

# 列生成法と行生成法を用いた容量制約のない ネットワークデザインの問題の近似解法

片山直登

## 1 はじめに

容量制約のないネットワークデザイン問題 (Uncapacitated Network Design Problem: *UND*) は, アーク上の容量の制約を考慮しないネットワークにおいて, 適切なネットワークの形状と多品種のフローを求める問題であり, 通信ネットワーク設計, 交通ネットワーク設計や輸送・配送ネットワーク設計などに様々な応用分野が存する問題である。

容量制約のないネットワークデザイン問題に対しては, 数多くの最適解法や近似解法が示されている。最適解法としては, Los-Lardinois [7] による分枝限定法, Magnanti-Mireault-Wong [8] によるベンダーズ分解原理, Holmberg-Hellstrand [3] によるLagrange緩和と分枝限定法を組み合わせた解法などがある。貪欲的な近似解法としては, バックワード法が数多く提案されている。Billheimer-Gray [2] はアーク削除時の目的関数値の変化量を事前に近似的計算する方法, Los-Lardinois [7] はアーク削除時の目的関数値の変化量を厳密に求め, 単位アーク費用当たりの変化量による方法を示している。また, Minoux [10] は初期のアーク削除時の目的関数値の近似的な変化量をリストに保持し, 削除対象時に再計算した変化量がリスト中で最大の場合のみ削除する方法, 片山 [6] は目的関数値の変化量を厳密に計算したMinouxの改良法を示している。緩和問題の解法と緩和解を用いた近似解法としては, Balakrishnan-Magnanti-Wong [1] による双対上昇法と双対ヒューリスティクス, 片山ら [5] によるLagrange緩和法とLagrangeヒューリスティクスがある。また, 多くのサーベイ [9] [11] [4] が提供されている。

近年, ネットワークデザイン問題に対して, パスフローを用いた定式化を行い, 列生成法を適用する研究が行われている。容量制約をもつネットワークデザイン問題に対して, Crainic-Gendreau-Farvolden [3] は列生成法と基底変換に対するタブーサーチ法

を組み合わせた近似解法を示している。陳-片山-久保 [8] は行生成法と容量スケールリング法を組合せた解法を提案している。また、久保 [18], 小松 [17] はサービスネットワーク設計問題に対する容量スケールリング法を提案している。

ノードは交換機・コンピュータ, 交差点・交通需要の発生吸収点や配送センター・消費地などを表し, アークは通信回線, 道路や輸送経路などのノード間をつなぐものを表している。また, フローは, 呼, データ, 自動車や荷物などのノード間を移動するものを表している。ここでは, 一般的な表現とは異なり, 始点・終点の異なるものを異なる品種と定義する。本研究では, 多品種を対象とする。そのため, 多品種とは, 始点と終点の組合せが異なる複数種類の運ぶべきモノが存在することを意味する。

ノード集合, デザイン費用とフロー費用をもつ容量制約と向きをもたないアーク集合, 需要をもつ品種集合が与えられているものとする。このとき, 容量制約のないネットワークデザイン問題は, 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を選択し, 各品種のフローを求める問題である。本研究では, このような容量制約のないネットワークデザイン問題に対して, 新たな強い妥当不等式を提案し, 列生成法と行生成法を組み合わせた線形緩和問題の解法を提案し, さらに緩和解をもとにした近似解法を提案する。また, 数値実験により, 提案する解法が従来解法よりも精度が高い解を算出できることを示す。

## 2 UNDの定式化

ノード集合を $N$ , 向きをもたないアーク集合を $A$ , 品種集合を $K$ と表す。ノード $i, j$ 間のアーク $(i, j)$ を設置するか否かを表す0-1のデザイン変数を $y_{ij}$ とし, アーク $(i, j)$ 上を $i \rightarrow j$ 方向に流れる品種 $k$ 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 $(i, j)$ を選択するときが発生するデザイン費用を $f_{ij}$ , 品種 $k$ の1単位の需要がアーク上を $i \rightarrow j$ 方向に通るときが発生するフロー費用を $c_{ij}^k$ とする。また, 品種 $k$ の始点を $O^k$ , 終点を $D^k$ , 需要を1とする。このとき, アークフロー変数を用いたUNDの定式化は, 次のように表される。

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

$$\text{条件 } \sum_{i \in N_n} x_{in}^k - \sum_{j \in N_n} x_{nj}^k = \begin{cases} -1 & \text{if } n = O^k \\ 1 & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N, k \in K \quad (2)$$

$$x_{ij}^k + x_{ji}^k \leq y_{ij} \quad \forall k \in K, (i, j) \in A \quad (3)$$

$$x_{ij}^k \geq 0, x_{ji}^k \geq 0 \quad \forall k \in K, (i, j) \in A \quad (4)$$

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

(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。(2)式はフロー保存式であり，品種の需要が始点から終点に移動することを保証する。(3)式は， $x_{ij}^k$ の上限が1であることを考慮した強制制約式である。これは，アーク  $(i, j)$  が存在するときのみ，1まで品種  $k$  のフローの存在を許す制約式であり，アーク  $(i, j)$  上の品種  $k$  のフロー量とデザイン変数の関係を表す。(4)式はアークフロー変数の非負制約であり，(5)式はデザイン変数の0-1条件である。

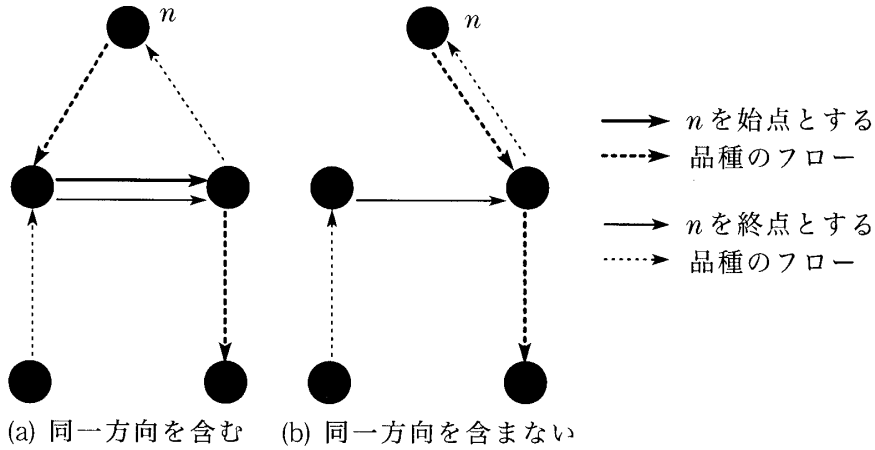


図1：同一ノードを始点，終点にする品質のフロー

問題の定式化に有効な妥当不等式を追加すると実行可能領域が狭くなり，緩和問題などを使って問題を解く場合に，妥当不等式が有効に機能する。そこで，UNDに対する有効な妥当不等式を示す。

ここでは，終点（または始点）を同じにする品種に対して，フロー費用が等しいか比例などの場合を想定する。この場合，終点（または始点）を同じにする品種のフローが同一のアーク上で互いに逆流しない最適解が存在する。このことから，Balakrishnan-Magnanti-Wong [1] は，次の妥当な強制制約式を示している。

$$x_{ij}^k + x_{ji}^h \leq y_{ij} \quad \forall k \in S_n, h \in S_n, n \in N, (i, j) \in A \quad (6)$$

$$x_{ij}^k + x_{ji}^h \leq y_{ij} \quad \forall k \in T_n, h \in T_n, n \in N, (i, j) \in A \quad (7)$$

ここで， $S_n$ はノード  $n$  を始点とする品種の集合， $T_n$ はノード  $n$  を終点とする品種の集合である。 $k = h$  のときには，(6)，(7)式の両方に(3)式を含むことになって冗長となるが，表現を簡潔にするためにこのように記述する。

図1(a)に示すように，同一ノードを始点とする品種と終点とする品種のフローが，同一のアーク上で同じ向きに流れるフローが存在するものとする。このとき，図1(b)に示すように，同じ向きに流れないような実行可能で，かつフロー費用が安いフローが存在する。このため，最適解の一つでは，同一ノードを始点とする品種と終点とする品種のフローが同一のアーク上で同じ向きに流れないことになる。したがって，次式も妥当な

強制制約式となる。

$$x_{ij}^k + x_{ij}^h \leq y_{ij} \quad \forall k \in S_n, h \in T_n, n \in N, (i, j) \in A \quad (8)$$

$$x_{ji}^k + x_{ji}^h \leq y_{ij} \quad \forall k \in S_n, h \in T_n, n \in N, (i, j) \in A \quad (9)$$

一方、パスに関するフローであるパスフロー変数を用いて、UNDの定式化を行うことができる。品種  $k$  の取り得るパス集合を  $P^k$ 、品種  $k$  のパス  $p$  のフロー量を表すパスフロー変数を  $z_p^k$  とする。また、パス  $p$  がアーク  $(i, j)$  を  $i \rightarrow j$  向きに通るとき 1、そうでないとき 0 を表す定数を  $\delta_{ij}^p$  とする。このとき、(6)~(9)式に対応する強制制約式を考慮したパスフロー変数を用いたUNDは、次のような問題PFとして定式化できる。

$$\text{最小化} \quad \sum_{(i,j) \in A} \sum_{k \in K} \sum_{p \in P^k} (\delta_{ij}^p c_{ij}^k + \delta_{ji}^p c_{ji}^k) z_p^k + \sum_{(i,j) \in A} f_{ij} y_{ij} \quad (10)$$

$$\text{条件} \quad \sum_{p \in P^k} z_p^k = 1 \quad \forall k \in K \quad (11)$$

$$\sum_{p \in P^k} \delta_{ij}^p z_p^k + \sum_{p \in P^h} \delta_{ji}^p z_p^h \leq y_{ij} \quad \forall k \in S_n, h \in S_n, n \in N, (i, j) \in A \quad (12)$$

$$\sum_{p \in P^k} \delta_{ij}^p z_p^k + \sum_{p \in P^h} \delta_{ji}^p z_p^h \leq y_{ij} \quad \forall k \in T_n, h \in T_n, n \in N, (i, j) \in A \quad (13)$$

$$\sum_{p \in P^k} \delta_{ij}^p z_p^k + \sum_{p \in P^h} \delta_{ij}^p z_p^h \leq y_{ij} \quad \forall k \in S_n, h \in T_n, n \in N, (i, j) \in A \quad (14)$$

$$\sum_{p \in P^k} \delta_{ji}^p z_p^k + \sum_{p \in P^h} \delta_{ji}^p z_p^h \leq y_{ij} \quad \forall k \in S_n, h \in T_n, n \in N, (i, j) \in A \quad (15)$$

$$z_p^k \geq 0 \quad \forall k \in K, p \in P^k \quad (16)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (17)$$

(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。ここで、 $\sum_{p \in P^k} \delta_{ij}^p z_p^k$  は  $x_{ij}^k$ 、 $\sum_{p \in P^k} \delta_{ji}^p z_p^k$  は  $x_{ji}^k$  と一致する。(11)式は、品種  $k$  のパスフロー量の合計が品種  $k$  の需要量に一致することを表す。(12)~(15)式は、(6)~(9)式に対応する強制制約式である。(16)式は、パスフローの変数の非負制約である。

(6)~(9)式の強制制約式を考慮したアークフローによる定式化では、 $O(|K||A|)$  個のアークフロー変数、 $O(|A|)$  個のデザイン変数と  $O(|A||N|^3)$  個の制約式をもつ組合せ最適化問題となる。また、パスフローによる定式化では、指数個のパスフロー変数、 $O(|A|)$  個のデザイン変数と  $O(|A||N|^3)$  個の制約式をもつ組合せ最適化問題となる。いずれにしても、一般的な最適解法を用いて、実用規模の問題を最適に解くことは困難である。

### 3 線形緩和問題と被約費用

$UND$ は0-1変数を含む組合せ最適化問題であるため、0-1条件を0から1の連続条件に線形緩和した問題を考え、この緩和問題を解くことにする。一般的には、線形緩和問題の最適解は小数値を含むため、 $UND$ の実行可能解とはならない。また、緩和問題の最適値は $UND$ の最適値ではなく下界値となる。

はじめに、パスフローにより定式化した $PF$ に対する線形緩和問題 $PFL$ を考える。 $PFL$ では、(17)式の代わりに、次式が制約式となる。

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

(11)式に対する双対変数を  $v$ 、(12)、(13)、(14)、(15)式に対する非負の双対変数を  $s$ 、 $t$ 、 $u$ 、 $w$ として、 $z$ に関するLagrange関数 $LF$ を作成する。

$$\begin{aligned} LF = & \sum_{(i,j) \in A} \sum_{k \in K} \sum_{p \in P^k} (\delta_{ij}^p c_{ij}^k + \delta_{ji}^p c_{ji}^k) z_p^k - \sum_{k \in K} \sum_{p \in P^k} z_p^k v^k \\ & + \sum_{(i,j) \in A} \sum_{n \in N} \sum_{k \in S_n} \sum_{h \in S_n} \left( \sum_{p \in P^k} \delta_{ij}^p z_p^k + \sum_{p \in P^h} \delta_{ji}^p z_p^h \right) s_{ij}^{kh} \\ & + \sum_{(i,j) \in A} \sum_{n \in N} \sum_{k \in T_n} \sum_{h \in T_n} \left( \sum_{p \in P^k} \delta_{ij}^p z_p^k + \sum_{p \in P^h} \delta_{ji}^p z_p^h \right) t_{ij}^{kh} \\ & + \sum_{(i,j) \in A} \sum_{n \in N} \sum_{k \in S_n} \sum_{h \in T_n} \left( \sum_{p \in P^k} \delta_{ij}^p z_p^k + \sum_{p \in P^h} \delta_{ji}^p z_p^h \right) u_{ij}^{kh} \\ & + \sum_{(i,j) \in A} \sum_{n \in N} \sum_{k \in S_n} \sum_{h \in T_n} \left( \sum_{p \in P^k} \delta_{ji}^p z_p^k + \sum_{p \in P^h} \delta_{ji}^p z_p^h \right) w_{ij}^{kh} \end{aligned}$$

$s$ 、 $t$ 、 $u$ 、 $w$ の添え字にはノード  $n$  ( $\in N$ ) が必要であるが、品種  $h$  と  $k$  は  $n$  に依存することなく特定できるので省略している。

品種  $k$  と始点を同じにする品種の集合を  $S^k$ 、品種  $k$  と終点を同じにする品種の集合を  $T^k$ 、品種  $k$  の始点と終点と同じである品種の集合を  $S_I^k$ 、品種  $k$  の終点と始点と同じである品種の集合を  $T_S^k$  とする。このとき、 $\sum_{n \in N} \sum_{k \in S_n} \sum_{h \in S_n}$  は  $\sum_{k \in K} \sum_{h \in S^k}$ 、 $\sum_{n \in N} \sum_{k \in T_n} \sum_{h \in T_n}$  は  $\sum_{k \in K} \sum_{h \in T^k}$ 、 $\sum_{n \in N} \sum_{k \in S_n} \sum_{h \in T_n}$  は  $\sum_{k \in K} \sum_{h \in S_I^k}$ 、 $\sum_{n \in N} \sum_{k \in S_n} \sum_{h \in T_n}$  は  $\sum_{k \in K} \sum_{h \in T_S^k}$  に置き換えることができる。 $LF$ を整理すると次のようになる。

$$LF = \sum_{k \in K} \sum_{p \in P^k} \left[ \sum_{(i,j) \in A} \left\{ \delta_{ij}^p (c_{ij}^k + d^k) + \delta_{ji}^p (c_{ji}^k + e^k) \right\} - v^k \right] z_p^k \quad (18)$$

ここで

$$\begin{aligned} d^k &= \sum_{h \in S^k} s_{ij}^{kh} + \sum_{h \in S_I^k} u_{ij}^{kh} + \sum_{h \in T^k} t_{ij}^{kh} + \sum_{h \in T_S^k} w_{ij}^{kh} \\ e^k &= \sum_{h \in S^k} s_{ij}^{hk} + \sum_{h \in S_I^k} u_{ij}^{kh} + \sum_{h \in T^k} t_{ij}^{hk} + \sum_{h \in T_S^k} w_{ij}^{hk} \end{aligned}$$

である。

以上のことから、適当な双対変数値が与えられたときに、品種  $k$ 、パス  $p$  に対するパスフロー変数の被約費用は(18)式の  $z_p^k$  の係数で与えられることになる。

#### 4 限定主問題

線形緩和問題は線形計画問題となるため、組合せ最適化問題よりも解き易い問題となる。しかし、パスフローによる定式化では、指数個の変数を含むため、直接、線形緩和問題を解くことは得策ではない。そこで、すべてのパスを対象とするのではなく、適当なパス集合からなる限定主問題から始めて、逐次、必要なパスを生成し、限定主問題に追加していく。生成するパスに対応する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、制約式も  $O(|A||N|^3)$  個と膨大な数となる。初期パス集合が与えられた場合、これらのパスを含まない強制制約式が多数存在する。これらの制約式は  $0 \leq d^k y_{ij}$  と左辺は定数 0 と表わされることになり、これらは無意味な制約式となる。そこで、初期パス集合に含まれるパスが当該品種、当該アークを含む強制制約式のみを制約式として使用する。列生成法により生成されたパスに対応するアークフロー変数が生成されたときに、逐次、新たに左辺の変数が生成された強制制約式を追加する。このような方法を列生成法とよぶ。

ここで、品種  $k (k \in K)$  の適当なパスの初期集合  $\bar{P} = (\bar{P}^k)$  が求められているものとする。このとき、パス集合が  $\bar{P}$  に制限されている次のような限定主問題  $R(\bar{P})$  を考える。

$$\begin{aligned} \text{最小化} \quad & \sum_{(i,j) \in A} \sum_{k \in K} \sum_{p \in \bar{P}^k} (\delta_{ij}^p c_{ij}^k + \delta_{ji}^p c_{ji}^k) z_p^k + \sum_{(i,j) \in A} f_{ij} y_{ij} \\ \text{条件} \quad & \sum_{p \in \bar{P}^k} z_p^k = 1 \quad \forall k \in K \end{aligned} \quad (19)$$

$$\sum_{p \in \bar{P}^k} \delta_{ij}^p z_p^k + \sum_{p \in \bar{P}^h} \delta_{ji}^p z_p^h \leq y_{ij} \quad \forall k \in S_n, h \in S_n, n \in N, (i,j) \in A, \text{ if } \alpha_{ij}^{hk} = 1 \quad (20)$$

$$\sum_{p \in \bar{P}^k} \delta_{ij}^p z_p^k + \sum_{p \in \bar{P}^h} \delta_{ji}^p z_p^h \leq y_{ij} \quad \forall k \in T_n, h \in T_n, n \in N, (i,j) \in A, \text{ if } \beta_{ij}^{hk} = 1 \quad (21)$$

$$\sum_{p \in \bar{P}^k} \delta_{ij}^p z_p^k + \sum_{p \in \bar{P}^h} \delta_{ij}^p z_p^h \leq y_{ij} \quad \forall k \in S_n, h \in T_n, n \in N, (i,j) \in A, \text{ if } \gamma_{ji}^{hk} = 1 \quad (22)$$

$$\sum_{p \in \bar{P}^k} \delta_{ji}^p z_p^k + \sum_{p \in \bar{P}^h} \delta_{ji}^p z_p^h \leq y_{ij} \quad \forall k \in S_n, h \in T_n, n \in N, (i,j) \in A, \text{ if } \gamma_{ji}^{hk} = 1 \quad (23)$$

$$\begin{aligned} z_p^k &\geq 0 \quad \forall k \in K, \quad p \in \bar{P}^k \\ 0 &\leq y_{ij} \leq 1 \quad \forall (i,j) \in A \end{aligned} \quad (24)$$

ここで

$$\alpha_{ij}^{hk} = \begin{cases} 1 & \text{if } \sum_{p \in \bar{P}^k} \delta_{ij}^p + \sum_{p \in \bar{P}^h} \delta_{ji}^p > 0 \\ 0 & \text{otherwise} \end{cases} \quad \forall k \in S_n, h \in S_n, n \in N, (i, j) \in A$$

$$\beta_{ij}^{hk} = \begin{cases} 1 & \text{if } \sum_{p \in \bar{P}^k} \delta_{ij}^p + \sum_{p \in \bar{P}^h} \delta_{ji}^p > 0 \\ 0 & \text{otherwise} \end{cases} \quad \forall k \in T_n, h \in T_n, n \in N, (i, j) \in A$$

$$\gamma_{ij}^{hk} = \begin{cases} 1 & \text{if } \sum_{p \in \bar{P}^k} \delta_{ij}^p + \sum_{p \in \bar{P}^h} \delta_{ij}^p > 0 \\ 0 & \text{otherwise} \end{cases} \quad \forall k \in S_n, h \in T_n, n \in N, (i, j) \in A$$

である定数であり、(20)~(23)式の左辺に含まれるパスフロー変数 $\delta$ が一つでも生成されているとき1、そうでないとき0である定数である。(20)~(23)式は、左辺の品種 $k, h$ 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、パス集合 $\bar{P}$ の要素数が比較的少なければ、汎用の数理最適化ソフトウェアを用いて比較的容易に解くことができる。

## 5 列生成法と行生成法

$R(\bar{P})$  はパスフロー変数が限定された問題である。そのため、被約費用が負であり、基底変換により目的関数値を減少できるようなパスおよびパスフロー変数を生成する必要がある。生成すべきパスは、 $R(\bar{P})$  の最適解において、被約費用が負であるパスである。そこで、 $R(\bar{P})$  が単体法などによって最適に解かれ、最適な双対変数値が求められているものとする。これらの最適双対変数値を用いて価格付け問題とよばれる問題を作成する。この価格付け問題を解くことによって、被約費用が負であるパスを生成することができる。

品種 $k$ のパス $p$ のパスフロー変数 $z_p^k$ に関する被約費用は、3節より、次のようになる。

$$\sum_{(i,j) \in A} \left\{ \delta_{ij}^p (c_{ij}^k + d^k) + \delta_{ji}^p (c_{ji}^k + e^k) \right\} - v^k \quad \forall p \in P^k, k \in K$$

品種 $k(k \in K)$ に対して、価格付け問題は被約費用が最小であるパスを見つける問題となる。

$$\bar{c}_{ij}^k = c_{ij}^k + d^k \quad \forall p \in P^k, k \in K$$

$$\bar{c}_{ij}^k = c_{ji}^k + e^k \quad \forall p \in P^k, k \in K$$

とすると、 $R(\bar{P})$  の品種 $k$ に対する価格付け問題は、長さが $\bar{c}^k$ である向きをもつアークからなるネットワーク上で、品種 $k$ の始点・終点間の最短距離を求める次のような最短路問題 $SP^k$ と等価になる。

$$\text{最小化 } \sum_{(i,j) \in A} (c_{ij}^k x_{ij}^k + c_{ji}^k x_{ji}^k)$$

$$\text{条件 } \sum_{i \in N_n} x_{in}^k - \sum_{j \in N_n} x_{nj}^k = \begin{cases} -1 & \text{if } n = O^k \\ 1 & \text{if } n = D^k \\ 0 & \text{otherwise} \end{cases} \quad \forall n \in N$$

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

この最短路問題を解き，“最短距離 $-v^k$ ”が負であれば，被約費用が負であるパスが見つかったことになる。 $SP^k$ は，Dijkstra法によって解くことができる。

品種 $k(\in K)$ について，このようなパスを生成して $\bar{P}^k$ に加えて，再度 $R(\bar{P})$ を解き直していく。

すべての品種に対して，“最短距離 $-v^k$ ”が負であるパスがなければ，すべての変数の被約費用が非負となり，線形緩和問題 $PFL$ が最適に解けたことになる。なお， $PFL$ の最適値は $PF$ の下界値となる。

#### ***PFL*の解法**

##### [ステップ1]

品種 $k(\in K)$ について適当な初期パス集合を求め， $\bar{P}^k$ とする。 $\bar{P}^k(k \in K)$ に含まれるパス $p$ 上のアーク $(i,j)$ に対して $a_{ij}^{kk}=1$ ， $\beta_{ij}^{kk}=1$ ， $\gamma_{ij}^{kk}=1$ とし，それ以外を0とする。 $l:=1$ とする。

##### [ステップ2]

$R(\bar{P})$ を解き，双対変数 $v, s, t, u, w$ を求め，デザイン変数 $\bar{y}^l$ を求める。

##### [ステップ3]

すべての品種 $k(\in K)$ に対して，以下の操作を行う。

- (a) アークの長さを $\bar{c}^k$ とした $SP^k$ を解き，品種 $k$ の始点・終点間の最短距離 $sp^k$ を求める。
- (b)  $sp^k - v^k < 0$ であれば，この最短路 $p$ を $\bar{P}^k$ に加え，パスフロー変数 $z_p^k$ を生成する。 $a_{ij}^{kk}$ ， $\beta_{ij}^{kk}$ ， $\gamma_{ij}^{kk}$ を更新する。
- (c)  $a_{ij}^{kk}$ ， $\beta_{ij}^{kk}$ ， $\gamma_{ij}^{kk}$ が0から1になれば，対応する強制制約式を生成し， $R(\bar{P})$ に追加する。

##### [ステップ4]

追加されたパスがあれば， $l:=l+1$ とし，ステップ2へ戻る。そうでなければ， $L:=l$ として，終了する。

ここで， $L$ は最終的な繰り返し回数である。



## 6 近似解法

前節で求めた列生成法と行生成法を用いて、線形緩和問題 $PFL$ を最適に解くことができ、 $PFL$ の最適値は $PF$ の下界値となる。次に、列生成法と行生成法によって生成された変数と制約式を用いて、 $PF$ を解くことにする。しかし、 $PFL$ で必要なパスは生成されているが、 $PF$ に対して必要なパスがすべて生成されているとは限らない。したがって、得られる解は $PF$ の最適解となるとは限らない。さらに、多くの0-1変数を含むために、直接的に解くことも困難である。

そこで、 $PFL$ のデザイン変数の緩和解をもとに問題の規模を縮小した問題を作成し、この問題を解くことによって、近似解を求めることにする。

$PFL$ の解法のステップ2において求められた $l$ 回目のデザイン変数解を $\bar{y}^l$ を用いて、次のように $\hat{y}^l$ を定める。

$$\hat{y}^l := \begin{cases} \bar{y}^l & l=1 \\ \rho\bar{y}^l + (1-\rho)\hat{y}^{l-1} & \forall l=2, \dots, L \end{cases}$$

ここで、 $\rho$ は $0 < \rho < 1$ である定数である。 $\hat{y}^l$ は、 $\bar{y}^l$ に対して指数平滑化したものであり、新しい繰り返し回数のデザイン変数の情報の重みを大きく、古い情報の重みは小さくして算出したものである。

最終的に得られた $\hat{y}^L$ が1に近ければ、 $PF$ の最適解が1である可能性が高く、逆に、0に近ければ $PF$ において0である可能性が高いと考える。適当な閾値を $\epsilon$  ( $0 \leq \epsilon \leq 0.5$ ) とする。アーク  $(i, j) (\in A)$  に対して、次のように、一部のデザイン変数を0か1に固定する。

$$y_{ij} = \begin{cases} 0 & \text{if } \hat{y}_{ij}^L \leq \epsilon \\ 1 & \text{if } \hat{y}_{ij}^L \geq 1 - \epsilon \\ \text{free} & \text{otherwise} \end{cases} \quad \forall (i, j) \in A \quad (25)$$

$PFL$ の最適解である $\bar{y}^l$ ではなく、指数平滑化した $\hat{y}^l$ をもとにデザイン変数解を設定しているが、これは次のような理由からである。 $\bar{y}^l$ はイテレーションのたびに変化するため、 $l=L$ のときに $\bar{y}_{ij}^L \leq \epsilon$ であっても、少し前のイテレーションでは $\bar{y}_{ij}^l > \epsilon$ である可能性がある。このような場合、0と固定してしまうと $PF$ の最適解を排除する可能性がある。そこで、過去のデザイン変数の情報も考慮した $\hat{y}^L$ を利用する。

以上のように、一部を0または1に固定して問題の規模を縮小し、かつ列生成法と行生成法によって生成された変数と制約式を用いた次のような問題 $FX(\hat{Y})$ を作成する。ここで、 $\hat{Y}$ は(25式により定義される領域である。

$$\begin{aligned}
& \text{最小化} && \sum_{(i,j) \in A} \sum_{k \in K} \sum_{p \in \hat{P}^k} (\delta_{ij}^p c_{ij}^k + \delta_{ji}^p c_{ji}^k) z_p^k + \sum_{(i,j) \in A} f_{ij} y_{ij} \\
& \text{条件} && \sum_{p \in \hat{P}^k} z_p^k = d^k \quad \forall k \in K \\
& && \sum_{p \in \hat{P}^k} \delta_{ij}^p z_p^k + \sum_{p \in \hat{P}^h} \delta_{ji}^p z_p^h \leq d^k y_{ij} \quad \forall k \in S_n, h \in S_n, n \in N, (i,j) \in A, \text{ if } \alpha_{ij}^{hk} = 1 \\
& && \sum_{p \in \hat{P}^k} \delta_{ij}^p z_p^k + \sum_{p \in \hat{P}^h} \delta_{ji}^p z_p^h \leq d^k y_{ij} \quad \forall k \in T_n, h \in T_n, n \in N, (i,j) \in A, \text{ if } \beta_{ij}^{hk} = 1 \\
& && \sum_{p \in \hat{P}^k} \delta_{ij}^p z_p^k + \sum_{p \in \hat{P}^h} \delta_{ij}^p z_p^h \leq d^k y_{ij} \quad \forall k \in S_n, h \in T_n, n \in N, (i,j) \in A, \text{ if } \gamma_{ij}^{hk} = 1 \\
& && \sum_{p \in \hat{P}^k} \delta_{ji}^p z_p^k + \sum_{p \in \hat{P}^h} \delta_{ji}^p z_p^h \leq d^k y_{ij} \quad \forall k \in S_n, h \in T_n, n \in N, (i,j) \in A, \text{ if } \gamma_{ji}^{hk} = 1 \\
& && z_p^k \geq 0 \quad \forall k \in K, p \in \hat{P}^k \\
& && y_{ij} \in \hat{Y} \quad \forall (i,j) \in A
\end{aligned}$$

ここで、 $\hat{P}^k (k \in K)$  はPFLの解法において、最終的に得られた $\bar{P}^k (k \in K)$  である。

このFX( $\hat{Y}$ ) を分枝限定法によって解く。得られる解はPFの近似解となる。freeであるデザイン変数が比較的少数であれば、数理最適化ソフトウェアを用いて解くことができる。

## 7 数値実験

数値実験によって、従来の解法と本解法による計算の比較を行う。アーク上のフロー費用とデザイン費用が比例し、デザイン費用/フロー費用=5、 $i < j$  であるすべてのノード間にアークと品種の需要があるようなユークリッド平面上の問題例（同一ノード数各10問）を用いる。ノード数は10~50とし、アーク数と品種数は表1に示す通りである。パラメータは、 $\rho = 0.2$ ,  $\varepsilon = 0.001$ と設定した。

上界値の比較のためにバックワード法 [10], Minoux法の改良法 [6], 一方、下界値の比較のために双対上昇法 [1] とLagrange緩和法 [5] を用いる。同時に、数理計画ソフトウェアCPLEXによる分枝限定法を用いて、直接、アークフローにより定式化した問題を解いている。数値実験には、Pen3.2GHzのパソコン（CPLEXは3.4GHz）、WindowsXP, VB.NET2003およびCPLEX9.1を使用する。

表1に、従来の解法と提案した近似解法（本解法）による上界値の平均誤差と平均計算時間を示す。誤差は、CPLEXによる最適値または最良の下界値との差の比率である。表2に、従来の解法と本解法による下界値の平均誤差と平均計算時間を示す。本解法による下界値は線形緩和問題PFLの最適値である。誤差は、CPLEXによる最適値または最良の上界値、または本解法による上界値との差の比率である。また、-は問題の規模

が大きいため、CPLEXを実行できなかったことを表す。

本解法では、ノード数30までの最適解を求めることができた。また、ノード数40以上では誤差が0.5%～0.7%となっている。ノード数40以上の問題の最適値は、CPLEXによって求めることができなかった。このため、ノード数40以上の誤差は、最適値ではなく下界値と差を使用しており、実際の最適値との誤差はもう少し小さく、最適解に近いと考えられる。また、近似解法の中では、すべてのノード数において、本解法が下界値、上界値、ともに最も良い解を求めることができた。今回使用した問題は、すべてのノード間にアークが存在する問題であるため、生成されるパス数が膨大なものとなり、本解法では膨大な計算時間が必要となった。

表 1：上界値と計算時間の比較

N	A	K	バックワード		Minoux改		CPLEX		本解法	
			誤差 (%)	時間 (s)	誤差 (%)	時間 (s)	誤差 (%)	時間 (s)	誤差 (%)	時間 (s)
10	45	45	0.2	0.0	0.2	0.0	0.0	0.2	0.0	1.0
20	190	190	0.4	0.1	0.6	0.0	0.0	65	0.0	74
30	435	435	0.3	0.6	0.5	0.0	0.0	18164	0.0	1022
40	780	780	0.9	3.7	1.1	0.0	-		0.6	9310
50	1225	1225	1.1	13.6	1.2	0.1	-		0.7	89822

表 2：下界値と計算時間の比較

N	双対上昇		Lagrange		CPLEX	本解法
	誤差 (%)	時間 (s)	誤差 (%)	時間 (s)	誤差 (%)	誤差 (%)
10	1.8	0.1	0.1	0.0	0.0	0.0
20	3.2	0.1	0.3	0.9	0.0	0.2
30	3.1	0.1	0.5	4.4	0.0	0.4
40	3.1	0.4	0.7	14.6	-	0.5
50	3.4	0.3	1.0	34.8	-	0.7

## 8. おわりに

本研究では、容量制約のないネットワークデザイン問題に対して、以下のような成果を示した。最適解において、同一ノードを始点とする品種と終点とする品種のフローが同一のアーク上で同じ向きに流れるフローが存在しないことを用いた新たな強い妥当不等式を提案した。これらの妥当不等式を含むパスフロー変数を用いた定式化に対して、列生成法と行生成法を組み合わせた線形緩和問題の解法を提案し、緩和解をもとにした近似解法を提案した。最後に、数値実験により、提案する解法が従来の近似解法よりも精度が高い解を算出できることを示した。

今後の課題として、不要なパスの削除などによる問題の縮小、分枝限定法に移行する際の自由変数の数の制御といった解法の改良が必要である。また、新たに提案した妥当

不等式に対するLagrange緩和法や、分枝限定法への列生成法の組み込みなども挙げられる。

#### 参考文献

- [1] A. Balakrishnan, T. L. Magnanti, and R. T. Wong. A dual-ascent procedure for large-scale uncapacitated network design. *Operations Research*, Vol. 37, pp. 716-740, 1989.
- [2] J. W. Billheimer and P. Gray. Network design with fixed and variable cost elements. *Transportation Science*, Vol. 7, pp. 49-74, 1973.
- [3] K. Holmberg and J. Hellstrand. Solving the uncapacitated network design problem by a Lagrangean heuristic and branch-and-bound. *Operations Research*, Vol. 46, pp. 247-259, 1998.
- [4] 片山直登. ネットワーク設計問題. 朝倉, 2008.
- [5] 片山直登, 岩田実, 柳下和夫, 三原一郎, 今澤明男, ラグランジュ緩和法を用いた容量制約のないネットワーク計画問題の解法. 土木計画学研究・論文集, Vol. 11, pp. 105-112, 1993.
- [6] 片山直登. ネットワークデザイン問題の近似解法. 流通経済大学流通情報学部開校記念論文集, pp. 171-191, 1997.
- [7] M. Los and C. Lardinois. Combinatorial programming, statistical optimization and the optimal transportation network problem. *Transportation Research B*, Vol. 16, pp. 89-124, 1982.
- [8] T. L. Magnanti, P. Mireault, and R. T. Wong. Tailoring benders decomposition for uncapacitated network design. *Mathematical Programming Study*, Vol. 26, pp. 112-155, 1986.
- [9] T. L. Magnanti and R. T. Wong. Network design and transportation planning: Models and algorithms. *Transportation Science*, Vol. 18, pp. 1-55, 1984.
- [10] M. Minoux. Multiflots de coût minimal avec fonctions de coût concaves. *Annales des Télécommunications*, Vol. 31, pp. 77-92, 1976.
- [11] M. Minoux. Network synthesis and optimum network design problems: Models, solution methods and applications. *Networks*, Vol. 19, pp. 313-360, 1989.