psi_{ij}$;
    $(i^*, j^*) \leftarrow \arg\max_{(i,j)\in L} \psi_{ij}$;
    $L \leftarrow L\backslash\{(i^*, j^*)\}$;
    Solve $MCFI(\bar{A}\backslash\{(i^*, j^*)\})$ by an MIP solver or Tabu Search;
    if $MCFI(\bar{A}\backslash\{(i^*, j^*)\})$ *is feasible* **then**
        Get $\phi(\bar{A}\backslash\{(i^*, j^*)\})$;
        $\psi_{i^*j^*} \leftarrow \phi(\bar{A}) - \phi(\bar{A}\backslash\{(i^*, j^*)\})$;
        if $\psi_{i^*j^*} \geq \max_{(i,j)\in L} \psi_{ij}$ **then**
            $\bar{A} \leftarrow \bar{A}\backslash\{(i^*, j^*)\}$;
        else if $\psi_{i^*j^*} > 0$ **then**
            $L \leftarrow L \cup \{(i^*, j^*)\}$;
        end
    end
until $|L| = 0$;
Get $\phi(\bar{A})$;
$UB_{MG} \leftarrow \phi(\bar{A})$;
Solve $UCNDA(\bar{A})$ by an MIP solver within time T;
Get UB_{MBB} and $\hat{A}$ with arcs such as solution $y = 1$ of $UCNDA(\bar{A})$;
$UB_{MBB} \leftarrow \min(UB_{MG}, UB_{MBB})$;
if $UB_{MG} = UB_{MBB}$ **then** $\bar{A} \leftarrow \hat{A}$;
Return $\bar{A}$, UB_{MG};

Algorithm 4: Tabu Search($\bar{A}, (i, j)$)

Set γ,ζ,η,Ite_{max};
$\bar{A} \leftarrow \bar{A} \backslash \{(i, j)\}$;
for $k \in K$ **do**
    Solve the shortest path problem for commodity k on the network accocisted with $\bar{A}$ and arc flow cost c;
    Add the demand d^k on the shortest path;
end
$Tabu \leftarrow 1$;
$l \leftarrow 0$;
$\phi(\bar{A} \backslash \{(i, j)\}) \leftarrow \infty$;
$\bar{x} \leftarrow current\ arc\ flows$;
repeat
    for $(i, j) \in \bar{A}$ **do**
        $\bar{c} \leftarrow c$;
        if $\bar{x}_{ij} > C_{ij}$ **then**
            for $k \in K$ **do**
                if $Tabu[k] \leq 0$ **then**
                    $\bar{c}^k_{ij} \leftarrow c^k_{ij} + \gamma \times \max\{0, \bar{x}_{ij} - C_{ij}\}$;
                    if $\bar{x}^k_{ij} > 0$ **then**
                        Delete d^k on the current path;
                        Solve the shortest path problem for k accocisted with $\bar{c}$;
                        Add the d^k on the shortest path;
                        $Tabu[k] \leftarrow \eta$;
                        $\bar{x} \leftarrow current\ arc\ flows$;
                    end
                end
            end
        end
    end
    if $\bar{x} \leq C$ **then**
        Get $\phi(\bar{A} \backslash \{(i, j)\})$;
        break;
    end
    $l \leftarrow l + 1$;
    $\gamma \leftarrow \gamma \times \zeta$;
    $Tabu \leftarrow Tabu - 1$;
until $l > Ite_{max}$;
Return $\phi(\bar{A} \backslash \{(i, j)\})$;

- SA：論文に不記載
- IPS，BPG：CPU INTEL Xeon 2.26GHz 8Core，RAM 24GByte
- PRL：CPU INTEL i7 2600 3.4GHz 4Core，RAM 16GByte
- GUR,PBB,MIP,TUB：CPU INTEL i7 3770K 3.5GHz 4Core，RAM 32GByte

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

GUR	SA	IP	BPG	RRL	RBB	MIP	TAB
1.03	14.51*	2.31	1.34	0.99	3.29	2.11	3.80

*:15 out of 31 instances

C 問題に対する解法別の上界値の平均誤差を表 1 に示す．なお，SA では 31 問中の比較的小さな 15 問の解のみが掲載されている．最適化ソルバーである GUR の平均誤差は 1.03%と小さい．また，SA の平均誤差では 14.51%と大きく，IP では 2.31%，BPG では 1.34%，PRL では 0.99%である．一方，限定分枝限定法のみの解法である RBB では 3.29%，MIP では 2.11%であるが，TAB では 3.80%と大きい．SA よりも優れているが，提案した解法は貪欲法であることから，SA 以外の従来法より解の精度が悪化している．

表 2 に C 問題の個別の問題ごとの最適値/下界値，CPLEX および各解法により得られた上界値，表 3 に平均誤差を示す．”O”は最適値，”L”は下界値であることを表している．また，ボールド体は最適値を表している．31 問の内で最適解を求めることができたのは，GUR と PRL では 14 問，SA では 0 問，IP では 9 問，BPG では 8 問である．一方，RBB では 1 問，MIP では 3 問題，TAB では 1 問と少ない．

C 問題に対する平均計算時間を表 3 に示す．SA では計算時間が未掲載であり，IP と BPG では 1800 秒の上限時間を設けている．GUR は最大 30 時間と設定していることもあり，52356.7 秒を要している．また，PRL では 14503.2 秒と大きな計算時間を要している．一方，提案した解法では，MIP では 3051.8 秒と比較的計算時間を要しているが，RBB では 146.9 秒，TAB では 87.8 秒と計算時間の大幅な短縮化に成功している．表 5 に，C 問題の個別の問題ごとの計算時間を示す．

R 問題に対する解法別の上界値の平均誤差を表 6 に示す．R 問題に対しては，従来の研究の結果はないため，GUR と提案した 3 つの解法の結果のみを示している．最適化ソルバーである GUR の平均誤差は 0.35%と小さく，RBB では 4.66%，MIP では 2.51%，TAB では 4.50%である．なお，r18/8/05 では $\alpha = 20$ と設定した．

表 7 から表 9 に，R 問題の個別の問題ごとの最適値/下界値，Gurobi および各解法により得られた上界値を示す．R 問題に対する平均計算時間を表 10 に示す．GUR では 25929.9 秒を要し，RBB では 64.1 秒，MIP では 497.8 秒，TAB では 41.5 秒と計算時間の大幅な短縮化に成功している．

表 2: Results for C-Category Problems

Problem	LB/OPT	GUB	SA	IP	BRG	PRL	RBB	MIP	TAB
20/230/040/V/L	423933 ^{O}	**423933**		**423933**	**423933**	**423933**	425195	**424633**	427814
20/230/040/V/T	398870 ^{O}	**398870**		**398870**	**398870**	**398870**	398870	399729	400933
20/230/040/F/T	668699 ^{O}	**668699**	670928	**668699**	**668699**	**668699**	**668699**	675433	**668699**
20/230/200/V/L	94644 ^{O}	**94644**	118785	95695	94843	**94644**	95697	94710	98725
20/230/200/F/L	137410 ^{L}	138084		141700	138476	138084	141821	138779	145339
20/230/200/V/T	98339 ^{L}	98610	112792	100884	98947	98610	100406	98958	100728
20/230/200/F/T	135051 ^{L}	137886	167006	141734	139889	137594	144290	138928	146213
20/300/040/V/L	430253 ^{O}	**430253**	433929	**430253**	430314	**430253**	**430253**	**430253**	431583
20/300/040/F/L	597059 ^{O}	**597059**		**597059**	**597059**	**597059**	609383	600645	615270
20/300/040/V/T	501766 ^{O}	**501766**	511384	**501766**	501889	**501766**	501784	**501766**	501784
20/300/040/F/T	643395 ^{O}	**643395**	656412	**643395**	**643395**	**643395**	645933	645440	644816
20/300/200/V/L	74766 ^{L}	76029	91702	76946	76333	75802	77743	76472	77680
20/300/200/F/L	114144 ^{L}	117600	151513	119590	118204	117617	126642	118058	123947
20/300/200/V/T	75881 ^{L}	76469	88550	77055	76986	76357	77352	77257	78043
20/300/200/F/T	106595 ^{L}	109806		110516	109776	109385	120403	110406	120076
30/520/100/V/L	54387 ^{O}	**54387**	57422	54427	**54387**	**54387**	54770	55545	55474
30/520/100/F/L	94614 ^{L}	95877		97199	96277	95877	99995	97874	101353
30/520/100/V/T	53812 ^{O}	**53812**		**53812**	**53812**	**53812**	55022	54905	55080
30/520/100/F/T	98860 ^{L}	99195		100391	99355	99204	104859	102427	105476
30/520/400/V/L	112688 ^{L}	114454	134587	116465	114867	114295	114861	115351	115816
30/520/400/F/L	147819 ^{L}	150875		156433	152837	150965	155361	152814	154430
30/520/400/V/T	115182 ^{L}	116514		118193	116730	116306	118034	116552	117735
30/520/400/F/T	150834 ^{L}	155961		161513	155982	155641	160462	155702	160872
30/700/100/V/L	47883 ^{O}	**47883**	51505	**47883**	**47883**	**47883**	48103	48236	48494
30/700/100/F/L	60384 ^{O}	**60384**	68179	61254	60384	**60384**	61981	61872	62232
30/700/100/V/T	47670 ^{O}	**47670**		47736	47686	**47670**	48761	50758	48434
30/700/100/F/T	56686 ^{O}	**56686**		56931	56809	**56686**	58204	59044	58021
30/700/400/V/L	97421 ^{L}	98750		100368	98850	98535	99707	99051	100611
30/700/400/F/L	131450 ^{L}	136952		144077	138376	137017	139808	137234	141304
30/700/400/V/T	94803 ^{L}	96526		98040	97352	96366	97526	97142	98288
30/700/400/F/T	128225 ^{L}	132526	168070	135512	133759	132112	135702	132859	137226

表 3: Gaps for C-Category Problems (%)

Problem	GUR	SA	IP	BRG	PRL	RBB	MIP	TAB
20/230/040/V/L	0.00		0.00	0.00	0.00	0.30	0.17	0.92
20/230/040/V/T	0.00		0.00	0.00	0.00	0.00	0.22	0.52
20/230/040/F/T	0.00	0.33	0.00	0.00	0.00	0.00	1.01	0.00
20/230/200/V/L	0.01	25.52	1.12	0.22	0.01	1.12	0.08	4.32
20/230/200/F/L	0.49		3.12	0.78	0.49	3.21	1.00	5.77
20/230/200/V/T	0.28	14.70	2.59	0.62	0.34	2.10	0.63	2.43
20/230/200/F/T	2.10	23.66	4.95	3.58	1.90	6.84	2.87	8.27
20/300/040/V/L	0.00	0.85	0.00	0.01	0.00	0.00	0.00	0.31
20/300/040/F/L	0.00		0.00	0.00	0.00	2.06	0.60	3.05
20/300/040/V/T	0.00	1.92	0.00	0.02	0.00	0.00	0.00	0.00
20/300/040/F/T	0.00	2.02	0.00	0.00	0.00	0.39	0.32	0.22
20/300/200/V/L	1.69	22.65	2.92	2.10	1.41	3.98	2.28	3.90
20/300/200/F/L	3.03	32.74	4.77	3.56	3.22	10.95	3.43	8.59
20/300/200/V/T	0.77	16.70	1.55	1.46	0.79	1.94	1.81	2.85
20/300/200/F/T	3.01		3.68	2.98	2.99	12.95	3.58	12.65
30/520/100/V/L	0.00	5.58	0.07	0.00	0.00	0.70	2.13	2.00
30/520/100/F/L	1.33		2.73	1.76	1.61	5.69	3.45	7.12
30/520/100/V/T	0.00		0.00	0.00	0.00	2.25	2.03	2.36
30/520/100/F/T	0.34		1.55	0.50	0.35	6.07	3.61	6.69
30/520/400/V/L	1.57	19.43	3.35	1.93	1.52	1.93	2.36	2.78
30/520/400/F/L	2.07		5.83	3.39	2.09	5.10	3.38	4.47
30/520/400/V/T	1.16		2.61	1.34	0.99	2.48	1.19	2.22
30/520/400/F/T	3.40		7.08	3.41	2.93	6.38	3.23	6.65
30/700/100/V/L	0.00	7.56	0.00	0.00	0.00	0.46	0.74	1.28
30/700/100/F/L	0.01	12.92	1.45	0.01	0.01	2.65	2.47	3.07
30/700/100/V/T	0.00		0.14	0.03	0.00	2.29	6.48	1.60
30/700/100/F/T	0.00		0.44	0.22	0.00	2.68	4.16	2.36
30/700/400/V/L	1.36		3.03	1.47	1.15	2.35	1.67	3.27
30/700/400/F/L	4.19		9.61	5.27	4.24	6.36	4.40	7.50
30/700/400/V/T	1.82		3.41	2.69	1.65	2.87	2.47	3.68
30/700/400/F/T	3.35	31.07	5.68	4.32	3.03	5.83	3.61	7.02

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

GUR	IP	BPG	RRL	RBB	MIP	TAB
52356.7	1800.0	1800.0	14503.2	146.9	3051.8	87.8

表 5: Computation Time for C-Category Problems (seconds)

Problem	GUR	PRL	RBB	MIP	TAB
20/230/040/V/L	0.6	1.5	0.3	1.0	0.4
20/230/040/V/T	0.8	1.7	0.5	3.2	1.2
20/230/040/F/T	0.8	2.0	0.5	2.2	1.0
20/230/200/V/L	75511.6	12791.0	113.0	179.3	75.2
20/230/200/F/L	108000.0	18849.8	114.8	214.6	74.2
20/230/200/V/T	108000.0	13606.2	106.5	141.8	68.6
20/230/200/F/T	108000.0	16842.4	118.2	216.4	75.5
20/300/040/V/L	0.7	1.3	0.2	1.0	0.3
20/300/040/F/L	1.4	4.6	0.9	2.1	1.4
20/300/040/V/T	1.0	2.5	0.6	2.1	1.4
20/300/040/F/T	1.0	2.9	0.4	2.0	0.7
20/300/200/V/L	108000.0	16462.9	111.9	169.1	77.1
20/300/200/F/L	108000.0	20637.6	128.1	321.7	76.5
20/300/200/V/T	108000.0	19963.0	115.0	1979.9	75.1
20/300/200/F/T	108000.0	15908.7	113.3	222.7	77.3
30/520/100/V/L	246.5	1916.4	8.3	14.1	8.7
30/520/100/F/L	108000.0	18193.7	105.5	110.5	67.3
30/520/100/V/T	32.3	118.6	6.5	50.1	10.9
30/520/100/F/T	108000.0	8801.5	103.6	41.4	66.3
30/520/400/V/L	108000.0	20006.5	481.9	12315.4	372.6
30/520/400/F/L	108000.0	19763.4	247.1	9398.2	163.0
30/520/400/V/T	108000.0	17956.9	285.0	14192.4	189.2
30/520/400/F/T	108000.0	18590.3	1158.2	12442.8	220.9
30/700/100/V/L	38.4	324.1	3.5	14.5	4.4
30/700/100/F/L	18333.1	6466.8	9.2	19.6	18.8
30/700/100/V/T	161.4	1353.2	6.9	89.3	11.7
30/700/100/F/T	2413.3	5744.8	12.2	57.5	26.3
30/700/400/V/L	108000.0	17846.4	136.4	8219.3	114.9
30/700/400/F/L	108000.0	29991.3	692.3	10061.1	372.1
30/700/400/V/T	108000.0	27030.8	165.0	12674.5	176.4
30/700/400/F/T	108000.0	24750.6	209.5	11445.8	292.1

表 6: Average Gap for R-Category Problems (%)

Gurobi	RBB	MIP	TAB
0.35	4.66	2.51	4.50

表 7: Results for R-Category Problems

Problem			LB/OPT	GUR	RBB	MIP	TAB
		01	201556 ^{L}	201572	202149	202149	202149
	1	05	351776 ^{O}	**351776**	360092	364962	361169
		10	498453 ^{O}	**498453**	518863	542052	513908
		01	260070 ^{O}	**260070**	260568	263140	260568
r10	2	05	449550 ^{O}	**449550**	461793	453859	467213
		10	647022 ^{O}	**647022**	666972	680871	686064
		01	1567316 ^{O}	**1567316**	**1567316**	1567364	1567316
	8	05	1822071 ^{O}	**1822071**	**1822071**	1823365	1822071
		10	2111012 ^{L}	2111143	2111143	2125436	2111143
		01	733745 ^{O}	**733745**	737094	742357	739725
	1	05	1289628 ^{L}	1289747	1308565	1306853	1334857
		10	1873569 ^{L}	1873739	1963039	1930698	1986339
		01	936095 ^{O}	**936095**	939419	945155	947301
r11	2	05	1700425 ^{L}	1700582	1745625	1780587	1759138
		10	2513658 ^{L}	2513905	2599861	2628661	2608088
		01	3502261 ^{L}	3502498	3503626	3503626	3503626
	8	05	4333820.4 ^{L}	4333940	4333940	4346397	4333940
		10	5107395 ^{O}	**5107395**	5116080	5107395	5116080
		01	1696049 ^{L}	1698932	1710660	1708250	1713489
	1	05	3447931 ^{L}	3486183	3560783	3613542	3612353
		10	5296273 ^{L}	5399202	5621578	5547438	5578265
		01	2459864 ^{L}	2459902	2460285	2462435	2497625
r12	2	05	4834613.2 ^{L}	4835012	4846977	4852520	4848252
		10	7240234.6 ^{L}	7240540	7283022	7257635	7268157
		01	8609022.3 ^{L}	8609660	8609733	8609842	8611193
	8	05	10487442.9 ^{L}	10487681	10487848	10487681	10487681
		10	12183155 ^{L}	12184305	12184305	12184008	12184305

表 8: Results for R-Category Problems

Problem			LB/OPT	GUR	RBB	MIP	TAB
		01	143036 ^{O}	**143036**	**143036**	143169	143900
	1	05	264356 ^{O}	**264356**	266415	267498	271811
		10	368938 ^{O}	**368938**	372228	372970	379099
		01	154548 ^{O}	**154548**	155955	155509	156051
r13	2	05	283471 ^{O}	**283471**	290760	298288	292277
		10	413528 ^{O}	**413528**	442869	431968	434481
		01	328229 ^{O}	**328229**	332528	328229	332528
	8	05	629793 ^{O}	**629793**	690008	663018	690008
		10	962571 ^{O}	**962571**	1072241	1052801	1072241
		01	405541 ^{O}	**405541**	406204	406285	405987
	1	05	754089 ^{O}	**754089**	762651	760063	761345
		10	1068814 ^{L}	1068882	1108695	1091210	1133686
		01	443948 ^{O}	**443948**	446840	447689	447606
r14	2	05	855516 ^{O}	**855516**	893220	867907	891433
		10	1231281 ^{L}	1231353	1342380	1240950	1345052
		01	888544 ^{O}	**888544**	952979	919112	908769
	8	05	1938401 ^{O}	**1938401**	2051170	2014619	2054587
		10	2982752 ^{L}	2982869	3072625	3135315	3080702
		01	1010451 ^{L}	1010549	1018707	1013139	1018913
	1	05	1960158 ^{L}	1977479	2039154	1986246	2074275
		10	2825323 ^{L}	2920202	3150290	2923646	3245981
		01	1170553 ^{L}	1170668	1175923	1185707	1174697
r15	2	05	2466320 ^{L}	2547091	2708055	2587471	2748907
		10	3788564.2 ^{L}	3959492	4499021	4179271	4624103
		01	3237673 ^{O}	**3237673**	3241292	3294140	3255409
	8	05	6382248.2 ^{L}	6382293	6401631	6383465	6436143
		10	9380744.6 ^{L}	9381609	9430742	9381609	9429831

表 9: Results for R-Category Problems

Problem			LB/OPT	GUR	RBB	MIP	TAB
r16	1	01	136161 ^{O}	**136161**	**136161**	136392	136908
		05	239570 ^{O}	**239570**	240926	242369	254630
		10	325671 ^{O}	**325671**	327470	326458	340521
	2	01	139552 ^{O}	**139552**	140157	140250	140157
		05	243281 ^{O}	**243281**	251210	245903	246239
		10	338266 ^{O}	**338266**	348242	348784	351987
	8	01	203353 ^{O}	**203353**	207082	208533	207082
		05	409683 ^{O}	**409683**	484782	436781	469552
		10	601110 ^{O}	**601110**	758242	618950	700719
r17	1	01	355129 ^{O}	**355129**	355340	35512	359780
		05	648505 ^{O}	**648505**	680419	662638	686271
		10	911499 ^{L}	911557	939982	922855	946614
	2	01	372505 ^{O}	**372505**	376934	375660	377219
		05	711258 ^{L}	711302	731635	721955	733433
		10	1017062 ^{L}	1029546	1102078	1046998	1102765
	8	01	569987 ^{O}	**569987**	591154	598009	631475
		05	1267160 ^{L}	1275470	1525210	1413887	1446108
		10	2028601 ^{L}	2045350	2503906	2282719	2407552
r18	1	01	832248 ^{L}	832330	842653	839076	833992
		05	1537916 ^{L}	1538055	1628182	1547901	1637427
		10	2170620 ^{L}	2181969	2383272	2204465	2319141
	2	01	925144 ^{L}	929423	953126	938364	956689
		05	1796727 ^{L}	1849499	2030319	1886925	1981318
		10	2643077 ^{L}	2751422	3164144	2778824	3187602
	8	01	1700898 ^{L}	1710389	1756552	1803261	1757601
		05	4218889.3 ^{L}	4272431	4935483	4498043	4606018
		10	6808830 ^{L}	6858396	8782424	6971415	7522391

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

Gurobi	RBB	MIP	TAB
25929.9	64.1	497.8	41.5

6　おわりに

本研究では，非分割フローを考慮した容量制約をもつ設計問題に対して，容量スケーリング法，限定分枝限定法，および貪欲解法を組合せた近似解法を提案した．

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．その結果，解の精度は従来の最良の解法に劣るが，非常に短い計算時間で実行可能解を算出することができた．今後，適切なパラメータチューニング，タブサーチ法の改良，およびさらなる全体的な計算時間の短縮化が必要である．

参考文献

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.

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.

Gendron, B., T. G. Crainic, A. Frangioni. 1997. Multicommodity capacitated network design. Tech. Rep. CIRRELT-98-14, Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation, Université de Montréal.

Hewitt, M., G. Nemhauser, M. W. P. Savelsbergh. 2013. Branch-and-price guided search for integer programs with an application to the multicommodity fixed charge network flow problem. *INFORMS Journal on Computing* **25** 302–316.

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

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

Minoux, M. 1976. Multiflots de coût minimal avec fonctions de coût concaves. *Annales des Télécommunications* **31** 77–92.

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

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.R.A Kazemzadeh. 2012. A simulated annealing algorithm for unsplittable capacitated network design. *International Journall of Industrial Engineering & Production Research* **23** 91–100.

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

片山直登. 2008. ネットワーク設計問題. 朝倉書店.

片山直登. 2010. 多品種を考慮したロジスティクスネットワーク設計問題の数理的解法に関する研究. 博士論文, 流通経済大学.

片山直登. 2013. 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. 流通経済大学流通情報学部紀要 **20** 1–19.

片山直登. 2015. 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** 1–24.