ネットワーク上の被覆型施設配置問題

大阪大学 *新 浩治 ARATA Koji
大阪大学 岩田 覚 IWATA Satoru
大阪大学 牧野 和久 MAKINO Kazuhisa
大阪大学 藤重 悟 FUJISHIGE Satoru

1. はじめに

各点の要求に対して複数個の施設からフローを供給するネットワークに関して，最小の施設数で各点の要求容量を満足する配置を求める問題を総合被覆型施設配置問題 (総合被覆問題) という．この問題を解く多項式時間のアルゴリズムが, 田村, 菅原, 仙石, 篠田 [2] によって提案されている．

本論文では，このアルゴリズムの正当性について，数理計画法に基づいた別証明を与える．また，このアルゴリズムの計算時間を高速化する．さらに，施設の設置費用を考慮した最小費用総合被覆問題を提起する．

2. 総合被覆問題

点集合 V, 枝集合 E からなる無向ネットワーク $N=(V,E,c,h)$ を考える．ここで，c は各枝 $e\in E$ の枝容量 $c(e)$ を表す非負関数であり，h は各点 $v\in V$ の要求容量 $h(v)$ を表す非負関数である．

点集合 X から点 v への最大フロー値を $f(X,v)$ と表す．点集合 V の空でない真部分集合をカットと呼び，カット W について，$g(W)=\sum\{c(u,v)\mid u\in W,\ v\in V-W\}$ をカット W の容量と呼ぶ．

点部分集合 $U\subseteq V$ が任意の $v\in V$ に対して $f(U,v)\geq h(v)$ となるとき，U は N の h-総合被覆であるという．総合被覆問題とは，総合被覆のうちで要素数最小なものを求める問題である．点部分集合 W が $W=V$ であるか，または $g(W)<\max\{h(v)|v\in W\}$ であるとき，W を未充足集合と呼ぶ．未充足集合で，真部分集合として未充足集合を含まないものを極小な未充足集合という．また，$W\subseteq V$ に対して，$h(W)=\max\{h(v)|v\in W\}$ と定義する．$h(v)=h(W)$ となる点 $v\in W$ を W の要求容量最大点という．

3. 貪欲アルゴリズム

田村, 菅原, 仙石, 篠田 [1] によるアルゴリズム A は以下の通りである．

Step 1 $V=\{v_1,\cdots,v_n\}$ を $h(v_1)\leq\cdots\leq h(v_n)$ となるように並べる．$U:=V$, $j:=1$ とする．

Step 2 $U-\{v_j\}$ が N の総合被覆ならば，$U:=U-\{v_j\}$.

Step 3 $j=n$ のとき U を出力して終了．
そうでなければ $j:=j+1$ として Step 2 へ． □

極小未充足集合を $W_1,\cdots,W_m$ と表す．行列 $A=(a_{ij})$ を，

$$a_{ij}=\begin{cases}1 & (v_j\in W_i)\\ 0 & (v_j\notin W_i)\end{cases}$$

によって定める．

無向フローネットワークの性質として，以下の補題1が成り立つことが知られている．

補題 1 N を無向フローネットワークとする．点集合 $X,\ Y\subseteq N$ に対して，

$$\begin{aligned}g(X)+g(Y) &\geq g(X\cup Y)+g(X\cap Y)\\ g(X)+g(Y) &\geq g(X-Y)+g(Y-X).\end{aligned}$$

補題1より，次の補題2が導かれる．

補題 2 行列 A において，行 i の最も右の非零列を $k(i)$ とする．このとき次のような部分行列が存在しない．

$$\begin{array}{ccc} & & \begin{array}{ccc} j & k(i_2) & k(i_1)\end{array}\\ j<k(i_1) & i_1 & \multirow{2}{*}{$\left(\begin{array}{ccc}1 & 0 & 1\\ 1 & 1 & 0\end{array}\right).$}\\ j<k(i_2) & i_2 & \end{array}$$

総合被覆問題は，v_jが解であるとき $x_j=1$ とする変数 $x_j\in\{0,1\}$ によって，整数計画法を用いて以下のように定式化される．

$$\begin{aligned}\text{Min.}\ & \textstyle\sum_{j=1}^{n}x_j\\ \text{s. t.}\ & \textstyle\sum_{j=1}^{n}a_{ij}x_j\geq 1 \quad (i=1,\cdots,m)\\ & x_j\in\{0,1\}\quad (j=1,\cdots,n).\end{aligned}$$

この問題の線形緩和問題

$$\begin{aligned}\text{Min.}\ & \textstyle\sum_{j=1}^{n}x_j\\ \text{s. t.}\ & \textstyle\sum_{j=1}^{n}a_{ij}x_j\geq 1 \quad (i=1,\cdots,m)\\ & x_j\geq 0\quad (j=1,\cdots,n)\end{aligned}$$

の双対問題は以下のようになる．

$$\begin{aligned}\text{Max.}\ & \textstyle\sum_{i=1}^{m}y_i\\ \text{s. t.}\ & \textstyle\sum_{i=1}^{m}a_{ij}y_i\leq 1 \quad (j=1,\cdots,n)\\ & y_i\geq 0\quad (i=1,\cdots,m).\end{aligned}$$

このように線形緩和問題に置き換えることで，線形計画法の双対定理を用いることができるようになる．

変数 x は明らかに線形緩和問題の実行可能解となる．線形緩和問題とその双対問題の相補性条件を満たすように，アルゴリズムの実行中に y を適当な (0,1)-ベクトルに定め，その y が双対問題の実行可能解であることを示す．

相補性条件は以下の通りである．

(CS1) $x_j\neq 0$ であるような j について

$$\textstyle\sum_{i=1}^{m}a_{ij}y_i=1.$$

(CS2) $1<\sum_{j=1}^{n}a_{ij}x_j$ であるような i について

$$y_i=0.$$

相補性条件 (CS1), (CS2) を満たすよう，以下の約束事に基づいて y を定める．

(1) y_i:=0 ($i=1,\cdots,m$) と初期設定する．

(2) 最適解として選ばれた点 v_j を要求容量最大点として含んでいる W に対応する y_iのうち，1つを y_i:=1 とし，それ以外の y_i=0 を確定する．

(3) その v_j を要求容量最大点以外として含んでいる W に対応する y_i=0 を確定する．

補題 3 y は双対問題の実行可能条件を満たす．

証明 $x_j=1$ の場合は (CS1) より明らか．$x_j=0$ の場合を証明する．

$x_j=0$ である点 v_j を共有する W が2つ存在し，W_1, W_2 とする．このとき W_1, W_2に対応する $y_1=y_2=1$ であることを仮定する．

この仮定の下では，W_1, W_2がお互い異なる要求容量最大点を持ち，かつ一方の最大点を，他方が共有していないことが言える．しかし，それは補題2より矛盾する．よって $a\cdot y_1$ と $a\cdot y_2$は同時に1とはならないため，$x_j=0$ においても制約条件は成り立つ． □

定理 1 アルゴリズム A の出力は最適解となる．

証明 (CS1), (CS2) と補題3よりアルゴリズム終了時の x は緩和問題における最適解である．このとき，この解 x は $x_j\in\{0,1\}$ であるため，元の問題の制約条件をも満たす．したがって元の問題の最適解となる． □

4. アルゴリズムの高速化

無向フローネットワークの点数を n, 枝数を m として，アルゴリズム A の計算時間を解析する．Step 1 の並べ替えの手間は $O(n\log n)$. Step 2 ~ Step 3 の部分は，2点間の最大フロー最小カットを求める手間を M とすると，1回のループの手間は $O(n\cdot M)$ となり，さらにループ回数が n であるので，全体の計算時間は $O(n^2M)$ となる．

このアルゴリズムは，まずネットワーク上のすべての点 V を実行可能解集合 U の初期値とし，要求容量の小さな点から順に，最適解として残すべきか検討している．カット表現を用いると，U に含まれる要求容量最小の点 v_t による集合 $U-\{v_t\}$ について，各点 $v_j\notin U-\{v_t\}$ との間の最小カットを調べ，カット容量が $h(v_j)$ を下回るような カット が存在すれば，$U-\{v_t\}$ とはせず U を保つことになる．

補題 4 点 v_tを U に残すかどうかの判定に要する計算量は $O(M)$ である．

証明 点 v_tを残すかどうか検討するために調べる，点集合 $U-\{v_t\}$ と v_jを分けるカットは，v_tを含むカット X と，v_tを含まないカット Y の2種類に大別される．

まず X のようなカットを最小カットとして持つ場合は，より以前の検討でカット値 $g(X)$ を調べているため，再び調べる必要はない．

次に Y のようなカットを最小カットとして持つ場合，v_tが最適解として必要でないときに，以下の不等式

$$g(Y)\geq f(U-\{v_t\},v_t)\geq h(v_t)\geq h(v_j)$$

が成り立つため，$U-\{v_t\}$ と v_tにおける最小カットのみを調べれば十分である．

よって，カットの計算量は $O(M)$ となる． □

以上より，アルゴリズム A の **Step 2** を変更する．

Step 2 $f(U-\{v_j\},\ v_j)\geq h(v_j)$ である場合は，
U:=$U-\{v_j\}$ とする．

1回のループにかかる計算量は，$O(M)$ となり，それを高々 n 回繰り返せばよいので $O(n\cdot M)$ となる．これは以前の $O(n^2\cdot M)$ を大きく改善している．

5. 最小費用総合被覆問題

最小費用総合被覆問題とは，総合被覆問題において，x の費用係数 $d_j\geq 0$ を導入して一般化した問題である．

最小費用総合被覆問題は次のように定式化できる．

$$\begin{array}{ll} \text{Min.} & \sum_{j=1}^{n} d_j x_j \\ \text{s. t.} & \sum_{j=1}^{n} a_{ij} x_j \geq 1 \quad (i=1,\cdots,m) \\ & x_j = 0,1 \quad (j=1,\cdots,n). \end{array}$$

定理 2 最小費用総合被覆問題は NP-困難である．

証明 以下のナップサック問題

$$\begin{array}{ll} \text{Min.} & \sum_{j=1}^{n} d_j x_j \\ \text{s. t.} & \sum_{j=1}^{n} c_j x_j \geq \sum_{j=1}^{n} c_j - b \\ & x_j = 0,1 \quad (j=1,\cdots,n) \end{array}$$

において，変数を

$$h_1=h_2=\cdots=h_{n-1}=0,\ \ h_n=\sum_{j=1}^{n} c_j - b$$
$$d_n=\sum_{j=1}^{n} d_j + 1$$

とおくことによって，要求容量 h_j，費用 d_j であり，点 v_n と他の点 $v_1,\cdots,v_{n-1}$のみを枝容量 c_j ($j=1,\cdots,n-1$) の枝でつないだグラフに帰着できるため，最小費用総合被覆問題は NP-困難である． □

この最小費用総合被覆問題に対して，要求容量と費用の比により点を並べ替えて，アルゴリズム A を適用することを検討している．

良い近似解を得るためのアルゴリズムの開発が，今後の研究課題である．

参考文献

[1] H. Tamura, M. Sengoku, S. Shinoda, and T. Abe: Some covering problems in location theory on flow networks, *IEICE Trans. Fundamentals*, **E75-A** (1992), 678-683.

[2] 田村, 菅原, 仙石, 篠田：無向フローネットワークにおける総合被覆問題について, 電子情報通信学会論文誌, **J81-A** (1998), 863-869.