1995年度日本オペレーションズ・リサーチ学会
秋季研究発表会

木構造ネットワーク上の部分木配置問題に対する高速解法

02003590 東京工業大学 *宇野 毅明 UNO Takeaki
02201890 東京工業大学 塩浦 昭義 SHIOURA Akiyoshi

ネットワーク上の施設配置に対して、今までに多くの問題が考えられている。またその応用も様々である。本稿ではその1つである木構造ネットワークへの部分木配置問題に対する線形時間の解法を提案する。

1．部分木配置問題

$G=(V,A)$ を木構造を持つ有向グラフとする。ただしここでの木構造グラフは根付き木ではなく、無向グラフの木を有向グラフに拡張したもの、即ち $(u,v)\in A$ に対して $(v,u)\in A$ が存在し、u から v へのパスは一意的に決まるものとする。まず、このグラフに対して各種の定義を行う。$c:A\rightarrow R$ を枝の費用の関数 、$l:A\rightarrow R$ を枝の長さとする。両者ともに負の値を許すとする。G の有向パス P に対して、その長さ $l(P)$ をそのパスに含まれる枝の長さの総和とする。G の根付き木 F に対して、F の頂点から v への有向パスで、枝の数が最小なものを $P(F,v)$ と表す。F から最も遠い頂点への距離を 最遠距離 $MD(F)=\max_{v\notin F} l(P(F,v))$ とする。また、F の費用 $c(F)$ を F に含まれる枝の費用の総和とする。

以上の関数を利用して、費用がある定数 C 以下であるような根付き部分木の施設の中で、最遠距離を最小にするものを求める問題を定式化する。

$$\begin{aligned} \text{min.} \quad & MD(F) \\ s.t. \quad & F \text{は} G \text{の根付き木}, \\ & c(F)\leq C. \end{aligned}$$

ここでは考えるネットワークを木構造のものに限定しているので、問題の範囲を狭めていることになる。しかし、一般のネットワーク上においてはこの問題は NP-hard に属することが知られているので [1]、問題の簡素化を図ることには意味がある。また、グラフの中の1部分、例えば最短路木や最小全張木に注目し、その部分についてのこの問題を解くことにより、元の問題のある意味での近似解や指標を得ることが出来る。

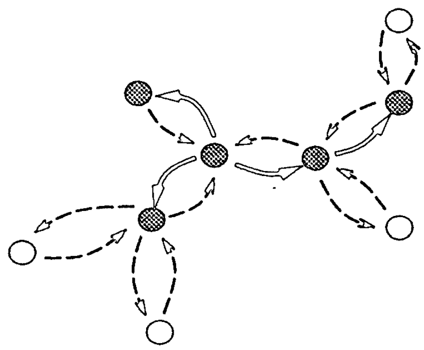

図1：木構造ネットワーク上での施設配置
太線が根付き部分木の施設

2．アルゴリズム

アルゴリズムを説明するにあたり、幾つかの準備を行う。関数 $md:A\rightarrow R$ を、$md((u,v))=\max_{(u,v)\in P(u,w)} l(P(u,w))$ と定義すると、上の問題を同値の問題に変形出来る。

$$\begin{aligned} \text{min.} \quad & \max_{a=(u,v),u\in F,v\notin F} md(a) \\ s.t. \quad & F \text{は} G \text{の根付き木}, \\ & c(F)\leq C. \end{aligned}$$

md は A の関数であるので、高々 $|A|$ 種類の値しか取りえない。この性質を利用し、md の値、即ち目的関数に対する2分探索を行うアルゴリズムを考える。

*Step*1 : $M=A$ とする。
*Step*2 : $\{md(a):a\in M\}$ の中で $md(a)$ の中央値 b を求める。
*Step*3 : 目的関数値が b 以下である木の中で、費用を最小化する F^* を求める。
*Step*4 : $c(F^*)\leq C$ であれば $md(a)>b$ なる

a を M から除く。そうでなければ
$md(a) < b$ となるものを捨てる。

*Step*5 : M の要素が1個になったら終了。
そうでなければ *Step*2 へ。

3．アルゴリズムの実現

*Step*1, 2, 4, 5 は $O(|V|)$ の時間で容易に実現できる。そこで *Step*3 の実現方法を示す。

枝 $a=(u,v)$ に対して、G から $(u,v),(v,u)$ を取り除いて出来る2つの連結成分のうち v を含むほうを $T(a)$ と表記する。

初めに、各枝 $a=(u,v)\in A$ について、$T(a)$ の v を根とし、$MD(F)\leq b$ である根付き木の中で費用最小なもの $F^*(a)$ を求める。これは、$T(a)$ で v に接続する枝 (v,w) について再帰的に同様の問題を解くことによって求められるので、深さ優先探索を行うことにより、$O(|V|)$ の時間で計算できる。

この $F^*((u,v))$ を使い、各頂点 u について、u を根とし最遠距離が b 以下の根付き木の中で費用を最小にするものを計算する。後はこれを最小にする頂点 u^* を発見すればよい。

以上の操作は $O(|V|)$ の時間がかかり、アルゴリズムは全体で $O(\log|V|)$ の反復を行うので、上記のアルゴリズムの時間計算量は $O(|V|\log|V|)$ となる。$O(|V|)$ の計算時間を達成するためには、アルゴリズムの各反復に置いて以下の縮約操作を行えばいい。

アルゴリズムの反復中にある頂点集合 $V'\subset V$ に誘導される部分グラフ G' のすべての枝が M に含まれなくなったとする。そのような部分グラフについては、頂点 $v_1,\ldots,v_r$ のそれぞれについて、その頂点を根とする根付き木の中で費用最小なものを、各 v_i をそれぞれ含む、含まない場合について求めておけば、*Step*3 の深さ優先探索で G' の枝についての探索を省略することが出来る。ただし、$v_1,\ldots,v_r$ は V' の頂点で、G' 以外の枝が接続しているものである。

この方法は、頂点集合 $\{v_1,\ldots,v_r\}$ の要素数が大きいときには効果的ではないが、その数が高々定数個の場合には効果的である。具体的には、M に含まれる枝が接続していない次数2の頂点 v が存在するとき、その頂点に隣接する枝を1つの疑似枝に縮約することが出来る。次数1の頂点に付いても、接続する枝が M に含まれていないときに縮約することができる。この操作により各反復で G の次数1,2の頂点の数を $|M|$ の定数倍以下にでき、その結果各反復で G の大きさを $|M|$ の定数倍以下にすることが出来る。その結果、アルゴリズムの計算時間は $O(|V|)$ となる。

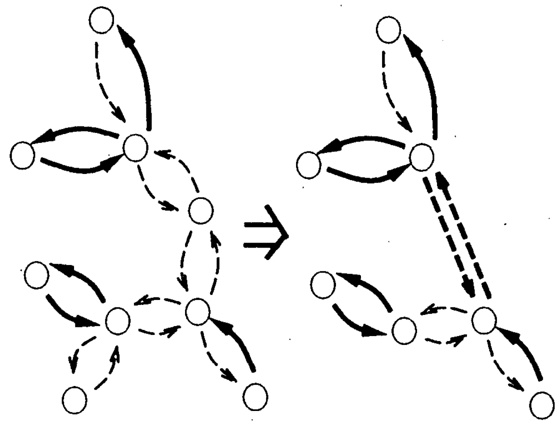

図2：縮約の様子。
太線が M の枝で太点線が疑似枝

4．おわりに

このアルゴリズムは少しの変更で目的関数の最大化をすることもでき、また枝の1部分だけを施設に含むことを許した問題に対しても $O(|V|)$ のオーダーを変化させることなく適用することが出来る。

参考文献

[1] S. L. Hakimi, E. F. Schmeichel, and M. Labbé. *On Locating Path- or Tree-Shaped Facilities on Networks. Networks* **23** (1993) 543–555.
[2] E. Minieka. *The Optimal Location of a Path or Tree in a Tree Network. Networks* **15** (1985) 309–321.