容量スケーリングと局所分枝を用いた容量制約をもつネットワーク設計問題の解法

Capacity Scaling and Local Branch Approaches for the Capacitated Network Design Problem

片山直登*
Naoto KATAYAMA*
*流通経済大学流通情報学部
*Ryutsu Keizai University
*Email:katayama@rku.ac.jp

要旨：容量制約をもつネットワーク設計問題は，輸送・ロジスティクスや通信の領域で発生するネットワークを設計する問題の基本的な問題であり，固定費用と変動費用およびアーク容量を考慮してネットワークの形状とモノの移動するパスを決定する問題である. 本研究では, この問題に対して容量スケーリング，列生成，行生成に加え，局所分枝による分枝限定法を用いた解法を提案する．容量スケーリング法は容量を変化させながら線形緩和問題を解き直す方法であり，局所分枝は現在の解の近傍に限定して探索する方法である．さらに，数値実験によって，従来の解法と比較し，提案した解法の有効性を示す.

1. はじめに

容量制約をもつネットワーク設計問題 (CND) は，アーク容量の制約をもつネットワークにおいて，多品種のモノが移動するときに，費用が最小となるようなネットワークの形状とモノが移動するパスを求める問題であり，ロジスティックスや通信ネットワーク分野に現れる重要な問題である[1].

CND に対しては数多くの研究がなされている. タブー探索法を用いた解法として，Crainic ら[2]は単体法に基づく方法, Ghamlouche ら[3]はサイクルに基づく方法, Ghamlouche ら[4]はパス再結合法とタブー探索法を組合せた方法，Crainic ら[5]は共同並列タブー探索法を提案している.

また，現在のフローに基づいて費用や容量を変更して近似解を求めるスケーリング法を用いた研究も行われており，Crainic ら[6]は費用を変更する傾斜スケーリング法, Katayama ら[7]はアークの容量を変更する容量スケーリング法を提案している.

最近になって，数理計画ソルバーなどの分枝限定法とメタヒューリスティクスを組合せた解法が開発され，成果を挙げている．Martín-Gonzáleza[8]は局所分枝を分枝限定法に組み込んだ解法，Hewitt ら[9]は限定問題を解くことにより広い近傍探索を行う IP 探索法, Chouman-Crainic[10]は MIP モデルとタブー探索法を組合せた解法を提案している.

本研究では, CND に対して, 容量スケーリング，列生成，行生成に加え，局所分枝を加えた分枝限定法を用いた解法を提案する．さらに，数値実験によって，従来の解法と比較し，提案した解法の有効性を示す.

2. 定式化

ノード集合 N と向きをもつアーク集合 A からなるネットワークが与えられている．品種集合を K，品種 k の取りうるパスの集合を P^k，品種 k の需要量を d^k とする．アーク (i,j) に関して，このアークを選択した際に生じる固定費用を f_{ij}，品種 k の単位当りのフロー費用を c_{ij}^k，アーク容量を C_{ij} とする. また，アーク (i,j) に関する 0-1 のデザイン変数を y_{ij}，品種 k のパス p 上のフロー量を表すフロー変数を x_p^k とする．アーク (i,j) がパス p に含まれるとき 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，パスフローを用いた CND の定式化 *CNDP* は次のようになる.

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

$$subject \ to \sum_{p\in P^k} x_p^k = d^k \quad \forall k\in K \quad (2)$$

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

$$\sum_{p\in P^k}\delta_{ij}^{p}x_p^k \le d^k y_{ij} \quad \forall (i,j)\in A, k\in K \quad (4)$$

$$x_p^k \ge 0 \quad \forall p\in P^k, k\in K \quad (5)$$

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

3. 容量スケーリング

なんらかの手段を用いて，*CNDP*の最適なフロー量を求めることができれば，アーク容量をこれらの量に変更しても，*CNDP*の最適値は変わらない．また，線形緩和問題において，アーク容量を最適なフロー量に近づけると，デザイン変数が 1 に近くなることが期待できる．しかし，最適なフロー量を求めること自体が問題の目的であるため，実際にはこれらの値を直接求めることはできない．そこで，線形緩和問題において,最適なフロー量に近づくように，少しずつアーク容量を変更しながら問題を解き直すことを繰り返す．容量スケーリング法は，このように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 変数解を導く近似解法である[7].

*CNDP*において，デザイン変数の 0-1 条件を 0 から 1 の連続数に緩和し，繰返し回数lにおいてアーク容量を$\boldsymbol{C}^l$とした問題 $LR(\boldsymbol{C}^l)$を考える．$LR(\boldsymbol{C}^l)$のデザイン変数解を$\tilde{\boldsymbol{y}}$とし，パラメータ$\lambda(0\le\lambda\le 1)$を用いて，次のように$l+1$回目のアーク容量を設定する．

$$C_{ij}^{l+1} = \lambda C_{ij}^{l}\tilde{y}_{ij} + (1-\lambda)C_{ij}^{l} \quad (i,j)\in A \quad (7)$$

このようにアーク容量を変更しながら，線形緩和問題を解き直す．一般的には，すべての 0-1 変数が 0 または 1 に収束する保証はない．そこで，適当な数値Bを設定し，0 または 1 に収束していない変数の数がB以下になったら，未収束の 0-1 変数に対して分枝限定法を適用する．

容量スケーリング法の流れを示す.

1) $\lambda\in(0,1)$，$\varepsilon\in(0,0.5)$，$MINN$，Bを設定する．
$C_{ij}^{1} = C_{ij}, (i,j)\in A$，$UB := \infty$，$l := 1$とする．
2) $LR(\boldsymbol{C}^l)$を解き，解$\tilde{\boldsymbol{y}}$を求める．
3) $(i,j)\in A$について，

$$y_{ij} = \begin{cases} 0 \ if \ \tilde{y}_{ij} < \varepsilon \\ 1 \ if \ \tilde{y}_{ij} > 1-\varepsilon \\ free \ otherwise \end{cases} \quad (8)$$

とし，$\boldsymbol{y}$の自由変数の数がB以下であれば，
a) $Z < UB$の条件を付加したアークフローによる定式化 *CNDA* を解き，解$\bar{\boldsymbol{y}}$を求める．ここで，Zは限定された *CNDA* の最適値である．
b) Zが存在すれば，$UB := Z$とおく．
4) $l > MINN$かつ$UB \ne \infty$であれば，終了する．
5) $l := l+1$，$C_{ij}^{l} := \lambda C_{ij}^{l-1}y_{ij} + (1-\lambda)C_{ij}^{l-1}, (i,j)\in A$とし，$B$を減少させ，2)へ戻る．

3)a)において，パスフローではなく，アークフローによる定式化を用いる理由は，パスフローによる定式化では最適解に含まれるパスが現在まで生成されていない可能性があるためである．

$LR(\boldsymbol{C}^l)$は線形計画問題ではあるが，指数オーダーのパスフロー変数と強制制約式である(4)式を含むため，直接解くことは容易ではない．そこで，パスフロー変数については列生成，強制制約式については行生成を用いると，効率的に解くことができる[7].

4. 局所分枝

容量スケーリング法で求められた解は必ずしも優れた解とは限らないため，得られた解をもとに局所分枝を用いた分枝限定法を行い，解を改善する．容量スケーリング法で求められたデザイン解を$\bar{\boldsymbol{y}}$としたとき，*CNDA* に次の二式を制約式として追加した *CNDAA* を考える．

$$\sum_{(i,j)\in A|\bar{y}_{ij}=1}(1-y_{ij}) + \sum_{(i,j)\in A|\bar{y}_{ij}=0} y_{ij} \le M \quad (9)$$

$$\sum_{(i,j)\in A|\bar{y}_{ij}=1}(1-y_{ij}) + \sum_{(i,j)\in A|\bar{y}_{ij}=0} y_{ij} \ge 1 \quad (10)$$

(9)式は$\bar{\boldsymbol{y}}$のM近傍の範囲を表し，(10)式は$\bar{\boldsymbol{y}}$を除外

する制約である．計算時間の上限 T の下で *CNDAA* を解くことによって，$\bar{y}$ より良い解 $\bar{y}'$ が見つかれば，$\bar{y} := \bar{y}'$ として，次の制約を付加して，局所分枝を繰り返す．

$$\sum_{(i,j)\in A|\bar{y}_{ij}=1}(1-y_{ij})+\sum_{(i,j)\in A|\bar{y}_{ij}=0} y_{ij} \geq M+1 \quad (11)$$

5. 数値実験

提案した解法の有効性を評価するために，Crainic らのベンチマーク問題[2]を用いて数値実験を行った．使用したコンピュータは，Pentium CoreDuo2 4Core 2.93GHz, RAM16GB, OS Ubuntu10.10，最適化ソルバー CPLEX12.2，言語 C++である．

ベンチマーク問題は，C 問題（簡単な 6 問を除く 37 問）と R 問題（153 問）である．C 問題では，局所分枝法(LBR)[8]，IP 探索法(IP) [9], MIP タブー探索法(MIP)[10]および容量スケーリング法(CAP)と比較した．また，R 問題では，共同並列タブーサーチ法（PARA）[5]，サイクルに基づくタブーサーチ法(CYCL)[4] および容量スケーリング法(CAP) と比較した．上界値の比較のために，CPLEX 上で *CNDA* を最大 72000 秒解くことにより，下界値 *LB* または最適値 *OPT* を求めた．

パラメーラ λ は 0.025～

表 1. C 問題の平均誤差(%)

LBR	IP	MIP	CAP	5-300	5-900	10-300	10-900	15-300	15-900	BEST
3.55	2.13	1.31	1.65	1.12	1.06	0.92	0.88	0.98	0.84	0.79

表 2. C 問題の結果

N/A/K/Type	OPT/LB		LBR	IP	MIP	CAP	CALB		GAP(%)
100/400/010/F/L	23949	*O*	24690	*23949*	24161	24459	*23949*	*	0.00
100/400/010/F/T	62245	*L*	67357	65885	67233	72169	*64607*	**	3.79
100/400/010/V/L	28423	*O*	*28423*	*28423*	*28423*	*28423*	*28423*	*	0.00
100/400/030/F/L	49018	*O*	49872	49694	49682	51956	*49018*	**	0.00
100/400/030/F/T	130852	*L*	141633	141365	144349	144314	*136446*	**	4.28
100/400/030/V/T	384802	*O*	384809	384836	384940	384880	*384802*	**	0.00
20/230/040/V/L	423848	*O*	*423848*	424385	*423848*	*423848*	*423848*	*	0.00
20/230/040/V/T	371475	*O*	*371475*	371779	*371475*	371906	*371475*	*	0.00
20/230/040/F/T	643036	*O*	*643036*	643187	643538	643666	*643036*	*	0.00
20/230/200/V/L	94213	*O*	95295	95097	94218	*94213*	*94213*	**	0.00
20/230/200/F/L	136975	*L*	143446	141253	138491	137851	*137642*	**	0.49
20/230/200/V/T	97914	*O*	98039	99410	98612	97968	*97914*	**	0.00
20/230/200/F/T	134811	*L*	141128	140273	136309	136302	*136031*	**	0.90
20/300/040/V/L	429398	*O*	*429398*	*429398*	*429398*	*429398*	*429398*	*	0.00
20/300/040/F/L	586077	*O*	*586077*	*586077*	588464	587800	*586077*	*	0.00
20/300/040/V/T	464509	*O*	*464509*	*464509*	*464509*	464569	*464509*	*	0.00
20/300/040/F/T	604198	*O*	*604198*	*604198*	*604198*	*604198*	*604198*	*	0.00
20/300/200/V/L	74375	*L*	76375	75319	75045	74913	*74830*	**	0.61
20/300/200/F/L	113144	*L*	119143	117543	116259	115876	*115751*	**	2.30
20/300/200/V/T	74991	*O*	76168	76198	74995	*74991*	*74991*	**	0.00
20/300/200/F/T	105846	*L*	109808	110344	109164	*107467*	*107467*	**	1.53
30/520/100/V/L	53958	*O*	54026	54113	54008	54012	*53958*	**	0.00
30/520/100/F/L	92653	*L*	96255	*94388*	93967	94743	94043		1.50
30/520/100/V/T	52046	*O*	52129	52174	52156	52270	*52046*	**	0.00
30/520/100/F/T	95852	*L*	101102	98883	97490	98867	*97377*	**	1.59
30/520/400/V/L	112422	*L*	114367	114042	112927	112846	*112786*	**	0.32
30/520/400/F/L	147183	*L*	157726	154218	149920	*149446*	*149446*	**	1.54
30/520/400/V/T	114556	*L*	115240	114922	114664	*114641*	*114641*	**	0.07
30/520/400/F/T	150215	*L*	168561	154606	152929	*152745*	*152745*	**	1.68
30/700/100/V/L	47603	*O*	*47603*	47612	*47603*	47614	*47603*	*	0.00
30/700/100/F/L	59321	*L*	60272	60700	60184	60192	*59995*	**	1.14
30/700/100/V/T	45822	*L*	45905	46046	45880	46169	*45875*	**	0.12
30/700/100/F/T	54597	*L*	55104	55609	54926	55339	*54904*	**	0.56
30/700/400/V/L	97001	*L*	103787	98718	97982	*97960*	*97960*	**	0.99
30/700/400/F/L	131233	*L*	169760	152576	135109	*135100*	*135100*	**	2.95
30/700/400/V/T	94333	*L*	96680	96168	95781	*95306*	*95306*	**	1.03
30/700/400/F/T	128027	*L*	144926	131629	130856	*130146*	*130146*	**	1.66

O：最適値，L：下界値，Italic：最良値，**：新たな最良値，*：同値

0.250，分枝限定法基準 B は 75, 局所分枝基準 M は 5, 10，15 とし，局所分枝の 1 回当りの計算時間の上限 T を 300 または 900 秒とした.

表 1 に C 問題に対する平均誤差を示す. 局所分枝法は 3.55%，IP 探索法は 2.13%，MIP タブー探索法は 1.31%であった．一方，容量スケーリング法は 1.65%と MIP タブー探索法に劣るが, 局所分枝法と組合せた解法は 0.79%であり，従来の最良の解法に比べて 0.52%改善することができた. 表 2 に C 問題の結果の詳細を示す．37 問の内，従来の最良値と同値が求められたものが 10 問，新たな最良値が求められたものが 26 問であり，最良値を求められなかったものが 1 問であった．なお，同値の 10 問はすべて最適値である．表 3 に平均計算時間を示す．コンピュータが異なるため，単純に比較することはできないが, 容量スケーリング法は 76.3 秒，局所分枝法と組合せた解法は 500～1500 秒程度となった. 局所分枝法や IP 探索法の最大で 3～4 倍程度，MIP タブー探索法の 1/4 程度である.

表 4 に R 問題に対する平均誤差を示す. 共同並列タブーサーチ法は 6.03%，サイクルに基づくタブーサーチ法は 3.64%であった．一方，容量スケーリング法は 0.65%，局所分枝法と組合せた解法は 0.19%となり，大幅に改善することができた．表 5 に平均計算時間を示す. 容量スケーリング法は 45.7 秒，局所分枝法と組合せた解法は 500～1400 秒程度であり，他の解法と比べて極端に長いことはない.

表 3. C 問題の平均計算時間（秒）

LBR	IP	MIP	CAP	5-300	5-900	10-300	10-900	15-300	15-900
574.6	408.1	6958.6	76.3	503.5	1550.4	644.4	1495.0	514.7	1390.5

表 4. R 問題の平均誤差 (%)

PARA	CYCL	CAP	CALB
6.03	3.64	0.65	0.19

6. おわりに

本研究では，CND に対して容量スケーリング，列生成，行生成に加え，局所分枝を加えた分枝限定法を用いた解法を提案した．数値実験によって，従来の解法と比較し，提案した解法により大半のベンチマーク問題で最良値を求めることができた.

参考文献

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

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

[3] Ghamlouche I., Crainic T.G., Gendreau M., Cycle-based neighborhoods for fixed-charge capacitated multicommodity network design, Operations Research, 2003, 51, 655-667.

[4] Ghamlouche I., Crainic T.G., Gendreau M., Path relinking, cycle-based neighbourhoods and capacitated multicommodity network design, Annals of Operations Research, 2004, 131, 109-134.

[5] Crainic T.G., Li Y., Toulouse M., A first multilevel cooperative algorithm for capacitated multicommodity network design, Computers & Operations Research, 2006, 33, 2602-2622.

[6] Crainic T.G., Gendron B., Hernu G., A slope scaling/ Lagrangean perturbation heuristics with long-term memory for multicommodity capacitated fixed-charge network design, Journal of Heuristics, 2004, 10, 525-545.

[7] Katayama N., Chen M., Kubo M., A capacity scaling heuristics for the multicommodity capacitated network design problem, Journal of Computational and Applied Mathematics, 2009, 232, 90-101.

[8] Rodríguez-Martín I., Salazar-González J.J., A local branching heuristics for the capacitated fixed-charge network design problem, Computers & Operations Research, 2010, 37, 575-581.

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

[10] Chouman M., Crainic T.G., A MIP-Tabu search hybrid framework for multicommodity capacitated fixed-charge network design, CRT-2010-31, CIRRELT, Université de Montréal, 2010.

表 5. R 問題の平均計算時間（秒）

PARA	CYCL	CAP	5-300	5-900	10-300	10-900	15-300	15-900
989.7	1575.2	45.7	539.5	288.8	532.6	211.0	493.6	1390.5