图论及其应用

北京邮电大学理学院 寇彩霞

北京邮電大學
Beijing University of Posts and Telecommunications

Ch3 树与最优树

Ch3 主要内容

- 树的概念
- 生成树、余树和健
- 生成树的计数及Caley公式
- 树的应用

3.4 树的应用——最优生成树问题（Optimal spanning tree）

- 赋权无向连通图G的所有生成树中权重之和最小的一个。
- 称为最优树（optimal tree）或
  最小树（minimum tree）
- 即：求图G的一棵生成树T，使得

$$w(T) = \min_{T} \sum_{e \in E(T)} w(e)$$

- 问题 设城市v_i与v_j间建立直接通信线路的费用为c_{ij}（≥ 0）。

则要建设连接v个城市的通讯网使造价最省

⇔在赋权图G中求一最小权连通生成子图；

⇔在赋权图G中求一最小权生成树(最优树)。

- 下面的**Kruskal**算法是在非赋权图中求生成树的“极大无圈子图”算法的改进，它是一种*贪心算法*（greedy algorithm）：

Kruskal's algorithm

(1) 选棱（**link**）e_1使$w(e_1)$最小；

(2) 若已选定 $e_1, e_2, \ldots, e_i$，则从 $E \setminus \{e_1, e_2, \ldots, e_i\}$中选取 e_{i+1} 使

(i)$G[\{e_1, e_2, \ldots, e_i\} \cup \{e_{i+1}\}]$无圈；

(ii)$w(e_{i+1})$是满足(i)之最小者。

(3) 若(2)不能再进行下去时，停止。否则，回到 **(2)**。

定理2.10 Kruskal算法求出的生成树 $T^* = G[\{e_1, e_2, \ldots, e_{\nu-1}\}]$ 是最优树。

证明：反证，假设T* 不是G的最优树。取G的一最优树T。令e_k为$\{e_1, e_2, \ldots, e_{\nu-1}\}$中（按顺序）第一个不属于T的边，且令T为最优树中使k为最大者。则$T+e_k$中唯一的圈C包含e_k，且C中必含一条边$e'_k \notin T^*$ (不然，$C \subseteq T^*$，矛盾。）但

$$T' = T + e_k - e'_k$$

也是G的生成树（？：因e'_k不是$T + e_k$的割边（定理2.3），从而T' 连通，且其边数=ν -1）。又，由于T的子图

$G[\{e_1, e_2, \ldots, e_{k-1}\} \cup \{e'_k\}]$ 也不含圈，由Kruskal算法知

$$w(e_k) \le w(e'_k)$$

$$\therefore \quad w(T') \le w(T),$$

即T'也是G的最优树，且$\{e_1, e_2, \ldots, e_{\nu-1}\}$中第一个不属于T'的边的下标 > k。这与k的取法相矛盾。

实现

先按权的不减顺序将边集重排成$a_1, a_2, \ldots, a_\varepsilon$。

关于算法中无圈性的判定，我们有一简单的办法：当$S=\{e_1, e_2, \ldots, e_i\}$已取定时，对候选边$a_j$有

$G[S \cup \{a_j\}]$ 无圈 $\Leftrightarrow$ a_j的两端点在林$G[S]$（此处当作生成子图）的不同分支中。

从而我们有求最优树的标记法：

开始：取a_1为候选边；并将v_k 标以 k，$k = 1,2,\ldots,\nu$ 。

若$S=\{e_1, e_2, \ldots, e_i\}$已取定，当候选边$a_j$ 的两端点有相同标号时，丢掉a_j永不再考虑，并改取a_{j+1}为 新候选边；否则选定$e_{i+1}=a_j$，并将$G[S]$中a_j两端点所在的二分支的顶点重新标号，标以两者中的最小者。

算法复杂度

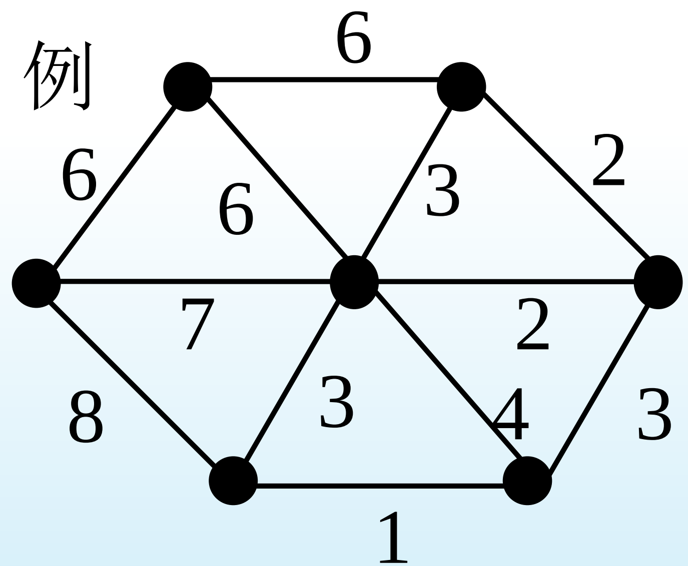
- 边排序　　　　　　　　$O(\varepsilon \cdot \log_2 \varepsilon)$

  比较边两端的标号　ε

  重新标号　　　　　　$O((\nu-1)\nu)$

- 故为好算法（$\le O(\nu^3)$）。

例



Prim算法

- 不断扩展一棵子树T=(S,E’)，直到S中包含原图的全部顶点，得到最优树T
- 每一次增加的一条边，使得这条边是S和S补形成的边割中权重最小的边。

Prim's algorithm （也是一种用贪心算法求最优树的一个著名算法）

Prim 算法：（令 S 为当前子树的顶点集合， E' 为当前子树的边的集合， $w(uv)$ 为边 uv 的权重， $d(v)$ 为点 v 到 S 的最小距离。）

0. $k=0$，任取 $v\in V(G)$，令 $v_0=v$，$S=\{v_0\}, E'=\varnothing$，$d(v_0)=0$，$d(v)=\infty\ (v\neq v_0)$，
1. 若 $S=V(G)$，结束（S 与 E' 分别为最优树的顶点集与边集）；否则，转 2。
2. 对任意的 $v\in V(G)\setminus S$，计算 $d(v)=min\{\,w(v_k v), d(v)\}$；对所有的 $v\in V(G)\setminus S$，计算 $\min\{d(v)\mid v\in V(G)\setminus S\}$，令 v_{k+1} 为最小值点，若 $d(v_{k+1})=\infty$，则图 G 不连通，结束（G 没有生成树）；否则，选取边 vv_{k+1}（其中 $v\in S$，$w(vv_{k+1})=d(v_{k+1})$），令 $S=S\cup\{v_{k+1}\}, E'=E'\cup\{vv_{k+1}\}$，令 $k:=k+1$，转 1。

Prim's algorithm

Prim 算法中的标号 $d(v)$ 可以用优先队列实现，算法复杂度为 $O(\upsilon\log_2\upsilon+\varepsilon)$，比 Kruskal 算法要快一些。

自学

- 树的应用——斯坦纳树问题（**Steiner tree problem**）
- 树的应用——最小树形图

3.4习题

- 2.4.1 用Kruskal算法解带约束的连线问题：用最小费用建立一连接若干城市的网络。

  但某些特定的 城市对 间要求有直通线路相连。

- 2.4.2 连通图 G 的树图是这样的一个图：其顶点集是G的全体生成树$\{T_1, T_2, \ldots, T_\tau\}$，

  且T_i和T_j相连 $\Leftrightarrow$ 它们恰有 v-2条公共边。

  证明：任何连通图的树图是连通的。

- 2.4.3* 任二最优树中，具有相同权的边数相等。
- 2.4.4 若赋权图中各边权互不相等，则其最优树是唯一的。