《論　文》

容量制約をもつネットワーク設計問題の高速な貪欲法

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，ノードおよび容量をもつアーク候補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，ネットワークのデザイン費用とフロー費用の合計が最小となるアークの選択と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．

ネットワーク設計問題に対するサーベイとして，Balakrishnan et al.（1997），Costa（2005），Crainic（2003），Gendron et al.（1997），Magnanti and Wong（1984），Minoux（1989），Wong（1984, 1985），Yaghini and Rahbar（2012）がある．また，*CND* はNP-困難な問題であることが知られている（Magnanti and Wong 1984）．このため，妥当不等式，緩和法，ヒューリスティクスやメタヒューリスティクスなど多く技法が開発されてきた．

多面体解析として，Magnanti et al.（1993）は整数丸め込みカット集合，３分割不等式および剰余容量不等式を提案している．また，Barahona（1996）は多重カット不等式を示し，Bienstock and Günlük（1996）はフローカット集合や３分割ファセットを示している．Atamtürk and Rajan（2002）は，単一アーク集合とそれに関連する分離問題と持ち上げ問題を示している．Chouman et al.（2003）はカバー不等式と最小基数不等式を示し，Kliewer and Timajev（2005）はカバー不等式と局所カットを示している．また，Costa et al.（2009）はBenders不等式，メトリック不等式およびカット集合不等式を示している．また，Chouman et al.（2009）はフローカバー不等式とフローパック不等式と，それらを生成するための効率的な分離問題を提案している．

CND に対する緩和問題と下界値を求める手法が数多く提案されている．片山 and 春日井（1993）はカットセットに対する整数丸め込み不等式を用いた双対上昇法を提案し，Gendron and Crainic（1994, 1996）は線形緩和問題とラグランジュ緩和問題を示している．また，Herrmann et al.（1996）は容量制約のないネットワーク設計問題に対する双対上昇法を拡張しており，Holmberg and Yuan（2000）はラグランジュ緩和問題と分枝限定法を組合せた解法を提案している．一方，Crainic et al.（2001）はバンドルタイプの劣勾配法を適用している．

ヒューリスティクスやメタヒューリスティクスといった適切な時間内に適切な解を求める近似解法が数多く開発されている．Gendron and Crainic（1994, 1996）は資源主導型分解法に基づく解法を提案している．また，タブサーチ法を用いた解法が数多く開発され，Crainic et al.（2000）や Zaleta and Socarrás（2004）は単体に基づいたタブサーチ法，Ghamlouche et al.（2003）はサイクルに基づいたタブサーチ法，Crainic et al.（2006）は多重レベル共同探索法を提案している．また，Ghamlouche et al.（2004），Alvarez et al.（2005）や Crainic and Gendreau（2007）は散布探索法とパス再結合法を適用している．一方，Crainic et al.（2004）はフロー費用を変化させる傾斜スケーリング法を開発している．

近年では，メタヒューリスティクスと最適化ソルバーを組合せた解法が主流である．Katayama et al.（2009）は容量スケーリング法と列生成法・行生成法を組合せた解法を開発し，Rodríguez-Martín and Salazar-González（2010）は最適化ソルバーを用いた局所分枝法を提案している．また，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解法を開発している．Ghamlouche et al.（2011）は学習メカニズムと局所探索を用いた解法を示している．一方，Yaghini et al.（2011, 2013）は単体法と列生成法を用いたシミュレーテッドアニーリング法を提案し，Paraskevopoulos et al.（2013）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）は蟻コロニーアルゴリズムを適用し，Katayama（2015）は容量スケーリング法と局所分枝法を用いた解法を提案している．

メタヒューリスティクスに代表される近年の研究では，精度の高い解の算出が議論の中心となっている．これらの解法においては，精度の高い解の算出するために多くの計算時間を必要としている．しかし，実用的な問題ではノード数，アーク数や品種数が膨大なものとなることから，適度な精度の解を算出する高速な解法の開発が必要となっている．さらには，様々なシナリオに柔軟に対応できるネットワーク設計を目的とするロバスト性を考慮したネットワーク設計問題や不確実性を考慮したネットワーク設計問題では，多くのシナリオやデータに対する膨大な数の基本的なネットワーク設計問題を繰り返し解くことが必要となることから，基本的な問題に対して短時間で解を算出できる高速な解法の開発が望まれている．

容量制約のないネットワーク設計問題に対する高速解法として，貪欲法であるデリー

ト法が知られており, Minoux（1989）はこの貪欲法を改良した解法を示し，片山（2010）は Minoux の貪欲法を改良した解法を提案している．容量制約をもつネットワーク設計問題に対して，片山（2001）は Minoux の貪欲法を改良した解法を提案している．この貪欲法では，その子問題である多品種フロー問題を繰り返し解くことが必要となる．この解法が提案された時期には数理最適化ソルバーは一般的なものではなかったため，多品種フロー問題を解くためにはラグランジュ緩和問題を劣勾配法で解くといった計算量を必要とする手法を用いる必要があった．しかし，今日では数理最適化ソルバーにより，多品種フロー問題は高速に解けるようになっている．このため，貪欲法と数理最適化ソルバーを組合せることによって，高速な近似解法が開発できる可能性がでてきている．

本研究では，*CND* に対して高速な貪欲法を用いた近似解法を提案し，さらには容量スケーリング法や限定した分枝限定法を組合せた高速な近似解法を提案する．また，数値実験を行い，提案した近似解法により高精度の近似解を短時間で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，品種 k の取り得るパス集合を P^k とする．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} とし，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．また，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，パスフロー集合 P に対してパスフロー変数を用いた *CND* の定式化 *CNDP* (A, P) は，次のように表される．

（*CNDP* (A, P)）

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

subject to

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

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

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

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

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

(1)式は，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．なお，$\omega\ (A, P)$ はアーク集合 A とパス集合 P が与えられたときの，*CND* の目的関数値である．(2)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．(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)式はパスフローの非負制約，(6)式はデザイン変数の 0−1 条件である．

CND には，2 種類の変数，デザイン変数 y とフロー変数 x が含まれている．そこで，デザイン変数 y が 1 となるアークを決定する問題，すなわち全体のアーク候補集合から選択するアーク集合を決定する問題と，選択アーク集合が決定したときにフローを求める多品種フロー問題の 2 段階の問題として，*CND* を考える．すべてのデザイン変数が固定された *CND* を考える．このとき，$y_{ij} = 1$ である選択されたアーク集合を $\bar{A}$ とする．選択したアーク集合が $\bar{A}$ である問題は，$\bar{A}$ からなるネットワーク上のフロー変数からなる多品種フロー問題を解くことにより，フローと全体の費用を求めることができる．

ここで，アーク集合 A の部分集合 $\bar{A}$ に対する多品種フロー問題を $MCF(\bar{A})$ とし，$MCF(\bar{A})$ の最適目的関数値であるフロー費用と $\bar{A}$ に含まれるアークのデザイン費用の和を $\phi(\bar{A})$ とおく．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 とし，品種 k の始点を O^k，終点を D^k とする．また，$Nn^+(\bar{A})$ はアーク $\bar{A}$ 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，$Nn^-(\bar{A})$ はノード n に入るアーク集合である．

このとき，*CND* 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 段階の問題と見なすことができる．このため，アークフロー変数を用いた *CND* の定式化 *CNDA* は，次のように表される．

（*CNDA*）

$$\min_{\forall \bar{A} \subseteq A} \quad \phi(\bar{A}) \tag{7}$$

subject to

（$MCF(\bar{A})$）

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

subject to

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

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

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

(7)式は，選択されたアーク集合 $\bar{A}$ に対する目的関数値であり，これを最小化することを表す．(8)式は，選択されたアーク集合 $\bar{A}$ に対するフロー費用と $\bar{A}$ に含まれるアークのデザイン費用の和であり，これを最小化することを表す．(9)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．(10)式の左辺はアーク (i, j) 上のフロー量の合計であり，これが容量 C_{ij} 以下となる容量制約式である．(11)式はアークフロー変数の上限と非負制約である．

3　貪欲法

3．1　貪欲法と改良貪欲法

容量制約をもたないネットワーク設計問題に対する基本的な解法として，目的関数が最も減少するであろう順にアークを削除していく貪欲法（Minoux 1989）があり，デリート法やフォワード法などとよばれている．容量制約をもつネットワーク設計問題である *CND* に対しても，同様の貪欲法を適用することができる．*CND* に対する貪欲法のアルゴリズムを Algorithm1 に示す．ψ_{ij} はアーク (i, j) を削除したときの目的関数値の減少量，π は $\bar{A}$ に含まれるアークの中で目的関数値の減少量の最大値であり，(i^*, j^*) は減少量が最大であるアークである．

この貪欲法では，特定の $\bar{A}$ からなるネットワーク上で削除するアークを求めるために，すべての $(i, j) \in \bar{A}$ に対して，$\bar{A} \setminus \{(i, j)\}$ 上のネットワークにおける $MCF(A \setminus \{(i, j)\})$ を解き，アーク (i, j) を削除したときの目的関数値の減少量を求めることが必要となる．続いて，減少量が最大であるアークを $\bar{A}$ から削除する．アークを削除するたびに，$\bar{A}$ に含まれるすべてのアークに対してこの計算を繰り返し行う必要がある．したがって，全体として $O(|A|^2)$ 回もの多品種フロー問題を解く必要があり，その計算量は膨大なものとなる．

容量制約をもたないネットワーク設計問題に対して，それぞれのアークを削除したと

きの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，Minoux（1989）は適時，近似的に求める貪欲法を提案している．この解法では，繰り返しのたびに目的関数値の減少量を計算するのではなく，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する．ここで，近似値は削除したアークの両端間の最短経路上にフローを流しかえたときの目的関数値の減少量とする。これはアークの両端間の最短経路問題を解くことにより求めることができる。そして，減少量が最大のアークを選択し，このアークに対する減少量を近似的に再計算し，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減少量の更新だけを行う方法である．減少量を近似的に算出することに加え，目的関数値の減少量の算出回数を大幅に削減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設計問題では効率的な解法となっている．

Minoux 法は高速であるものの，減少量を近似的に算出するため，得られる解の精度が良くないことが知られている．そこで，片山直登（2010）は，Minoux 法において減少量を近似的ではなく，効率的でかつ厳密に求める改良法を示しており，計算時間は増加するものの高精度解の解を算出することに成功している．

一方，容量制約をもつネットワーク設計問題に対しては，片山直登（2001）が Minoux 法を改良した解法を提案している．この研究では，多品種フロー問題を劣勾配法で解いているために，非常に多くの計算時間を必要とし，大規模な問題に対しては，必ずしも実用的でないものとなっている．近年，最適化問題を解くためのソフトウエアーである最適化ソルバーが普及している．最適化ソルバーにより，線形計画問題である多品種フロー問題は高速に解くことができるため，最適化ソルバーと貪欲法を組合せることによって，大規模な問題を高速に解けることが期待できる．

容量制約をもつネットワーク設計問題に対する Minoux 法を改良した方法を改良貪欲法とよぶことにする．このアルゴリズムを Algorithm2 に示す．ψ_{ij} はアーク (i, j) を削除したときの目的関数値の減少量であり，L は減少量が正であるアークの集合である．また，π は L に含まれるアークの中で目的関数値の減少量の最大値，(i^*, j^*) は ψ_{ij} が最大であるアークである．アーク (i^*, j^*) を L から取り出した後，$\psi_{i^*j^*}$ を再計算する．この値が L に含まれるアークの減少量の中で最大値以上であればアーク (i^*, j^*) を削除し，そうでなければ L に戻す．

この解法では，アーク集合 A における目的関数値の減少量の初期の計算回数は $O(|A|)$ であり，それ以降はアーク (i, j) が実際に削除の対象となる場合に限り目的関数値の減少量を計算するだけである．このため，多品種フロー問題を解く回数は最悪の場合には $O(|A|^2)$ であるが，経験的には $O(|A|)$ となる．

3．2　容量スケーリング法と限定した分枝限定法

改良貪欲法は，効率的に近似解を算出できる可能性がある．この解法では，ネットワークから特定のアークを削除したときの目的関数値の減少量を基準とし，減少量の大きい順にアークを削除していくことが手順となる．このため，大規模なネットワーク上で品種数や需要量の少ない問題において，アーク集合 A からなるネットワーク上で多品種フロー問題を解いた場合には，大半のアーク上のフロー量は 0 となることが予想される．フロー量が 0 であるアークを削除しても，目的関数の減少量はそのアークのデザイン費用のみであることから，フロー量が 0 でかつデザイン費用の高いアークから順に削除されることになる．このため，これらのアークが最適解に含まれる場合では，得られる近似解の精度が大きく悪化する可能性がある．一方，改良貪欲法が効率的であるとは言っても，アーク数の多い問題では非常に多くの 多品種フロー問題を繰り返し解く必要がある．これらの問題を解決するために，初期ネットワークに対して容量スケーリング法を適用し，事前に対象となるアークを絞ることにする．

容量スケーリング法は，線形緩和問題を解き，そのデザイン変数解の値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．容量スケーリングを行うことにより，パスフローが分散し，かつ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 2009）．そこで，アーク集合 A に対して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し，これらのアークのみを改良貪欲法の対象にする．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ることになり，効率的に問題規模を縮小することが可能となる．もちろん，0 に収束したデザイン変数が最適解において 1 である可能があることに注意する．なお，すべての変数が 0 または 1 に収束するためには多くの計算時間が必要なため，0 に収束しないデザイン変数が一定数以下となったら容量スケーリングを終了し，これらのデザイン変数に対応するアークを，改良貪欲法の対象となるアーク候補集合に限定する．

CNDP (A, P) において，(3)式にあるアーク容量を C_{ij}^{l} とし，(6)式の上限である 1 を C_{ij}/C_{ij}^{l}，0 を下限に置き換えた線形緩和問題を *CNDPL* (A, C^{l}) とする．なお，l は容量スケーリングの繰り返し回数である．

（*CNDPL* (A, C^{l})）

$$\min \sum_{(i,j)\in A}\sum_{k\in K} c_{ij}^{k}\sum_{p\in P^{k}}\delta_{ij}^{p}x_{p}^{k}+\sum_{(i,j)\in A} f_{ij}y_{ij} \tag{12}$$

subject to

$$\sum_{p\in P^{k}} x_{p}^{k}=d^{k} \quad \forall k\in K \tag{13}$$

$$\sum_{k\in K}\sum_{p\in P^{k}}\delta_{ij}^{p}x_{p}^{k}\le C_{ij}^{l}y_{ij} \quad \forall (i,j)\in A \tag{14}$$

$$\sum_{p \in P^k} \delta_{ij}^p x_p^k \leq d^k y_{ij} \quad \forall (i,j) \in A, k \in K \tag{15}$$

$$x_p^k \geq 0 \quad \forall p \in P^k, k \in K \tag{16}$$

$$y_{ij} \leq C_{ij}/C_{ij}^l \quad \forall (i,j) \in A \tag{17}$$

$l-1$ 回目の繰り返しである *CNDPL* (A, C^{l-1}) のデザイン変数解を $\tilde{y}$ とすると，l 回目のアーク容量を次のように変更する．

$$C_{ij}^l := \lambda C_{ij}^{l-1} \tilde{y}_{ij} + (1-\lambda) C_{ij}^{l-1} \tag{18}$$

ここで，λ はスケーリングパラメータであり，0 から 1 の間の定数である．

容量スケーリング法のアルゴリズムを Algorithm3 に示す．ϵ は収束判定基準，ITE_{min} は容量スケールの最小繰り返し回数，ITE_{max} は容量スケールの最大繰り返し回数である．また，*ArcNum* は収束していないアーク数の上限値であり，収束していないアーク数が *ArcNum* 以下となった場合，容量スケーリングを終了する．なお，パスフローを用いた定式化では，必要なパスを生成する列生成法と必要な強制制約式を生成する行生成法を用いる．容量スケーリング法ならび列生成法と行生成法の詳細については，Katayama（2009）を参照のこと．

容量スケーリング法を適用した結果，容量スケーリングをしたネットワーク上におけるパスフロー変数および対応する強制制約が生成される．生成されたパスの集合を $\bar{P}$ とおき，0 に収束しないデザイン変数に対するアーク集合を $\bar{A}$ とする．このとき，アーク集合 $\bar{A}$ とパス集合 $\bar{P}$ で構成される問題は *CNDP* $(\bar{A}, \bar{P})$ となる．$\bar{A}$ とパス集合 $\bar{P}$ は，それぞれ A と P の部分集合であるため，*CNDP* $(\bar{A}, \bar{P})$ の最適解は *CNDP* (A, P) の近似解，最適値は *CNDP* (A, P) の上界値となる．

CNDP $(\bar{A}, \bar{P})$ を最適または近似的に解くことによって，*CNDP* (A, P) の近似解を求めることができる．本来の *CNDP* (A, P) に比べると，*CNDP* $(\bar{A}, \bar{P})$ の規模は相対的に小さいため，最適化ソルバーの分枝限定法を用いると短時間で *CNDP* $(\bar{A}, \bar{P})$ の最適解を算出できる可能性がある．もちろん，短時間で最適解を算出できる保証はないため，計算時間の上限 BBTime を設定して，*CNDP* $(\bar{A}, \bar{P})$ を解くことにする．ここでは，この方法を限定した分枝限定法とよぶ．

CNDP $(\bar{A}, \bar{P})$ はパス変数が限定されているため，$\bar{A}$ に対する厳密な問題 *CNDP* $(\bar{A}, P)$ と比べると，最適解に含まれるすべてのパスが含まれていない可能性があり，*CNDP* $(\bar{A}, \bar{P})$ の目的関数値は *CNDP* $(\bar{A}, P)$ の目的関数値よりも大きな値となる．そこで，*CNDP* $(\bar{A}, \bar{P})$ で得られたデザイン解で $y_{ij} = 1$ である集合を $\tilde{A}$ とし，$\tilde{A}$ を用いた *MCF* $(\tilde{A})$ を解くことによって ϕ $(\tilde{A})$ を求め，の目的関数値を算出する．これは，

CNDP ($\bar{A}$, $\bar{P}$) ではパス変数が限定されており，*MCF* ($\bar{A}$) の方がより良い目的関数値を得られる可能性があるためである．

また，スケーリングを行う以前の容量が $C^l = C$ である線形緩和解において，正となるデザイン変数の数が少数で *ArcNum* 以下である場合には，これらのデザイン変数をもつアークも改良貪欲法の限定アーク集合に含め，限定分枝限定法の対象となるアークの範囲を増加させることにする．

全体のアルゴリズムを Algorithm4 に示す．始めに容量スケーリング法でパスを生成しかつ対象アークを限定し，容量スケーリング法により限定されたアーク集合とパス集合を用いて限定された分枝限定法により近似解を算出する．続いて，改良貪欲法を用いて近似解を算出する．なお，容量スケーリング法および限定された分枝限定法にはパスフローを用いた定式化 *CNDP* を用い，改良貪欲法にはアークフローを用いた定式化 *MCF* を用いることに注意する．アークフローを用いた定式化 *MCF* を用いるのは，フロー問題のみを解く場合には，アークフローを用いた定式化がより高速に解けるためである．

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主な実験条件は以下の通りである．

- 使用 OS および言語：UBUNTU 14.10，C++
- 最適化ソルバー：Gurobi 6.05
- CPU INTEL i7 3774 3.4GHz 4Core，RAM 32GByte
- 使用コア数：線形計画問題 1 コア，分枝限定法 4 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：0.025～0.250
- スケーリング法の終了判定アーク数 ArcNum：50，100，150，200
- 限定した分枝限定法の計算時間の上限 BBTime：10 秒

近似解の誤差を算出するために，アークフローによる定式化を数理計画ソルバー CPLEX（最大30時間）により解くことにより得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，2010年以降の主な研究である Rodriguez の局所分枝法（Rodríguez-Martín and Salazar-González 2010），Hewitt の IP 探索法（Hewitt et al. 2010），Chouman の MIP タブー探索法（Chouman and Crainic 2010），ハイブリッドシミュレーテッドアニーリング

法（Yaghini et al. 2013），ならびに Katayama の容量スケーリング法および容量スケーリング・局所分枝法（Katayama 2015）の結果を併記した．R 問題に対して，詳細な解を公開している研究は少ないため，サイクルに基づいたタブサーチ法（Ghamlouche 2003），および Katayama の容量スケーリング法および容量スケーリング・局所分枝法の結果を併記した．

表 1 に C 問題に対する近似解の平均誤差を示す．IPS は IP 探索法，MIP は MIP タブー探索法，HSA はハイブリッドシミュレーテッドアニーリング法，CS は容量スケーリング法，CPLB は容量スケーリング・局所分枝法による結果である．なお，HSA における問題30/520/400/FT の解は誤りのため集計から除いている．一方，提案した解法である MG は改良貪欲法のみの解法，MGCS は容量スケーリング法と改良貪欲法の組合せ，RBB は容量スケーリング法，改良貪欲法および限定された分枝限定法の組合せである．なお，MGCS におけるパラメータは $\lambda = 0.200$，ArcNum = 150，RBB におけるパラメータは $\lambda = 0.200$，ArcNum = 200 とした．また，BEST は MGCS および RBB においてパラメータを変更した中での最良値である．

従来の研究では，IP 探索法が誤差1.86%，MIP タブー探索法および容量スケーリング法が誤差1.04%，ハイブリッドシミュレーテッドアニーリング法が誤差0.91%，容量スケーリング・局所分枝法が誤差0.43%となっている．一方，提案した解法では，改良貪欲法が誤差4.48%と大きいが，容量スケーリング法との組合せでは1.71%，限定された分枝限定法との組合せでは1.17%，最良値では1.03%であった．従来の解法である容量スケーリング・局所分枝法と比べると 3 倍程度の誤差があるが，IP 探索法よりも優れた解を算出し，MIP タブー探索法と同程度の解を算出している．

表 2 に，C 問題に対する個別の目的関数値を示す．太字は最適値，斜体文字は従来の最良値である．最良値は容量スケーリング・局所分枝法のものである．提案した解法の MGCS および RBB では37問の内 1 問，BEST では 6 問の最適値を算出している．提案した解法が最適値を算出できる数が少ないのは，高精度よりも高速性を追求していることが要因である．

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．平均誤差の最も小さい容量スケーリング・局所分枝法は，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．また，IP 探索法が406.1秒，MIP タブー探索法が6958.6秒，ハイブリッドシミュレーテッドアニーリングが4587.2秒である．最も短時間である容量スケーリング法では262.4秒である．一般的に，メタヒューリスティクスは時間制限により終了させることが多いため，良い解を算出するために計算時間が大きくなっている．また，CS である容量スケーリング法には，容量スケーリング法に加え，パスフローおよびアーク

フロー用いた定式化に対する限定された分枝限定法を含んでいるために，計算時間が大きくなっている．一方，提案した解法では，改良貪欲法が51.2秒，容量スケーリング法との組合せでは29.2秒，限定された分枝限定法との組合せでは42.5秒，RBB における最良値（計算時間が最小）では23.5秒であった．最良値の計算時間は小さいが，適切なパラメータを選択する必要があるので，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．なお，このときの平均誤差は1.95%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，従来の研究に比べ，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．提案した解法の計算時間は従来の解法に比べて圧倒的に短いことが分かる．MG では最大が373.4秒，MGCS では230.2秒，RBB では268.2秒，最良値では161.4秒となっている．

表 5 に R 問題に対する近似解の平均誤差を示す．CYC はサイクルに基づいたタブサーチ法である．なお，MGCS におけるパラメータは $\lambda = 0.150$，ArcNum = 200，RBB におけるパラメータは $\lambda = 0.200$，ArcNum = 200とした．

従来の研究では，サイクルに基づいたタブサーチ法が誤差5.74%，容量スケーリング法が誤差0.61%，容量スケーリング・局所分枝法が誤差0.17%となっている．一方，提案した解法では，改良貪欲法が誤差4.35%と大きいが，容量スケーリング法との組合せでは1.42%，限定された分枝限定法との組合せでは0.86%，最良値では0.67%であった．従来の解法である容量スケーリング・局所分枝法と比べると大きな誤差ではあるが，サイクルに基づいたタブサーチ法よりも優れた解を算出している．

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．太字は最適値，斜体文字は従来の最良値である．最良値は容量スケーリング・局所分枝法のものであり，提案した解法の MGCS では81問の内 5 問，RBB では11問，BEST では14問の最適値を算出している．提案した解法が最適値を算出できる数が少ないのは，高精度よりも高速性を追求していることが要因である．

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容量スケーリング・局所分枝法は，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．また，サイクルに基づいたタブサーチ法が2933.8秒であり，最も短時間である容量スケーリング法では119.1秒である．ここでも，メタヒューリスティクスの計算時間が大きくなっている．一方，提案した解法では，改良貪欲法が9.1秒，容量スケーリング法との組合せでは13.2秒，限定された分枝限定法との組合せでは18.4秒，RBB における最良値（計算時間が最小）では10.1秒であった．最良値の計算時間は小さいが，適切なパラメータを選択する必要があるので，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．なお，このときの平均誤差は2.12%である．提案した

解法では，従来の研究に比べ，大幅に計算時間が短縮できることが分かる．

表 9 および10に個別の問題の計算時間を示す．提案した解法の計算時間は従来の解法に比べて圧倒的に短いことが分かる．MG では最大が108.9秒，MGCS では112.9秒，RBB では133.4秒，最良値では125.1秒となっている．

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高速な貪欲法を提案し，さらに容量スケーリング法ならびに限定された分枝限定法を組合せた解法を提案した．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容量スケーリング・局所分枝法と比較すると誤差は大きいが，他の解法と同程度の誤差の解を求めることができた．また，計算時間は，従来の解法と比べて大きく削減することができた．

今後は，ロバスト性を考慮したネットワーク設計問題や不確実性を考慮したネットワーク設計問題に対して，提案した高速解法を適用することが課題である．

本研究は科学研究費基盤研究 C（課題番号25350454）による成果の一部である．

参考文献

A. M. Alvarez, J. L. González-Velarde, and K. De-Alba. Scatter search for network design problem. *Annals of Operations Research*, 138(1): 159-178, 2005.

A. Atamtürk and D. Rajan. On splittable and unsplittable flow capacitated network design arc-set polyhedra. *Mathematical Programming*, 92(2): 315-333, 2002.

A. Balakrishnan, T. L. Magnanti, and P. Mirchandani. Network design. In M. Dell'Amico, F. Maffioli, and S. Martello, editors, *Annotated Bibliographies in Combinatorial Optimization*, pages 311-334. John Wiley & Sons, New York, 1997.

F. Barahona. Network design using cut inequalities. *SIAM journal on Computing*, 6: 823-837, 1996.

D. Bienstock and O. Günlük. Capacitated network design – polyhedral structure and computation. *INFORMS journal on Computing*, 8: 243-259, 1996.

M. Chouman and T. G. Crainic. A MIP-tabu search hybrid framework for multicommodity capacitated fixed-charge network design. Technical Report CIRRELT-2010-31, Centre de recherche sur les transports, Université de Montréal, 2010.

M. Chouman, T. G. Crainic, and B. Gendron. A cutting-plane algorithm based on cutset inequalities for multicommodity capacitated fixed charge network design. Technical

Report CIRRELT-2003-16, Centre de recherche sur les transports, Université de Montréal, 2003.

M. Chouman, T. G. Crainic, and B. Gendron. A cutting-plane algorithm for multicommodity capacitated fixed-charge network design. Technical Report CIRRELT-2009-20, Centre de recherche sur les transports, Université de Montréal, 2009.

A. M. Costa. A survey on benders decomposition applied to fixed-charge network design problems. *Computers and Operations Research*, 32: 1429-1450, 2005.

A. M. Costa, J. F. Cordeau, and B. Gendron. Benders, metric and cutset inequalities for multicommodity capacitated network design. *Computational Optimization and Applications*, 42: 371-392, 2009.

T. G. Crainic. Long-haul freight transportation. In R. W. Hall, editor, *Handbook of Transportation Science*, pages 451-516. Kluwer Academic Publishers, 2003.

T. G. Crainic and M. Gendreau. A scatter search heuristic for the fixed-charge capacitated network design problem. In K. F. Doerner, M. Gendreau, P. Greistorfer, W. Gutjahr, R. F. Hartl, and M. Reimann, editors, *Metaheuristics*, pages 25-40. Springer, 2007.

T. G. Crainic, M. Gendreau, and J. M. Farvolden. A simplex-based tabu search for capacitated network design. *INFORMS journal on Computing*, 12: 223-236, 2000.

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

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. *Journal of Heuristics*, 10: 525-545, 2004.

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

B. Gendron and T. G. Crainic. Relaxations for multicommodity capacitated network design problems. Technical Report CIRRELT-965, Centre de recherche sur les transports, Université de Montréal, 1994.

B. Gendron and T. G. Crainic. Bounding procedures for multicommodity capacitated fixed charge network design problems. Technical Report CIRRELT-96-06, Centre de recherche sur les transports, Université de Montréal, 1996.

B. Gendron, T. G. Crainic, and A. Frangioni. Multicommodity capacitated network design. Technical Report CIRRELT-98-14, Centre de recherche sur les transports, Université de Montréal, 1997.

I. Ghamlouche, T. G. Crainic, and M. Gendreau. Cycle-based neighbourhoods for fixed-charge

capacitated multicommodity network design. *Operations Research*, 51: 655-667, 2003.

I. Ghamlouche, T. G. Crainic, and M. Gendreau. Path relinking, cycle-based neighborhoods and capacitated multicommodity network design. *Annals of Operations Research*, 131: 109-134, 2004.

I. Ghamlouche, T. G. Cainic, and M. Gendreau. Learning mechanisms and local search heuristics for the fixed charge capacitated multicommodity network design. *International Journal of Computer Science Issues*, 8(6): 21-32, 2011.

J. W. Herrmann, G. Ioannou, I. Minis, and J. M. Proth. A dual ascent approach to the fixed-charge capacitated network design problem. *European journal of Operational Research*, 95: 476-490, 1996.

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.

K. Holmberg and D. Yuan. A Lagrangian heuristic based branch-and-bound approach for the capacitated network design problem. *Operations Research*, 48: 461-481, 2000.

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

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.

G. Kliewer and L. Timajev. Relax-and-cut for capacitated network design. In G. S. Brodal and S. Leonardi, editors, *Algorithms - ESA 2005: Lecture Notes in Computer Science*, pages 47-58. Springer, 2005.

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

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

T. L. Magnanti, P. Mirchandani, and R. Vachani. The convex hull of two core capacitated network design problems. *Mathematical Programming*, 60: 233-250, 1993.

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

DC. Paraskevopoulos, T. Bektas, T. Crainic, and CN. Potts. A cycle-based evolutionary algorithm for the fixed-charge capacitated multi-commodity network design problem. Technical Report CIRRELT-2013-08, Centre de recherche sur les transports, Université

de Montréal, 2013.

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.

R. T. Wong. Introduction and recent advances in network design: Models and algorithms. In M. Florian, editor, *Transportation Planning Models*, pages 187-225. Elsevier Science, North Holland, Amsterdam, 1984.

R. T. Wong. Location and network design. In M. O'heEigeartaigh, J. Lenstra, and A. RinnooyKan, editors, *Combinatorial Optimization Annotated Bibliographies*, pages 129-147. John Wiley & Sons, New York, 1985.

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

M. Yaghini, M. Karimi, and M. Rahbar. A hybrid simulated annealing and simplex method for fixed-cost capacitated multicommodity network design. *International Journal of Applied Metaheuristic Computing*, 2(4): 13-28, 2011.

M. Yaghini, M. Rahbar, and M. Karimiand. A hybrid simulated annealing and column generation approach for capacitated multicommodity network design. *Journal of the Operational Research Society*, 64: 1010-1020, 2013.

N. C. Zaleta and A. M. A. Socarrás. Tabu search-based algorithm for capacitated multicommodity network design problems. In *14th International Conference on Electronics, Communications and Computers*, pages 144-148, 2004.

片山直登．予算制約をもつ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紀要，5: 29-40. 2001.

片山直登．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，2010.

片山直登 and 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，44: 164-175, 1993.

Algorithm 1: Greedy Algorithm

$\bar{A} \leftarrow A$;
Solve $MCF(\bar{A})$;
Get $\phi(\bar{A})$;
repeat
  for $(i,j) \in \bar{A}$ **do**
    Solve $MCF(\bar{A}\backslash\{(i,j)\})$;
    if $MCF(\bar{A}\backslash\{(i,j)\})$ *is feasible* **then**
      Get $\phi(\bar{A}\backslash\{(i,j)\})$;
      $\psi_{ij} \leftarrow \phi(\bar{A}) - \phi(\bar{A}\backslash\{(i,j)\})$;
    else
      $\psi_{ij} \leftarrow -\infty$;
    end
  end
  $\pi \leftarrow \max_{(i,j)\in\bar{A}} \psi_{ij}$;
  if $\pi > 0$ **then**
    $(i^*, j^*) \leftarrow \arg\max_{(i,j)\in\bar{A}} \psi_{ij}$;
    $\bar{A} \leftarrow \bar{A}\backslash\{(i^*, j^*)\}$;
  end
until $\pi \leq 0$
Get $\phi(\bar{A})$;
$UB \leftarrow \phi(\bar{A})$;
Return $\bar{A}$, UB;

Algorithm 2: Modified Greedy Algorithm(A)

$\bar{A} \leftarrow A$;
$L \leftarrow ()$;
Solve $MCF(\bar{A})$;
Get $\phi(\bar{A})$;
for $(i,j) \in \bar{A}$ **do**
  Solve $MCF(\bar{A}\backslash\{(i,j)\})$;
  if $MCF(\bar{A}\backslash\{(i,j)\})$ *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**
      $L.push((i,j))$;
    end
  end
end
repeat
  $\pi \leftarrow \max_{(i,j)\in L} \psi_{ij}$;
  $(i^*,j^*) \leftarrow \arg\max_{(i,j)\in L} \psi_{ij}$;
  $L.pop((i^*,j^*))$;
  Solve $MCF(\bar{A}\backslash\{(i^*,j^*)\})$;
  if $MCF(\bar{A}\backslash\{(i^*,j^*)\})$ *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
      $L.push((i^*,j^*))$;
    end
  end
until $\pi \leq 0$ *or* $|L| = 0$
Get $\phi(\bar{A})$;
$UB_{MG} \leftarrow \phi(\bar{A})$;
Return $\bar{A}$, UB_{MG};

Algorithm 3: Capacity Scaling(A)

Set $\bar{P}$, λ, ϵ, ITE_{min}, ITE_{max}, $ArcNum$;
Solve $CNDPL(A, C)$);
Get the solution $\tilde{y}$ of $CNDPL(A, C)$;
Add paths to $\bar{P}$ by Column Genaration;
$\hat{A} \leftarrow ()$;
Get AN which is the number of arcs such that $y_{ij} > \epsilon$;
if $AN < ArcNum$ **then**
  for $(i, j) \in A$ **do**
    if $\tilde{y}_{ij} > \epsilon$ **then**
      $\hat{A}.push((i, j))$;
    end
  end
end
$C^1 := C$; $l := 1$;
repeat
  Solve $CNDPL(A, C^l)$;
  Get the solution $\tilde{y}$ of $(CNDPL(A, C^l))$;
  Add paths to $\bar{P}$ by Column Genaration;
  $\bar{A} \leftarrow ()$;
  for $(i, j) \in A$ **do**
    $C^l_{ij} := \lambda C^{l-1}_{ij} \tilde{y}_{ij} + (1 - \lambda) C^{l-1}_{ij}$;
    if $\tilde{y}_{ij} > \epsilon$ **then**
      $\bar{A}.push((i, j))$;
      $\hat{A}.push((i, j))$;
    end
  end
until $l \geq ITE_{min}$ *and* $|\bar{A}| \leq ArcNum$, *or* $l \geq ITE_{max}$
Return $\bar{A}$,$\hat{A}$,$\bar{P}$;

Algorithm 4: Combined Algorithm

Set A;
$\bar{A}$,$\hat{A}$,$\bar{P}$ = Capacity Scaling(A);
Solve $CNDP(\hat{A}, \bar{P})$ and get $\tilde{A}$ which is the selected arc set;
Get $\phi(\tilde{A})$;
$UB_{BB} = \phi(\tilde{A})$;
UB_{MG} = Modified Greedy Algorithm($\bar{A}$);
$\bar{A}, UB = \min(UB_{BB}, UB_{MG})$;
Return $\bar{A}$, UB;

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

IPS	MIP	HSA	CS	CSLB	MG	MGCS	RBB	BEST
1.86	1.04	0.91*	1.04	0.43	4.48	1.71	1.17	1.03

*:exculde 30/520/400/FT

表 2 ：Results for C-Category Problems

N/A/K/FC	IPS	MIP	HSA	CS	CSLB	MG	MGCS	RBB	BEST
100/400/010/VL	**28423**	**28423**	**28423**	**28423.0**	**28423.0**	28615.0	28615.0	28599.0	28599.0
100/400/010/FL	**23949**	24161	23949	24459.0	**23949.0**	31041.0	24492.0	24022.0	**23949.0**
100/400/010/FT	65885	67233	65172	70903.0	*63753.0*	77785.0	73494.0	70258.0	70258.0
100/400/030/VT	384836	384940	**384802**	384809.0	**384802.0**	385139.0	384809.0	384809.0	384809.0
100/400/030/FL	49694	49682	49250	49319.0	**49018.0**	64882.0	51552.0	49632.0	49632.0
100/400/030/FT	141365	144349	141014	142122.0	*136803.0*	146116.0	144888.0	141356.0	140964.0
20/230/040/VL	424385	**423848**	**423848**	**423848.0**	**423848.0**	426841.0	424697.0	424470.0	424470.0
20/230/040/VT	371779	**371475**	**371475**	371906.0	**371475.0**	371978.0	371642.0	371642.0	**371475.0**
20/230/040/FT	643187	643538	**643036**	644339.0	**643036.0**	652729.0	645259.0	645259.0	644132.0
20/230/200/VL	95097	94218	94283	**94213.0**	**94213.0**	95951.0	95126.0	94530.0	94247.0
20/230/200/FL	141253	138491	137842	137763.7	**137642.3**	141166.0	140084.0	139115.3	138343.0
20/230/200/VT	99410	98612	**97914**	97968.0	**97914.0**	98946.0	99550.0	98429.0	98338.0
20/230/200/FT	140273	136309	137072	136080.0	*135863.1*	141098.8	136423.0	136423.0	135888.0
20/300/040/VL	**429398**	**429398**	**429398**	**429398.0**	**429398.0**	430690.0	**429398.0**	**429398.0**	**429398.0**
20/300/040/FL	**586077**	588464	586077	587512.0	**586077.0**	603953.0	588464.0	588464.0	588464.0
20/300/040/VT	**464509**	**464509**	464627	464569.0	**464509.0**	464509.0	464628.0	464628.0	**464509.0**
20/300/040/FT	**604198**	**604198**	604201	**604198.0**	**604198.0**	611296.0	604208.0	604198.0	**604198.0**
20/300/200/VL	75319	75045	74902	74900.0	*74811.0*	76660.5	75756.5	75246.5	75013.0
20/300/200/FL	117543	116259	116431	115798.0	*115526.0*	116779.5	116150.8	116150.8	115981.0
20/300/200/VT	76198	74995	**74991**	**74991.0**	**74991.0**	76165.0	74995.0	74995.0	74995.0
20/300/200/FT	110344	109164	108638	*107315.0*	*107167.0*	108025.5	107466.5	107466.5	107466.5
30/520/100/VL	54113	54008	53983	53983.0	**53958.0**	54444.0	54122.0	54065.0	54065.0
30/520/100/FL	94388	93967	94066	94313.0	*93967.0*	99121.0	95007.0	94544.0	94265.0
30/520/100/VT	52174	52156	52247	52210.0	**52046.0**	52855.0	52279.0	52279.0	52073.0
30/520/100/FT	98883	97490	98543	97818.0	97107.0	101947.0	99917.0	98460.0	98275.0
30/520/400/VL	114042	112927	113720	112837.1	*112774.4*	114051.1	112848.1	112848.1	112848.1
30/520/400/FL	154218	149920	151009	149195.7	*149151.0*	150289.7	150063.8	149921.0	149423.9
30/520/400/VT	114922	114664	115581	**114640.5**	**114640.5**	114897.8	114742.5	114693.1	**114640.5**
30/520/400/FT	154606	152929	**	152634.8	*152476.7*	155326.0	152937.2	153202.3	152723.9
30/700/100/VL	47612	**47603**	**47603**	47614.0	**47603.0**	48616.0	47760.0	47717.0	47668.0
30/700/100/FL	60700	60184	60391	60076.0	*59958.0*	63073.0	60412.0	60091.0	60091.0
30/700/100/VT	46046	45880	45956	46066.0	**45871.5**	46557.0	46160.0	46085.0	46085.0
30/700/100/FT	55609	54926	54975	55115.0	54912.0	57529.0	55251.0	55251.0	55095.0
30/700/400/VL	98718	97982	99316	97875.0	*97853.4*	99023.0	97983.9	97983.9	97960.0
30/700/400/FL	152576	135109	133976	134723.3	*134553.7*	136609.4	135365.0	135365.0	134946.5
30/700/400/VT	96168	95781	95538	95274.6	*95249.6*	95861.7	95434.1	95482.0	95362.0
30/700/400/FT	131629	130856	131473	130051.2	*129990.0*	131998.7	130345.1	130292.0	130147.3

** ERROR, the upper bound is less than the lower bound.

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

IPS	MIP	HSA	CS	CSLB	MG	MGCS	RBB	BEST
408.1	6958.6	4587.2	262.4	6679.4	51.2	29.2	42.5	23.5

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

N/A/K/FC	IPS	MIP	HSA	CS	CSLB	MG	MGCS	RBB	BEST
100/400/010/VL	35	49	163	7.5	78.0	0.0	0.1	0.1	0.1
100/400/010/FL	9	109	194	8.7	10423.1	0.0	0.1	0.2	0.2
100/400/010/FT	813	918	169	1.0	2.0	0.1	0.3	1.2	0.3
100/400/030/VT	330	85	428	119.4	2165.0	0.7	0.7	0.7	0.7
100/400/030/FL	886	2068	937	9.3	18731.2	0.3	3.9	10.6	5.4
100/400/030/FT	888	145	1207	1.2	68.9	1.4	1.6	13.0	1.3
20/230/040/VL	4	116	118	0.4	0.9	0.1	0.1	0.1	0.1
20/230/040/VT	41	37	386	0.5	3.2	0.1	0.1	0.1	0.1
20/230/040/FT	45	63	168	0.6	9.8	0.1	0.2	0.1	0.2
20/230/200/VL	822	8328	2460	66.3	6061.1	11.7	14.9	27.2	16.3
20/230/200/FL	691	15006	1100	323.8	6722.5	12.1	27.5	40.9	28.0
20/230/200/VT	821	3965	2351	59.5	2705.9	10.9	10.5	22.5	10.7
20/230/200/FT	156	835	1249	107.8	9707.3	21.0	19.0	33.1	17.7
20/300/040/VL	19	83	197	0.3	0.8	0.1	0.1	0.1	0.1
20/300/040/FL	29	46	283	1.1	8.4	0.1	0.2	0.2	0.1
20/300/040/VT	24	161	111	0.5	3.8	0.1	0.2	0.2	0.2
20/300/040/FT	68	35	184	0.5	3.5	0.1	0.2	0.1	0.1
20/300/200/VL	802	4476	2964	360.2	10622.0	13.6	11.9	24.3	12.1
20/300/200/FL	686	9109	1046	379.2	14044.3	20.1	28.5	42.2	27.7
20/300/200/VT	388	1913	4022	43.3	2549.6	13.6	10.0	22.0	9.2
20/300/200/FT	396	542	2486	318.5	8720.0	17.0	18.5	30.6	16.5
30/520/100/VL	218	2415	1372	7.4	1680.4	2.2	2.4	3.8	1.6
30/520/100/FL	226	2925	928	280.5	7500.0	2.7	10.2	22.0	12.1
30/520/100/VT	455	2521	9127	4.1	4048.5	2.6	2.8	4.3	1.8
30/520/100/FT	815	4161	1008	1311.1	15791.7	2.8	7.0	18.8	7.3
30/520/400/VL	394	22797	12904	163.3	9501.6	153.1	91.8	112.8	44.4
30/520/400/FL	750	5769	9731	736.1	11131.2	200.2	102.1	119.1	76.0
30/520/400/VT	621	38793	18000	20.2	8262.8	155.0	69.3	96.1	48.8
30/520/400/FT	466	8556	9346	2467.6	19593.7	373.4	143.9	251.1	161.4
30/700/100/VL	32	3938	5783	3.9	64.4	1.8	1.6	1.9	1.3
30/700/100/FL	741	5650	3992	183.6	7631.5	2.1	5.3	16.4	5.5
30/700/100/VT	371	4263	4201	9.5	8806.6	3.4	3.5	9.3	2.1
30/700/100/FT	387	3018	6844	31.8	7639.7	2.4	4.2	14.9	3.7
30/700/400/VL	222	35241	18000	314.5	13311.9	129.5	65.5	91.5	47.8
30/700/400/FL	860	21429	13982	2084.3	16946.6	303.4	230.2	268.2	127.8
30/700/400/VT	365	15372	18000	104.7	10743.0	143.2	66.3	115.3	104.1
30/700/400/FT	225	32531	14286	177.1	13656.0	292.9	127.1	158.3	75.1

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

CYC	CS	CSLB	MG	MGCS	RBB	BEST
5.74	0.61	0.17	4.35	1.42	0.86	0.67

表 6 ：Results for R-Category Problems

Group	C	F	CYC	CS	CSLB	MG	MGCS	RBB	BEST
		F01	200613	**200087.0**	**200087.0**	202976.0	200484.0	200407.0	200407.0
	C1	F05	350573	348998.0	**346813.5**	363371.0	352545.0	351126.0	350735.0
		F10	507118	492409.0	**488015.0**	510348.0	492120.0	490875.0	490875.0
		F01	232473	229549.0	**229196.0**	231331.0	231179.0	**229196.0**	**229196.0**
r10	C2	F05	432913	412915.0	**411664.0**	441635.0	417011.0	416098.0	414876.0
		F10	640621	612598.0	**609104.0**	674960.0	616201.0	622971.0	616201.0
		F01	488737	487014.0	**486895.0**	488857.0	488585.0	486951.0	486951.0
	C8	F05	980010	957839.0	**951056.0**	963807.0	962291.0	957188.0	956414.0
		F10	1487270	1426215.0	**1421746.0**	1426048.0	1429463.0	1427598.0	1421862.0
		F01	725416	**714431.0**	**714431.0**	717085.0	715466.0	714432.0	**714431.0**
	C1	F05	1306090	1264665.0	**1263713.0**	1279486.0	1274747.0	1267751.0	1267626.0
		F10	1914040	1844097.0	**1843611.0**	1892247.0	1859212.0	1855297.0	1844097.0
		F01	876894	871323.5	**870451.0**	872862.0	871930.0	870784.0	870784.0
r11	C2	F05	1694860	1625443.0	**1623640.0**	1666402.0	1628629.0	1625191.0	1625191.0
		F10	2607690	2419709.0	**2414060.0**	2495473.0	2430628.0	2423220.0	2419709.0
		F01	2295790	2295439.0	**2294912.0**	2294912.0	2295439.0	**2294912.0**	**2294912.0**
	C8	F05	3568430	3508336.0	**3507100.0**	3508961.0	**3507100.0**	**3507100.0**	**3507100.0**
		F10	4621900	**4579353.0**	**4579353.0**	4580239.0	4580239.0	**4579353.0**	**4579353.0**
		F01	1713670	1639565.0	**1639443.0**	1642590.3	1640888.5	**1639443.0**	**1639443.0**
	C1	F05	3746250	3409218.0	**3396050.0**	3531315.0	3476239.0	3423560.8	3417612.0
		F10	6070200	5297992.0	**5228711.0**	5537455.7	5449426.3	5273253.3	5273253.3
		F01	2326230	**2303557.0**	**2303557.0**	2308881.0	2306043.5	2306043.5	2306043.5
r12	C2	F05	4967940	**4669799.0**	**4669799.0**	4669799.0	4677482.5	**4669799.0**	**4669799.0**
		F10	7638050	7101247.0	**7100019.0**	7121070.0	7104545.0	**7100019.0**	**7100019.0**
		F01	7637250	**7635270.0**	**7635270.0**	7635270.0	**7635270.0**	**7635270.0**	**7635270.0**
	C8	F05	10121700	**10067742.0**	**10067742.0**	10079161.0	**10067742.0**	**10067742.0**	**10067742.0**
		F10	12079300	**11967768.0**	**11967768.0**	11967768.0	**11967768.0**	**11967768.0**	**11967768.0**
		F01	144138	143036.0	**142947.0**	145080.0	**142947.0**	**142947.0**	**142947.0**
	C1	F05	270316	265049.0	**263800.0**	273816.0	268554.0	267074.0	266453.0
		F10	374999	369084.0	**365836.0**	434001.0	383704.0	374101.0	370400.0
		F01	151513	151170.0	**150977.0**	153590.0	152508.0	151742.0	151269.0
r13	C2	F05	291510	284283.0	**282682.0**	324476.0	285981.0	285415.0	285415.0
		F10	420028	412045.0	**406790.0**	489750.0	420708.0	412440.0	409626.0
		F01	212451	209317.0	**208088.0**	210975.0	208656.3	209879.0	208656.3
	C8	F05	484112	462629.0	**444826.0**	461006.0	452710.0	451442.0	450795.0
		F10	758715	717975.0	**697967.0**	747963.5	743893.0	720788.0	720788.0
		F01	415119	403420.0	**403414.0**	407376.0	404706.0	404310.0	403529.0
	C1	F05	803356	752530.0	**749503.0**	766455.0	755057.0	755593.0	754757.0
		F10	1155840	1064986.0	**1063098.0**	1108017.0	1093378.0	1078679.0	1076340.0
		F01	453204	**437607.0**	**437607.0**	438849.0	438234.5	438261.0	**437607.0**
r14	C2	F05	912456	850303.0	**849163.0**	866457.0	858997.0	855219.0	853202.0
		F10	1333440	1216473.0	**1214609.0**	1265720.0	1238355.0	1218178.0	1216473.0
		F01	702226	**668216.3**	**668216.3**	678779.6	671153.0	670635.7	670267.0
	C8	F05	1748930	1620125.9	**1613428.8**	1688082.0	1627952.0	1639308.0	1627952.0
		F10	2882710	2662036.3	**2602690.0**	2750810.0	2666918.5	2650638.0	2650638.0

表 7 ：Results for R-Category Problems

Group	C	F	CYC	CS	CSLB	MG	MGCS	RBB	BEST
r15	C1	F01	1049360	**1000787.0**	**1000787.0**	1007622.0	1003371.5	1002006.0	1001491.0
		F05	2158720	1972762.0	**1966206.0**	2008755.8	1968199.0	1968199.0	1968199.0
		F10	3135760	2887346.0	*2884076.5*	2983338.0	2922139.0	2922139.0	2902001.0
	C2	F01	1215130	1149298.2	**1148604.0**	1157701.0	1151939.7	1149949.8	1149188.7
		F05	2756680	2476699.7	*2476246.0*	2552541.0	2516026.0	2490344.0	2477502.0
		F10	4384640	3837263.0	3831870.4	4202576.0	3871287.0	3860121.0	3856335.5
	C8	F01	2355730	2300475.0	**2297919.0**	2302279.3	2301208.8	2299127.0	2298492.5
		F05	5926330	5583619.0	**5573412.8**	5596404.3	5582658.0	5578736.0	5577946.5
		F10	9180920	**8696932.0**	**8696932.0**	8699691.5	8701696.0	8700994.0	8699691.5
r16	C1	F01	136538	**136161.0**	**136161.0**	139603.0	136722.0	136722.0	136722.0
		F05	247682	240221.0	**239500.0**	277032.0	246355.0	242632.0	240356.0
		F10	338807	325839.0	**325671.0**	388965.0	341665.0	331926.0	326458.0
	C2	F01	139973	**138532.0**	**138532.0**	141602.0	139961.0	**138532.0**	**138532.0**
		F05	246014	**241801.0**	**241801.0**	272129.0	249853.0	243456.0	243456.0
		F10	355610	340496.0	**337762.0**	403801.0	337908.0	337908.0	337908.0
	C8	F01	172268	173508.0	**169233.0**	173603.0	171933.0	171236.0	171179.0
		F05	365214	357334.0	**348167.0**	386824.0	356904.0	356904.0	355611.0
		F10	569874	537140.0	*529988.0*	600937.0	552994.7	537096.0	537096.0
r17	C1	F01	370090	**354138.0**	**354138.0**	361765.0	355392.0	354503.0	354223.0
		F05	688554	651335.0	**645488.0**	677035.0	664491.0	658438.0	653017.0
		F10	971151	915958.0	**910518.0**	991407.0	932461.0	922602.0	920776.0
	C2	F01	380850	370622.0	**370590.0**	373139.0	373371.0	371308.0	371269.0
		F05	753188	709247.0	**706746.5**	749007.0	730519.0	711220.0	709316.0
		F10	1108180	1023343.0	*1019917.0*	1050036.5	1033116.0	1036240.5	1033068.0
	C8	F01	524038	504227.5	**501634.5**	507744.0	506265.0	503837.0	503770.0
		F05	1195140	1108344.0	*1105083.0*	1167845.0	1115103.4	1111467.5	1108224.3
		F10	1945080	1794926.0	*1781436.0*	2065964.0	1796565.8	1792591.5	1790088.0
r18	C1	F01	872888	829416.3	**828117.0**	831098.0	830366.0	831135.0	830366.0
		F05	1716680	1535033.0	**1533675.0**	1593223.0	1547348.0	1546469.0	1544783.0
		F10	2377560	2185109.0	2174276.0	2208324.0	2217118.0	2217118.0	2208324.0
	C2	F01	975396	921752.0	**919325.0**	927146.0	924557.0	922552.0	921565.0
		F05	2037950	1829540.0	*1826245.0*	1850468.0	1850572.0	1833652.0	1833652.0
		F10	2966370	2706736.0	*2703852.0*	2730882.0	2742956.0	2736706.0	2730882.0
	C8	F01	1622520	1481165.3	*1477395.0*	1499345.3	1486926.8	1484621.9	1481353.3
		F05	4576750	3906817.0	3896893.2	4027423.0	3953603.3	3921596.5	3914562.6
		F10	7504310	6397172.5	**6361906.0**	6613019.3	6520939.0	6390734.7	6376447.0

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

CYC	CS	CSLB	MG	MGCS	RBB	BEST
2933.8	119.1	3310.5	9.1	13.2	18.4	10.1

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

Group	C	F	CYC	CS	CSLB	MG	MGCS	RBB	BEST
		F01	124.2	0.3	0.6	0.1	0.1	0.1	0.1
	C1	F05	135.1	3.3	8.5	0.1	0.3	0.3	0.3
		F10	161.9	5.1	17.8	0.2	0.4	0.7	0.7
		F01	80.3	1.3	5.5	0.2	0.2	0.2	0.1
r10	C2	F05	196.4	1.4	59.6	0.2	0.2	0.3	0.3
		F10	175.4	6.0	140.7	0.2	0.4	0.7	0.5
		F01	464.7	1.7	11.0	0.7	0.7	0.9	0.7
	C8	F05	414.6	2.4	22.0	0.6	0.9	1.3	0.5
		F10	332.9	2.9	23.3	0.7	1.1	1.6	0.4
		F01	392.6	0.8	10.2	1.2	0.9	1.1	0.7
	C1	F05	631.1	26.0	573.8	1.5	2.1	9.1	1.8
		F10	674.6	57.2	2334.4	2.0	2.7	11.1	2.9
		F01	580.6	2.1	83.5	1.9	1.6	2.2	1.3
r11	C2	F05	1069.8	9.0	303.4	3.1	2.9	3.9	1.7
		F10	1244.4	18.2	875.9	2.2	3.4	8.2	2.5
		F01	1126.3	1.6	5.1	1.9	2.1	2.7	2.1
	C8	F05	1160.2	2.3	8.3	1.8	1.5	1.6	1.5
		F10	786.5	1.6	5.4	0.8	0.8	1.1	0.8
		F01	3012.1	4.8	83.5	8.3	5.1	5.7	4.2
	C1	F05	3274.9	155.1	5311.2	19.9	23.5	33.0	9.5
		F10	3612.7	158.3	7333.6	23.1	22.1	30.8	11.2
		F01	4700.4	3.6	77.3	13.7	15.6	17.2	10.3
r12	C2	F05	7932.1	7.6	114.7	15.6	13.2	14.8	5.5
		F10	4728.5	7.7	132.1	8.7	7.4	9.1	4.1
		F01	4728.3	3.3	7.3	6.2	6.6	6.7	6.9
	C8	F05	2978.3	1.0	4.3	2.5	2.7	3.1	2.9
		F10	2284.0	10.1	13.6	1.7	1.6	1.7	1.8
		F01	277.4	0.3	1.1	0.1	0.1	0.1	0.1
	C1	F05	274.7	6.3	40.7	0.1	0.8	1.5	1.3
		F10	259.2	20.0	61.1	0.2	2.2	3.4	3.5
		F01	252.3	0.5	3.8	0.1	0.1	0.2	0.2
r13	C2	F05	285.3	6.7	74.0	0.2	0.6	1.4	0.9
		F10	310.8	24.0	356.2	0.1	1.8	4.6	2.2
		F01	393.5	2.7	225.0	0.3	0.4	0.6	0.3
	C8	F05	483.3	11.1	4011.8	0.4	0.6	0.9	0.4
		F10	504.5	13.9	5376.8	0.5	0.9	10.8	0.7
		F01	615.2	1.7	14.6	1.0	0.9	1.1	0.7
	C1	F05	822.8	66.5	1834.9	1.6	5.0	10.9	6.7
		F10	765.6	155.2	3223.1	1.6	12.1	24.8	16.5
		F01	665.4	3.1	25.9	1.2	1.1	1.3	0.9
r14	C2	F05	971.3	74.3	4905.4	1.8	4.2	15.0	4.5
		F10	862.7	93.4	4769.8	1.8	7.1	17.7	7.3
		F01	1814.6	8.0	3995.8	6.9	5.9	11.3	2.8
	C8	F05	3793.7	147.2	8440.1	7.7	15.4	29.1	2.9
		F10	3858.4	58.3	6806.1	6.3	20.3	26.3	5.0

表10：Results for R-Category Problems

Group	C	F	CYC	CS	CSLB	MG	MGCS	RBB	BEST
r15	C1	F01	1767.1	3.6	208.6	9.2	4.3	5.6	3.3
		F05	2521.4	184.3	13158.1	11.3	19.7	33.8	22.4
		F10	2618.4	416.8	11905.3	10.6	34.3	48.6	38.6
	C2	F01	2794.5	28.8	3049.4	11.3	8.0	13.7	4.8
		F05	5980.8	349.0	9029.4	26.2	22.7	35.3	16.9
		F10	5133.2	233.2	21993.4	23.9	26.4	37.6	26.9
	C8	F01	15947.9	10.3	2465.6	63.0	58.4	61.1	21.4
		F05	13398.5	11.2	316.4	55.2	25.1	21.2	11.0
		F10	12288.1	31.2	103.8	14.7	8.4	8.7	6.1
r16	C1	F01	404.9	0.3	0.6	0.1	0.1	0.1	0.1
		F05	416.9	11.8	59.3	0.1	1.8	2.8	3.0
		F10	148.5	35.6	114.3	0.1	5.7	7.8	9.0
	C2	F01	376.8	0.6	1.9	0.2	0.1	0.1	0.2
		F05	430.9	9.9	33.5	0.1	2.1	2.8	2.7
		F10	421.6	37.0	212.6	0.2	6.2	8.3	7.3
	C8	F01	477.1	2.4	2973.7	0.2	0.3	0.3	0.4
		F05	520.9	43.4	6730.1	0.2	0.8	1.1	1.0
		F10	577.9	73.0	7734.3	0.2	1.8	12.1	1.7
r17	C1	F01	909.2	2.4	15.2	1.2	1.2	1.3	1.0
		F05	998.6	148.7	1145.0	1.6	12.5	23.5	19.8
		F10	999.4	341.6	7501.1	1.6	37.9	52.1	54.5
	C2	F01	927.3	4.1	69.8	1.5	1.4	1.7	1.2
		F05	1070.0	85.4	5679.2	1.6	10.2	19.4	12.8
		F10	1168.9	426.0	7823.3	1.9	21.7	35.1	26.8
	C8	F01	1315.7	37.8	6135.1	3.3	3.6	13.7	2.0
		F05	2508.8	144.3	8979.1	4.6	8.2	19.5	6.3
		F10	2813.2	279.7	12489.6	4.6	13.3	24.6	8.1
r18	C1	F01	2339.0	29.0	2455.5	10.0	8.0	16.5	6.3
		F05	2889.6	662.8	6817.3	10.9	61.2	77.2	69.3
		F10	2918.6	1081.4	12505.6	10.8	112.1	133.4	125.1
	C2	F01	2677.5	115.0	9410.5	12.0	8.9	20.7	7.7
		F05	4335.1	522.0	14321.7	12.9	35.6	48.3	37.6
		F10	4234.0	2166.6	12970.2	12.4	42.2	54.9	44.7
	C8	F01	13514.9	96.3	14508.3	108.9	71.6	84.0	33.8
		F05	28883.1	213.7	13431.7	97.1	112.9	118.0	44.1
		F10	32695.2	245.1	25608.2	65.2	82.7	112.8	11.5