《論　文》

容量スケーリング法を用いた容量制約をもつ多品種フローネットワークデザイン問題の近似解法

片　山　直　登

1　はじめに

容量制約をもつ多品種フローネットワークデザイン問題(Multicommodity Capacitated Network Design Problem:*MCND*)は，多品種フローを含むアーク上の容量の制約をもつネットワークにおいて，適切なネットワークの形状とフローの経路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在している。本研究では，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。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物が存在することを意味する。ノードは交換機・コンピュータ，交差点・交通需要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流れるものを表している。アークの容量は，通信回線の容量，道路の交通容量や輸送能力のように，アークは単位時間当たりの処理能力を表している。

本研究では，容量制約をもつネットワークデザイン問題に対して，弱い定式化と強い定式化を用いた容量スケーリング法を提案する。

容量制約をもつ多品種フローネットワークデザイン問題は*NP-hard*であることが知られており，経験的には*NP-hard*の問題群のなかでも困難な問題である。この問題に対しては数多くの研究がなされている。

Lagrange 緩和法を用いた多くの研究が行われている。片山―春日井[21]は，双対上昇法との比較の中で，フロー保存条件Lagrange 緩和法を使用している。Gendron-Crainic[6) 7) 8)] およびGendron-Crainic-Frangioni[9] は，連続緩和，フロー保存条件Lagrange 緩和，容量制約Lagrange 緩和などとそれらの解法を示している。Crainic- Frangioni-

Gendron[2] は劣勾配法に対してBundle 法を適用した解法を示している。Holmberg-Yuan[13] はLagrange 緩和法と分枝限定法を組合せた解法を示している。

一方，Herrmann-Ioannou-Minis[12] は，容量をもたないネットワークデザインに対するBalakrishnan-Magnanti-Wong の双対上昇法[1] を拡張した解法を示しているが，Gendron[5] はこの解法が誤りであることを指摘している。片山[20] は，Herrmann-Ioannou-Minis とは異なるタイプの容量制約をもたないモデルに対する双対上昇法を拡張した解法を示している。

近似解法も数多く提案されている。Gendron-Crainic[7][8] は多品種フロー問題の解法であるリソース分解法を利用した貪欲解法を示している。Crainic-Gendreau-Farvolden[3] は，パスフローによる定式化を用いて，シンプレックス法の基底解変更にタブーサーチ法を適用した近似解法を示している。Ghamlouche-Crainic-Gendreau[11][10] は，サイクル部のフローを移動する隣接探索法を用いた解法，およびパス再リンク探索を用いた解法を示している。片山[19] は貪欲的なMinoux タイプの近似解法を提案している。

スロープスケーリング法は，フロー費用やアーク容量を変化させながら線形緩和問題を繰り返し解き，デザイン変数の収束解を求める近似解法である。Kim-Pardals[14][15][16] は，区分的線形フロー費用や非線形フロー費用をもつネットワークフロー問題の解法を示している。Crainic-Gendron-Hernu[4] は，フロー費用を変化させる弱い定式化に基づいた解法，Lagrange 緩和法，長期メモリを用いた解法を示している。一方，久保[18] や小松[17] はサービスネットワーク設計問題に対してパスフローによる定式化に基づいた容量を変化させる容量スケーリング法を提案している。

2　問題の定式化

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)上を流れる品種kのフロー量を表すフロー変数をx_{ij}^kとする。アーク(i, j)上に設置できるアークの容量をC_{ij}で表す。アーク(i, j)のデザイン費用をf_{ij}，品種kの単位当たりのルーティング費用をc_{ij}^kとする。品種kの始点をO^k，終点をD^k, OD フロー量をd^kとする。このとき，アークフローを用いた$MCND$の定式化は，次のように表される。

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

st

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

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

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

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

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

（1）式は，ルーティング費用とデザイン費用の総和の最小化を表す。（2）式はフロー保存式である。（3）式は，アーク(i, j)が存在するときにのみフローの存在を許し，かつそのアークのフロー量は容量以下であることを表す。（4）式は，アーク(i, j)上を流れる品種kのフロー量がアーク(i, j)存在するときに最大d^k，存在しないとき０であることを表す。（５）式はデザイン変数の０－１条件，（６）式はフロー変数の範囲を表す。

3 Crainic-Gendron-Hernuのスロープスケーリング法

スロープスケーリング法は，線形緩和問題を解くことによって得られた解をもとにフロー費用を変化させ，この操作を解が収束するまで繰り返し，収束解をもとにデザイン変数を設定し，近似解を算出する方法である。Crainic-Gendron-Hernu は，基本的なスロープスケーリング法とLagrange 摂動を用いたスロープスケーリング法を示している。これらの２つのスロープスケーリング法[4] を紹介する。

3.1 基本的なスロープスケーリング法

Crainic-Gendron-Hernu のスロープスケーリング法では，品種k，アーク(i, j)に対する線形パラメータρ_{ij}^kを用いてフロー費用を変換した問題を作成し，さらにデザイン変数を線形緩和した多品種フロー問題 (Multicommodity Capacitated Network Flow Problem:*MCNF*) を利用する。線形パラメータρ が与えられたとき，$MCNF(\rho)$ は次のように表すことができる。

$$MCNF(\rho): \quad z(\rho) = min \sum_{(i,j) \in A} \sum_{k \in K} (c_{ij}^k + \rho_{ij}^k) x_{ij}^k \tag{7}$$

st

$$\sum_{k \in K} x_{ij}^k \leq C_{ij} \qquad \forall (i, j) \in A \tag{8}$$

$$x_{ij}^k \geq 0 \quad \forall k \in K,\ (i,j) \in A \tag{9}$$

（2）式

はじめに$\rho=\rho(0)$を初期値とした$MCNF(\rho)$を解く。ここで，$\rho_{ij}^k(0)=f_{ij}/C_{ij}$, $\forall(i,j)\in A$, $k\in K$である。

スロープスケーリング法ではρを変化させながら$MCNF(\rho)$を繰り返し解くため，現在の繰り返し回数をtとおく。t回目の$MCNF(\rho(t))$の最適解を$\tilde{x}$とする。$\rho(t+1)$は，$\tilde{x}$の値を反映して，次のように修正する。もし，$\sum_{k\in K}\tilde{x}_{ij}^k>0$であれば，次式のようにデザイン費用を考慮するように設定する。

$$\sum_{k\in K}(c_{ij}^k+\rho_{ij}^k(t+1))\tilde{x}_{ij}^k=\sum_{k\in K}c_{ij}^k\tilde{x}_{ij}^k+f_{ij} \tag{10}$$

$\sum_{k\in K}\tilde{x}_{ij}^k=0$であれば，$\rho(t+1)$に$\rho(t)$をそのまま使用する。$\rho(t)$についてまとめると，$t>0$に対して次式となる。

$$\rho_{ij}^k(t)=\begin{cases} f_{ij}/\sum_{k\in K}\tilde{x}_{ij}^k & if\ \tilde{x}_{ij}^k>0 \\ \rho_{ij}^k(t-1) & otherwise \end{cases} \quad \forall(i,j)\in A, k\in K \tag{11}$$

一定回数の繰り返しの後に得られた$MCNF$の最適解を$\tilde{x}$としたとき，次式のようにデザイン変数を設定すれば，$MCND$の実行可能解を導くことができる。

$$\tilde{y}_{ij}=\begin{cases} 1 & if\ \sum_{k\in K}\tilde{x}_{ij}^k>0 \\ 0 & otherwise \end{cases} \quad \forall(i,j)\in A \tag{12}$$

$MCND$の上界値は次式によって求めることができる。

$$\sum_{(i,j)\in A}\sum_{k\in K}c_{ij}^k\tilde{x}_{ij}^k+\sum_{(i,j)\in A}f_{ij}\tilde{y}_{ij} \tag{13}$$

3.2　Lagrange摂動を用いたスロープスケーリング法

π，$\alpha(\geq 0)$，$\beta(\geq 0)$をそれぞれ（2），（3），（4）式に対するLagrange乗数・双対変数とする。このとき，Crainic-Gendron-Hernuは最短パス緩和SP(Shortest Path Relaxation)と連続ナップサック緩和CK(Continuous Knapsack Relaxation)の２つ緩和問題を用いている。

$$SP: \quad z(\alpha,\beta) = min \quad \sum_{(i,j)\in A}\sum_{k\in K}(c_{ij}^k + \alpha_{ij} + \beta_{ij}^k)x_{ij}^k + \sum_{(i,j)\in A}(f_{ij} - C_{ij}\alpha_{ij} - \sum_{k\in K} d^k\beta_{ij}^k)y_{ij} \tag{14}$$

st

（2），（5），（9）式

この問題SPは品種kごとの最短路問題に分離することができ，容易に最適に解くことができる。

$$CK: \quad z(\pi) = min \quad \sum_{(i,j)\in A}\sum_{k\in K}(c_{ij}^k + \pi_i^k - \pi_j^k)x_{ij}^k + \sum_{(i,j)\in A} f_{ij}y_{ij} + \sum_{k\in K} d^k(\pi_{D(k)}^k - \pi_{O(k)}^k) \tag{15}$$

st

（3），（4），（5），（9）式

この問題CKはアーク(i, j)ごとのデザイン変数が1の場合の連続ナップサック問題と0の場合の自明な問題に分けることができ，容易に最適に解くことができる。

Lagrange 乗数π はBundle 法などの劣勾配法で定めることができ，α，β はLagrange 緩和解から導くことができる。Lagrange 乗数・双対変数の情報を次のように定めておく。

$$\Delta_{ij}^k = \pi_i^k - \pi_j^k + \alpha_{ij} + \beta_{ij}^k \quad \forall(i,j)\in A, k\in K \tag{16}$$

t 回目の繰り返しにおいて，スロープスケーリング法の線形パラメータ$\rho(t)$を次式のように定める。

$$\rho_{ij}^k(t) = \begin{cases} \Delta_{ij}^k + (f_{ij} - \sum_{(i,j)\in A}\Delta_{ij}^k\tilde{x}_{ij}^k)/\sum_{k\in K}\tilde{x}_{ij}^k & if\ \tilde{x}_{ij}^k > 0 \\ \rho_{ij}^k(t-1) & otherwise \end{cases} \quad \forall(i,j)\in A, k\in K \tag{17}$$

さらに，Crainic-Gendron-Hernu は，$\rho(t)$に対して長期メモリを設定して，集中・多様化探索を図っている。

4　容量スケーリング法

容量スケーリング法は，線形緩和問題を解くことによって得られた解をもとにアーク容量を変化させ，この操作を解が収束するまで繰り返し，収束解をもとにデザイン変数を設定し，近似解を算出する方法である。本研究では，$MCND$の弱い定式化に対する容量スケーリング法と強い定式化に対する容量スケーリング法を示す。

4.1　弱い定式化と強い定式化

弱い定式化とは制約式（4）を取り除いた定式化である。この定式化の線形緩和による下界値は，（4）式を含めた定式化に比べて大きく劣ることが知られている。一方，強い定式化は制約式（4）を含む定式化であり，一般に，この形式の定式化の線形緩和による下界値は良いことが知られている 。

*MCND*の弱い定式化に対して，デザイン変数を０以上の連続数に緩和し，t回目の繰り返しにおけるアーク容量を$C(t)$とした問題$MCNDLW(t)$を考える。

$$MCNDLW(t): \quad min \quad \sum_{(i,j)\in A} \sum_{k\in K} c_{ij}^k x_{ij}^k + \sum_{(i,j)\in A} f_{ij} y_{ij} \tag{18}$$

st

$$\sum_{i\in N} x_{in}^k - \sum_{j\in N} x_{nj}^k = \begin{cases} -d^k & if\ n = O^k \\ d^k & if\ n = D^k \\ 0 & otherwise \end{cases} \quad \forall n \in N,\ k \in K \tag{19}$$

$$\sum_{k\in K} x_{ij}^k \le C_{ij}(t) y_{ij} \qquad \forall (i,j) \in A \tag{20}$$

$$y_{ij} \ge 0 \qquad \forall (i,j) \in A \tag{21}$$

$$0 \le x_{ij}^k \le d^k \quad \forall k \in K,\ (i,j) \in A \tag{22}$$

一方，*MCND*の強い定式化に対して，デザイン変数の０以上の連続数に緩和し，アーク容量を$C(t)$とした問題$MCNDLS(t)$を考える。

$$MCNDLS(t): \quad min \quad \sum_{(i,j)\in A} \sum_{k\in K} c_{ij}^k x_{ij}^k + \sum_{(i,j)\in A} f_{ij} y_{ij} \tag{23}$$

st

$$x_{ij}^k \le d^k y_{ij} \quad \forall k \in K,\ (i,j) \in A \tag{24}$$

(19)，(20)，(21)，(22) 式

4.2　容量スケーリング法

*MCND*LWまたは*MCND*LSの解である最適なデザイン変数を$\tilde{y}$とおく。このとき，$t+1$回目のアーク容量$C(t+1)$を次式で定義する。

$$C_{ij}(t+1) := \lambda C_{ij}(t)\tilde{y}_{ij} + (1-\lambda) C_{ij}(t) \quad \forall (i,j) \in A \tag{25}$$

ここで，λは０から１のパラメータである。

容量スケーリング法の流れを示す。

［ステップ1］ $t:=0$, $C_{ij}(1):=C_{ij}$, $\forall(i,j)\in A$, $ub_{min}:=\infty$, $thres:=thres_0$ とする。

［ステップ2］ $t:=t+1$とする。$MCNDLW(t)$または$MCNDLS(t)$を解き，$\tilde{y}$を求める。

［ステップ3］ $C_{ij}(t+1):=\lambda C_{ij}(t)\tilde{y}_{ij}+(1-\lambda)C_{ij}(t)$, $\forall(i,j)\in A$とする。

［ステップ4］ $t\ MOD\ t_{fs}=0$ または $t=t_{max}$であれば［ステップ5］へ，そうでなければ［ステップ2］へ戻る.

［ステップ5］

$$\bar{y}_{ij}=\begin{cases}1 & if\ \tilde{y}_{ij}>=thres\\ 0 & otherwise\end{cases}\qquad \forall(i,j)\in A$$

として，問題$MCNDFS(\bar{y})$を解き，上界値$ub(t)$を求める。$ub_{min}>ub(t)$であれば，$ub_{min}:=ub(t)$とする。

［ステップ6］ 問題$MCND$FSが実行不可能であれば$thres:=\max(thres-\delta,0)$，そうでなければ$thres:=\min(thres+\delta,1)$とする。

［ステップ7］ $t=t_{max}$であれば終了，そうでなければ［ステップ2］へ戻る。

ここで，$MCNDFS(\bar{y})$はデザイン変数を$\bar{y}$としたときの上界値および実行可能解を算出する問題であり，次のように表す。

$$MCNDFS(\bar{y}):\quad min\quad \sum_{(i,j)\in A}\sum_{k\in K}c_{ij}^k x_{ij}^k+\sum_{(i,j)\in A}f_{ij}\bar{y}_{ij} \tag{26}$$

st

$$\sum_{k\in K}x_{ij}^k\le C_{ij}\bar{y}_{ij}\qquad \forall(i,j)\in A \tag{27}$$

$$x_{ij}^k\le d^k\bar{y}_{ij}\quad \forall k\in K,\ (i,j)\in A \tag{28}$$

(19)，(22) 式

表1：問題の性質

問題名	ノード数	アーク数	品種数	費用特性	容量特性
p33	20	230	40	V	L
p34	20	230	40	F	L
p35	20	230	40	V	T
p36	20	230	40	F	T
p37	20	229	200	V	L
p38	20	229	200	F	L
p39	20	229	200	V	T
p40	20	229	200	F	T

表２：計算結果

	CPLEX				強い定式化			弱い定式化		
問題名	上界値	下界値	誤差	計算時間	上界値	誤差	計算時間	上界値	誤差	計算時間
p33	494257	494257	0.0	0:00:01	494292	0.0	0:00:09	498073	0.8	0:00:14
p34	642921	642921	0.0	0:00:00	642921	0.0	0:00:09	682315	6.1	0:00:25
p35	473228	473228	0.0	0:00:05	473609	0.1	0:01:09	474304	0.2	0:01:20
p36	573952	573952	0.0	0:00:18	578030	0.7	0:06:25	595664	3.8	0:00:33
p37	114961	110748	3.8	10:00:00	119595	8.0	5:15:28	136334	23.1	0:04:10
p38	151582	146527	3.4	10:00:00	154053	5.1	5:18:47	183903	25.5	0:04:16
p39	106523	103910	2.5	10:00:00	110376	6.2	6:09:54	120517	16.0	0:02:26
p40	140931	134065	5.1	10:00:00	143220	6.8	2:01:21	163817	22.0	0:05:50

また，*thres*は実行可能解の探索においてデザイン変数を０に設定するか１に設定するかの閾値であり，δ はその変化量である。$MCNDFS(\bar{y})$は必ずしも実行可能ではないため，実行不可能であれば閾値を*thres*を減少させて実行可能性を増加させる。一方，実行可能であれば閾値を*thres*を増加させることによって，よりデザイン費用の少ない解を探索可能にしている。ub_{min}は最良の上界値，t_{max}は繰り返し回数，$thres_0$ は閾値の初期値，t_{fs}は上界値・実行可能解を探索する周期である。

5　数値実験

提案した容量スケーリング法を評価するために数値実験を行った。使用したデータは，OR-Library 内 のMulticommodity Network Design Problems(http://www.di.unipi.it/ di/groups/optimize/Data/MMCF.html#NetDesMMCF) にあるMulgenI の問題p33 からp40 である。これらの問題の性質を表１に示す。表内のVはフロー費用の重みが大きい，Fはデザイン費用の重みが大きい，Lはアーク容量が大きい，Tはアーク容量が小さいことを表している。

容量スケーリング法における線形計画問題は，GNU LineraProgramming Kit Ver.4.4(GLPK) を用いて解く。ただし，線形計画問題を解く際の基底変換の回数の上限を10 万回に設定する。また，提案した解法を評価するために，汎用の数理最適化ソフトウエアであるCPLEXVer.7.1 を用いて，*MCND*を組合せ最適化問題として直接に解く。ただし，CPLEX の計算時間の上限を10 時間に設定する。GLPKの単体法は，CPLEX のものよりも10 から100 倍遅いことが報告されている。

$\lambda := 0.1$，$thres0 := 0.25$，$\delta := 0.05$，$t_{fs} := 1$，$t_{max} := 100$ に設定して，計算を行う。計算結果を表２ に示す。CPLEXの上界値と下界値が一致している場合は最適値であり，一

致していない場合は最適値を求めることができていない。計算時間は（時：分：秒）であり，容量スケーリング法では最良の上界値が求められるまでの時間である。誤差はCPLEXの下界値に対する差(%)である。

p36までの品種数40の問題では，CPLEXでは最適解を短時間で求めることができた。容量スケーリング法の強い定式化では最適解または誤差0.7 % 以内の上界値を求めることができたが，弱い定式化では。p34とp36の誤差が大きくなっている。p37以降の品種数200の問題では，CPLEXでは2.5から5.1 %の誤差の上界値，強い定式化では5.1から7.9 %の誤差の上界値を求めることができた。一方，弱い定式化では16.0から25.5% と誤差の大きな上界値となった。CPLEX，強い定式化ともに膨大な計算時間が必要となった。ただし，容量スケーリング法ではGLPKを用いているためにCPLEXに比べて単体法が10から100倍遅くなることから，容量スケーリング法による計算時間とCPLEXによる計算時間を直接比較することはできない。

6 おわりに

本研究では，容量制約をもつネットワークデザイン問題に対して，弱い定式化と強い定式化を用いた容量スケーリング法による近似解法を提案し，数値実験により解法の有効性を示した。強い制約を用いた容量スケーリング法では，優れた上界値を算出することができたが，品種数が多いと膨大な計算時間を必要とする。このため，GLPKコードからCPLEXコードへの変換や，パスフローによる定式化と列生成法を用いた方法などによって，計算時間の短縮を図る必要がある。

参考文献

[1] A. Balakrishnan, T. Magnanti, and R. Wong. A dual-ascent procedure for large-scale uncapacitated network design. *Operations Research*, Vol. 37, pp. 716-740, 1989.

[2] T.G. Crainic, A. Frangioni A., and B. Gendron. Bundle-based relaxation methods for multicommodity capacitated fixed charge network design problems. *Discrete Applied Mathematics*, Vol. 112, pp. 73-99, 2001.

[3] T.G. Crainic, M. Gendreau, and J.M. Farvolden. A simplex-based tabu search for capacitated network design. *INFORMS Journal on Computing*, Vol. 12, pp. 223-236, 2000.

[4] T.G. Crainic, B. Gendron, and G. Hernu. A slope scaling/lagrangean perturbation heuristic with long-term memory for multicommodity capacitated fixed-charge network design. Technical Report CRT-2003-05, Centre de recherche sur les transports,Universite de Montreal, 2003.

[5] B. Gendron. A note on" a dual-ascent approach to the fixed-charge capacitated network design

problem" .Technical Report CRT-99-38, Centre de recherche sur les transports,Universite de Montreal, 1999.

[6] B. Gendron and T.G. Crainic. Parallel implementations of bounding procedures for multi-commodity capacitated network design. Technical Report CRT-94-45, Centre de recherche sur les transports,Universite de Montreal, 1994.

[7] B. Gendron and T.G. Crainic. Relaxations for multicommodity capacitated network design problems. Technical Report CRT-965, Centre de recherche sur les transports,Universite de Montreal, 1994.

[8] B. Gendron and T.G. Crainic. Bounding procedures for multicommodity capacitated fixed charge network design problems. Technical Report CRT-96-06, Centre de recherche sur les transports,Universite de Montreal, 1996.

[9] B. Gendron, T.G. Crainic, and A. Frangioni. Multicommodity capacitated network design. Technical Report CRT-98-14, Centre de recherche sur les transports,Universite de Montreal, 1997.

[10] I. Ghamlouche, T.G. Crainic, and M. Gendreau. Path relinking for fixed-charge capacitated multicommodity network design. TechnicalReport CRT-2001-01, Centre de recherche sur les transports,Universite de Montreal, 2002.

[11] I. Ghamlouche, T.G. Crainic, and M. Gendreau. Cycle based neighborhood structures for fixed-charge capacitated multicommodity network design. *Operations Research*, Vol. 51, pp. 655-667, 2003.

[12] J.W. Herrmann, G. Ioannou, and I. Minis. A dual ascent approach to the fixed-charge capacitated network design problem. *European Journal of Operational Research*, Vol. 95, pp. 476-490, 1996.

[13] K. Holmberg and D. Yuan. A lagrangian heuristic based branch-and-bound approach for the capacitated network design problem. *Operations Reserach*, Vol. 48, pp. 461-481, 2000.

[14] D. Kim and P.M. Pardalos. A solution approach to the fixed charge network flow problem using a dynamic slope scaling procedure. *Operations Research Letters*, Vol. 24, pp. 195-203, 1999.

[15] D. Kim and P.M. Pardalos. Adynamic domain contraction algorithm for nonconvex piecewise linear network flow problems. *Journal of Global Optimization*, Vol. 17, pp. 225-234, 2000.

[16] D. Kim and P.M. Pardalos. Dynamic slope scalingand trust interval techniques for solving concave piecewise linear network flow problems. *Networks*, Vol. 35, pp. 216-222, 2000.

[17] 小松馨．サービス・ネットワーク設計問題に対する列生成法．海洋大学卒業論文，2003.

[18] 久保幹雄．MIP ソルバーを利用したメタ解法の設計に関する報告書．Working Paper，海洋大学，2003.

[19] 片山直登．容量制約をもつネットワークデザイン問題の貪欲解法．流通経済大学流通情

報学部紀要，Vol. 6, No. 1, pp. 57-67, 2001.
[20] 片山直登．容量制約をもつネットワークデザイン問題の双対上昇法．流通経済大学流通情報学部紀要，Vol. 8, No. 2, pp. 23-32, 2004.
[21] 片山直登，春日井博．容量制約をもつ多品種流ネットワークデザイン問題. 日本経営工学会誌，Vol. 44, pp. 164-175, 1993.