容量制約をもつ最小木問題のモデルとアルゴリズム

片　山　直　登

1．はじめに

通信ネットワーク設計問題は，通信ネットワークのデザインと通信のルーティングを求める問題である。通信ネットワーク設計問題の中で，特にホストコンピュータなどを中心としたマルチポイントやマルチドロップなどと呼ばれる閉路を含まない通信ネットワーク設計問題は，ローカルアクセスネットワークデザイン問題などと呼ばれ，次のような条件をもつモデルとして扱うことができる。ネットワークの形状が木に限定される。通信は，ホストコンピューター端末間の1対多間に発生する。費用は，通信回線のデザイン費用のみであり，通信のルーティング費用は発生しない。通信回線には回線容量が与えられる。このような条件を考慮した問題は，容量制約をもつ最小木問題(Capacitated Minimum Spanning Tree Problem, *CMST*）としてモデル化できることが知られている[9)]。

本論文は，容量制約をもつ最小木問題に関する定式化，解法および妥当不等式など，従来の研究のサーベイである。2節では *CMST* の定式化を示し，3節では *CMST* の性質を示す。4節では *CMST* に対する近似解法を示し，5節では *CMST* に対する最適解法および妥当不等式を示す。

2．容量制約をもつ最小木問題の定式化

ホストコンピュータ，集線装置，端末などのように点で表すことのできるものをノードと呼び，通信回線などのように点と点を結ぶ線をアークと呼ぶ。通信データのように，ネットワーク上には決められたノード間を移動するものが存在し，単位時間当たりの移動量を需要量，この品種の需要の出発ノードを始点，到着ノードを終点と呼ぶ。一般的

には品種はものの種類を表すが，ここでは同一の始点・終点をもつものを同一の品種とし，始点または終点が異なるものは異なる品種として区別する。アーク上の品種の流れをフローと呼ぶ。*CMST* では，1ノードを始点とし，これ以外のノードを終点とする多品種を対象とする。特に，始点をセンターと呼び，集線装置やホストコンピュータなどに対応する。一方，終点ノードは端末装置に対応する。通信回線容量のように，単位時間当たりに処理できる量の上限値をアーク容量と呼ぶ。

このとき，容量制約をもつ最小木問題は，1始点-多終点間の多品種の需要量とアーク容量が与えられたときに，フロー保存条件とアーク容量制約などを満足する条件の下で，デザイン費用の合計を最小にする全域木を求める問題となる。

一方，全域木からセンターに直接接続するアークを取り除くと，全域木は部分木に分割される。これらの部分木，一般の部分木，および適当な手順にしたがって構築中の連結成分をコンポーネントと呼ぶ。コンポーネントとセンターを直接または間接に接続するアーク上のフローは，そのコンポーネントに含まれるノードを終点とする需要量の合計であり，コンポーネント内の最大値をとる。したがって，*CMST* は，各コンポーネントに含まれるノードの需要量の合計が容量制約以下となるように，デザイン費用を最小化する全域木を求める問題と見ることもできる。

ノード集合を N，センターを含まないノード集合を N'，向きのあるアーク集合を A，センターと N' 内のノード間のアークを含まないアーク集合を A'，品種の集合を K とし，センターのノード番号を0とおく。ノード i, j 間を結ぶアークを (i, j)，アーク (i, j) 上の品種 k のフロー量の比率を表すフロー変数を x_{ij}^{k} とする。アーク (i, j) を設置するか否かを表すデザイン変数を y_{ij}，アーク (i, j) のデザイン費用を f_{ij} とし，アーク (i, j) のアーク容量を C とする。また，センターを始点，ノード i を終点とする需要量を d_i とし，特に $d_0=0$ とする。

このとき，*CMST* は次のように定式化することができる。

$$(CMST_1)\quad \text{最小化}\quad \sum_{i\in N}\sum_{j\in N'} f_{ij}y_{ij} \tag{1}$$

$$\text{条件}\quad \sum_{i\in N}x_{in}^{k}-\sum_{j\in N}x_{nj}^{k}=1 \quad for\ all\ n\in N',\ k\in K \tag{2}$$

$$\sum_{i\in N}y_{ij}=1 \quad for\ all\ j\in N' \tag{3}$$

$$\sum_{k\in N'}d_k x_{ij}^{k}\le (C-d_i)\,y_{ij} \quad for\ all\ i\in N,\ j\in N' \tag{4}$$

$$0\le x_{ij}^{k}\le y_{ij} \quad for\ all\ i\in N,\ j\in N',\ k\in K \tag{5}$$

$$y_{ij}\in\{0,1\} \quad for\ all\ i\in N,\ j\in N' \tag{6}$$

(1)式は，目的関数であるデザイン費用の合計であり，これを最小化することを表す。(2)式は，センター以外のノードに対して，入るフロー量と出るフロー量の差が，そのノードの需要量であることを表す。(3)式は，センター以外のノードに入るアークは1本であることを表す。(4)式は，アーク (i, j) が存在するときに，センター以外のノード i から

出るアーク上を流れるフロー量の合計は，最大でもアーク容量から i の需要量を引いた量であることを表す。(5)式は，アーク $(i,\ j)$ が設置されたときのみアーク $(i,\ j)$ 上のフローが存在することと，フロー変数が非負であることを表す。(6)式は，デザイン変数が0-1変数であることを表す。

一方，*CMST* は，部分木に含まれるノードの需要量の合計を容量制約以下にする問題と見ることができる。この考えを用いると，フロー変数を使用しない定式化を行うことができる。

$$(CMST_2)\quad \text{最小化} \quad \sum_{i\in N}\sum_{j\in N'} f_{ij}y_{ij}$$

$$\text{条件} \quad \sum_{i\in N} y_{in} + \sum_{j\in N'} y_{nj} \geq 1 \quad \text{for all } n\in N' \tag{7}$$

$$\sum_{i\in N}\sum_{j\in N'} y_{ij} = |N|-1 \tag{8}$$

$$\sum_{i\in S}\sum_{j\in S} y_{ij} \leq |S| - \left\lceil \sum_{i\in S} d_i/C \right\rceil \quad \text{for all } S\subset N',\ |S|\geq 2 \tag{9}$$

$$y_{ij}\in\{0,\ 1\} \quad \text{for all } i\in N,\ j\in N'$$

(7)式は，センター以外のノードに出入りするアークは1本以上であることを表す。(8)式は，アークの総数が $|N|-1$ 本であることを表している。S ($\subset N'$) 内のノードが1つの部分木を構成すれば，$\sum_{i\in S}\sum_{j\in S} y_{ij} = |S|-1$ である。しかし，容量との関係から，S 内のノードは少なくとも $\left\lceil \sum_{i\in S} d_i/C \right\rceil$ 個の部分木に分割する必要がある。(9)式は，この性質を表す。

一方，アークは最大でも C の需要量しか処理できないので，センターに接続するアークは少なくとも $\left\lceil \sum_{i\in N'} d_i/C \right\rceil$ 本は必要となるので，

$$\sum_{i\in N'} y_{0i} \geq \left\lceil \sum_{i\in N'} d_i/C \right\rceil \tag{10}$$

が成り立ち，この式はセンターの度数制約と呼ばれ，*CMST* の妥当な不等式となる。

3．容量制約をもつ最小木問題の性質

CMST は，容量に関する制約である(4)，(9)や(10)式を緩和（取り除く）すると，その緩和問題は最小木問題となる。最小木問題は容易に最適に解くことができ，その目的関数値は *CMST* の下界値となる。

CMST に対して，Chandy-Lo[3] は次の2つの定理を示している。*CMST* の最適木を *CT*，容量制約緩和問題である最小木問題の最適木を T とする。

定理3.1　アーク $(0,\ i)$ が T に含まれていれば，このアークは *CT* に含まれる。

定理3.2　アーク $(i,\ j)$ が T に含まれ，かつノード i と j が *CT* の同じコンポーネント

に含まれれば，アーク (i, j) は CT に含まれる。

これらの定理と容量制約緩和解による下界値を用いると，分枝限定法を効率的に行うことができる。

Papadimitriou[19] は，充足可能性問題を用いて，次の 2 つの定理を示している。$CMST$ 決定問題は，総費用の目標値を F としたとき，総費用 F となる $CMST$ の解が存在するか否かを判定する問題である。

定理3.3　$CMST$ 決定問題は，$\mathcal{NP}$-完全である。

定理3.4　ユークリッド距離をデザイン費用とする $CMST$ 決定問題は，$\mathcal{NP}$-完全である。

これらの定理は，$CMST$ に対する効率的な解法が存在しない可能性を示している。

4 節に示す Q 反復最適巡回路分割法において，最小木法や最近挿入法などの多項式オーダーでかつ最適値の 2 倍以内の巡回路が求められる方法を用いて巡回路を求めるものとする。このとき，Altinkemer-Gavish[1] は，次の 2 つの定理を示している。

定理3.5　すべての品種の需要量が同一であるとする。Q 反復巡回路分割法による解の目的関数値を ϕ_e^h とし，$CMST$ の最適値を ϕ_e^* する。このとき，次式が成り立つ。

$$\phi_e^h/\phi_e^* \leq 3-2/C$$

一方，品種の需要量が一般的な場合には，次の定理が成り立つ。

定理3.6　最適巡回路分割法による解の目的関数値を ϕ^h とし，$CMST$ の最適値を ϕ^* する。このとき，次式が成り立つ。

$$\phi^h/\phi^* \leq 4-4/C$$

最適巡回路分割法の計算量は $O(|N|^2)$ であり，これらの定理は $CMST$ の最適値の定数倍の解を求める多項式オーダーの解法が存在することを意味する。

4．近似解法

ここでは，$CMST$ に対する近似解法を示す。

修正 Kruskal 法と修正 Prim 法

容量制約を緩和した問題は，最小木問題になる。そこで，最小木問題の解法にコンポーネントに含まれる需要の合計が容量を超えないような手順を付加すれば，$CMST$ の近似解を求めることができる。

Boorstyn-Frank[2] は，最小木の解法である Kruskal 法を改良した方法を提案している。Kruskal 法と同様に，A' に含まれるアークについて，デザイン費用の安い順にアークを付加していく。このとき，コンポーネントに含まれるノードの需要量の合計が容量以下でかつ閉路を含まないようにする。もし，コンポーネントに含まれるノードの需要量の合計が容量を超えれば，コンポーネントに含まれるノードとセンターを結ぶアークの中で，デザイン費用が最小のアークを付加し，このコンポーネントを対象からはずす。この手法の計算量は，$O(|N|^2\log|N|)$ である。

Prim 法を改良しても近似解を求めることができる。Prim 法を用いて，センターを始点とした最小木を形成していく。その際に，構築されていくコンポーネントに含まれるノードの需要量の合計が容量を超える場合には，そのアークは付加しないものとする。この手法の計算量は，$O(|N|^2)$ である。

Esau-Williams 法

Esau-Williams[6] は，トレードオフ基準値を用いた近似解法を示している。A' に含まれるアークを順々に付加して，コンポーネントを構築していく方法を考える。このとき，ノード i を含むコンポーネント内のノードとセンターを結ぶアークの中で最小のデザイン費用をノード i の重み w_i とする。ノード i を含むコンポーネントに含まれるノード集合を COM_i とすると，$w_i=\min\{f_{0j}|j\in COM_i\}$ と表すことができる。

アーク (i, j) を付加するときの基準値 t_{ij} をデザイン費用と重みの差 $(t_{ij}=f_{ij}-w_i)$ とする。この t_{ij} の小さい順にアークの付加を検討していく。t_{ij} は，ノード i，j 間の接続と，i とセンター間の接続とのトレードオフを表している。基準値が小さい（負である）ことは，ノード i を j へ接続する方が，i を COM_i に含まれるノードを経由してセンターに接続するよりも費用が安いことになる。

アークを付加する際には，コンポーネントに含まれるノードの需要量の合計が容量を超えず，閉路を含まないようにする。アークを付加した後には 2 つのコンポーネントが連結され，w_i が変化するので，w_i と t_{ij} を更新する必要がある。付加できるアークがなくなったら，各コンポーネントに対して，コンポーネントに含まれるノードとセンター間のアークの中で，最小デザイン費用をもつアークを付加する。この手法の計算量は，$O(|N|^2\log|N|)$ である。

統一アルゴリズム

Kershenbaum-Chou[17] は，Esau-Williams 法における重みと，その更新方法を一般化することによって，従来の手法を統一的に表現できるようにしている。次のように，ノード i の重み w_i を定義する。

$$w_i = \alpha\{\beta f_{0i} + (1-\beta) f_{ij}\}$$

ここで，j はノード i の最近隣接ノードなどであり，α と β は $\alpha \geq 0$，$0 \leq \beta \leq 1$ である定数である。

修正 Krauskal 法は $\alpha=0$ とし，重み $w_i=0$ とした場合に対応する。Esau-Williams 法は，$\alpha=1$，$\beta=1$ とし，アーク (i, j) を付加したときに $w_i := w_j$ と更新したものとなる。修正 Prim 法は，重み w_i の初期値を $w_0 := 0$，$w_j := -\infty$ $(j \in N')$ とし，アーク (i, j) を付加したときに $w_j := 0$ と更新したものとなる。

スイープ法

Sharma[21] は，スイープ法を用いた手法を示している。センターを中心とした平面において，センターとなす角度によってコンポーネントを構成する方法であり，配送経路問題のスイープ法と同じ考えである。センターとの角度順にノードをソートし，この順にしたがって容量を超えない範囲でノードをコンポーネントに分割する。すべてのノードが分割されるまで，この操作を繰り返す。その後，コンポーネントに含まれるノードとセンターを対象とした最小木を求め，全域木を構成する。この手法の計算量は，$O(|N|\log|N|)$ である。

2次オーダー貪欲アルゴリズム

Esau-Williams 法などの貪欲的近似解法を1次オーダー貪欲アルゴリズム（First Order Greedy Algorithm, *FOGA*）と呼び，*FOGA* の解を初期解とし，適当な条件を設定した *FOGA* を行なうことによって，近似解を求める方法を2次オーダー貪欲アルゴリズム（Second Oorder Greedy Algorithm, *SOGA*）と呼ぶ。

Karnaugh[15] は，*SOGA* である禁止アルゴリズムと結合アルゴリズムを示している。*FOGA* では，アークを付加し，2つのコンポーネントを連結していくことを繰り返す。*FOGA* で連結した繰り返し回に，その2つのコンポーネントの連結を禁止するという条件を付けて，新たに *FOGA* を行う。このとき，初期解と異なる解が生成され，解を改善できる可能性がある。禁止アルゴリズムは，このように特定の繰り返し回に連結を禁止する条件を付けて *FOGA* を行う方法である。$|N|-1$ 種類の禁止方法が設定できるので，$|N|-1$ 個の近似解が生成され，これらの最良の解を採用する。

ノード i から最も近い（デザイン費用の安い）ノードを i_1，ノード i よりもセンターに

近いノードの中で i に最も近いノードを i_2 とする。各ノードに対して，*FOGA* で求めた木にアーク $(i,\ i_1)$ が含まれていなければ，このアークを木に含むという条件を加えて *FOGA* を行う。さらに，*FOGA* で求めた木にアーク $(i,\ i_2)$ が含まれていなければ，このアークを木に含むという条件を加えて *FOGA* を行い，これらの最良の解を採用する。この方法は，結合アルゴリズムと呼ばれている。

Kershenbaum-Boorstyn-Oppenhein[16] は，より効率的な *SOGA* を提案している。経験的に，*CMST* の最適木に含まれるアークの大半が，最小木に含まれていることが分かっている。そこで，最小木に含まれているが，*FOGA* で求めた *CMST* の木に含まれていないいくつかのアークを木に含む条件のもとで *FOGA* を行えば，解を最善できる可能性がある。最小木に含まれているが *FOGA* で求めた木に含まれていないアークの集合を A_0 とし，A_0 の部分集合を A_1 とする。A_1 に含まれるアークを木に含み，$A_0 \backslash A_1$ に含まれるアークを木に含まないという条件を付けて，*FOGA* を行う。A_1 を $|A_1| \leq 2$ であるものに制限すれば，計算量は $O(|N|^3 \log|N|)$ となる。

Gavish[11] は，この他に *CMST* に対する近似解法のサーバイを行っている。

アーク交換法

Elias-Ferguson[5] は，次のようなアーク交換を利用した近似解法を示している。適当な最小木を初期解とする。この最小木は実行可能解とは限らない。ここで，ノード i の親ノードを p_i とする。このとき，アーク $(p_i,\ i)$ が容量制約を満たさないノード i の集合を N_1，容量制約を満たすノード i の集合を N_2 とする。

N_1 に含まれるノード i を選び，アーク $(p_i,\ i)$ を取り除く。さらに，ノード i に最も近いノードまたは先祖ノードを k とし，N_2 に含まれる i の子孫ノードを l とし，アーク $(k,\ l)$ を加える。この交換の中で，交換後，全域木となるものを考える。これらのアーク交換の中で，最も費用が増加しない交換を選び，実行する。実行可能解が求められるまで，この操作を繰り返す。

パラレルセービング法

Gavish-Altinkemer[11),12)] は，セービング法を拡張した解法を示している。*FOGA* では，各繰り返しにおいて１本のアークを貪欲に選択し，２つのコンポーネントを連結する。これに対して，パラレルセービング法は，２つのコンポーネントを連結したときの費用の減少量をもとに，各繰り返しにおいて同時に複数の（センターを含む）コンポーネントの連結を行う方法である。

初期のコンポーネントは，各ノード（センターを含む）とする。ある繰り返しにおいて，コンポーネント p とコンポーネント q を連結するときの費用の減少量をセービング値 S_{pq} とし，$p,\ q\ (p<q)$ に対して次のように計算する。

$$S_{pq}=\begin{cases}0 & \textit{if } p=0\\ f(COM_p)+f(COM_q)-f(COM_p\cup COM_q) & \textit{otherwise}\end{cases}$$

ここで，コンポーネント p に含まれるノード集合を COM_p，$COM_p\cup\{0\}$ 上の最小木の費用を $f(COM_p)$ とする。

各コンポーネントを１つのノードとみなし，$\sum_{i\in COM_p}d_i+\sum_{i\in COM_q}d_i\leq C$ を満たすすべての $(p,\ q)$ について重み S_{pq} をもつアーク $(p,\ q)$ を用いた補助ネットワークを作成する。ここで，アーク容量を C とする。このネットワークに対して最大マッチング問題を解くと，実行可能なコンポーネントの連結が得られるので，これらのコンポーネントを連結する。全域木が得られるまで，この操作を繰り返すと，最終的に実行可能な木が得られる。

巡回路分割法

Altinkemer-Gavish[1] は，$CMST$ の最適値の定数倍の解を求める多項式オーダーの解法を開発している。アークのデザイン費用を長さとした $G(N',\ A')$ 上で，巡回セールスマン問題の巡回路が求められているものとする。巡回路上の１番目のノードから時計周りの順番に，需要量の合計が容量制約以内になるように，巡回路を分断してコンポーネントを形成する。コンポーネント内の巡回路上の最初のノードをセンターに接続すると実行可能な木が得られる。さらに，スタートするノードを巡回路上の２番目，３番目，…，$|N'|$ 番目と変更して，それぞれに対して同様にして解を求め，これらの中の最良解を求める方法を Q 反復巡回路分割法と呼ぶ。

一方，巡回路上のノードに通し番号 $(1,\ 2,\ \cdots,\ |N'|)$ をつけ，次のような長さ e をもつネットワーク上のノード 0 から $|N'|$ までの最短路問題を作成する.

$$e_{ij}=\begin{cases}f_{0,i+1}+\sum_{l=i+1}^{j-1}f_{l,l+1} & \textit{if } \sum_{l=i+1}^{j}d_i\leq C\\ \infty & \textit{otherwise}\end{cases}$$

このとき，ノード 0 から $|N'|$ までの最短路は，巡回路の最小費用の分割を形成する。最短路にアーク $(i,\ j)$ が含まれる場合，ノード $(i+1,\ \cdots,\ j)$ が１つのコンポーネントを構成することになる。この方法によって，巡回路をコンポーネントに分割する方法を最適巡回路分割法と呼ぶ。

3節で示したように，これらの解法は最適値の定数倍の目的関数値をもつ近似解を提供する。

Lagrange ヒューリスティック法

５節で示す Lagrange 緩和法と劣勾配法を用いると，その繰返し毎に最小木が得られ

る。しかし，これらの最小木は実行可能な木とは限らない。Lagrange ヒューリスティック法は，これらの最小木のアークを交換して木を改良し，実行可能解を求めていく方法である。

アークの交換方法として，Gavish[10] は次の方法を示している。容量制約を満足しないコンポーネントに含まれるすべてのアーク (i, j) に対して，緩和問題の最小木に含まれないアーク (k, l) を見つける。ただし，このアーク (k, l) は，アーク (i, j) を取り除き，アーク (k, l) を付加したときに，コンポーネントが容量制約を満たし，かつ全域木を形成するアークの中で，費用最小のアークとする。これらの (i, j) と (k, l) の組み合わせの中で，費用の増加量 $f_{kl}-f_{ij}$ が最小のものを選び，アーク (i, j) を取り除き，アーク (k, l) を付加する。実行可能解が得られるまで，この操作を繰り返す。この手法の計算量は，$O(|N|)^3$ となる。

タブサーチ法

Sharaiha-Gendreau-Laporte-Osman[20] は，タブサーチ法を用いた解法を提案している。このタブサーチ法における近傍探索法は，次の通りである。根を i とする部分木を T_i，ノード i の親ノードを p_i とする。ノード l は T_i に含まれるノードで，k はセンターまたは l の近傍ノードとする。このとき，アーク (p_i, i) を取り除き，アーク (k, l) を加える交換を行う。取り除いたアーク (p_i, i) に対して探索禁止条件を設定し，このアークは一定回の付加を禁止する。さらに，解が実行可能でない場合には，容量オーバー分をペナルティーとして目的関数に加えることも提案している。

5．最適解法と妥当不等式

この節では，*CMST* に対する最適解法である分枝限定法，Benders 分解法および Lagrange 緩和法と，強い下界値を生成するための妥当不等式を示す。

容量制約緩和法

CMST 決定問題は $\mathcal{NP}$-完全であるので，*CMST* の厳密解を求める方法として分枝限定法などが用いられる。そのためには，緩和問題と下界値が必要となる。最も基本的な緩和問題は，容量制約条件を取り除く緩和問題である。この緩和問題は最小木問題となるため，容易に *CMST* の下界値を求めることができる。

Chandy-Lo[3] は，この下界値を用いた分枝限定法を提案し，3 節に示した 2 つの定理を用いて分枝操作と限定操作を効率化している。また，Chandy-Russell[4]，Elias-Ferguson[5]，Kershenbaum-Boorstyn-Oppenhein[16] も，*CMST* に対する分枝限定法を提案している。

Benders 分解法

すべての需要が1であるモデルに対して，Gavish[8) は次に示す Benders カットを用いた Benders 分解法を示している。定式化 $CMST_1$ において，デザイン変数 y を固定したフロー変数 x に対する問題を作成する。この問題および双対問題は容易に解くことができ，次の 2 種類の Benders カットが得られる。

$$\sum_{i\in N_1}\sum_{j\in N_1}y_{ij}\geq\lceil\ |N'\backslash N_1|/C\rceil$$
$$\sum_{i\in N_2}\sum_{j\in N'\backslash N_2}y_{ij}\geq\lceil\ |N_2|/C\rceil$$

ここで，主問題の解 y において，センターからのパスが存在するノード集合を N_1 ($\subset N'$) とする。また，主問題の解 y において，センターに接続するアークの中で容量を越えるアークを取り除いたときに，センターからのパスが存在しないノード集合を N_2 ($\subset N'$) とする。一方，切り上げはカットを強化するために使用している。

フロー保存制約を用いた Lagrange 緩和法

すべての需要が 1 であるモデルに対して，Gavish[9) はフロー保存制約を Lagrange 緩和して下界値を求める解法を示している。はじめに，使用する定式化を示しておく。x_{ij} はアーク (i, j) 上のフロー量を表す変数である。

$$\begin{array}{lll}
(CMST_3)\quad 最小化 & \sum_{i\in N}\sum_{j\in N'}f_{ij}y_{ij} & \\
\quad\quad 条件 & \sum_{i\in N}x_{in}-\sum_{j\in N}x_{nj}=1\quad for\ all\ n\in N' & (11)\\
 & \sum_{i\in N}y_{ij}=1\quad for\ all\ j\in N' & (12)\\
 & y_{ij}\leq x_{ij}\leq (C-1)\,y_{ij}\quad for\ all\ i\in N,\ j\in N' & (13)\\
 & y_{1j}\leq x_{1j}\leq Cy_{1j}\quad for\ all\ j\in N' & (14)\\
 & \sum_{i\in N'}y_{1j}\geq\lceil\ |N'|/C\rceil & (15)\\
 & x_{ij}\geq 0,\ y_{ij}\in\{0, 1\}\quad for\ all\ i\in N,\ j\in N' & (16)
\end{array}$$

乗数 v を用いて，フロー保存式(11)を Lagrange 緩和すると次の緩和問題 LG_1 を得る。

$$\begin{array}{ll}
(LG_1)\quad 最小化 & \sum_{i\in N}\sum_{j\in N'}f_{ij}y_{ij}+\sum_{i\in N}\sum_{j\in N}(v_i-v_j)\,x_{ij}+\sum_{j\in N'}v_j\\
\quad\quad 条件 & (12),\ (13),\ (14),\ (15)\ and\ (16)
\end{array}$$

ただし，x_{i0} と x_{j0} はダミー変数で，$v_0=0$ とおく。デザイン変数 x を固定すると，フロー変数 y は，v の値の比較によって簡単に求められる。さらに，デザイン変数 x は次の問題の最適解となる。

$$(LG_2)\quad \text{最小化} \quad \sum_{i\in N}\sum_{j\in N'}(f_{ij}+\gamma_{ij})\,y_{ij}$$
$$\text{条件} \quad (12)\ and\ (15)$$
$$y_{ij}\in\{0,1\}\quad for\ all\ i\in N,\ j\in N'$$

ここで,

$$\gamma_{ij}=\begin{cases}(v_i-v_j)(C-1) & \text{if } (v_i-v_j)<0\ and\ i\in N',\ j\in N'\\ -v_jC & \text{if } -v_j<0\ and\ j\in N'\\ (v_i-v_j) & \text{otherwise}\end{cases}$$

である。この緩和問題 LG_2 は貪欲的に解くことができ，$CMST$ の下界値を求めることができる。乗数 v は劣勾配法などで設定することができる。

Gavish[9] は，補助フロー変数を使った定式化を用いて，さらに強い Lagrange 緩和法も提案している。

ビンパッキング制約を用いた Lagrange 緩和法

Gavish[10] は，ビンパッキング制約を用いた定式化を示し，この制約を Lagrange 緩和して下界値を求める解法を示している。

適当なノード集合を S $(\subset N)$，ビンの長さ (容量) を C とし，ビンに詰める要素 i $(\in S)$ の長さ (需要) を d_i とする。このビンパッキング問題を解いて得られた最適ビン数を L_S とする。このとき，S に含まれるノードは L_S 個未満の連結成分には分割できないため，S 内には高々 $|S|-L_S$ 本のアークしか存在しない。したがって，次式は妥当な不等式となる。

$$\sum_{i\in S}\sum_{j\in S}y_{ij}\leq|S|-L_S$$

この制約をビンパッキング制約と呼ぶ。ビンパッキング制約は，ビンパッキング問題を解くことによって求めることができる。ビンパッキング問題自体は $\mathcal{NP}$-完全であるが，ここで対象とするビンパッキング問題の要素数は少ないため，動的計画法などによって解くことができる。

ビンパッキング制約とセンターの度数制約を用いた定式化 $CMST$ を示しておく。

$$(CMST_4)\quad \text{最小化} \quad \sum_{i\in N}\sum_{j\in N'}f_{ij}y_{ij}$$
$$\text{条件} \quad \sum_{i\in S}\sum_{j\in S}y_{ij}\leq|S|-L_S\quad for\ all\ S\in N',\ |S|\geq 2 \qquad (17)$$
$$(7),\ (8),\ (10)\ and\ (6)$$

ビンパッキング制約の数は非常に多いために，あらかじめすべてを列挙しておくことは得策ではない。そこで，ビンパッキング制約を取り除いた緩和問題から始め，その緩

和解を満足しないビンパッキング制約を探索し，逐次，追加していく方法を考える。

(10)式があるために，ビンパッキング制約を緩和した問題はセンターに度数制約をもつ最小木問題となるが，この問題は Gabow-Tarjan のアルゴリズム[7]を用いて $O(|N|^2)$ で解くことができる。

既に追加したビンパッキング制約に対応するノード集合 S の集合を Ω とし，ビンパッキング制約(17)に対して非負の乗数 v を用いて，Lagrange 緩和問題 LG_3 を作成する。

$$(LG_3)\quad \text{最小化}\quad \sum_{i\in N}\sum_{j\in N'}f_{ij}y_{ij}+\sum_{S\in\Omega}\sum_{i\in S}\sum_{j\in S}v_S y_{ij}-\sum_{S\in\Omega}v_S(|S|-L_S)$$
$$\text{条件}\quad (7),\ (8),\ (10)\ and\ (6)$$

適当な Lagrange 乗数 v が与えられたとき，LG_3 は容易に最適に解くことができ，$CMST$ の下界値を求めることができる。

Ω に含まれる S に対する Lagrange 乗数を $\tilde{v}$ と固定する。LG_2 の最適解を y^*，その木を T_s とする。次に，T_s において，ビンパッキング制約を満たさないノード集合の一つを S' とし，このビンパッキング制約を Lagrange 緩和する。

$$(LG_3')\quad \text{最小化}\quad \sum_{i\in N}\sum_{j\in N'}f_{ij}y_{ij}+\sum_{S\in\Omega}\sum_{i\in S}\sum_{j\in S}\tilde{v}_S y_{ij}$$
$$-\sum_{S\in\Omega}\tilde{v}_S(|S|-L_S)+v_{S'}\{\sum_{i\in S'}\sum_{j\in S'}y_{ij}-(|S'|-L_{S'})\}$$

ここで，

$$v_{S'}=\min_{\{(i,j)|i\in S',\ j\in N\setminus S'\}}\{f_{ij}+\sum_{S\in\Omega_{ij}}v_S\}-\max_{\{(i,j)|i\in S',\ j\in S'\}}\{f_{ij}+\sum_{S\in\Omega_{ij}}v_S\}$$

とすれば，下界値は $v_{S'}\{\sum_{i\in S'}\sum_{j\in S'}y^*_{ij}-(|S'|-L_{S'})\}$ だけ増加することができる。この Lagrange 乗数設定法の計算量は $O(|N|^2)$ であり，効率的に Lagrange 乗数を設定することができる。

Malik-Yu の妥当不等式

Malik-Yu[18]の妥当不等式を示す。妥当不等式は実行可能解を満足する不等式である。強い妥当不等式であれば，線形緩和した問題に制約条件として追加すると，有効な制約式となり，下界値を増加することができる。

N' の部分集合 S には，少なくとも $\lceil \sum_{i\in S}d_i/C\rceil$ 個のコンポーネントが存在する。このため，S から $N\setminus S$ をつなぐアークの最小数も $\lceil \sum_{i\in S}d_i/C\rceil$ となる。したがって，次式は $CMST$ の妥当不等式となる。

$$\sum_{i\in S}\sum_{j\in N'\setminus S}y_{ij}\geq\lceil \sum_{i\in S}d_i/C\rceil\qquad S\subset N'$$

S に含まれるノードによってコンポーネントを形成したときに，センターを除くノー

ドをこのコンポーネントに追加できない，すなわち $\sum_{i\in S}d_i+\min_{i\in N'\setminus S}d_i>C$ を S が満足するものとする。このとき，S 内のノードが連結して1つのコンポーネントを形成すれば，$\sum_{i\in S}\sum_{j\in S}y_{ij}=|S|-1$ であり，このコンポーネント内のノードからセンターへつなぐことが必要である。一方，S 内のノードが非連結であれば，S 内のアーク数は $|S|-2$ 本以下である。したがって，次式は $CMST$ の妥当不等式となる。

$$\sum_{i\in S}\sum_{j\in S}y_{ij}\leq|S|-2+\sum_{j\in N'}y_{0j}\quad S\subset N',\ \sum_{i\in S}d_i+\min_{i\in N'\setminus S}d_i>C$$

S が前述の条件を満たすとき，S に含まれるノードがすべて連結していれば，$\sum_{i\in S}\sum_{j\in S}y_{ij}=|S|-1$ かつ $\sum_{i\in S}\sum_{j\in N'\setminus S}y_{ij}=0$ である。一方，非連結であれば $\sum_{i\in S}\sum_{j\in S}y_{ij}<|S|-1$ となるが，S 内のノードから S 以外のノードをつなぐことができるアーク数の最大値は $|N|-|S|$ 本であるので，$\sum_{i\in S}\sum_{j\in N'\setminus S}y_{ij}\leq|N|-|S|$ である，したがって，次式は $CMST$ の妥当不等式となる。

$$\sum_{i\in S}\sum_{j\in N'\setminus S}y_{ij}\leq(|N|-|S|)(|S|-1-\sum_{i\in S}\sum_{j\in S}y_{ij})$$
$$S\subset N',\ \sum_{i\in N'}d_i+\min_{i\in N'\setminus S}d_i>C$$

これらの妥当不等式を Lagrange 緩和した緩和問題は最小木問題となるため，容易に解くことができる。これらの妥当不等式は数が非常に多いため，実際には，緩和解が満足しない妥当不等式を逐次生成し，Lagrange 緩和を行うことになる。Lagrange 乗数は劣勾配法によって設定することができる。

2N 制約と Gouveia の妥当不等式

Gouveia[13) は，すべての需要が1であるモデルに対して，$2|N|$ 個の制約式を使った定式化と妥当不等式を示している。はじめに，フロー変数 x を用いた定式化を示しておく。ここで，d_i はノード i を終点とする需要量を表すが，$i=0$ のときは0，それ以外は1である定数とする。

z_{ij}^q をアーク $(i,\ j)$ 上のフロー量が q であるとき1，そうでないときに0であるフロー変数とし，$Q_i=\{1,\ \cdots,\ C-d_i\}$ とする。このとき，y_{ij}，x_{ij} と z_{ij}^q の関係は次のようになる。

$$y_{ij}=\sum_{q\in Q_i}qz_{ij}^q\quad for\ all\ i\in N,\ j\in N' \tag{18}$$
$$x_{ij}=\sum_{q\in Q_i}qz_{ij}^q\quad for\ all\ i\in N,\ j\in N' \tag{19}$$

この z を用いると，$2|N'|$ 個の制約式を用いて $CMST$ を定式化することができる。

$$(CMST_5)\quad \text{最小化}\quad \sum_{i\in N}\sum_{j\in N'}\sum_{q\in Q_i} f_{ij} z_{ij}^q$$

$$\text{条件}\quad \sum_{j\in N}\sum_{q\in Q_i} z_{ij}^q = 1 \quad for\ all\ i\in N' \tag{20}$$

$$\sum_{i\in N}\sum_{q\in Q_i} q z_{in}^q - \sum_{j\in N'}\sum_{q\in Q_n} q z_{nj}^q = 1 \quad for\ all\ n\in N' \tag{21}$$

$$z_{ij}^q \in \{0,\ 1\} \quad for\ all\ q\in Q_i,\ i\in N,\ j\in N' \tag{22}$$

ここで，次の妥当な制約式を追加しておく。

$$\sum_{i\in S}\sum_{j\in S}\sum_{q\in Q_i} z_{ij}^q \leq |S|-1 \quad for\ all\ S\subset N'\ and\ |S|>2 \tag{23}$$

この定式化に対して，次の 4 種類の式は妥当な不等式となる。

$$\sum_{i\in N}\sum_{j\in N'} z_{ij}^q \leq \lfloor |N'|/q \rfloor \quad q=1,\ \cdots,\ C-1 \tag{24}$$

$$\sum_{i\in N'}\sum_{j\in N'} z_{ij}^q \leq \lfloor |N'|/(q+1) \rfloor \quad q=\lceil C/2 \rceil,\ \cdots,\ C-1 \tag{25}$$

$$\sum_{j\in N'} z_{0j}^C \leq \lfloor |N'|/C \rfloor \tag{26}$$

$$\sum_{p=q}^{C}\sum_{j\in N'} z_{0j}^p \leq \lfloor |N'|/q \rfloor \quad q=1,\ \cdots,\ C \tag{27}$$

(24)式は，フロー量 q をもつアークはその子孫に q 個のノードを持つため，その本数は高々 $\lfloor |N'|/q \rfloor$ 本であることを意味している。$\lceil C/2 \rceil$ 以上のフローを持つセンターに接続しないアークは，相異なるコンポーネントに含まれる。さらに，このようなアークが存在するためには，その一方の端点と子孫の q 個の計 $q+1$ 個のノードが必要である。したがって，(25)式は妥当不等式となる。(26)式は，センターに接続し，フロー量 C をもつアークはその子孫に C 個のノードを持つため，高々 $\lfloor |N'|/C \rfloor$ 本であることを意味している。(27)式は，q 個以上のノードをもつコンポーネントは $\lfloor |N'|/q \rfloor$ 個以下であることを意味している。

制約式(21)式に対して乗数 v を用いて Lagrange 緩和し，妥当不等式の内の(25)式を加え，さらに乗数 u を用いて Lagrange 緩和した緩和問題 LG_4 を作成する。

$$(LG_4)\quad \text{最小化}\quad \sum_{i\in N}\sum_{j\in N'}\sum_{q\in Q_1} w_{ij}^q z_{ij}^q + \sum_{j\in N'} v_j + \sum_{q=\lceil |C|/2 \rceil}^{C-1} \lfloor |N'|/(q+1) \rfloor u_q$$

$$\text{条件}\quad (20),\ (22)\ and\ (23)$$

ここで，

$$w_{ij}^q = \begin{cases} f_{ij} + q(v_i - v_j) & if\ i\in N',\ j\in N',\ q=1,\ \cdots,\ \lceil |C|/2 \rceil -1 \\ f_{ij} + q(v_i - v_j) - u_q & if\ i\in N',\ j\in N',\ q=\lceil |C|/2 \rceil,\ \cdots,\ C-1 \\ f_{ij} - q v_j & if\ i=0,\ j\in N',\ q=1,\ \cdots,\ C \end{cases}$$

とする。この問題は，アーク $(i,\ j)$ の重みを $\min_{q\in Q_i} w_{ij}^q$ とした最小 Arborescence 問題となり，比較的容易に解くことができ，$CMST$ の下界値を求めることができる。また，Lagrange 乗数は劣勾配法などで設定することができる。

Hall の妥当不等式

Hall[14] は，マルチスター不等式とルート－切断集合不等式を示し，これらの不等式を用いた切除平面法を示している。

ある $S\ (\subset N)$ に対して，S 内のノードが 1 つのコンポーネントを構成する場合には，

$$\sum_{i\in S}\sum_{j\in S}Cy_{ij}=C(|S|-1)\leq C|S|-\sum_{u\in S}d_u$$

が成り立つ。一方，$i\ (\in S)$ かつ $j\ (\in S)$ であるアークが存在しない場合は，$u\ (\in S)$ に対して $\sum_{v\in N\setminus S}d_v(y_{uv}+y_{vu})+d_u\leq C$ であり，

$$\sum_{u\in S}\sum_{v\in N\setminus S}d_v(y_{uv}+y_{vu})\leq C|S|-\sum_{u\in S}d_u$$

が成り立つ。このとき，アーク (i, j) $(i, j\in S)$ を付加して i と j を連結すると，$\sum_{i\in S}\sum_{j\in S}Cy_{ij}$ の値は C 増加するが，$\sum_{v\in N\setminus S}d_v(y_{iv}+y_{vi})+d_i+\sum_{v\in N\setminus S}d_v(y_{uj}+y_{vj})+d_j$ の取りうる最大値は $2C$ から C になり，C だけ減少する。このことから，次式は妥当不等式となることが分かる。

$$\sum_{i\in S}\sum_{j\in S}Cy_{ij}+\sum_{u\in S}\sum_{v\in N/S}d_v(y_{uv}+y_{vu})\leq C|S|-\sum_{u\in S}d_u$$

$U_p^v=\{u\in N|d_u>(C-d_v)/p\}$ とする。このとき，U_p^v に含まれる p 個のノードと v をアークで接続すると容量を超えて実行不可能となる。さらに，U_p^v に含まれるノードの需要の内で，小さい順に $p-1$ 番目までのものを $\delta_1, \cdots, \delta_{p-1}$ とし，$d_w>C-d_v-(\delta_1+\cdots+d_{p-1})$ を満たすノードを $w\ (\notin U_p^v)$ とする。このとき，U_p^v に含まれる $p-1$ 個のノードと v をアークで接続し，かつ w と v をアークで接続すると容量を超えて実行不可能となる。このことから，次式は妥当不等式となる。

$$y_{wv}+y_{vw}+\sum_{u\in U_p^v}(y_{uv}+y_{vu})\leq p-1$$

これらの 2 つ妥当不等式は，マルチスター不等式と呼ばれている。

$W_s=\{u\in N\setminus S|d_u+\sum_{v\in S}d_v\leq C\}$ とする。$S\ (\subset N)$ 内のノードを両端点するアークの数は高々 $|S|-1$ 本である。また，S 内のノードが一つのコンポーネントに含まれる場合には，S 内のノードは容量を満足できる W_s に含まれるノード（センターを含む）に接続する必要がある。このことから，次式は妥当不等式となる。

$$\sum_{i\in S}\sum_{j\in S}y_{ij}\leq|S|-2+\sum_{v\in S}\sum_{v\in W_S}y_{uv}$$

この妥当不等式は，ルート-切断集合不等式と呼ばれている。

最小費用フロー緩和法

Toth-Vigo[22] は，ノードの部分集号間のアーク集合に関する緩和問題が最小費用フロ

ー問題に帰着できることを示し，この緩和問題を用いた分枝限定法を示している。

はじめに，ここで用いる定式化を示しておく。

$$
\begin{array}{lll}
(CMST_6) & \text{最小化} & \sum_{i\in N}\sum_{j\in N'} f_{ij}y_{ij} \\
& \text{条件} & \sum_{j\in N'} y_{0j} \geq \lceil \sum_{i\in N'} d_i/C \rceil \\
& & \sum_{i\in N'\setminus S}\sum_{j\in S} y_{ij} \geq L_S \quad for\ all\ S\in N',\ |S|\geq 2 \\
& & (3),\ (6)\text{式}
\end{array}
$$

N' を $|M|\ (>2)$ 個のノード集合 $S_h\ (\in M, |S_h|>0)$ に分割し，分割したノード集合内のノード間をつなぐアーク集合を A_1，分割したノード集合間のノードをつなぐアーク集合を A_2 とする。

$$A_1=\bigcup_{h\in M}\{(i,\ j)\,|\,i\in S_h,\ j\in S_h\},\quad A_2=A\setminus A_1$$

$\sum_{(i,j)\in A_1} f_{ij}y_{ij}$ の下界値を θ_1，$\sum_{(i,j)\in A_2} f_{ij}y_{ij}$ の下界値を θ_2 とし，$CMST$ の下界値を θ とすると，$\theta=\theta_1+\theta_2$ の関係となる。

$\theta_1=0$ としておく。次に，θ_2 に関して，次の緩和問題 RP を作成する。

$$
\begin{array}{llll}
(RP) & \text{最小化} & \theta_2=\sum_{(i,j)\in A_2} f_{ij}y_{ij} & \\
& \text{条件} & \sum_{i|(i,j)\in A_2} y_{ij}\leq 1 \quad for\ all\ i\in N' & (28) \\
& & \sum_{j\in N'} y_{0j} \geq \lceil \sum_{i\in N'} d_i/C \rceil & \\
& & \sum_{i\in N'\setminus S_h}\sum_{j\in S_h} y_{ij} \geq L_{S_h} \quad for\ all\ S_h\in M & (29) \\
& & y_{ij}\in\{0,\ 1\} \quad for\ all\ (i,\ j)\in A_2 &
\end{array}
$$

この問題は，(3)式に対応する(28)式の等号を不等号とし，ビンパッキング制約(29)式を S_h $(\in M)$ に限定した問題である。したがって，θ_2 は $\sum_{(i,j)\in A_2} f_{ij}y_{ij}$ の妥当な下界値となる。

この問題は，補助ネットワーク上の最小費用フロー問題と見なすことができる。この補助ネットワークは，次に示すノードとアークから構成される。

・すべてのノード $i\ (\in N')$ に対して，ノード i^+，i^- をおく。

・センター0に対するソースノード 0^+ をおき，供給量を $\lceil \sum_{i\in N'} d_i/C \rceil$ とする。

・すべての $S_h\ (\in M)$ に対して，ノード a^h，b^h をおく。

・供給量 $|N|-1-\lceil \sum_{i\in N'} d_i/C \rceil$ をもつソースノード s と，需要量 $|N|-1$ をもつシンクノード t をおく。

・すべてのアーク $(i,\ j)\ (\in A_2)$ に対して，費用 f_{ij} と容量1をもつアーク $(i^+,\ j^-)$ をおく。

・すべてのノード $j\ (\in N')$ に対して，費用 f_{0j} と容量1をもつアーク $(0,\ j^-)$ をおく。

・すべての $S_h\ (\in M)$，$i\ (\in S_h)$ に対して，費用0と容量 ∞ をもつアーク $(a_h,\ i^+)$ をおく。

・すべての S_h ($\in M$)，i ($\in S_h$) に対して，費用 0 と容量 1 をもつアーク (i^-, b_h) をおく。

・すべての S_h ($\in M$) に対して，費用 0 と容量 $|S_h|-L_{S_h}$ をもつアーク (a_h, b_h) をおく。

・すべての S_h ($\in M$) に対して，費用 0 と容量 ∞ をもつアーク (s, a_h) をおく。

・すべての S_h ($\in M$) に対して，費用 0 と容量 $|S_h|$ をもつアーク (b_h, t) をおく。

・費用 0 と容量 ∞ をもつアーク (s, 0) をおく。

この補助ネットワークにおけるアーク (i^+, j^-) 上の最小費用フローを $flow_{ij}$, (0, j^-) 上の最小費用フローを $flow_{0j}$ とし，緩和問題 RP の最適解を $\tilde{y}$ とする。このとき，$\tilde{y}_{ij}=flow_{ij}$ ($i\in N$, $j\in N'$) であり，$\theta_2=\sum_{(i,j)\in A_2} f_{ij}\tilde{y}_{ij}$ となる。緩和問題 RP を解くための計算量は，最小費用フロー問題の計算量であり，$O(|N|^3)$ である。

異なる分割 S_h ($\in M$) に対して，異なる補助ネットワークが対応し，異なる下界値が得られる。ここで，$S_h=\{h\}$ ($h\in N$) とおけば，RP は容量制約を取り除いた緩和と同じ緩和問題となる。一方，S_h が複数のノードを含めば，S_h 内のアークの下界値 θ_1 を考慮することはできないが，その容量制約を考慮することができる。この解法は，実際には Lagrange 緩和法などと組み合せて使用し，その残余費用をアーク費用とした問題に適用し，下界値を上昇させるために用いられる。

6．おわりに

本論文では，通信ネットワーク設計問題などに発生する基本的な問題である容量制約をもつ最小木問題に対する従来の研究のサーベイを行った。最新の研究では，$CMST$ の近似解法としてはメタヒューリスティック解法であるタブサーチ法など，最適解法としては強い妥当不等式を生成し，これと Lagrange 緩和法を組み合わせる解法などが開発されている。

今後の開発されるべき $CMST$ の近似解法として，シミュレーテッドアーニリング法や対象とする問題の構造を考慮したメタヒューリスティック解法が挙げられる。また，最適解法としては，大規模な組合せ問題に効果を挙げている分枝カット法などが考えられる。また，強い妥当不等式やファセットを生成するための多面体理論の展開も必要であると考える。

(参考文献)

1）K. Altinkemer and B. Gavish. Heuristics with constant error guarantees for the design of tee networks. *Management Science*, Vol. 34, pp. 331-341, 1988.

2）R. Boorstyn and H. Frank. Large-scale network topological optimization. *IEEE Transactions on Communications*, Vol. 25, pp. 29-47, 1977.

3) K. Chandy and T. Lo. The capacitated minimum spanning tree. *Networks*, Vol. 3, pp. 173-181, 1973.

4) K. Chandy and R. Russell. The design of multipoint linkages in a teleprocessing tree network. *IEEE Transactions on Computers*, Vol. C-21, pp. 1062-1066, 1972.

5) D. Elias and M. Ferguson. Topological design of multipoint teleprocessing networks. *IEEE Transactions on Communications*, Vol. 22, pp. 1753-1763, 1974.

6) L. Esau and K. Willams. On teleprocessing system design. Part II. *IBM System Journal*, Vol. 5, pp. 142-147, 1966.

7) H. Gabow and R. Tarjan. Efficient algorithms for a family of matroid intersection problems. *Journal of Algorithms*, Vol. 5, pp. 80-131, 1984.

8) B. Gavish. Topological design of centralized computer networks—formulation and algorithms. *Networks*, Vol. 12, pp. 355-377, 1982.

9) B. Gavish. Formulations and algorithms for the capacitanted minimal directed tree problem. *Journal of ACM*, Vol. 30, pp. 118-132, 1983.

10) B. Gavish. Augmented Lagrangean based algorithms for centralized network design. *IEEE Transactions on Communications*, Vol. 33, pp. 1247-1257, 1985.

11) B. Gavish. Topological design of telecommunication netwwrks—local access design methods. *Annals of Operations Research*, Vol. 33, pp. 17-71, 1991.

12) B. Gavish and K. Altinkemer. A parallel saving heuristic for the topological design of local access tree networs. *Proceeding IEEE-INFOCOM '86*, pp. 130-139, 1986.

13) L. Gouveia. A 2n constraint formulation for the capacitated minimal spanning tree problem. *Operations Research*, Vol. 45, pp. 130-141, 1995.

14) L. Hall. Experience with a cutting plane algorithm for the capacitated spanning tree problem. *INFORMS Jounal on Computing*, Vol. 8, pp. 219-234, 1996.

15) M. Karnaugh. A new class of algorithms for multipoint network optimization. *IEEE Transactions on Communications*, Vol. 24, pp. 500-505, 1976.

16) A. Kershenbaum, R. Boorstyn, and R. Oppenhein. Second order greedy algorithms for centralized teleprocessing network design. *IEEE Transactions on Communications*, Vol. 28, pp. 1835-1838, 1980.

17) A. Kershenbaum and W. Chou. A unified algorithm for designing multidrop teleprocessing networks. *IEEE Transactions on Communications*, Vol. 22, pp. 1762-1773, 1974.

18) K. Malik and G. Yu. A branch and bound algorithm for the capacitated minimum spanning tree problem. *Networks*, Vol. 23, pp. 525-532, 1993.

19) C. Papadimitriou. The complexity of the capacitated tree problem. *Networks*, Vol. 8, pp. 217-230, 1978.

20) Y. Sharaiha, M. Gendreau, G. Laporte, and I. Osman. A tabu search algorithm for the capacitated shortest spanning tree problem. *Networks*, Vol. 29, pp. 161-171, 1997.

21) R. Sharma. Design of an economical multidrop network topology with capacity constaints. *IEEE Transactions on Communications*, Vol. 31, pp. 590-591, 1983.

22) P. Toth and D. Vigo. An exact algorithm for the capacitated shortest spanning arborescence. *Annals of Operations Research*, Vol. 61, pp. 121-141, 1995.