《图与网络分析读书笔记》

赵宁双 PB19010444

March 23, 2021

Contents

1 图的基本概念 **1**

2 树 **6**

2.1 树及其性质 6

2.2 图的子支撑树 9

2.3 最小支撑树问题 10

1 图的基本概念

定义 1.1. 图：由一些点以及一些点之间的连线 *(不带箭头或者带箭头)* 所组成

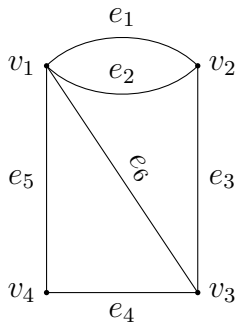
定义 1.2. 边：图中两点之间不带箭头的连线　　弧：图两点之间带箭头的连线

定义 1.3. 无向图 (图)：由点和边组成的图，记为 $G=(V,E)$，其中 V,E 分别是 G 的点集合和边集合。一条连接点 $v_i,v_j\in V$ 的边记为 $[v_i,v_j]$*(*或者 $[v_j,v_i]$*)*。

定义 1.4. 有向图：由点和弧构成的图，记为 $D=(V,A)$, 其中 V,A 分别是 G 的点集合和弧集合。一条由 v_i 指向 v_j 的弧记作 (v_i,v_j)。

图1.1是一个无向图,

$$V=\{v_1,v_2,v_3,v_4\},E=\{e_1,e_2,e_3,e_4,e_5,e_6\}$$



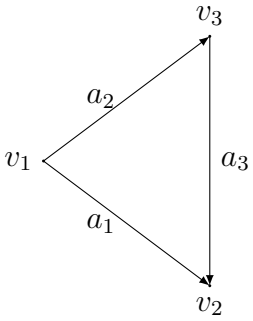


图 1.1 无向图 (G) 图 1.2 有向图 (D)

其中

$$e_1 = [v_1, v_2], e_2 = [v_1, v_2], e_3 = [v_2, v_3], e_4 = [v_3, v_4], e_5 = [v_1, v_4], e_6 = [v_1, v_3]$$

图1.2是一个有向图,

$$V = \{v_1, v_2, v_3\}, A = \{a_1, a_2, a_3\}$$

其中

$$a_1 = (v_1, v_2), a_2 = (v_1, v_3), a_3 = (v_3, v_2)$$

定义 1.5. 图 G 或者图 D 的点数记为 $\boldsymbol{p}$ **(**$\boldsymbol{G}$**)** 或者 $\boldsymbol{p}$ **(**$\boldsymbol{D}$**)**，边 (弧) 数记为 **q(G)(q(D))**。不会引起混淆的情况下记为 p,q。

例如图1.1的点数 q(G)=4; 图1.2的点数 q(D)=3.

接下来继续对图的一些概念做出一些定义，先考虑无向图 G=（V，E）

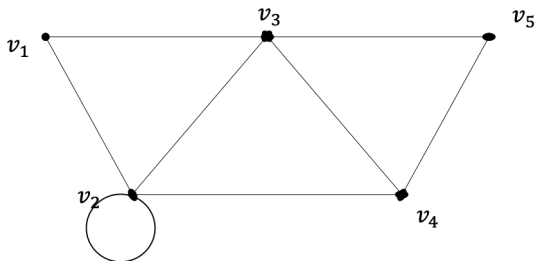


图 1.3 环的例子

定义 1.6. 若边 $e = [u, v] \in E$, 则称 u,v 是 e 的**端点**, 也称 u，v 是**相邻**的。称 e 是点 u (以及点 v) 的**关联边**。若图 G 中，某个边 e 的两个端点相同，则称 e 是**环** (如图1.6中的 $[v_2, v_2]$)，若两

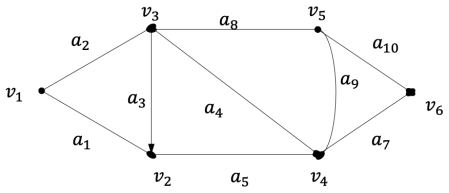


图 1.4 简单图

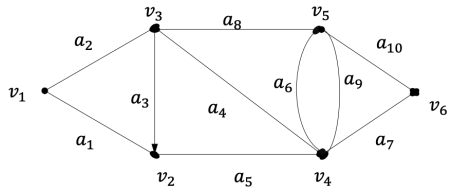


图 1.5 多重图

个点之间又多于一条的边，称这些边为**多重边** *(*如图*1.1*中的 e_1, e_2*)*。一个无环、无多重边的图称为**简单图**，但允许有多重边的图称为**多重图**（如图*1.5*)。

定义 1.7. 次：以点 v 为端点的边的个数，记为 $d_G(v)$ 或者 $d(v)$。如图*1.1*中，$d(v_1) = 4, d(v_2) = 3, d(v_3) = 3, d(v_4) = 2$。注意，环作为边在计算时算两遍。

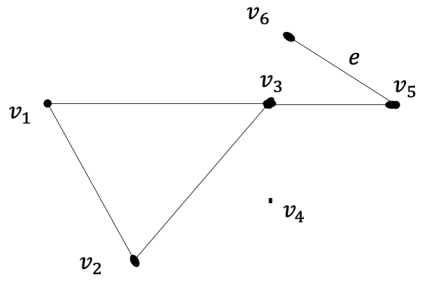


图 1.6 悬挂点、悬挂边与孤立点

定义 1.8. 悬挂点 次为 *1* 的点，如图 *1.6* 中的 v_6。

定义 1.9. 悬挂边 悬挂点的关联边，如图 *1.6* 中的 e。

定义 1.10. 孤立点 次为 *0* 的点，如图 *1.6* 中的 v_4。

定义 1.11. 次为奇数的点称作**奇点**，否则称作**偶点**。

定理 1.1. 图 *G=(V,E)* 中，所有点的次之和是边数的两倍，即

$$\sum_{v \in V} d(v) = 2q$$

证明. 在计算个点的次时，每条边被它的端点各用了一次。 □

定理 1.2. 任意一个图中，奇点的个数为偶数

证明. 设 V_1 和 V_2 分别是 G 中奇点和偶点的集合，由定理 1.1 有

$$\sum_{v\in V_1} d(v) + \sum_{v\in V_2} d(v) = \sum_{v\in V} d(v) = 2q$$

由于 $\sum_{v\in V_2} d(v)$ 是偶数，$\sum_{v\in V} d(v)$ 也是偶数，故 $\sum_{v\in V_1} d(v)$ 也必然是偶数，从而 V_1 的点数是偶数。 □

定义 1.12. 给定一个图 $G=(V, E)$，一个点、边的交错序列 $(v_{i_1}, e_{i_1}, v_{i_2}, e_{i_2}, \ldots, v_{i_{k-1}}, e_{i_{k-1}}, v_{i_k})$，如果满足 $e_{i_t} = [v_{i_t}, v_{i_{t+1}}](t = 1, 2, \ldots, k-1)$，则成为一条联结 v_{i_1} 和 $v_{i_{k-1}}$ 的**链**，记为 $(v_{i_1}, v_{i_2}, \ldots, v_{i_k})$，而 $v_{i_1}, v_{i_2}, \ldots, v_{i_k}$ 称为链的**中间点**。若 $v_{i_1} = v_{i_k}$，则称该链为一个**圈**。若 $v_{i_1}, v_{i_2}, \ldots, v_{i_k}$ 各不相同，则称之为**初等链**；若圈 $(v_{i_1}, v_{i_2}, \ldots, v_{i_k}, v_{i_1})$ 中 $v_{i_1}, v_{i_2}, \ldots, v_{i_k}$ 都是不同的，则称之为**初等圈**；若链（圈）中含有的边均不相同，则称之为**简单链（圈）**。

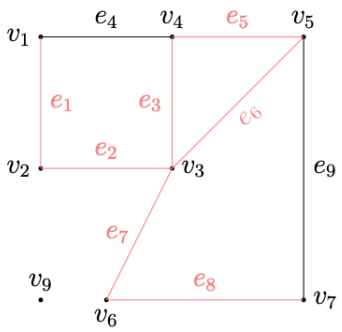


图 1.7 简单链

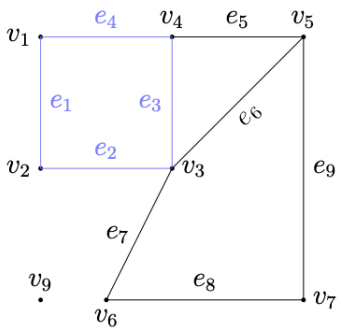


图 1.8 初等圈

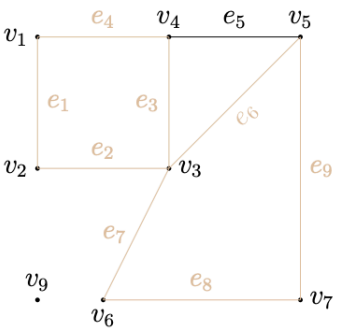


图 1.9 简单圈

例如上图中，$(v_1, v_2, v_3, v_4, v_5, v_3, v_6, v_7)$ 是以一条简单链，，但不是初等链。这个图中，不存在联结 v_1 和 v_9 的链。$(v_1, v_2, v_3, v_4, v_1)$ 是一个初等圈；$(v_4, v_1, v_2, v_3, v_5, v_7, v_6, v_3, v_4)$ 是一个简单圈但不是初等圈。

定义 1.13. 图 G 中，若任何两点之间，至少有一条链，则 G 是**连通图**，否则称为**不连通图**。若 G 是不连通图，他的每一个连通的部分称为**连通分图**（也称作**分图**）。如图 *1.7* 是一个不连通图。

定义 1.14. 给了一个图 $G = (V, E)$，如果图 $G' = (V', E')$，使得 $V = V'$ 即 $E' \subseteq E$，则称 G' 是 G 的一个**子支撑图**

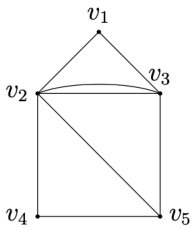


图 1.10 G

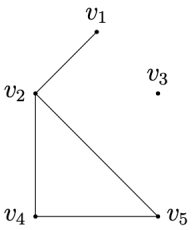


图 1.11 $G - v_3$

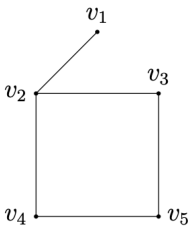


图 1.12 G 的一个支撑子图

定义 1.15. 设 $v \in V(G)$, 用$G - v$ 表示从图 G 中去掉点 v 以及 v 的关联边后得到一个图。

接下来讨论有向图的问题。设给了一个有向图 $D = (V, A)$, 从 D 中去掉所有弧上的箭头，就得到一个无向图，称之为 D 的**基础图**，记之为 G(D)。

给 D 中的一条弧 $a = (u, v)$, 称 u 为 a 的**始点**，v 为 a 的**终点**，称弧 a 是从 u 指向 v 的

设 $(v_{i_1}, a_{i_1}, v_{i_2}, a_{i_2}, \ldots, v_{i_{k-1}}, a_{i_{k-1}}, v_{i_k})$ 是 D 中的一个点弧交错序列，如果这个序列在基础图 G（D）中所对应的点边序列是一套链，则称这个点弧交错序列是 D 的一条链。类似，可以定义圈和初等链（圈）。

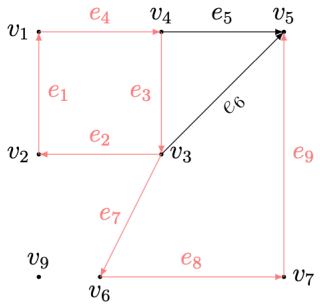


图 1.13 简单链

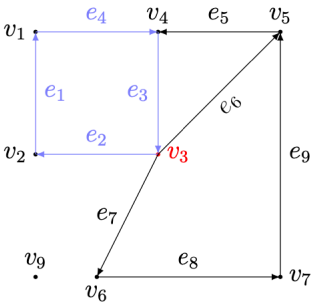


图 1.14 初等圈

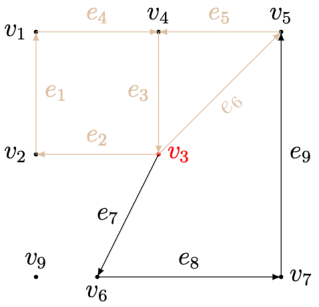


图 1.15 简单圈

如果 $(v_{i_1}, a_{i_1}, v_{i_2}, a_{i_2}, \ldots, v_{i_{k-1}}, a_{i_{k-1}}, v_{i_k})$ 是 D 中的一条链，并且对 $t = 1, 2, \ldots, k-1$ 均有 $a_{i_t} = (v_{i_t}, v_{i_{t+1}})$, 称之为**从** v_{i_1} **到** v_{i_k} **的一条路**。若路的第一个点和最后一个点相同，则称之为**回路**。类似可以定义**初等路（回路）**。

例如图 1.16 中，$(v_3, (v_3, v_2), v_2, (v_2, v_4), v_4, (v_4, v_5), v_5, (v_5, v_3))$ 是一个回路，$(v_1, (v_1, v_3), v_3, (v_3, v_4), v_4, (v_4, v_6), v_6)$ 是从 v_1 到 v_6 的路，$(v_1, (v_1, v_3), v_3, (v_5, v_3), v_5, (v_5, v_6), v_6)$ 是一条链，但不是路。

对于无向图，链和路的概念是一致的。

类似于无向图，可定义简单有向图、多重有向图，图 1.16 是一个简单有向图。以后除了特别交代意外，说到图，均指简单图。

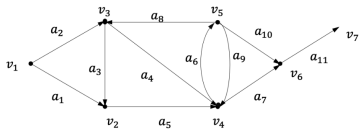


图 1.16

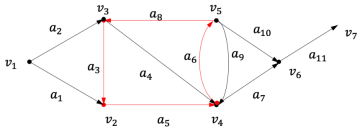


图 1.17 从 v_1 到 v_6 的路

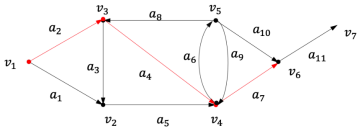


图 1.18 链

2 树

2.1 树及其性质

定义 2.1. 树一个无圈的连通图。

例如2.1是一个工厂的组织机构示意图。如果用图表示，该工厂的组织机构就是一个树。(如图2.2所示)

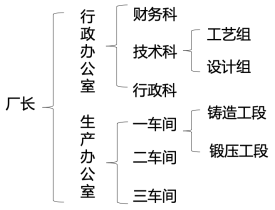


图 2.1

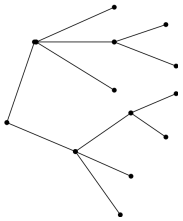


图 2.2

下面介绍一下树的基本性质

定理 2.1. 设图 $G=(V,E)$ 是一个树，$p(G) \geqslant 2$，则 G 中至少有两个悬挂点。

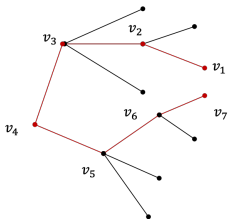


图 2.3 P 的寻找方法，k=7

证明. *step1.* 令 $P=(v_1,v_2,\ldots,v_k)$ 是 G 中**含边数最多的一条初等链**，因 $p(G)\geqslant 2$，并且 G 是连通的，故链 P 中至少有一条边，从而 v_1 与 v_k 是不同的。(如图2.1)

step2. 证明 v_1 是悬挂点，$d_(v_1)=1$。

反证法. 如果 $d(v_1)\geqslant 2$, 则存在边 $[v_1,v_m]$, 使得 $m\neq 2$。

①若 v_m 不在 P 上，那么 $(v_m,v_1,v_2,\ldots,v_k)$ 是 G 中的一条初等链，而它所含的边数比 P 多一条，这与 P 是含边数最多的初等链矛盾。

②若点 v_m 在 P 上，那么 $(v_1,v_2,\ldots,v_m,v_1)$ 是 G 中的一条圈，这与树的定义矛盾。

于是必有 $d_(v_1)=1$, 即 v_1 是悬挂点。同理可证 v_k 也是悬挂点，因而 G 至少有两个悬挂点。 □

定理 2.2. 图 $G=(V,E)$ 是一个树的充分必要条件是 G 不含圈，且恰有 p-1 条边。

证明. **必要性** 设 G 是一个树，根据定义，G 不含圈，故只要证明 G 恰有 p-1 条边。

对点数 p 进行数学归纳法。

①p=1,2 时，结论显然成立。

②假设对点数 $p\leqslant n$ 时，结论成立。

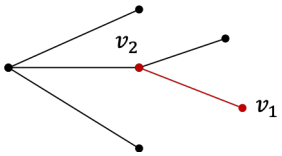


图 2.4 树 G

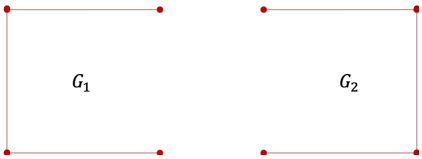


图 2.5 假设树 G 不连通

③设树 G 含 n+1 个点。由定理2.1，G 含悬挂点，设 v_1 是 G 的一个悬挂点，考虑图 $G-v_1$，易见 $p(G-v_1)=n, q(G-v_1)=q(G)-1$。因 $G-v_1$ 是 n 个点的树，由归纳假设得，$q(G-v_1)=n-1$，于是

$$q(G)=q(G-v_1)+1=(n-1)+1=n=p(G)-1$$

充分性 只要证明 G 是连通的。

反证法. 设 G 是不连通的 (如图2.1)，G 含 s 个连通分图 $G_1,G_2,\ldots,G_s(s\geqslant 2)$。因每个 $G_i(i=1,2,\ldots,s)$ 是连通的，并且不含圈，故每个 G_i 是树。设 G_i 有 p_i 个点，则由必要性，G_i 有 p_i-1 条边，于是

$$q(G)=\sum_{i=1}^{s}q(G_i)=\sum_{i=1}^{s}(p_i-1)=\sum_{i=1}^{s}p_i-s=p(G)-s\leqslant p(G)-2\neq p(G)-1$$

这与 $q(G)=p(G)-1$ 的假设矛盾。 □

定理 2.3. 图 $G=(V,E)$ 是一个树的充分必要条件是 G 是连通图，并且

$$q(G)=p(G)-1$$

证明. **必要性** 设 G 是树，根据定义，G 是连通图，由定理 2.2,$q(G)=p(G)-1$

充分性 只要证明 G 不含圈，对点数进行归纳。

①$p(G)=1,2$ 时，结论显然成立。

②设 $p(G)=n(n\geqslant 1)$ 时结论成立。

③现设 $p(G)=n+1$.

首先证明 G 必有悬挂点。若不然，因 G 是连通的，且 $p(G)\geqslant 2$, 故对每个点 v_i, 有 $d(v_i)\geqslant 2$。从而

$$q(G)=\frac{1}{2}\sum_{i=1}^{p(G)}d(v_i)\geqslant p(G)$$

这与 $q(G)=p(G)-1$ 矛盾，故 G 必有悬挂点。

设 v_1 是 G 的一个悬挂点，考虑 $G-v_1$，这个图仍然是连通的，$q(G-v_1)=q(G)-1=p(G)-2=p(G-v_1)-1$, 由归纳假设知 $G-v_1$ 不含圈，于是 G 也不含圈。 □

定理 2.4. 图 G 是树的充分必要条件是任意两个顶点之间恰有一条链。

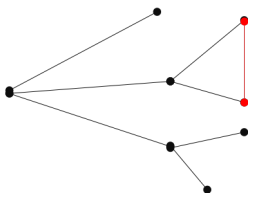


图 2.6 如果 G 有圈

证明. **必要性** 因 G 是连通的，故任两个点之间至少有一条链。但如果某两个点之间有两条链的话，那么图中含有圈，这与树的定义矛盾，从而任两点之间恰有一条链。

充分性 设图 G 中任两个点之间恰有一条链，那么易见 G 是连通的。如果 G 中含有圈，那么这个圈上的两个顶点之间有两条链，这与假设矛盾。故 G 不含圈，于是 G 是树。 □

由这个定理，很容易推出如下结论：

(1) 从一个树中去掉任意一边，则余下的图是不连通的。由此可知, 在点集合相同的所有图中，树是含边数最少的连通图。

(2) 在树中不相邻的两个点之间添上一条边，则恰好得到一个圈。如果再从这个圈上任意去掉一条边，则可以得到一个树。

2.2 图的子支撑树

定义 2.2. 设图 $T=(V,E')$ 是图 $G=(V,E)$ 的子支撑图，如果图 $T=(V,E')$ 是一个树，则称 T 是 G 的一个**支撑树**。

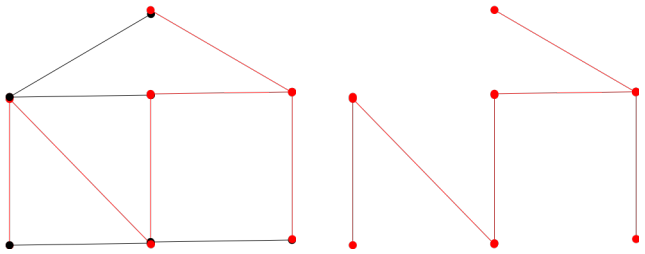


图 2.7 G

图 2.8 G 的支撑子树

若 $T=(V,E')$ 是 $G=(V,E)$ 的一个支撑树，则显然，树 T 中边的个数是 $p(G)-1$, G 中不属于树 T 的边数是 $q(G)-p(G)+1$

定理 2.5. 图 G 有支撑树的充分必要条件是图 G 是连通的。

证明. **必要性**是显然的。

充分性 设图 G 是连通图，如果 G 不含圈，那么 G 本身是一个树，从而 G 是他自身的一个支撑树。现设 G 含圈，任取一个圈，从圈中任意地去掉一条边，得到图 G 的一个支撑子图 G_1。如果 G_1 不含圈，那么 G_1 是 G 的一个支撑树（因为易见 G_1 是连通的）；如果 G_1 仍含圈，那么从 G_1 中任取一个圈，从圈中再任意去掉一条边，得到 G 的一个子支撑图 G_2，如此重复，最终可以得到 G 的一个子支撑图 G_k, 它不含圈，于是 G_k 是 G 的一个支撑树。 □

定理 2.5 的充分性证明，提供了一个寻求连通图的的支撑树的方法。这就是任取一个圈，从圈中去掉一边，对余下的图重复这个步骤，直到不含圈为止，即得到一个支撑树，这种方法称为” **破圈法**”

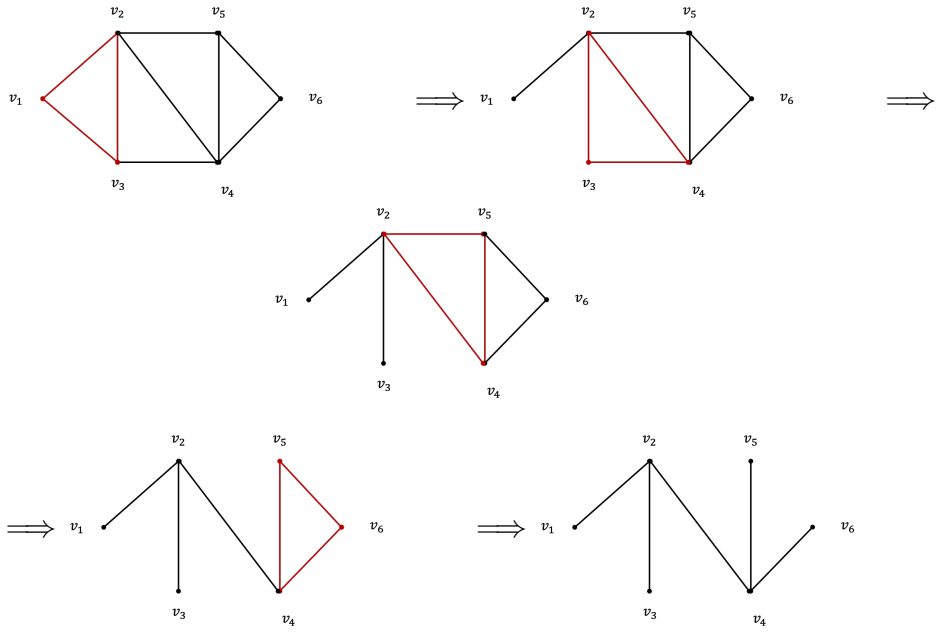


图 2.9 破圈法

2.3 最小支撑树问题

定义 2.3. 给图 $G=(V,E)$, 对 G 中的每一条边 $[v_i,v_j]$，相应地有一个数 ω_{ij}，则称这样的图 G 为**赋权图**，ω_{ij} 成为边 $[v_i,v_j]$ 上的**权**。

这里所说的权，是指与边有关的数量指标。根据实际问题需要，可以赋予它不同的含义，例如表示距离、时间、费用等。

设有一个连通图 $G=(V,E)$, 每一边 $e=[v_i,v_j]$, 有一个非负权

$$\omega(e)=\omega_{ij}(\omega_{ij}\geqslant 0)$$

定义 2.4. 如果 $T=(V,E')$ 是 G 的一个支撑树，称 E' 中所有边的权之和为支撑树 T 的权，记为 $\omega(T)$。即

$$\omega(T)=\sum_{[v_i,v_j]\in T}\omega_{ij}$$

如果支撑树 T^* 的权 $\omega(T^*)$ 是 G 的所有支撑树的权中最小者，则称 T^* 是 G 的**最小子支撑树**（简称**最小树**）。即

$$\omega(T^*)=min_T\omega(T)$$

式中对 G 的所有支撑树 T 取最小。

最小支撑问题就是要求给定连通赋权图 G 的最小子支撑树。

比如，给定了中科大的几个校区公交站点，已知每对站点之间的通行时间。不考虑路通不通的情况，要求建造一个连接各个站点的交通网，使总的建造费用最小，这个问题就是赋权图上的最小树问题。

求解最小树问题有两种方法。

1、避圈法（Kruskal）

首先选一条最小权的边，以后每一步中，总从已选边不构成圈的那些未选边中，选一条权最小的。（每一步中，如果有两条或者两图以上的边都是权最小的边，则从中任选一条。）

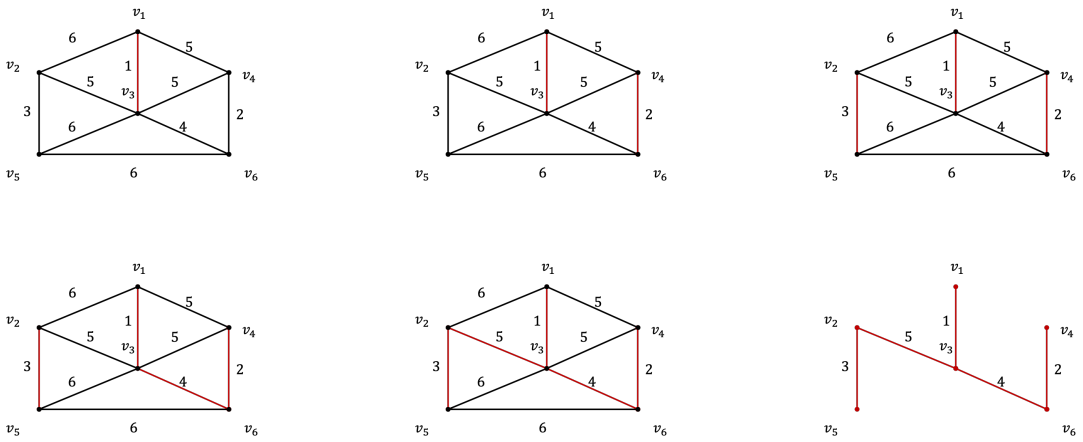


图 2.10 避圈法

算法具体步骤: 给定赋权图 $G=(V,E)$

(1) 令 $i=1, E_0=\emptyset, (\emptyset$ 表示空集)

(2)
- 如果 $i=p(G)$, 那么 $T=(V,E_{i-1})$ 是最小支撑树，算法终止
- 如果 $i<p(G)$, 选一条边 $e_i \in E \setminus E_{i-1}$, 使得 e_i 是使 $(V,E_{i-1}\cup\{e\})$ 不含圈的所有边 $e(e \in E \setminus E_{i-1})$ 中权最小的边。如果这样的边不存在，则说明图 G 不含支撑树，从而也就没有最小支撑树，算法终止。否则，令 $E_i=E_{i-1}\cup\{e_i\}$

(3) 把 i 换成 $i+1$, 转入第二步。

证明. 令 $G=(V,E)$ 是连通赋权图，我们只要证明 $T=(V,E_i)$ 是最小支撑树。

首先根据定理 2.2，可知 $T=(V,E_i)$ 是支撑树。为了方便起见，即

$$E(T)=\{e_1,e_2,\ldots,e_{p-1}\}$$

式中，$p = p(G)$。

下证 T 是最小支撑树。

反证法. 假设 T 不是最小支撑树，在 G 的支撑树中，令 H 是与 T 的公共边数最大的最小支撑树。因 T 与 H 不是同一个支撑树，故 T 中至少有一条边不在 H 中。令 $e_i(1 \leqslant i \leqslant p-1)$ 是第一个不属于 H 的边，把 e_i 放入 H 中，必得到一个且仅一个圈，记这个圈为 C。因为 T 是不含圈的，故 C 中必有一条边不属于 T，记这条边为 e。在 H 中去掉 e，增加 e_i，就得到 G 的另外一个支撑树 T_0, 可见

$$\omega(T_0) = \omega(H) + \omega(e_i) - \omega(e)$$

因为 $\omega(H) \leqslant \omega(T_0)$（因为 H 是最小支撑树），推出 $\omega(e) \leqslant \omega(e_i)$。但根据算法，$e_i$ 是使 $(V, \{e_1, e_2, \ldots, e_i\})$ 不含圈的权最小的边，而 $(V, \{e_1, e_2, \ldots, e_{i-1}, e\})$ 也是不含圈的，故必有 $\omega(e) = \omega(e_i)$, 从而 $\omega(T_0) = \omega(H)$。这就是说 T_0 也是 G 的一个最小支撑树，但是 T_0 与 T 的公共边数比 H 和 T 的公共边数多一条，这说明与 H 的选取矛盾。 □

2、破圈法

任取一个圈，从圈中去掉一条权最大的边（如果有两条或者两条以上的边都是权最大的边，则任意去掉其中一条）。在余下的图中，重复这个步骤，直至得到一个不含圈的图为止，这时候的树就是最小树。

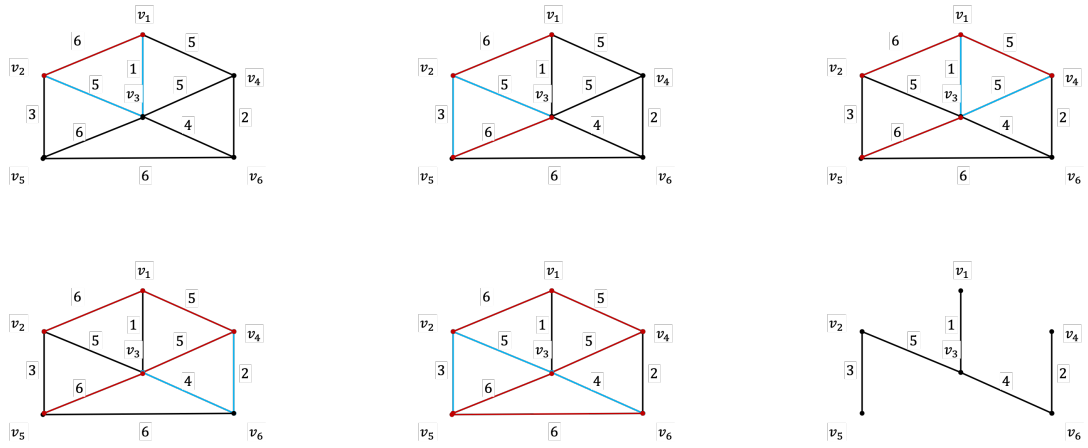


图 2.11 破圈法