图与网络分析

第六章

树与最小生成树

第2节

树

- 定义
  - 无圈的连通图称为树图，简称树(tree)，记作 $T=(V,E)$.
    - 树中次为1的点称为树叶（悬挂点），次大于1的点称为分支点。
- 应用
  - 树是图论中结构最简单但由十分重要的图，在自然科学和社会科学的许多领域都有广泛的应用。
  - 企业组织结构、一些不重要的通信网络等也可表示为树状结构。

树

- 定理1

  任何顶点数≥ 2的树中至少有两个悬挂点。

- **证明**:

  设图$G=(V,E)$是一个树， G的顶点数$p(G)\geq 2$，并令$P=(v_1,v_2,\cdots,v_k)$是G中最长的初等链。

  由于是连通的，且$p(G)\geq 2$，故链P中至少有一条边，从而v_1与v_k是不同的。下面证明v_1是悬挂点。

  用反证法。假设$d(v_1)\geq 2$, 则G中存在边$[v_1,v_m]$,使$m\neq 2$。

  若点v_m不在P上，则$(v_m,v_1,v_2,\cdots,v_k)$是G中的一条初等链，且其长度比P多1，这与P为G中最长的初等链矛盾。

  若点v_m在P上，则$(v_1,v_2,\cdots,v_m,v_1)$是G中的一个圈，这与树的定义矛盾。故v_1是悬挂点。

  同理可证v_k也是悬挂点，因而树G中至少有两个悬挂点。

树

下述性质中的每一个命题都可作为树的定义。

- 性质

设图$T=(V,E)$有$|V|=n, |E|=m$，则下列关于树的说法是等价的：

（1） T是一个树；

（2） T无圈，且$m=n-1$；

（3） T连通，且$m=n-1$；

（4） T无圈，但每增加一新边即得唯一一个圈；

（5） T连通，但任舍去一条边就不连通；

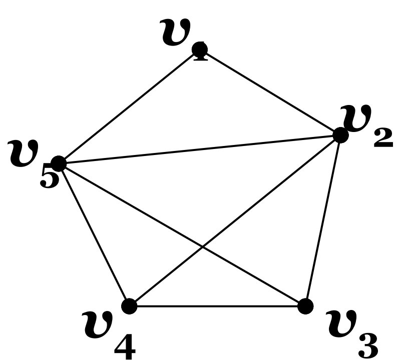
（6） T中任意两点，有唯一一条链相连。

树

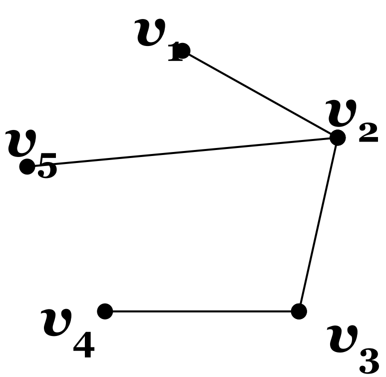
- 备注
  - 在点集相同的所有图中，树是含边数最少的连通图；在树中任意不相邻的两点间加一条边，则恰好得到一个圈，若在该圈上任意去掉一条边，则又可得到一个树。
  - 在树的任意两点之间有且仅有一条链。因此，树是最“脆弱”的连通图，从树图中任意去掉一条边，则剩下的图不再连通，故一些重要的网络不能设计成树的结构。

生成树

- 定义
  - 若图G的一个生成子图是一个树，则称该树为G的一个**生成树**(**支撑树**，spanning tree) 。
  - 设T为图G的一个生成树，图G中属于该生成树的边称为**树枝**，不在该生成树中的边称为**弦**。
- 示例



图G



图G的一个生成树

Haiyan Lu
Jiangnan University

生成树

定理2中充分性的证明给出了一种求连通图的生成树的方法——“破圈法”。
也可以用“避圈法”来求生成树。

- **定理2**

  图G有生成树的充分必要条件是图G是连通图。

- 证明：

  必要性显然成立。

  充分性。设图G是连通图。若G不含圈，则G本身是一个树，从而是其自身的一个生成树。

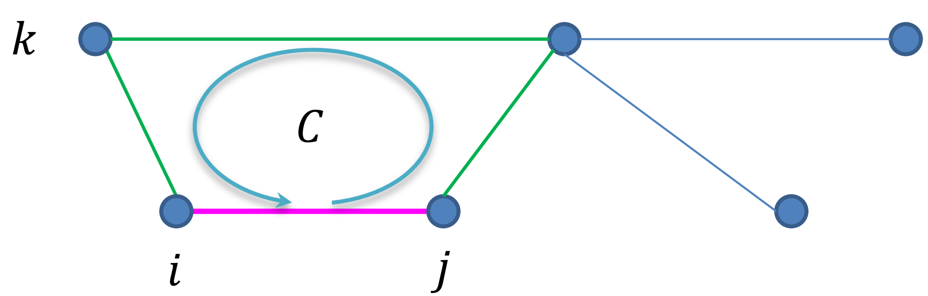
  若G含圈，任取一圈，从中任意去掉一条边，则可得到图G的一个生成子图G_1。若G_1不含圈，则G_1是G的一个生成树；若G_1仍含圈，则再从G_1中任取一个圈，从该圈中任意去掉一条边，可得到图G的一个生成子图G_2，如此重复，最终可得到图G的一个生成子图G_k，它不含圈，从而G_k是图G的一个生成子图。

最小生成树

- 定义
  - 给定图$G=(V,E)$，对G中的每一条边$[v_i,v_j]$赋予一个数w_{ij}（称为**权**），则称这样的图G为**赋权图(或网络)**，记作$G=(V,E,w)$. 根据实际问题需要，权可以是距离、时间、费用等。
  - 设$G=(V,E,w)$为一赋权连通图，每一边$[v_i,v_j]$的权$w_{ij}\geq 0$. 若$T=(V,E')$为G的一个生成树，称E'中所有边的权之和为**生成树T的权**，记作$w(T)$, 即
$$w(T)=\sum_{[v_i,v_j]\in E'} w_{ij}$$
  - 若生成树T^*是G的所有生成树中权中最小者，则称T^*是G的**最小生成树**(minimum spanning tree)，即
$$w(T^*)=\min_T w(T)$$

求最小生成树的算法

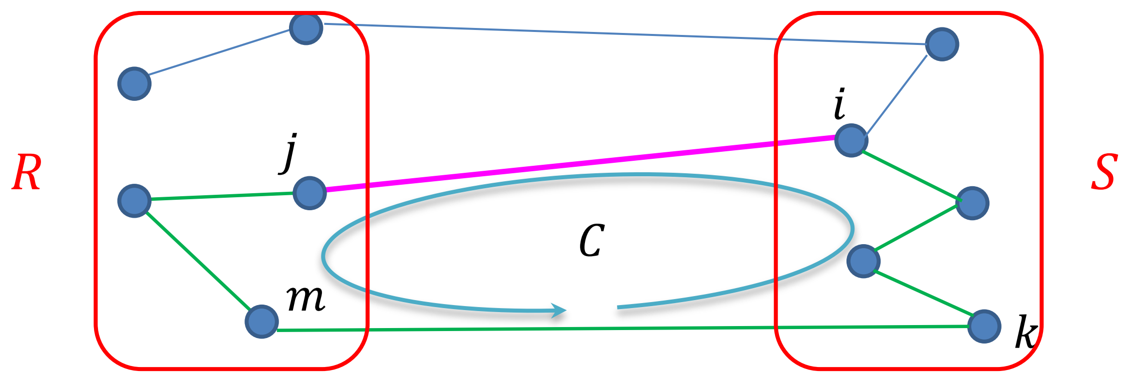
- 理论基础
  - **定理3** 设i是赋权图连通图$G=(V,E,w)$中任一顶点，若j是G中与i相邻的点中距离i最近的，即$w_{ij}=\min_{[i,l]\in E} w_{il}$，则边$[i,j]$一定含在图$G$的最小生成树$T$内。
  - 证明：**反证法**。
    假设边$[i,j]$不在G的最小生成树T中(**如下图所示**)，则将该边加到T中，必定出现一个圈C。设$[i,k]$为圈C上与顶点i关联的T中的边，由已知条件可知$w_{ik}>w_{ij}$。因此将$[i,k]$去掉可得到一个权更小的生成树T'，这与T为G的最小生成树矛盾，因此结论得证。



求最小生成树的算法

- 理论基础
  - **推论** 将连通图G的顶点集V分成两个不交的集合R和S，则两集合之间所有连线中的最短边一定含在G的最小生成树T内。
  - 证明：**反证法**。

    假设边$[i,j]$是R和S之间所有连线中的最短边，但不在G的最小生成树T中(如下图所示)，则将该边加到T中，必定出现一个圈C，且圈C中至少还有一条边$[m,k]$，其两个端点分别在R和S内，由已知条件可知$w_{mk} > w_{ij}$。因此将$[m,k]$去掉可得到一个权更小的生成树T'，这与T为G的最小生成树矛盾，因此结论得证。



求最小生成树的算法

Dijkstra 避圈法

- 避圈法
  - 设 $G=(V,E,w)$为赋权连通图,$V=\{v_1,\dots,v_n\}$,对于任意$[v_i,v_j]\in E, w_{ij}\geq 0$;
  - (1) 从G中任选一点, 不妨设为v_1, 令$T=\emptyset$, $R=\{v_1\}$, $S=\{v_2,\dots,v_n\}$;
  - (2) 从R与S的连线中找出最小边，不妨设为$[v_i,v_j]$，令$T=T\cup\{[v_i,v_j]\}$，$R=R\cup\{v_j\}$，$S=S\backslash\{v_j\}$;
  - (3) 重复(2)，直到$R=V$为止，此时得到的T即为所求。

求最小支撑树的算法

- 避圈法
  - **备注**
    - 上述避圈法是由**Dijkstra**于1959年提出的，因此被命名为**Dijkstra避圈法**，其求解效率比**Kruskal避圈法**(见《运筹学》第四版，清华大学出版社，302页)高。
      - **Kruskal避圈法**：每次从未选的边中选取边e，使它与已选边不构成圈，且e是未选边中权最小的，直到选够$n-1$条边为止。
    - **Dijkstra避圈法的实质**是在关于某支撑树T的$n-1$个基本割集中，取每个割集的最小边构成一个支撑树T，由前面的理论基础可知，该支撑树T为图G的最小支撑树。

求最小支撑树的算法

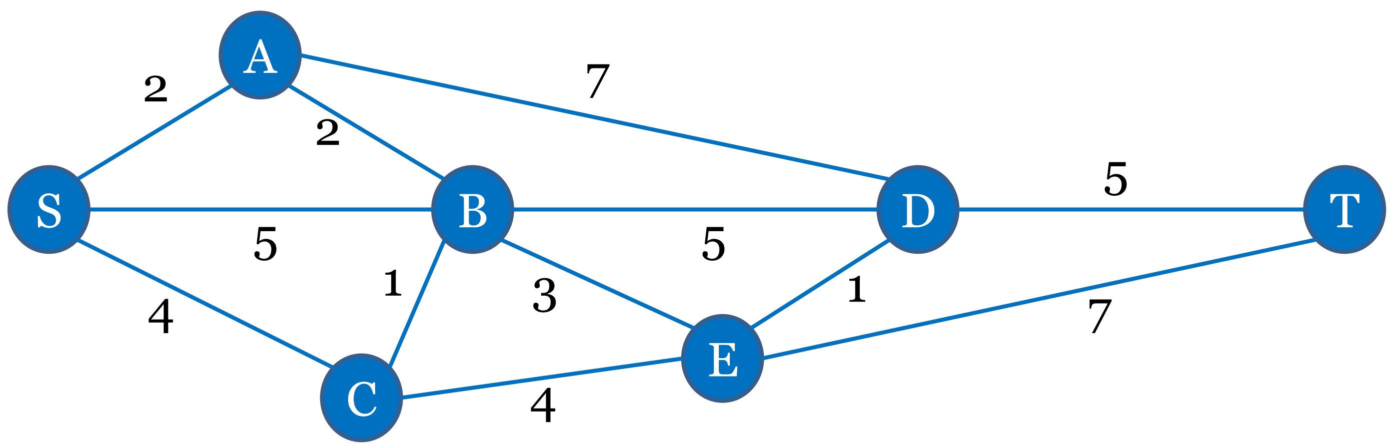
- 避圈法
  - 备注
    - **Fundamental Cuts(基本割集)**
      - Let T be a spanning tree of the graph G. The deletion of any tree arc of the spanning tree T produces a disconnected graph containing two subtrees T_1 and T_2 . Arcs whose endpoints belong to the different subtrees constitute a cut. We refer to any such cut as a fundamental cut of G with respect to the tree T.
      - Since a spanning tree contains $n-1$ arcs, the network has $n-1$ fundamental cuts with respect to any spanning tree.

求最小支撑树的算法

- 破圈法
  - 从赋权连通图G中任取一个圈（如果有），从该圈中去掉一条权最大的边（如果有多条权最大的边，则任选其中一条）。
  - 在余下的图中重复该步骤，直至得到一个不含圈的图为止，此时的图即为最小生成树。

求最小支撑树的算法

- 应用举例
  - **例2** 如下图所示，S、A、B、C、D、E、T代表村镇，它们间的连线表明了村镇间的现有道路情况，连线旁边的数字代表道路的长度。现要求沿图中道路架设电线，使上述村镇全部通上电，应如何架设使总的线路长度为最短？



求最小支撑树的算法

- 应用举例
  - **解**：

要使全部村镇通电，所架设线路各连接点之间须连通；又线路总长度须为最短，则该线路必不含圈。因此，所架设的线路为所给连通图的**最小生成树**。

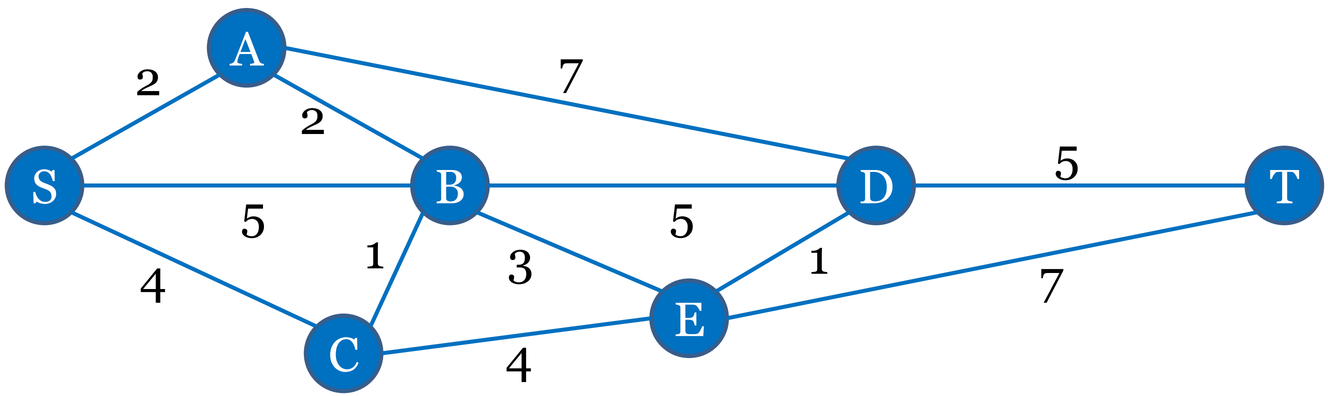
下面分别用两种方法给出相应的最小生成树：

- 方法一：避圈法(Dijkstra)
- 方法二：破圈法

求最小支撑树的算法

- 应用举例
  - 解:

方法一：避圈法(Dijkstra)

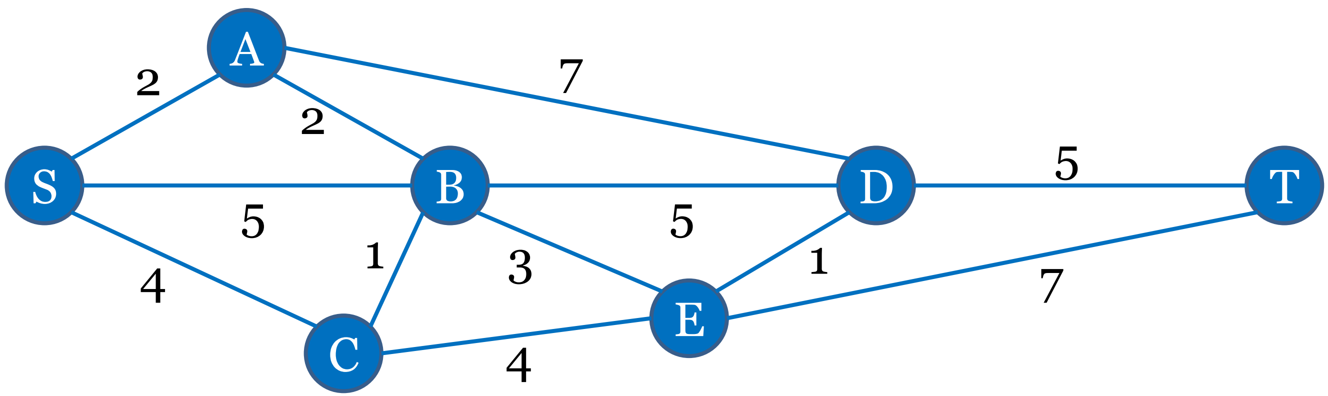


所得最小生成树T^*的权为$w(T^*) = 2 + 2 + 1 + 3 + 1 + 5 = 14$，即所求最优架设线路的总长度为14.

求最小支撑树的算法

- 应用举例
  - 解：

方法二：破圈法



所得最小生成树T^*的权为$w(T^*) = 2 + 2 + 1 + 3 + 1 + 5 = 14$，即所求最优架设线路的总长度为14.

Thank you!

谢谢！