

运筹学





2014/5/4

2

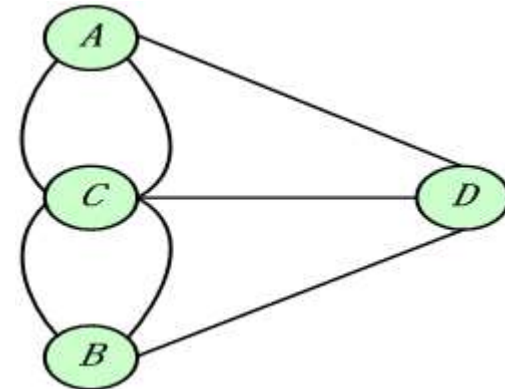
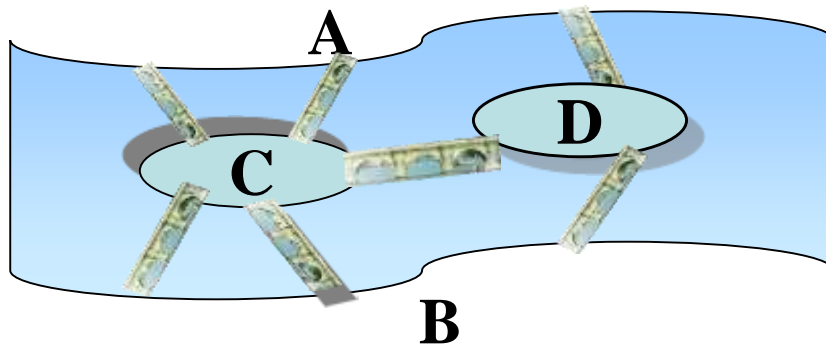
图与网络分析

Chp.10 Graph Theory and Network Analysis

- 图论是应用十分广泛的运筹学分支，它有**200**多年历史，大体可划分为三个阶段：
 - 第一阶段是从十八世纪中叶到十九世纪中叶，处于萌芽阶段，多数问题围绕游戏而产生，最有代表性的工作是所谓的**Euler**七桥问题，即一笔画问题。
 - 第二阶段是从十九世纪中叶到二十世纪中叶，这时，图论问题大量出现，如**Hamilton**问题，地图染色的四色问题以及可平面性问题等，这时，也出现用图解决实际问题，如**Cayley**把树应用于化学领域，**Kirchhoff**用树去研究电网络等。
 - 第三阶段是二十世纪中叶以后，由生产管理、军事、交通、运输、计算机网络等方面提出实际问题，以及大型计算机使大规模问题的求解成为可能，特别是以**Ford**和**Fulkerson**建立的网络流理论，与线性规划、动态规划等优化理论和方法相互渗透，促进了图论对实际问题的应用。

- 著名的**Euler**七桥问题

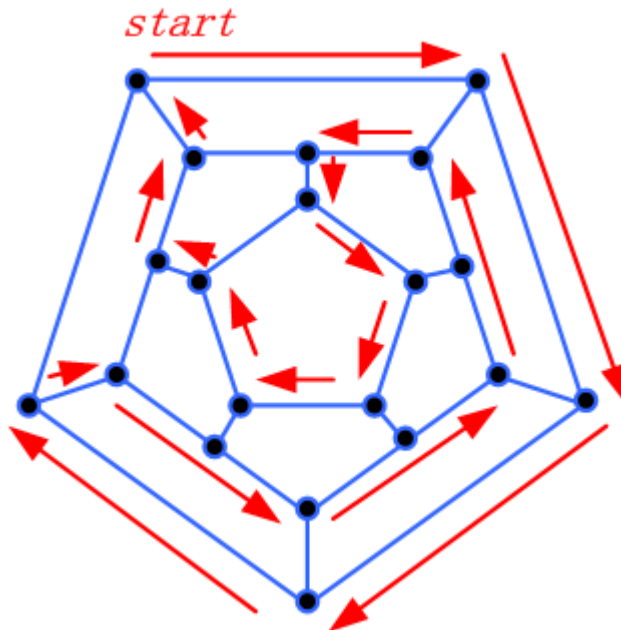
- 哥尼斯堡（现名加里宁格勒）是欧洲一个城市，**Pregei**河把该城分成两部分，河中有两个小岛，十八世纪时，河两边及小岛之间共有七座桥，当时人们提出这样的问题：有没有办法从某处（如**A**）出发，经过各桥一次且仅一次最后回到原地呢？



数学家**Euler**在1736年巧妙地给出了这个问题的答案，并因此奠定了图论的基础，**Euler**把**A**、**B**、**C**、**D**四块陆地分别收缩成四个顶点，把桥表示成连接对应顶点之间的边，问题转化为从任意一点出发，能不能经过各边一次且仅一次，最后返回该点。这就是著名的**Euler**问题。

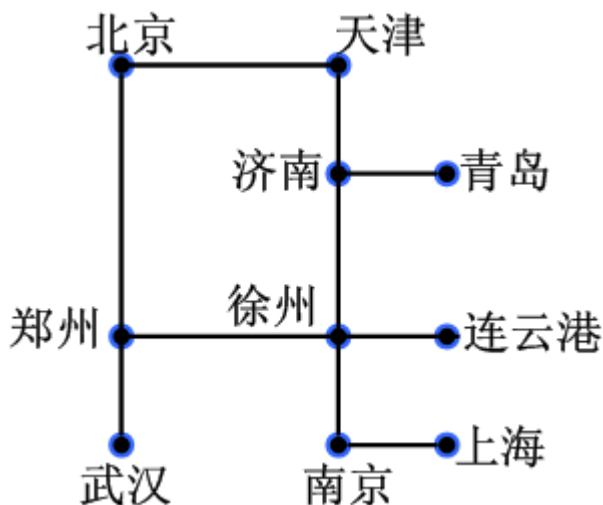
- **Hamilton问题**

- **Hamilton回路**是十九世纪英国数学家**Hamilton**提出，给出一个正**12**面体图形，共有**20**个顶点表示**20**个城市，要求从某个城市出发沿着棱线寻找一条经过每个城市一次而且仅一次，最后回到原处的周游世界线路（并不要求经过每条边）。



Hamilton问题的图解表示

- 图是反映对象之间关系的一种工具，图中的点表示了对象，点之间的连线表示了对象之间的对称的或者不对称的关系。
 - 例10.1 下图是北京、上海等十个城市间的铁路交通图，反映了它们之间的铁路分布情况。点代表城市，点和点之间的连线代表两个城市之间的铁路线：



类似的，还有电话线分布图、煤气管道图、航空线图等等。

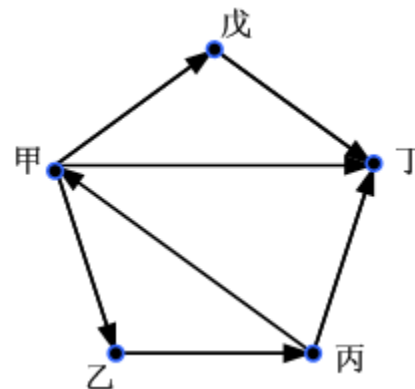
10.1 图的基本概念(cont.)

2014/5/4

7

- 例10.2 有甲、乙、丙、丁、戊五个球队，它们之间比赛情况以及胜负情况也可以用图表示出来。

甲胜了乙、丁和戊，输给了丙；
乙胜了丙，输给了甲；
丙胜了甲、丁，输给了乙；
丁输给了甲、丙和戊；
戊胜了丁，输给了甲。



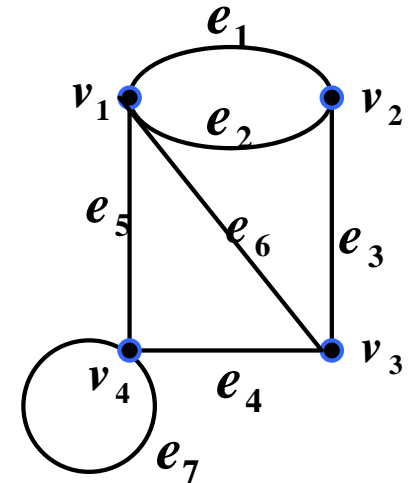
- 一个图是由一些点及一些点之间的连线（不带箭头或带箭头）所组成的。
 - 两点之间的不带箭头的连线称为边，带箭头的连线称为弧。
 - 如果一个图 G 是由点及边所构成的，则称之为无向图（也简称为图），记为 $G=(V,E)$ ， V 、 E 分别是 G 的点集合和边集合。一条联结点 v_i , $v_j \in V$ 的边记为 $[v_i, v_j]$ （或 $[v_j, v_i]$ ）。

10.1 图的基本概念(cont.)

2014/5/4

8

- 如果一个图 D 是由点及弧所构成的，则称之为有向图，记为 $D=(V,A)$ ， V 、 A 分别是 D 的点集合和弧集合。一条方向由 v_i 指向 v_j 的弧记为 (v_i, v_j) 。
 - 如例10.1中的图就是无向图，10.2中的图就是有向图。
- 图 G 或 D 中的点数记为 $p(G)$ 或 $p(D)$ ，边（弧）数记为 $q(G)$ 或 $q(D)$ 。
- 无向图
 - 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 （及点 v ）的关联边。若图 G 中，某个边的两个端点相同，则称 e 是环。若两个点之间有多于一条的边，称这些边为多重边。一个无环、无多重边的图称为简单图，一个无环但允许有多重边的图称为多重图。



- 以点 v 为端点的边的个数称为的次（度），记为 $d_G(v)$ 或 $d(v)$ 。
称次为1的点为悬挂点，它的关联边称为悬挂边；
称次为0的点为孤立点。
- 两个定理
 - 定理1: 图 $G=(V,E)$ 中，所有点的次之和是边数的两倍，即 $\sum_{v \in V} d(v) = 2q$
 - 定理2: 任一个图中，奇点的个数为偶数。（其中，奇点是次为奇数的点，否则称为偶点。）
- 给定一个图 $G=(V,E)$ ，一个点、边交错的序列 $(v_{i_1}, e_{i_1}, v_{i_2}, e_{i_2}, \dots, v_{i_{k-1}}, e_{i_{k-1}}, v_{i_k})$ ，如果满足 $e_{i_t} = [v_{i_t}, v_{i_{t+1}}] (t=1, 2, \dots, k-1)$ ，则称为一条联结 v_{i_1} 和 v_{i_k} 的链，记为 $(v_{i_1}, v_{i_2}, \dots, v_{i_k})$ ，有时，称点 $v_{i_2}, v_{i_3}, \dots, v_{i_{k-1}}$ 为链的中间点。链 $v_{i_1}, v_{i_2}, \dots, v_{i_k}$ 中，若 $v_{i_1} = v_{i_k}$ ，则称之为一个圈，记为 $(v_{i_1}, v_{i_2}, \dots, v_{i_{k-1}}, v_{i_1})$ 。若点 $v_{i_1}, v_{i_2}, \dots, v_{i_k}$ 都是不同的，则称之为初等链；若圈 $(v_{i_1}, v_{i_2}, \dots, v_{i_{k-1}}, v_{i_1})$ 中， $v_{i_1}, v_{i_2}, \dots, v_{i_{k-1}}$ 都是不同的，则称之为初等圈。若链（圈）中含的边均不相同，则称之为简单圈。

10.1 图的基本概念(cont.)

2014/5/4

10

- 图 G 中，若任何两个点之间至少有一条链，则称 G 是**连通图**，否则称为**不连通图**。若 G 是不连通图，它的每个连通的部分称为**连通分图**（也称**分图**）。
- 给定图 $G=(V,E)$ ，如果 $G'=(V',E')$ ，使 $V'=V$ 及 $E'\subseteq E$ ，则称 G' 是 G 的一个**支撑子图**。
- 设 $v\in V(G)$ ，用 $G-v$ 表示从图中去掉点 v 及其关联边后得到的一个图。

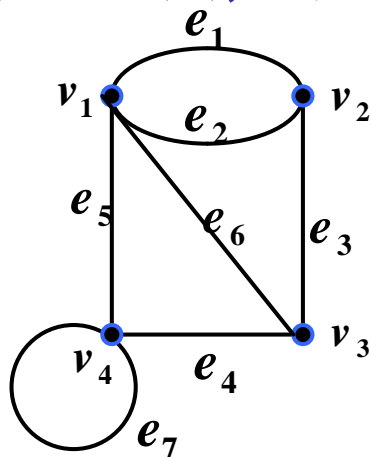
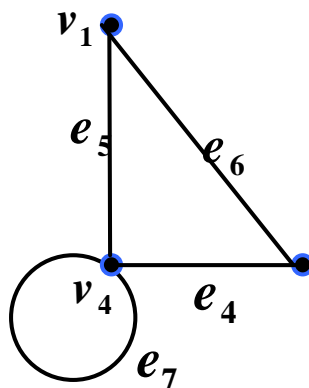


图 G



$G-v_2$

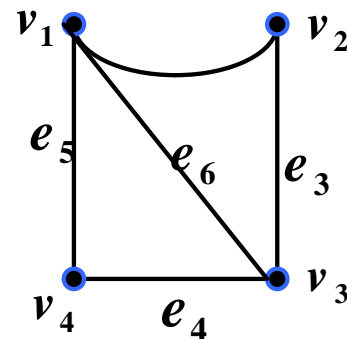
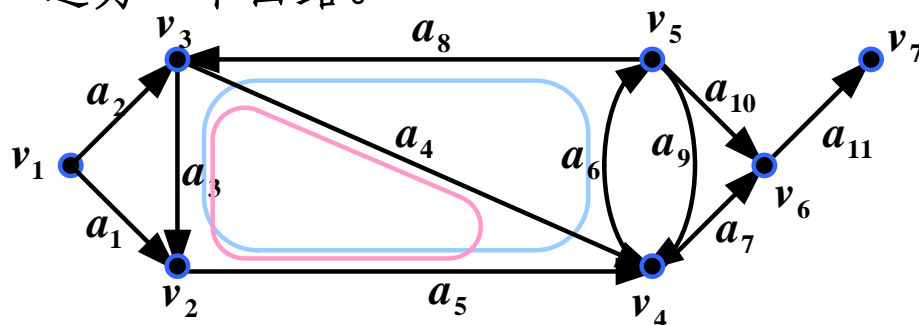


图 G 的一个支撑子图

• 有向图

- 给定有向图 $D=(V,A)$, 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**, 记为 $G(D)$ 。给定 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 $(v_{i_1}, a_{i_1}, v_{i_2}, a_{i_2}, \dots, v_{i_{k-1}}, a_{i_{k-1}}, v_{i_k})$ 是 D 中一个点、弧交错的序列, 如果这个序列在基础图 $G(D)$ 中所对应的点边序列是一条链, 则称这个点弧交错序列是 D 的一条链。如果 $(v_{i_1}, a_{i_1}, v_{i_2}, a_{i_2}, \dots, v_{i_{k-1}}, a_{i_{k-1}}, v_{i_k})$ 是 D 中的一条链, 并且满足 $a_{i_t} = [v_{i_t}, v_{i_{t+1}}] (t=1, 2, \dots, k-1)$, 称之为从 v_{i_1} 到 v_{i_k} 的一条路。若 $v_{i_1} = v_{i_k}$, 则称之为一个回路。



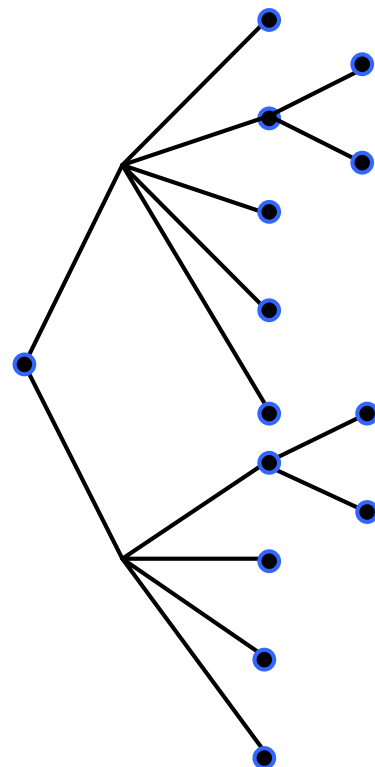
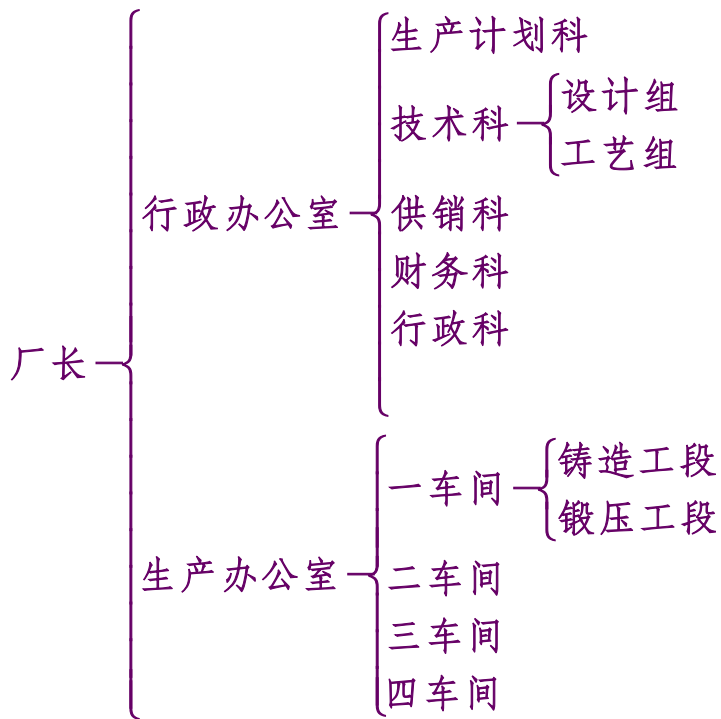
蓝色圈 $v_2 a_5 v_4 a_6 v_5 a_8 v_3 a_3 v_2$ 是一个回路;

红色圈 $v_2 a_5 v_4 a_4 v_3$ 不是回路, 是一个圈。

• 树及其性质

— 一个无圈的连通图称为**树**。如果一个无圈的图中每一个分支都是树，则称图为**森林**。

• 例10.3 某个工厂的组织机构如图。



如果用图表示，该工厂的组织机构就是一个树

– 定理1 设图 $G=(V,E)$ 是一个树, $p(G) \geq 2$,则 G 中至少有两个悬挂点。

• 证明:

令 $P=(v_1, v_2, \dots, v_k)$ 是 G 中含边数最多的一条初等链, 因 $p(G) \geq 2$, 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。

现在证明 v_1 是悬挂点: 用反证法, 如果 $d(v_1) \geq 2$, 则存在边 $[v_1, v_m]$, 使 $m \neq 2$ 。若点 v_m 不在 P 上, 那么 $(v_m, v_1, v_2, \dots, v_k)$ 是 G 中的一条初等链, 它的长度比 P 大, 与 P 是含边数最多的初等链矛盾。若点 v_m 在 P 上, 那么 $(v_1, v_2, \dots, v_m, v_1)$ 是 G 中的一个圈, 这与树的定义矛盾。于是必有 $d(v_1) = 1$, 即 v_1 是悬挂点。同理也可以证明 v_k 是悬挂点。因此 G 至少有两个悬挂点。

– 定理2 图 $G=(V,E)$ 是一个树的充要条件是 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。对点数 p 施行数学归纳法, $p=1, 2$ 时显然成立。假设对 $p \leq n$ 时结论成立, 设 G 含 $n+1$ 个点, 由定理, G 含悬挂点。设 v_1 是 G 的一个悬挂点。考虑 $G-v_1$, 易得 $p(G-v_1) = n$, $q(G-v_1) = q(G) - 1$ 。因 $G-v_1$ 是 n 个点的树, 由归纳假设 $q(G-v_1) = n-1$, 于是 $q(G) = n = p(G) - 1$ 。

充分性：

只要证明 G 是连通的即可。用反证法，设 G 是不连通的。含个连通分图 G_1, G_2, \dots, G_s 。因每个 $G_i (i=1, 2, \dots, s)$ 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

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

与假设矛盾，因此 G 是连通的。

— 定理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：根据定理易得。

充分性：只要证明 G 不含圈，对点数施行归纳。 $p(G)=1, 2$ 时显然成立。设 $p(G)=n (n \geq 1)$ 时结论成立，现设 $p(G)=n+1$ ，首先证明 G 有悬挂点。若 G 无悬挂点，则因为 G 连通，且 $p(G) \geq 2$ ，故对每个点 v_i ，有 $d(v_i) \geq 2$ ，从而

$$q(G) = \frac{1}{2} \sum_{i=1}^{p(G)} d(v_i) \geq p(G), \text{ 这与 } q(G) = p(G) - 1 \text{ 矛盾，故必有悬挂点。设 } v_1 \text{ 是}$$

G 的一个悬挂点，考虑 $G - v_1$ ，这个图仍是连通的， $q(G - v_1) = q(G) - 1 = p(G) - 2 = p(G - v_1) - 1$ ，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。

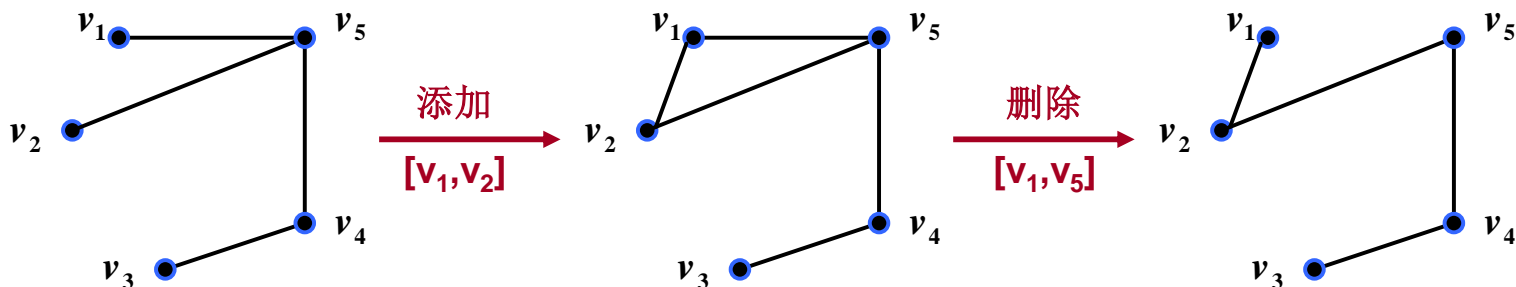
• 证明：

必要性——因 G 是连通的，故任意两点之间至少有一条链，但如果某两个点之间有两条链，则图 G 中含有圈，与树的定义矛盾；

充分性——设 G 中任意两点之间恰有一条链，那么易见 G 是连通的。如果 G 中含有圈，则这个圈上的两个顶点之间有两条链，与题设矛盾，因此 G 是无圈的，由树的定义可知 G 是树。

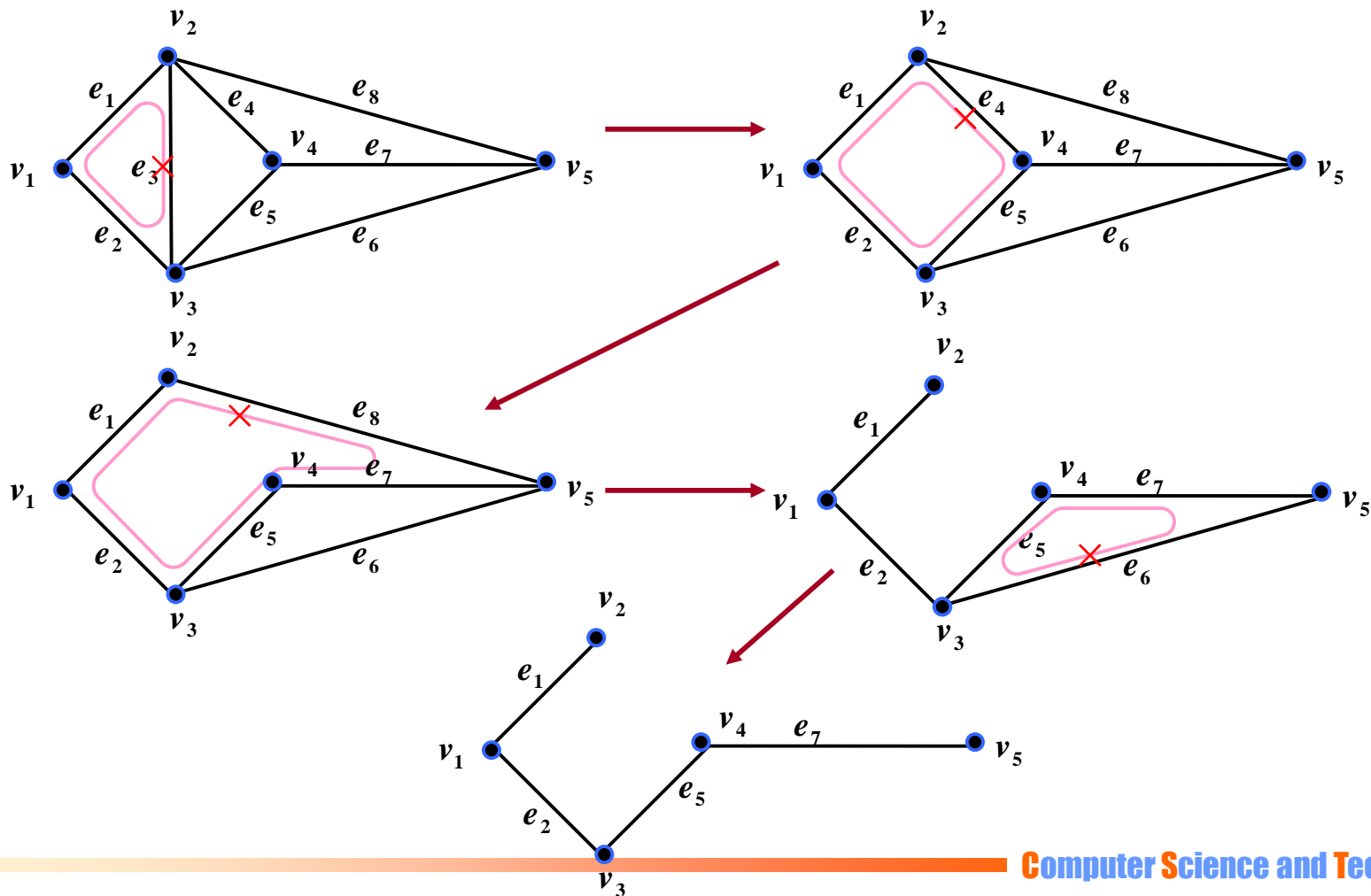
— 定理4的两个推论：

- 从一个树中去掉任意一条边，则余下的图是不连通的。
- 在树中不相邻的两个点间添上一条边，则恰好得到一个圈。进一步的，如果再从这个圈上任意去掉一条边，就得到一个树。

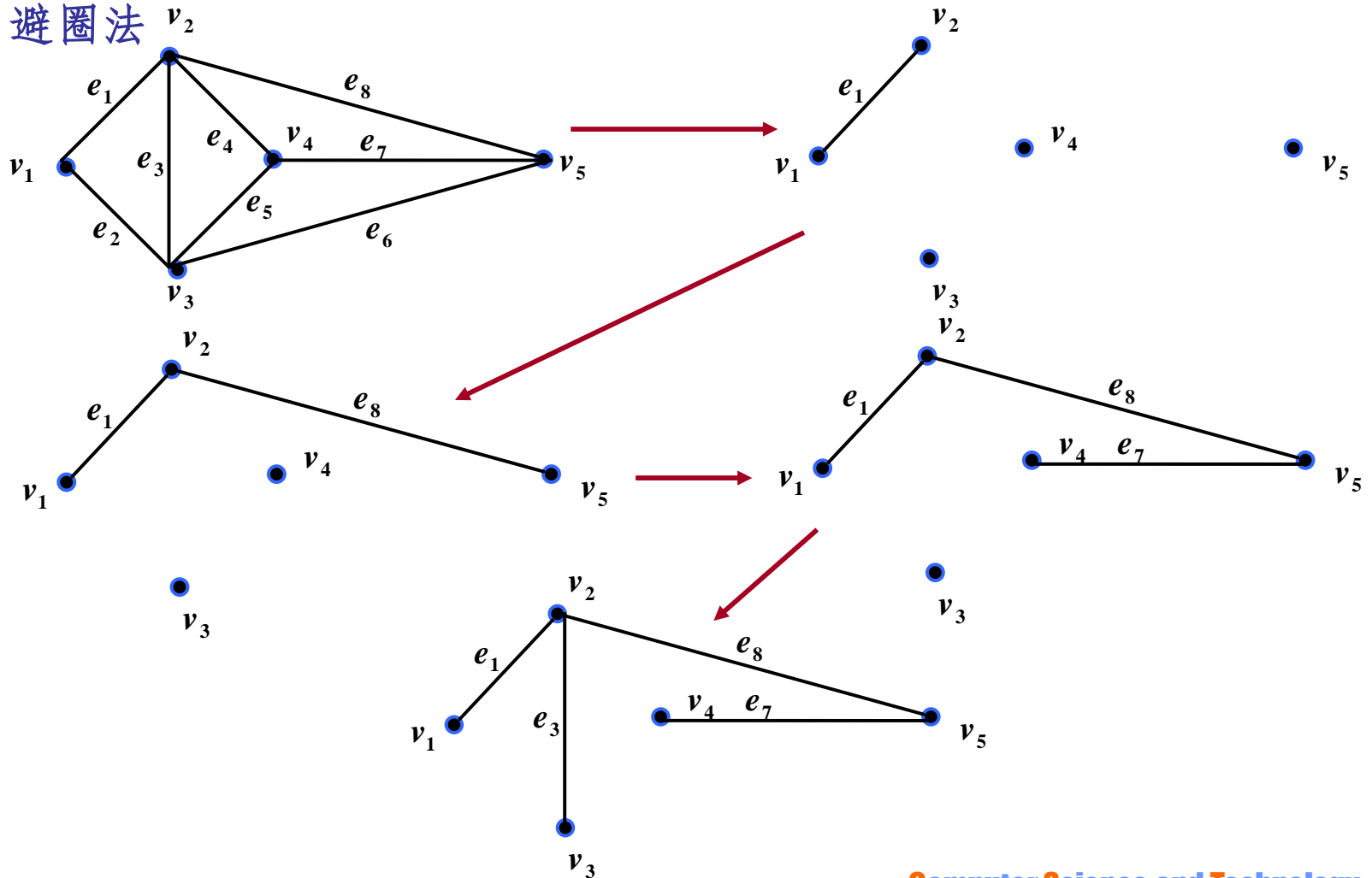


- 图的支撑树
 - 设图 $T = (V, E')$ 是图 $G = (V, E)$ 的支撑子图，如果 $T = (V, E')$ 是一个树，则称 T 是 G 的一个支撑树。
 - 若 $T = (V, E')$ 是图 $G = (V, E)$ 的支撑树，则树 T 中的边数为 $p(G)-1$ ， G 中不属于树 T 的边数是 $q(G) - p(G) + 1$ 。
 - 定理5 图 G 有支撑树的充分必要条件是图 G 是连通的。
 - 证明：必要性显然成立。充分性：设图 G 是连通图，则若 G 不含圈，则 G 本身就是它自身的支撑树。现设 G 含圈，任取一个圈，从圈中任意去掉一个边，得到图 G 的支撑子图 G_1 ；对 G_1 重复上面的过程，直至得到一个支撑子图 G_k 不含圈为止， G_k 就是 G 的支撑树。
 - 注意：上面的证明提供了一个寻求连通图的支撑树的方法，称为“破圈法”。
 - 也可以用另一种方法寻求连通图的支撑树，在图中任取一条边 e_1 ，找一条与它不够成圈的边 e_2 ，再找一条与 $\{e_1, e_2\}$ 不够成圈的边 e_3 ，重复寻找，直到不能进行为止。这时由取出的边构成了一个支撑树，称为“避圈法”。

— 破圈法



— 避圈法



- 最小支撑树问题

- 给定图 $G=(V,E)$ ，对 G 中的每一条边 $[v_i,v_j]$ ，相应的有一个数 w_{ij} ，则称这样的图 G 为**赋权图**，称 w_{ij} 为边 $[v_i,v_j]$ 上的**权**。

- 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 $w(T)$ ，即 $w(T) = \sum_{[v_i,v_j] \in T} w_{ij}$ 。如果支撑树 T^* 的权 $w(T^*)$ 是 G 的所有支撑树的

权中最小者，则称 T^* 是 G 的最小支撑树（简称最小树），即 $w(T^*) = \min_T w(T)$ 。

- **避圈法(Kruskal):**

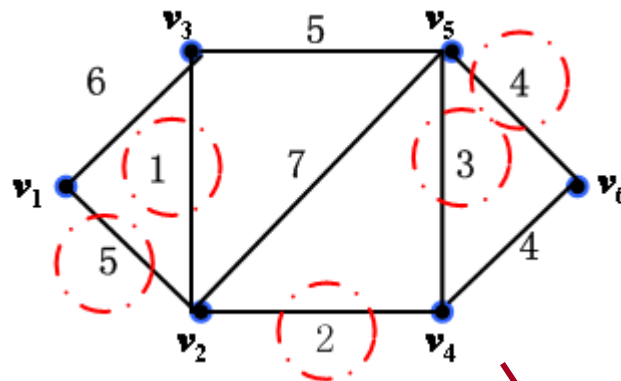
- 开始选一条最小权的边，以后每一步中，总从未被选取的边中选一条权最小的边，并使之与已选取的边不构成圈（若在一步中有两条或两条以上权值最小的边则任意选取一条）。

- **破圈法:**

- 任取一个圈，从圈中去掉权值最大的边（如果有两条或两条以上权值最大的边则任意去掉一条，在余下的图中重复这个步骤，直到得到不含圈的图为止。

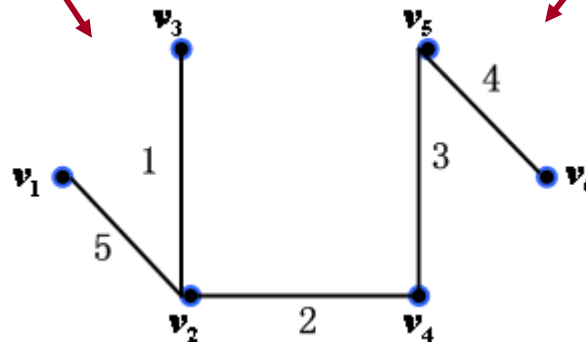
- 例10.4 某工厂内联结六个车间的道路网如图所示，已知每条道路的长，要求沿道路架设六个车间的电话线网，使电话线总长最小。
- 解：该问题转化为求图的最小支撑树问题。

解法1：避圈法

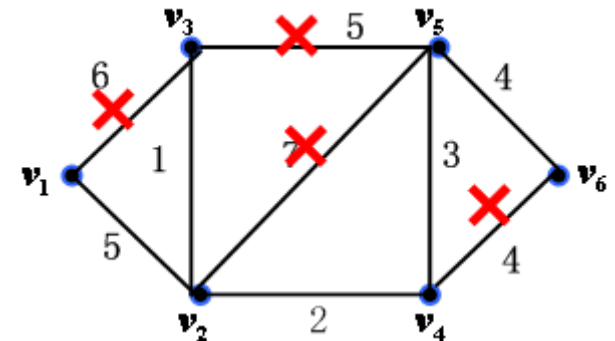


按规则依次选取

$[v_2, v_3]$, $[v_2, v_4]$,
 $[v_4, v_5]$, $[v_5, v_6]$,
 $[v_1, v_2]$



解法2：破圈法

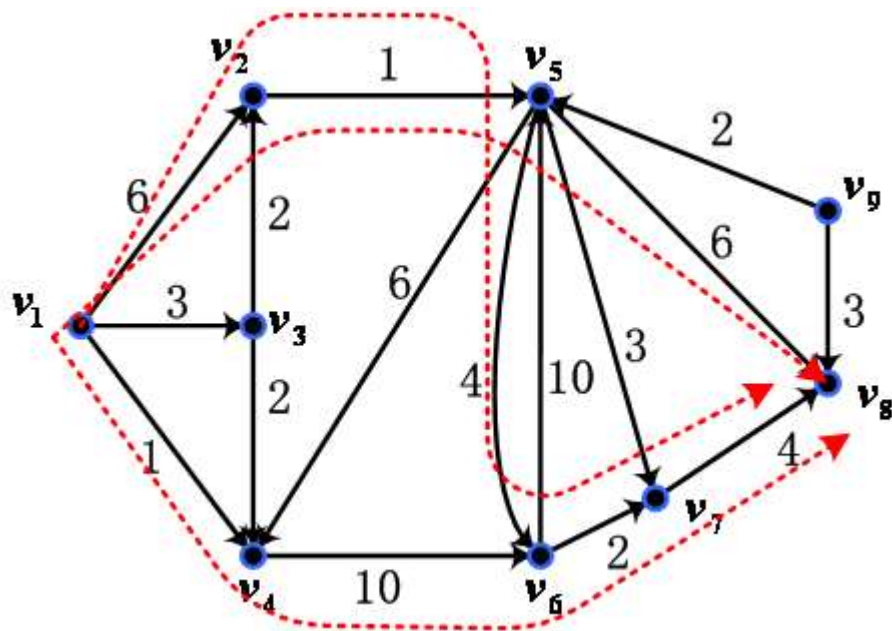


按规则依次删除

$[v_2, v_5]$, $[v_1, v_3]$,
 $[v_3, v_5]$, $[v_4, v_6]$

• 问题的提出

- 例10.5 已知如图所示的单行线交通网，每弧旁的数字表示通过这条单行线所需要的费用，现在要从 v_1 出发，通过这个交通网到 v_8 去，求使总费用最小的旅行路线。



可见从 v_1 到 v_8 的旅行路线是很多的，图中列出了三条路线（红色虚线）：

$v_1v_2v_5v_6v_7v_8$ 费用：17；

$v_1v_2v_5v_8$ 费用：13；

$v_1v_4v_6v_7v_8$ 费用：17。

还有很多其他的路线这里并未一一列出。

- 给定一个赋权有向图, 即给了一个有向图 $D = (V, A)$, 对每一个弧 $a = (v_i, v_j)$, 相应的有权 $w(a) = w_{ij}$, 又给定 D 中的两个顶点 v_s, v_t , 设 P 是 D 中从 v_s 到 v_t 的一条路, 定义路 P 的权是 P 中所有弧的权之和, 记为 $w(P)$ 。若有一条从 v_s 到 v_t 的路 P_0 , 使得 $w(P_0) = \min_P w(P)$, 则称 P_0 是从 v_s 到 v_t 的最短路。路 P_0 的权称为从 v_s 到 v_t 的距离, 记为 $d(v_s, v_t)$ 。显然, $d(v_s, v_t)$ 与 $d(v_t, v_s)$ 不一定相等。

- 最短路算法

- Dijkstra方法 (Dijkstra在1959年提出)

- 基本思想: 从 v_s 出发, 逐步的向外探寻最短路。执行过程中, 与每个点对应, 记录下一个数 (称为这个点的标号), 它或者表示从 v_s 到该点的最短路的权 (称为 **P** 标号), 或者是从 v_s 到该点的最短路的权的上界 (称为 **T** 标号), 方法的每一步是去修改 **T** 标号, 并且把某一个具 **T** 标号的点改变为具 **P** 标号的点, 从而使 D 中具 **P** 标号的顶点数多一个, 这样, 至多经过 $p-1$ 步, 就可以求出从 v_s 到各点的最短路。
 - Dijkstra方法是在 $w_{ij} \geq 0$ 的情况下公认最好的算法。

– Dijkstra方法具体步骤:

用 P, T 分别表示某个点的 P 标号、 T 标号。其中:

P : 表示从 v_s 到该点的最短路的权;

T : 表示从 v_s 到该点的最短路的权的上界;

S_i : 第 i 步时, 具有 P 标号点的集合。

为了在求出从 v_s 到各点的距离的同时也求出最短路径, 给每个点 v 以一个 λ 值。

算法终止时, 如果:

$\lambda(v) = m$: 表示在从 v_s 到 v 的最短路径上, v 的前一个点是 m ;

$\lambda(v) = M$: 表示图中不含从 v_s 到 v 的路径;

$\lambda(v) = 0$: 表示 $v = v_s$ 。

开始($i = 0$)令 $S_0 = \{v_s\}$, $P(v_s) = 0$, $\lambda(v_s) = 0$, 对每一个 $v \neq v_s$,

令 $T(v) = +\infty$, $\lambda(v) = M$, 令 $k = s$;

(1)如果 $S_i = V$, 算法终止, 这时, 对每个 $v \in S_i$, $d(v_s, v) = P(v)$; 否则转入(3)。

(2)考察每个使 $(v_k, v_j) \in A$ 且 $v_j \notin S_i$ 的点: 如果 $T(v_j) > P(v_k) + w_{kj}$, 则把 $T(v_j)$ 修改为 $P(v_k) + w_{kj}$, 把 $\lambda(v_j)$ 修改为 k ; 否则转入(3)。

(3)令 $T(v_{j_i}) = \min_{v_j \notin S_i} \{T(v_j)\}$: 如果 $T(v_{j_i}) < +\infty$, 则把 v_{j_i} 的 T 标号变为 P 标号,

$P(v_{j_i}) = T(v_{j_i})$, 令 $S_{i+1} = S_i \cup \{v_{j_i}\}$, $k = j_i$, 把 i 换成 $i + 1$, 转入(1); 否则终止,

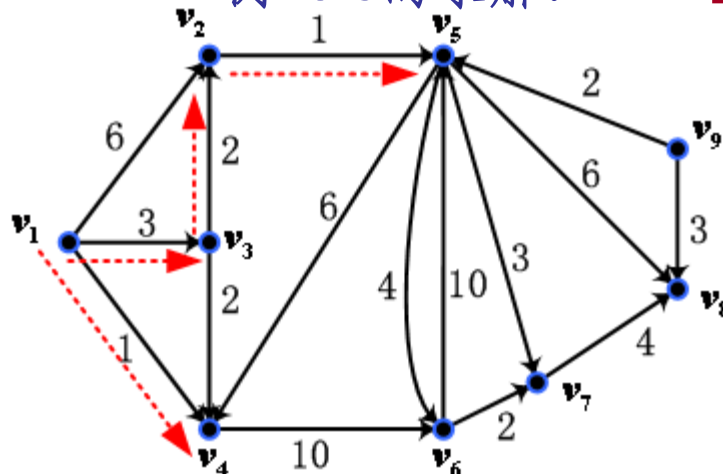
这时对每一个 $v \in S_i$, $d(v_s, v) = P(v)$, 而对于每一个 $v \notin S_i$, $d(v_s, v) = T(v)$ 。

10.3 最短路问题(cont.)

2014/5/4

25

— 例10.5的求解:



$i = 0$: $S_0 = \{v_1\}$, $P(v_1) = 0$, $\lambda(v_1) = 0$, 对每一个 $v_i \neq v_1$, 令 $T(v_i) = +\infty$, $\lambda(v_i) = M$, 令 $k = 1$;

$(v_1, v_2) \in A$ 且 $v_2 \notin S_0$, $T(v_2) > P(v_1) + w_{12}$, 把 $T(v_2)$ 修改为 $P(v_1) + w_{12} = 6$, 把 $\lambda(v_2)$ 修改为 1; 同理, 把 $T(v_3)$ 修改为 $P(v_1) + w_{13} = 3$, 把 $\lambda(v_3)$ 修改为 1; 把 $T(v_4)$ 修改为 $P(v_1) + w_{14} = 1$, 把 $\lambda(v_4)$ 修改为 1。

在所有的 T 标号中 $T(v_4)$ 最小, 于是令 $P(v_4) = 1$, $S_1 = \{v_1, v_4\}$, $k = 4$ 。

$i = 1$: $(v_4, v_6) \in A$ 且 $v_6 \notin S_1$, $T(v_6) > P(v_4) + w_{46}$, 把 $T(v_6)$ 修改为 $P(v_4) + w_{46} = 11$, 把 $\lambda(v_6)$ 修改为 4; 在所有的 T 标号中 $T(v_3)$ 最小, 于是令 $P(v_3) = 3$, $S_2 = \{v_1, v_3, v_4\}$, $k = 4$ 。

$i = 2$: $(v_3, v_2) \in A$ 且 $v_2 \notin S_2$, $T(v_2) > P(v_3) + w_{32}$, 把 $T(v_2)$ 修改为 $P(v_3) + w_{32} = 5$; 在所有的 T 标号中 $T(v_2) = 5$ 最小, 于是令 $P(v_2) = 5$, $S_3 = \{v_1, v_2, v_3, v_4\}$, $k = 2$ 。

$i = 3$: $(v_2, v_5) \in A$ 且 $v_5 \notin S_3$, $T(v_5) > P(v_2) + w_{25}$, 把 $T(v_5)$ 修改为 $P(v_2) + w_{25} = 6$; 在所有的 T 标号中 $T(v_5) = 6$ 最小, 于是令 $P(v_5) = 6$, $S_4 = \{v_1, v_2, v_3, v_4, v_5\}$, $k = 5$ 。

10.3 最短路问题(cont.)

2014/5/4

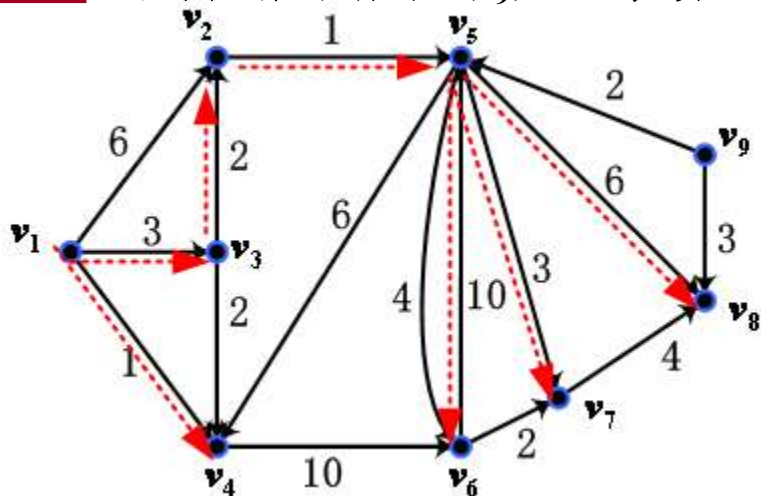
26

$i = 4$: $(v_5, v_6), (v_5, v_7), (v_5, v_8) \in A$ 且 $v_6, v_7, v_8 \notin S_4$, 把 $T(v_6)$ 、 $T(v_7)$ 、 $T(v_8)$ 修改为 10, 9, 12, 其中 $T(v_7) = 9$ 最小, 令 $P(v_7) = 9$, $S_5 = \{v_1, v_2, v_3, v_4, v_5, v_7\}$, $k = 7$ 。

$i = 5$: $(v_7, v_8) \in A$ 且 $v_8 \notin S_5$, $T(v_8) < P(v_7) + w_{78}$, 故 $T(v_8)$ 不变。在所有的 T 标号中 $T(v_6) = 10$ 最小, 令 $P(v_6) = 10$, $S_6 = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$, $k = 6$ 。

$i = 6$: 从 v_6 出发没有弧指向不属于 S_6 的点, 故直接查看在 S_6 中的 T 标号中 $T(v_8) = 12$ 最小, 令 $P(v_8) = 12$, $S_7 = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$, $k = 8$ 。

$i = 7$: 这时仅有的 T 标号 $T(v_9) = +\infty$, 算法终止。



可以得出：从 v_1 到 v_8 的最短路径为
 $v_1 \ v_3 \ v_2 \ v_5 \ v_8$ 长度 12

- 对于赋权无向图来说, 求从一个顶点到各个顶点的最短路也可以用 **Dijkstra** 算法来求解, 只需要将边看作是两条权值都等于该边权值的弧即可。

— 赋权图出现负权时的算法

设从任一点 v_i 到任一点 v_j 都有一条弧 (如果在 D 中, $(v_i, v_j) \notin A$, 则添加弧 (v_i, v_j) 令 $w_{ij} = +\infty$)。

从 v_s 到 v_j 的最短路总是从 v_s 出发, 沿着一条路到某点 v_i , 再沿 (v_i, v_j) 到 v_j 的 (这里 v_i 可以是 v_s 本身), 则从 v_s 到 v_i 的这条路必定是从 v_s 到 v_i 的最短路, 所以 $d(v_s, v_j)$ 必满足:

$$d(v_s, v_j) = \min_i \{d(v_s, v_i) + w_{ij}\}$$

为了求得这个方程的解 $d(v_s, v_1), d(v_s, v_2), \dots, d(v_s, v_p)$, 可用如下递推公式:

开始时, 令 $d^{(1)}(v_s, v_j) = w_{sj}, j = 1, 2, \dots, p$

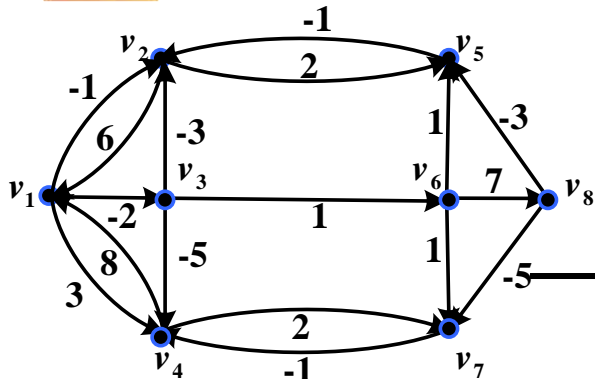
对 $t = 2, 3, \dots, d^{(t)}(v_s, v_j) = \min_i \{d^{(t-1)}(v_s, v_i) + w_{ij}\}, j = 1, 2, \dots, p$

若进行到第 k 步时, 对所有 $j = 1, 2, \dots, p$, 有 $d^{(k)}(v_s, v_j) = d^{(k-1)}(v_s, v_j)$,

则 $\{d^{(k)}(v_s, v_j)\}_{j=1,2,\dots,p}$ 就是 v_s 到各点的最短路。

10.3 最短路问题(cont.)

– 例10.6 求图中所示赋权有向图中从 v_1 到各点的最短路。



注意：表中没有填数字的部分代表 $+\infty$

	w_{ij}								$d^{(t)}(v_1, v_j)$			
	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	$t=1$	$t=2$	$t=3$	$t=4$
v_1	0	-1	-2	3					0	0	0	0
v_2	6	0			2				-1	-5	-5	-5
v_3		-3	0	-5		1			-2	-2	-2	-2
v_4				0			2		3	-7	-7	-7
v_5		-1			0					1	-3	-3
v_6					1	0	1	7		-1	-1	-1
v_7				-1			0			5	-5	-5
v_8					-3		-5	0			6	6

- 在求解过程中可以类例 *Dijkstra* 方法，给每一个点以值开始，

$$\lambda(v_s) = 0, \lambda(v_i) = s, i \neq s.$$

在迭代过程中，如果 $d^{(t)}(v_s, v_j) = \min_i \{d^{(t-1)}(v_s, v_i) + w_{ij}\} = d^{(t-1)}(v_s, v_{i_0}) + w_{i_0j}$,

这时把 $\lambda(v_i)$ 修改为 i_0 。迭代终止时，根据各点的 λ 值，可以得到从 v_s 到各点的最短路。

— 定义 设 **D** 是赋权有向图，**C** 是 **D** 中的一个回路，如果 **C** 的权 $w(\mathbf{C})$ 小于零则称 **C** 是 **D** 中的一个负回路。

— 推论：

- (1) 如果 **D** 是不含负回路的赋权有向图，那么从 v_s 到任一个点的最短路必可取为初等路，从而最多包括 $p-2$ 个中间点。
- (2) 上述递推公式中的 $d^{(t)}(v_s, v_j)$ 是在至多包含 $t-1$ 个中间点的限制条件下从 v_s 到 v_j 的最短路的权。

— 由 (1)(2) 可知：当 **D** 中不含负回路时，上述算法至多经过 $p-1$ 次迭代必定收敛，即对所有的 $j=1, 2, \dots, p$ ，均有 $d^{(k)}(v_s, v_j) = d^{(k-1)}(v_s, v_j)$ ，从而求出从 v_s 到各个顶点的最短路的权。

- 如果经过 $p-1$ 次迭代, 存在某个 j , 使得 $d^{(p)}(v_s, v_j) \neq d^{(p-1)}(v_s, v_j)$, 则说明 D 中含有负回路, 显然, 这时从 v_s 到 v_j 的路的权是没有下界的
 - 为了加快收敛速度, 可以利用如下的递推公式:

$$d^{(1)}(v_s, v_j) = w_{sj}$$

$$d^{(t)}(v_s, v_j) = \min \left\{ \min_{i < j} \{ d^{(t)}(v_s, v_i) + w_{ij} \}, \min_{i \geq j} \{ d^{(t-1)}(v_s, v_i) + w_{ij} \} \right\}$$

$$j = 1, 2, \dots, p, t = 2, 3, \dots$$

- J.Y.Yen提出了一个改进的递推算法:

$$d^{(1)}(v_s, v_j) = w_{sj}$$

$$\text{对 } t = 2, 4, 6, \dots, \text{按 } d^{(t)}(v_s, v_j) = \min \left\{ d^{(t-1)}(v_s, v_j), \min_{i < j} \{ d^{(t)}(v_s, v_i) + w_{ij} \} \right\}$$

$$\text{对他 } 3, 5, 7, \dots, \text{按 } d^{(t)}(v_s, v_j) = \min \left\{ d^{(t-1)}(v_s, v_j), \min_{i > j} \{ d^{(t)}(v_s, v_i) + w_{ij} \} \right\}$$

$$j = 1, 2, \dots, p, t = 2, 3, \dots$$

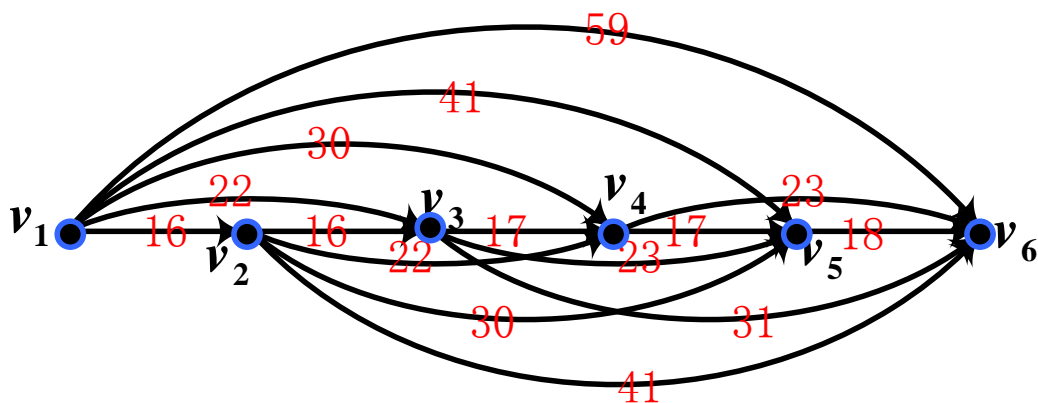
同样的, 当对所有的 $j = 1, 2, \dots, p$ 有 $d^{(k)}(v_s, v_j) = d^{(k-1)}(v_s, v_j)$ 时, 计算结束。

— 应用举例

- 例10.7 设备更新问题。某企业使用一台设备，在每年年初，需要决定购置新的还是使用旧的。购置新设备需要支付购置费用，使用旧的需要维修费用。如何制定一个几年之内的设备更新计划使得总的支付费用最少。以五年为例，已知该种设备在年初的价格以及使用不同时间（年）所需的维修费用如下表：

第1年	第2年	第3年	第4年	第5年	使用年数	0—1	1—2	2—3	3—4	4—5
11	11	12	12	13	维修费用	5	6	8	11	18

解：用点 v_i 代表第 i 年年初购进一台设备这种状态， v_6 代表第五年年底。从 v_i 到 $v_{i+1} \sim v_6$ 各画一条弧，弧 (v_i, v_j) 代表第 i 年年初购进的设备一直使用到第 j 年年初。

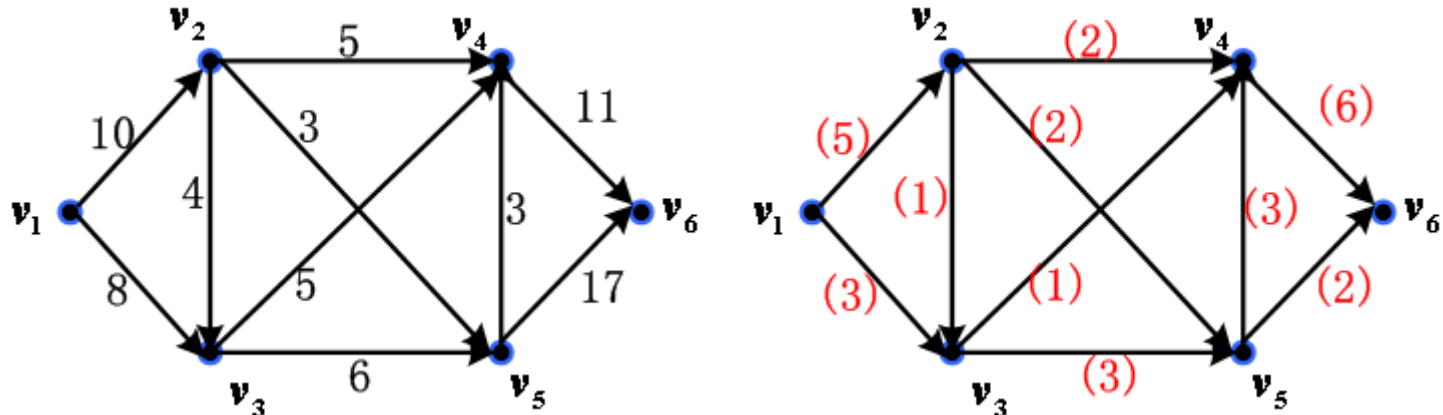


这样一来，制定一个最优的设备更新计划的问题就等价于寻求从 v_1 到 v_6 的最短路的问题。

按求解最短路的计算方法， $\{v_1, v_3, v_6\}$ 和 $\{v_1, v_4, v_6\}$ 均为最短路。即有两个最优方案：在第一年、第三年（或第四年）各置备一台新设备。

问题的提出:

- 例10.8 左图是联结某产品产地 v_1 和销地 v_6 的交通网，每一弧 (v_i, v_j) 代表从 v_i 到 v_j 的运输线，产品经这条弧由 v_i 输送到 v_j ，弧旁的数字表示这条运输线的最大通过能力。

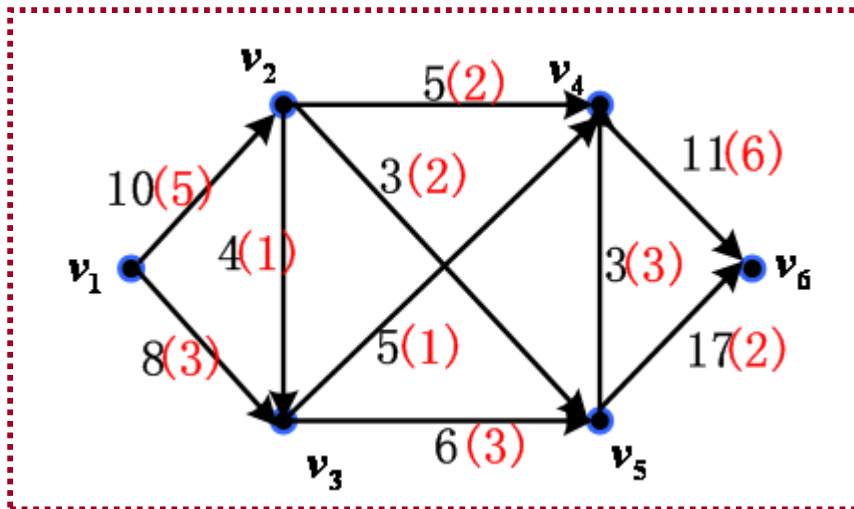


- 右图中给出了一个运输方案，红色的数字代表在这个方案中，每条运输线上的运输数量，该方案使8个单位的产品从 v_1 运送到 v_6 。
- 公路系统中的车辆流、控制系统中的信息流、供水系统中的水流、金融系统中的现金流都是类似的问题。

- 基本概念与基本定理

- 网络与流

- 定义1: 给定一个有向图 $D=(V,A)$, 在 V 中指定了一点, 称为发点, 记为 v_s , 和另一点, 称为收点, 记为 v_t , 其余的点叫中间点。对于每一个弧 $(v_i,v_j) \in A$, 对应有一个 $c(v_i,v_j) \geq 0$, 或简写为 c_{ij} , 称为弧的容量。通常把这样的 D 叫做一个网络, 记作 $D=(V,A,C)$ 。网络上的流, 是指定义在弧集合 A 上的一个函数 $f=\{f(v_i,v_j)\}$, 并称 $f(v_i,v_j)$ 为弧上的流量, 或简写为 f_{ij} 。



上例的图就是一个网络, 其中:

- ✓ v_1 是发点, v_6 是收点, 其他的点是中间点;
- ✓ 每条弧上的黑色数字为该弧的容量 c_{ij} ;
- ✓ 红色数字所示的运输方案为该网络上的一个流, 每个弧上的运输量就是该弧上的流量, 即 $f_{12}=5$, $f_{24}=2$, $f_{13}=3$, $f_{34}=1$ 等等。

- 可行流与最大流

- 运输网络的实际问题对流有两个要求：一是每个弧上的流量不能超过该弧的最大通过能力（即弧的容量）；二是中间点的流量为零，即该点的进量和输出量之差为零。
- **定义2**：满足下述条件的流 f 称为可行流：

1) 容量限制条件：对每一弧 $(v_i, v_j) \in A$ 有 $0 \leq f_{ij} \leq c_{ij}$

2) 平衡条件：

对于中间点：流出量=流入量，即对每个 $(i \neq s, t)$ 有 $\sum_{(v_i, v_j) \in A} f_{ij} - \sum_{(v_j, v_i) \in A} f_{ji} = 0$

对于发点 v_s ，记 $\sum_{(v_s, v_j) \in A} f_{sj} - \sum_{(v_j, v_s) \in A} f_{js} = v(f)$

于是对于收点 t ， $\sum_{(v_i, v_j) \in A} f_{ij} - \sum_{(v_j, v_i) \in A} f_{ji} = -v(f)$

式中 $v(f)$ 称为这个可行流的流量 即发点的净输出量 或收点的净输入量)

- 可行流总是存在的，比如令所有弧的流量 $f_{ij}=0$ ，就得到一个可行流，称为**零流**。其流量 $v(f)=0$ 。

10.4 网络最大流问题(cont.)

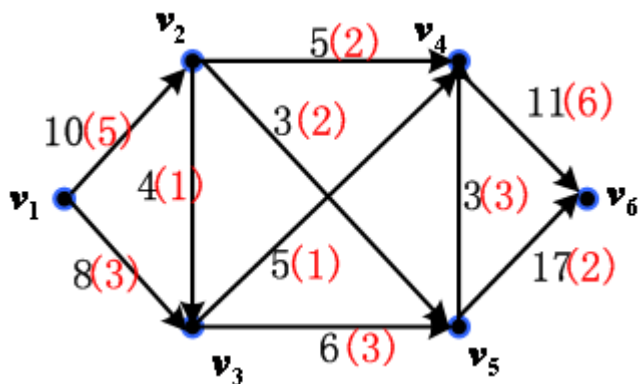
最大流问题就是求一个流 $\{f_{ij}\}$ 使其流量 $v(f)$ 达到最大，并且满足 $0 \leq f_{ij} \leq c_{ij}, (v_i, v_j) \in A$

$$\sum f_{ij} - \sum f_{ji} = \begin{cases} v(f), i = s \\ 0, i \neq s, t \\ -v(f), i = t \end{cases}$$

- 最大流问题是一个特殊的线性规划问题，即求一组 $\{f_{ij}\}$ ，在满足上述条件下使 $v(f)$ 达到最大。

— 增广链

- 若给定一个可行流 $f = \{f_{ij}\}$ ，我们把网络中使 $f_{ij} = c_{ij}$ 的弧称为 **饱和弧**，使 $f_{ij} < c_{ij}$ 的弧称为 **非饱和弧**。把 $f_{ii} = 0$ 的弧称为 **零流弧**， $f_{ij} > 0$ 的弧称为 **非零流弧**。



上例图中：

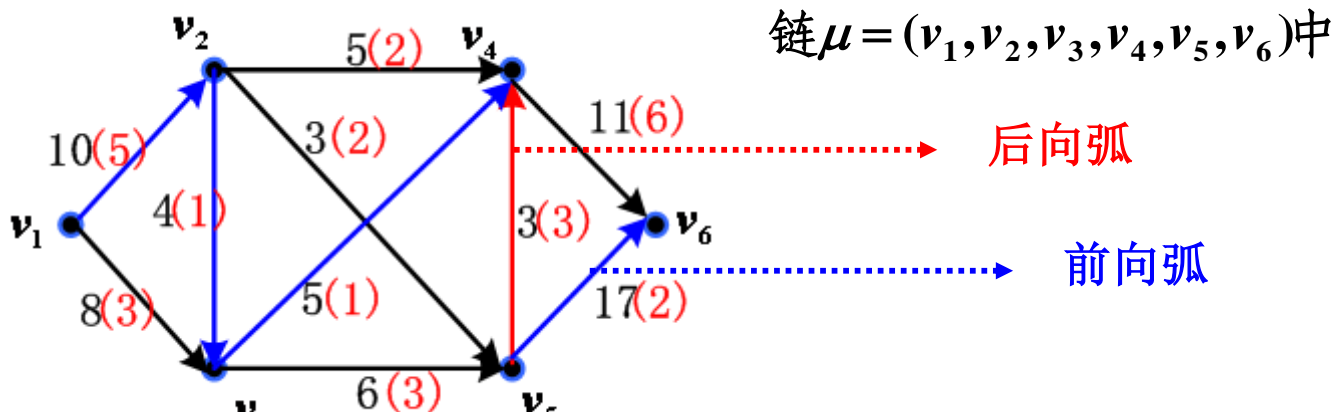
(v_5, v_4) 是饱和弧；其余为非饱和弧；所有的弧都是非零流弧。

10.4 网络最大流问题(cont.)

2014/5/4

36

- 若 μ 是网络中联结发点 v_s 和收点 v_t 的一条链，定义链的方向是从 v_s 到 v_t ，则链上的弧被分为两类：一类是弧的方向与链的方向一致，叫做**前向弧**，前向弧的全体记作 μ^+ ；另一类弧与链的方向相反，叫做**后向弧**，后向弧的全体记作 μ^- 。
- 定义3**：设 f 是一个可行流， μ 是从 v_s 到 v_t 的一条链，若 μ 满足下列条件，称之为（关于可行流 f 的）一条**增广链**：
在弧 $(v_i, v_j) \in \mu^+$ 上， $0 \leq f_{ij} < c_{ij}$ ，即 μ^+ 中每一弧是非饱和弧
在弧 $(v_i, v_j) \in \mu^-$ 上， $0 < f_{ij} \leq c_{ij}$ ，即 μ^- 中每一弧是非零流弧



上例中，链 $\mu = (v_1, v_2, v_3, v_4, v_5, v_6)$ 是一条增广链。因为在 μ^+ 和 μ^- 中的弧满足增广链的条件。
例如 $(v_1, v_2) \in \mu^+$ ， $f_{12} = 5 < c_{12} = 10$ ； $(v_5, v_4) \in \mu^-$ ， $f_{54} = 3 > 0$ 。

— 截集与截量

- 设 $S, T \subset V$, $S \cap T = \Phi$, 把始点在 S , 终点在 T 中的所有弧构成的集合记为 (S, T) 。

- **定义4:** 给定网络 $D = (V, A, G)$, 若点集 V 被剖分成两个非空集合 V_1 和 \bar{V}_1 ,

使 $v_s \in V_1$, $v_t \in \bar{V}_1$, 则把弧集 (V_1, \bar{V}_1) 称为是 (分离 v_s 和 v_t 的) 截集。

显然, 若把某一截集的弧从网络中丢去, 则从 v_s 到 v_t 便不存在路。所以直观上截集是从 v_s 到 v_t 的必经之路。

- **定义5:** 给一截集 (V_1, \bar{V}_1) , 把截集 (V_1, \bar{V}_1) 中所有弧的容量之和称为这个截集的容量, 简称为截量, 记做 $c(V_1, \bar{V}_1)$, 即

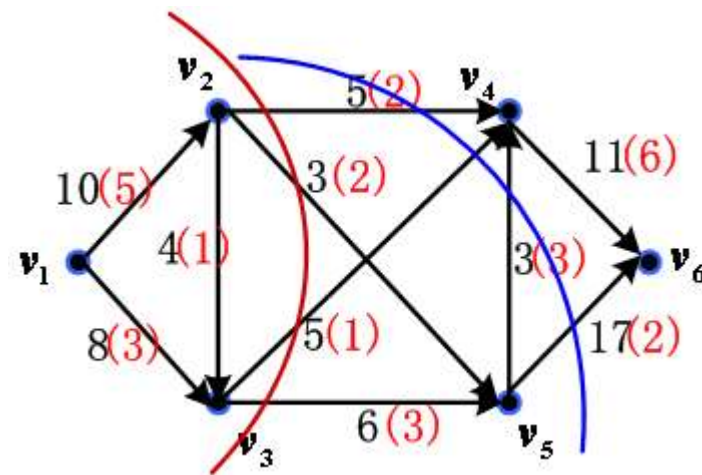
$$c(V_1, \bar{V}_1) = \sum_{(v_i, v_j) \in (V_1, \bar{V}_1)} c_{ij}$$

截集1: $(V_1, \bar{V}_1) = \{(2,4), (2,5), (3,4), (3,5)\}$

$$c(V_1, \bar{V}_1) = 5 + 3 + 5 + 6 = 19$$

截集2: $(V_1, \bar{V}_1) = \{(2,4), (5,6), (3,4), (5,4)\}$

$$c(V_1, \bar{V}_1) = 5 + 5 + 3 + 17 = 30$$



对于任一个可行流的流量 $v(f)$ 都不会超过任一截集的容量, 即 $v(f) \leq c(V_1, \bar{V}_1)$ 。

显然, 若对于一个可行流 f^* , 网络中有一个截集 (V_1^*, \bar{V}_1^*) 使 $v(f^*) = c(V_1^*, \bar{V}_1^*)$, 则 f^* 必是最大流, 而 (V_1^*, \bar{V}_1^*) 必定是 D 的所有截集中, 容量最小的一个, 即最小截集。

- 定理1: 可行流 f^* 是最大流, 当且仅当不存在关于 f^* 的增广链。

证明: 略。

- 最大流量最小截量定理: 任一个网络 D 中, 从 v_s 到 v_t 的最大流的流量等于分离 v_s, v_t 的最小截集的容量。
- 定理1提供了寻求网络中最大流的办法, 只要判断给定可行流 f 有无增广链, 就可以按照定理1证明的上半部分中的方法进行改进, 得到一个流量增大的新的可行流。调整方法如下:

若 D 中存在关于 f^* 的增广链 μ , 令 $\theta = \min\{\min_{\mu^+}(c_{ij} - f_{ij}^*), \min_{\mu^-} f_{ij}^*\}$

$$\text{令 } f_{ij}^{**} = \begin{cases} f_{ij}^* + \theta, (v_i, v_j) \in \mu^+ \\ f_{ij}^* - \theta, (v_i, v_j) \in \mu^- \\ f_{ij}^*, (v_i, v_j) \notin \mu \end{cases}$$

f_{ij}^{**} 是一个可行流, 且 $v(f^{**}) = v(f^*) + \theta > v(f^*)$ 。

- 寻求最大流的标号法(Ford-Fulkerson方法)

- 标号过程:

- 在这个过程中网络中的点是标号点或者未标号点。标号点又分为已检查和未检查两种。每个标号点的标号包括两部分：第一个标号表明它的标号是从哪一点得到的以便找出增广链；第二个标号是为了确定增广链的调整量 θ 。
 - 标号过程开始，先给 v_s 标上 $(0, +\infty)$ ，这时 v_s 是标号而未检查的点，其余都是未标号点。一般的，取一个标号而未检查的点 v_i ，对一切未标号点 v_j :

- (1)若在弧 (v_i, v_j) 上， $f_{ij} < c_{ij}$ ，则给 v_j 标号 $(v_i, l(v_j))$ 。

- 这里 $l(v_j) = \min[l(v_i), c_{ij} - f_{ij}]$ 。这时点 v_j 成为标号而未检查的点

- (2)若在弧 (v_j, v_i) 上， $f_{ji} > 0$ ，则给 v_j 标号 $(-v_i, l(v_j))$ 。

- 这里 $l(v_j) = \min[l(v_i), f_{ji}]$ 。这时点 v_j 成为标号而未检查的点。

- 重复上述步骤，一旦 v_t 被标上号，表明得到一条从 v_s 到 v_t 的增广链 μ ，转入调整过程。若所有标号都是已检查过而标号过程进行不下去时，则算法结束，说明这时的可行流就是最大流。

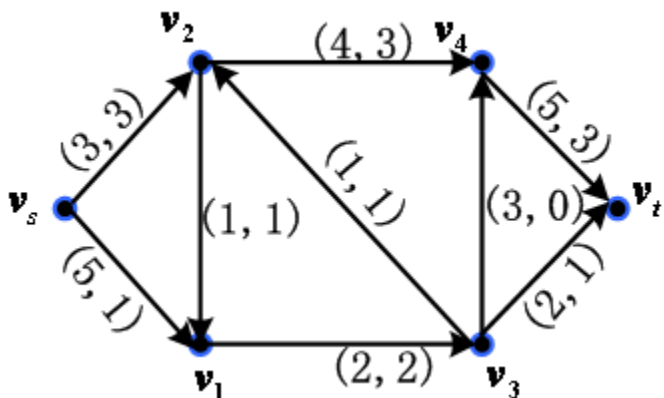
— 调整过程:

- 首先利用“反向追踪法”，按照 v_s 以及其它点的第一个标号，找出增广链 μ 。
令调整量 θ 是 $l(v_t)$ ，即 v_t 的第二个标号。

$$\text{令 } f'_{ij} = \begin{cases} f_{ij} + \theta, (v_i, v_j) \in \mu^+ \\ f_{ij} - \theta, (v_i, v_j) \in \mu^- \\ f_{ij}, (v_i, v_j) \notin \mu \end{cases}$$

去掉所有的标号，对新的可行流 $f' = \{f'_{ij}\}$ ，重新进入标号过程

- 例10.9 求下图所示网络的最大流。弧旁的数是 (c_{ij}, f_{ij}) 。



(1) 首先 v_s 标 $(0, +\infty)$

(2) 弧 (v_s, v_1) , $f_{s1} < c_{s1}$, $l(v_1) = \min[l(v_s), c_{s1} - f_{s1}] = \min[+\infty, 4]$, v_1 标 $(v_s, 4)$; 弧 (v_s, v_2) , $f_{s1} = c_{s1}$, 不满足标号条件。

(3) 弧 (v_1, v_3) , $f_{13} = c_{13}$, 不满足标号条件。弧 (v_2, v_1) , $f_{21} > 0$, $l(v_2) = \min[4, 1]$, v_2 标 $(-v_1, 1)$ 。

10.4 网络最大流问题(cont.)

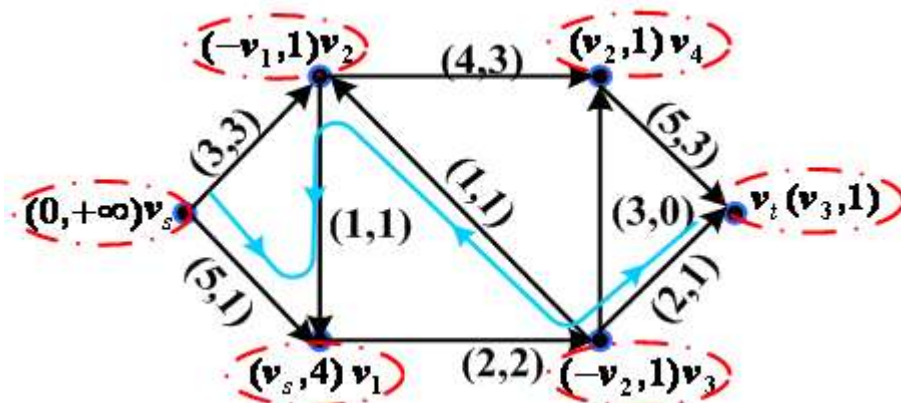
2014/5/4

41

(4)弧 (v_2, v_4) , $f_{s1} < c_{s1}$, $l(v_4) = \min[l(v_2), c_{24} - f_{24}] = \min[1, 1]$, v_4 标 $(v_2, 1)$;

弧 (v_3, v_2) , $f_{32} > 0$, $l(v_3) = \min[1, 1]$, v_3 标 $(-v_2, 1)$ 。

(5)弧 (v_3, v_t) , $f_{3t} < c_{3t}$, $l(v_t) = \min[l(v_3), c_{3t} - f_{3t}] = \min[1, 1]$, v_t 标 $(v_3, 1)$;



按点的第一个标号用反向追踪找到一条增广链 如上图蓝色线路所示。

其中： $\mu^+ = \{(v_s, v_1), (v_3, v_t)\}$, $\mu^- = \{(v_2, v_1), (v_3, v_2)\}$

按 $\theta = l(v_t) = 1$ 调整 f :

μ^+ 上： $f_{s1} + \theta = 2, f_{3t} + \theta = 2$; μ^- 上： $f_{21} - \theta = 0, f_{32} - \theta = 0$

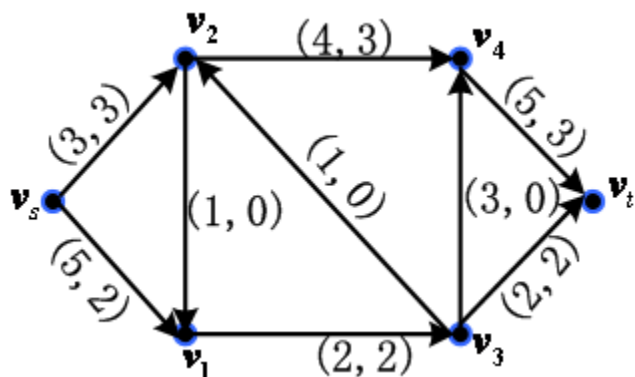
其余的 f_{ij} 不变。

10.4 网络最大流问题(cont.)

2014/5/4

42

- 第一次标号与调整得到如下可行流：

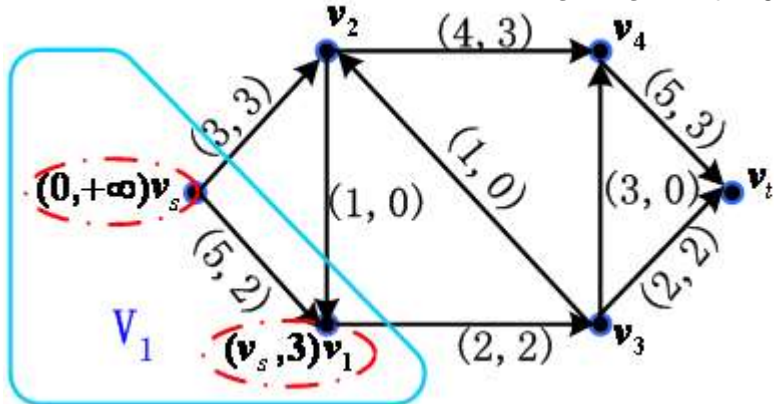


(1) 首先 v_s 标 $(0, +\infty)$

(2) 弧 (v_s, v_1) , $f_{s1} < c_{s1}$, $l(v_1) = \min[l(v_s), c_{s1} - f_{s1}] = \min[+\infty, 3]$, v_1 标 $(v_s, 3)$; 弧 (v_s, v_2) , $f_{s1} = c_{s1}$, 不满足标号条件。

(3) 弧 (v_1, v_3) , $f_{13} = c_{13}$, 不满足标号条件。弧 (v_2, v_1) , $f_{21} = 0$, 不满足标号条件。

- 此时标号过程无法继续，算法结束，上图中的可行流就是所求最大流。最大流量为： $v(f) = f_{s1} + f_{s2} = f_{4t} + f_{3t} = 5$ 。



与此同时，可找到最小截集 (V_1, \bar{V}_1) ，其中 V_1 为标号点集合（左图蓝框中部分）， \bar{V}_1 为未标号点集合。

弧集合 (V_1, \bar{V}_1) 即为最小截集。

- **Ford-Fulkerson**方法的缺点：
 - 当网络的容量可以取无理数时，用该方法可能找不到最大流。
 - 在容量全是正整数时，有时需要进行许多次迭代才能得到最大流，并且迭代的次数不仅依赖于网络中的顶点数 n 和弧数 m ，还依赖于容量和最大流的值。
- 上述**Ford-Fulkerson**方法的缺点和选取可扩充路时的任意性有关，如果在选择可扩充路时加上一定的限制，这些缺点可以克服。
- **Edmonds-Karp**方法
 - **Edmonds-Karp**方法是在1972年提出的一种方法，基本轮廓和**Ford-Fulkerson**方法相同，唯一的不同之处在于迭代的每一步如果存在增广链，那么必须在“最短”的（即包含的弧数最少的）增广链上来改进可行流。

- 给定网络 $D = (V, A, C)$, 每一弧 (v_i, v_j) 上, 除了已给容量 c_{ij} 外, 还给了一个单位流量的费用 $b(v_i, v_j) \geq 0$ (简记为 b_{ij})。所谓最小费用最大流问题就是要求一个最大流 f , 使流的总输送费用 $b(f) = \sum_{(v_i, v_j) \in A} b_{ij} f_{ij}$ 取极小值。
 - 考虑当沿着一条关于可行流 f 的增广链 μ , 以 $\theta = 1$ 调整 f , 得到新的可行流 f' 时, $v(f') = v(f) + 1$, $b(f') - b(f) = [\sum_{\mu^+} b_{ij} (f'_{ij} - f_{ij}) - \sum_{\mu^-} b_{ij} (f'_{ij} - f_{ij})] = \sum_{\mu^+} b_{ij} - \sum_{\mu^-} b_{ij}$
 $\sum_{\mu^+} b_{ij} - \sum_{\mu^-} b_{ij}$ 称为这条增广链 μ 的“费用”。
 - 若 f 是流量为 $v(f)$ 的所有可行流中费用最小者, 而 μ 是关于 f 的所有增广链中费用最小的增广链, 那么沿 μ 去调整 f , 得到的可行流 f' , 就是流量为 $v(f')$ 的所有可行流中的最小费用流。当 f' 是最大流时, 就得到了最小费用最大流。

10.5 最小费用最大流问题(cont.)

2014/5/4

45

- 由于 $b_{ij} \geq 0$, 所以 $f = 0$ 必是流量为 0 的最小费用流。构造一个赋权有向图 $W(f)$, 它的顶点是原网络 D 的顶点, 把 D 中的每一条弧 (v_i, v_j) 变成两个相反方向的弧 (v_i, v_j) 和 (v_j, v_i) , 定义 $W(f)$ 中的弧的权为:

$$w_{ij} = \begin{cases} b_{ij}, & \text{若 } f_{ij} < c_{ij} \\ +\infty, & \text{若 } f_{ij} = c_{ij} \end{cases}, \quad w_{ji} = \begin{cases} -b_{ij}, & \text{若 } f_{ij} > 0 \\ +\infty, & \text{若 } f_{ij} = 0 \end{cases}, \quad \text{长度为 } +\infty \text{ 的弧可以从 } W(f) \text{ 中略去。}$$

于是在网络 D 中求关于 f 的最小费用增广链就等价于在赋权有向图 $W(f)$ 中寻求从 v_s 到 v_t 的最短路。算法如下:

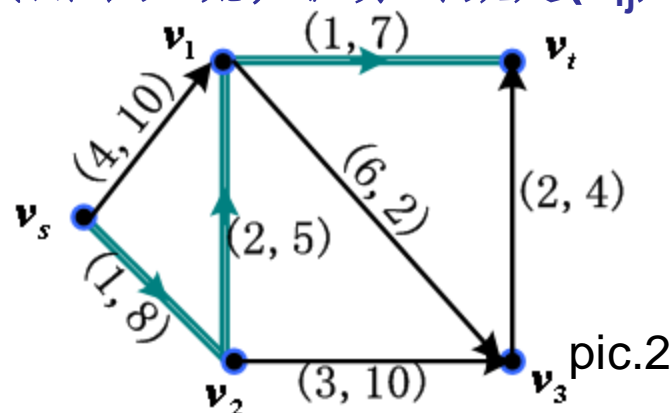
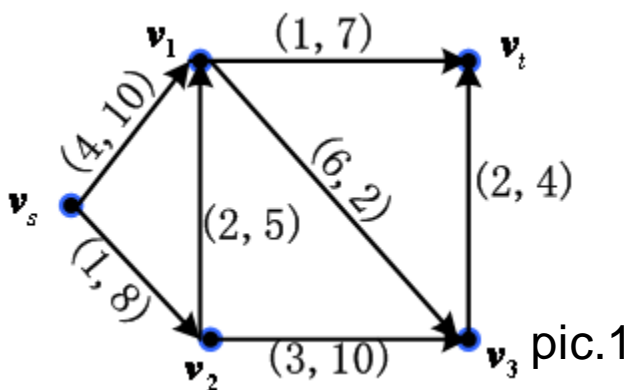
开始取 $f^{(0)} = 0$, 一般若在第 $k-1$ 步得到最小费用流 $f^{(k-1)}$, 则构造赋权有向图 $W(f^{(k-1)})$, 在 $W(f^{(k-1)})$ 中, 寻求从 v_s 到 v_t 的最短路。若不存在最短路 (即最短路权是 $+\infty$), 则 $f^{(k-1)}$ 就是最小费用最大流; 若存在最短路, 则在网络 D 中得到相应的增广链 μ , 在 μ 上对 $f^{(k-1)}$ 进行调整, 调整量为:

$$\theta = \min[\min_{\mu^+} (c_{ij} - f_{ij}^{(k-1)}), \min_{\mu^-} f_{ij}^{(k-1)}]$$

$$\text{令 } f_{ij}^{(k)} = \begin{cases} f_{ij}^{(k-1)} + \theta, & (v_i, v_j) \in \mu^+ \\ f_{ij}^{(k-1)} - \theta, & (v_i, v_j) \in \mu^- \\ f_{ij}^{(k-1)}, & (v_i, v_j) \notin \mu \end{cases} \text{ 得到新的可行流 } f^{(k)}, \text{ 再对 } f^{(k)} \text{ 重复上述步骤。}$$

10.5 最小费用最大流问题(cont.)

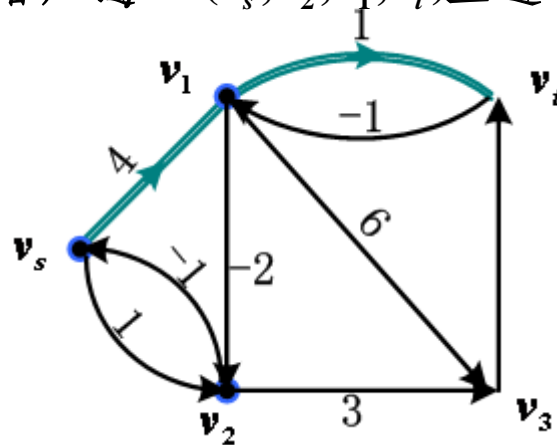
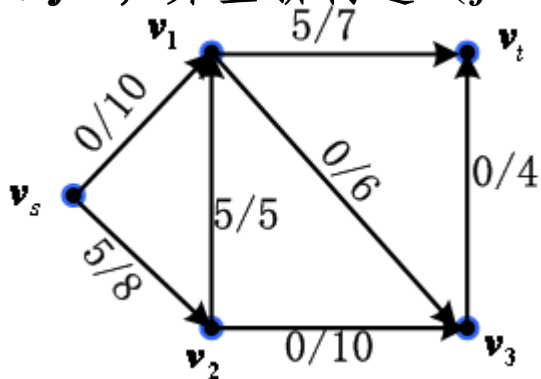
– 例10.10 求下图中的最小费用最大流，弧旁的数是 (b_{ij}, c_{ij}) 。



(1) 取 $f^{(0)} = 0$ 为初始可行流。

(2) 构造赋权有向图 $W(f^{(0)})$ ，并求出从 v_s 到 v_t 的最短路 (v_s, v_2, v_1, v_t) (如 *pic.2*)。

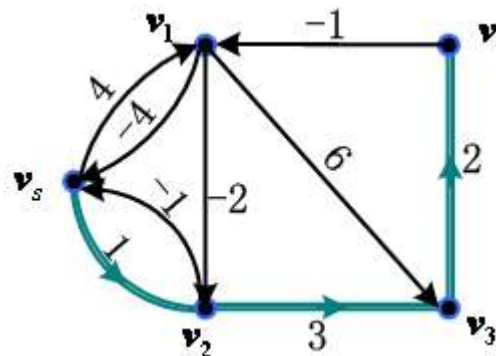
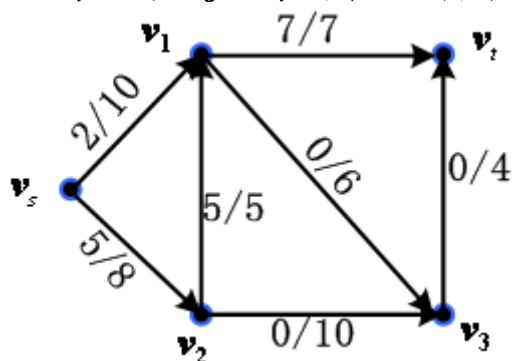
(3) 在原网络 D 中，与这条最短路径相应的增广链 $\mu = (v_s, v_2, v_1, v_t)$ 上进行调整：
 $\theta = 5$ ，得到 $f^{(1)}$ ，并重新构造 $W(f^{(1)})$ 。



10.5 最小费用最大流问题(cont.)

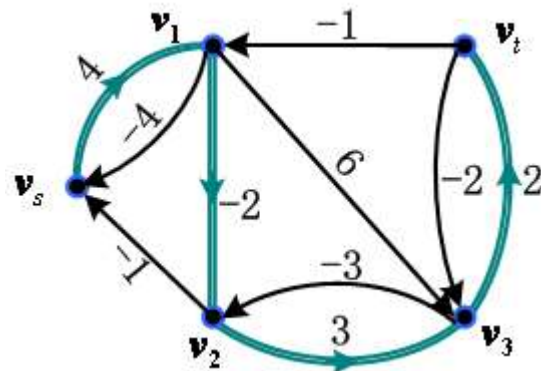
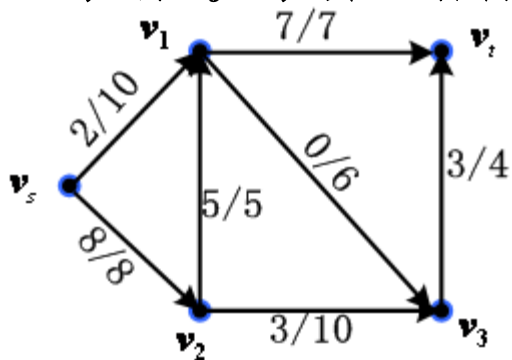
(4)构造赋权有向图 $W(f^{(1)})$, 并求出从 v_s 到 v_t 的最短路(v_s, v_1, v_t)。47

(5)在原网络 D 中, 与这条最短路径相应的增广链 $\mu = (v_s, v_1, v_t)$ 上进行调整
 $\theta = 2$, 得到 $f^{(2)}$, 并重新构造 $W(f^{(2)})$ 。



(6)构造赋权有向图 $W(f^{(2)})$, 并求出从 v_s 到 v_t 的最短路(v_s, v_2, v_3, v_t)。

(7)在原网络 D 中, 与这条最短路径相应的增广链 $\mu = (v_s, v_2, v_3, v_t)$ 上进行调整
 $\theta = 3$, 得到 $f^{(3)}$, 并重新构造 $W(f^{(3)})$ 。



10.5 最小费用最大流问题(cont.)

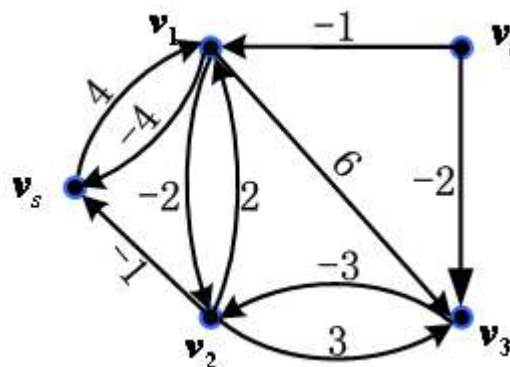
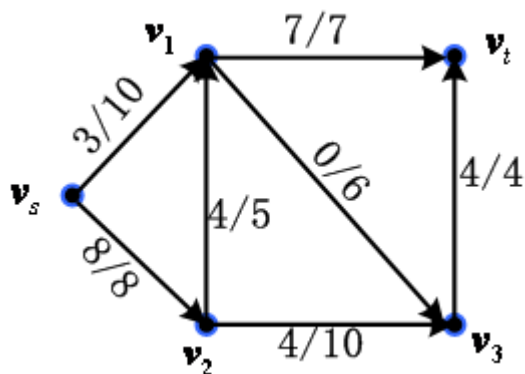
2014/5/4

48

(8)构造赋权有向图 $W(f^{(3)})$, 并求出从 v_s 到 v_t 的最短路(v_s, v_1, v_2, v_3, v_t)。

(9)在原网络 D 中, 与这条最短路径相应的增广链 $\mu = (v_s, v_1, v_2, v_3, v_t)$ 上进行调整:

$\theta = 1$, 得到 $f^{(4)}$, 并重新构造 $W(f^{(4)})$ 。



(10)注意: 在赋权有向图 $W(f^{(4)})$ 中, 已经不存在从 v_s 到 v_t 的最短路。

所以 $f^{(4)}$ 即为最小费用最大流。

- 问题的提出:

- 一个邮递员送信, 要走完他负责投递的全部街道, 完成任务后回到邮局, 应该按照怎样的路线走, 所走的路径最短。

- 将上述问题抽象为图的语言:

给定一个连通图, 在每边 e_i 上赋予一个非负的权 $w(e_i)$, 要求一个圈 (未必是简单圈) 过每个边至少一次, 并使圈的总权最小。

- 上述问题是我国管梅谷同志在**1962**年首先提出的, 因此在国际上通称为**中国邮递员问题**。

- 一笔画问题

- 给定一个连通多重图 G , 若存在一条链, 过每边一次, 且仅一次, 则称这条链为欧拉链。若存在一个简单圈, 过每边一次且仅一次, 称这个圈为欧拉圈。一个图若含有欧拉圈, 则称为欧拉图。显然, 一个图若能一笔画出, 这个图必是欧拉圈或含有欧拉链。

— 定理3 连通多重图 G 是欧拉图，当且仅当 G 中无奇点。

• 证明：

必要性是显然的。下面证明充分性：

不妨设 G 至少有三个点，对边数 $q(G)$ 进行数学归纳，因 G 是连通图，不含奇点，故 $q(G) \geq 3$ 。

首先， $q(G) = 3$ 时， G 显然是欧拉图。考察 $q(G) = n + 1$ 的情况，因 G 是不含奇点的连通图，并且 $p(G) \geq 3$ ，故存在三个点 u, v, w ，使 $[u, v], [w, v] \in E$ 。从 G 中丢去边 $[u, v], [w, v]$ 增加新边 $[u, w]$ ，得到新的多重图 G' 。 G' 有 $q(G) - 1$ 条边，并且仍不含奇点， G' 至多有两个分图。若 G' 是连通的，那么根据归纳假设， G' 有欧拉圈 C' 。把 C' 中的 $[w, u]$ 这一条边换成 $[w, v], [u, v]$ ；即得 G 中的欧拉圈。现设 G' 有两个分图 G_1, G_2 ，设 v 在 G_1 中，根据归纳假设， G_1, G_2 分别有欧拉圈 C_1, C_2 。则把 C_2 中的 $[u, w]$ 这条边换成 $[u, v]$ ， C_1 及 $[v, w]$ 就得到 G 的欧拉圈。

— 推论 连通多重图 G 有欧拉链，当且仅当 G 恰有两个奇点。

• 证明：

必要性是显然的。下面证明充分性：

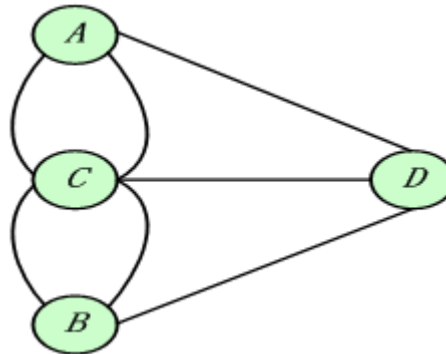
不妨设 G 恰有两个奇点 u, v ，在 G 中增加一个新点 w 以及新边 $[w, v], [w, u]$ 得到连通多重图 G' 。由定理三， G' 有欧拉圈 C' ，从 C' 中丢去点 w 及点 w 的关联边 $[w, v], [w, u]$ ，即得 G 中的一条联结 u, v 的欧拉链。

10.6 中国邮递员问题(cont.)

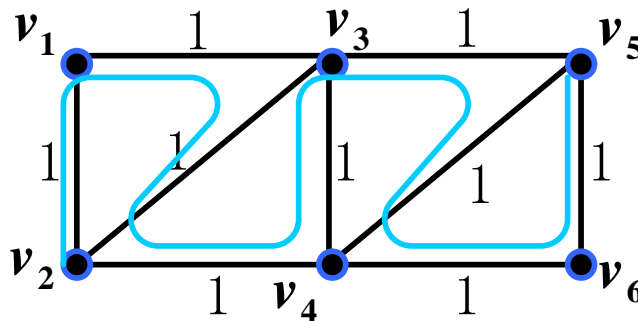
2014/5/4

51

- 上面的定理和推论提供了识别一个图是否能一笔画出的较为简单的办法。
 - 七桥问题的解：因为四个顶点皆为奇点，所以不能一笔画出：



- 下图因为有两个奇点，所以存在一条欧拉链，可以一笔画出：

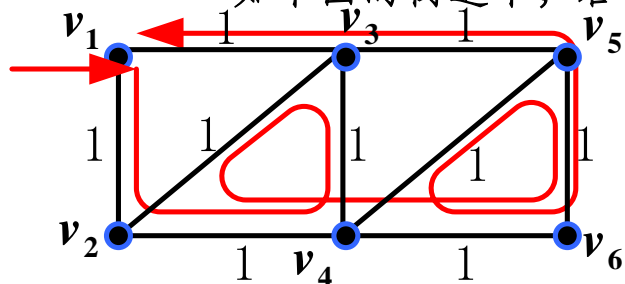


$$v_2 \rightarrow v_1 \rightarrow v_3 \rightarrow v_4 \rightarrow v_6 \rightarrow v_5$$

• 奇偶点图上作业法

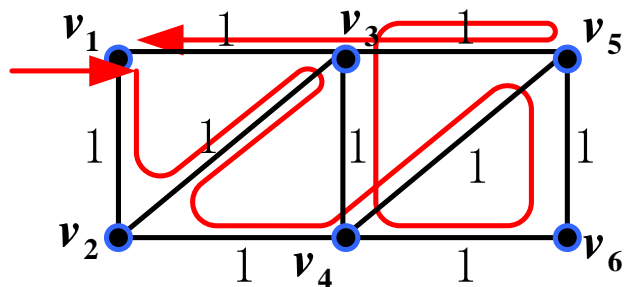
- 根据上面的定理和推论，若在邮递员所负责的范围内，街道图中没有奇点，那么他就可以从邮局出发，走过每条街道一次且仅一次，最后回到邮局。这样他走的路程也就是最短路程。

• 如下图的街道中，若 v_1 是邮局，则邮递员可以按照以下路线来投递信件：



方案1: $v_1 \rightarrow v_2 \rightarrow v_4 \rightarrow v_3 \rightarrow v_2 \rightarrow v_4$
 $\rightarrow v_6 \rightarrow v_5 \rightarrow v_4 \rightarrow v_6 \rightarrow v_5 \rightarrow v_3 \rightarrow v_1$
 总权为12

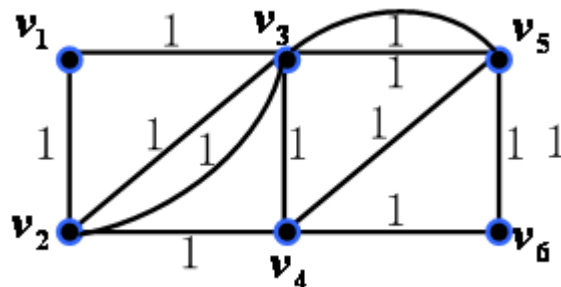
在边 $[v_2, v_4]$, $[v_4, v_6]$, $[v_6, v_5]$ 上各重复走了一次



方案2: $v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_2 \rightarrow v_4 \rightarrow$
 $v_5 \rightarrow v_6 \rightarrow v_4 \rightarrow v_3 \rightarrow v_5 \rightarrow v_3 \rightarrow v_1$
 总权为11

在边 $[v_3, v_2]$, $[v_3, v_5]$ 上各重复走了一次

-



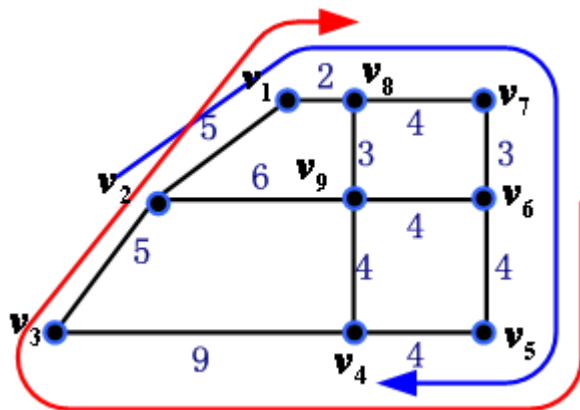
- # Computer Science and Technology

— 第一步：确定第一个可行方案

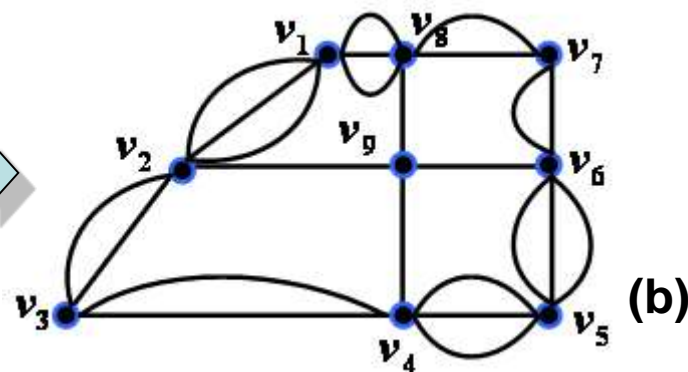
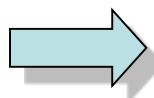
- 在任何一个图中，奇点个数必为偶数。若图中有奇点，由图的连通性可知，可以把它们配成对，将每对奇点之间的链的所有边作为重复边加到图中去。可见得到的新图中必无奇点，即得到了第一个可行方案。
- 例10.11 下图中有四个奇点， v_2, v_4, v_6, v_8 ，将 v_2, v_4 为一对， v_6, v_8 为一对。任取它们之间的链取重复边加入图中，形成新图。

取 v_2, v_4 之间的链为 $v_2, v_1, v_8, v_7, v_6, v_5, v_4$]

取 v_6, v_8 之间的链为 $v_6, v_5, v_4, v_3, v_2, v_1, v_8$]



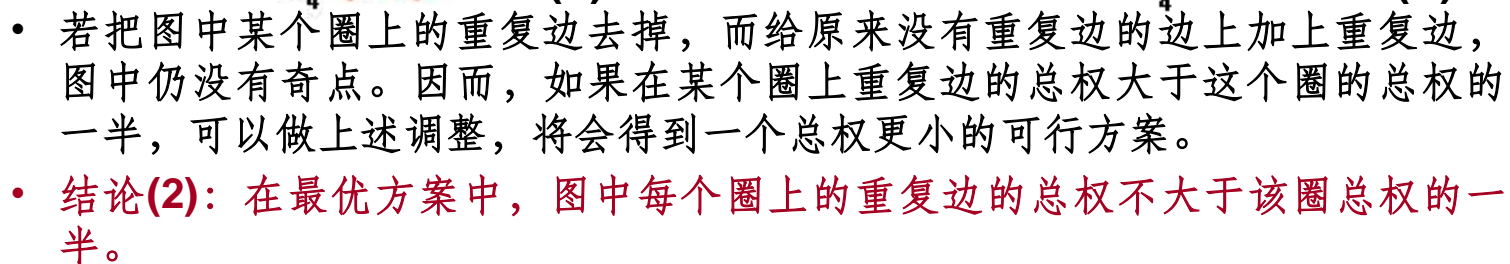
(a)



(b)

- 在上面的新图中没有奇点，故为欧拉图，对应于这个可行方案的重复边总权为： $2w_{12} + w_{23} + w_{34} + 2w_{45} + 2w_{56} + w_{67} + w_{78} + 2w_{18} = 51$

- 一般情况下, 若 $[v_i, v_j]$ 上有两条以上的重复边时, 从中去掉偶数条, 就能得到一个总权更小的可行方案。
- **结论(1):** 在最优方案中, 图的每一边上最多有一条重复边。

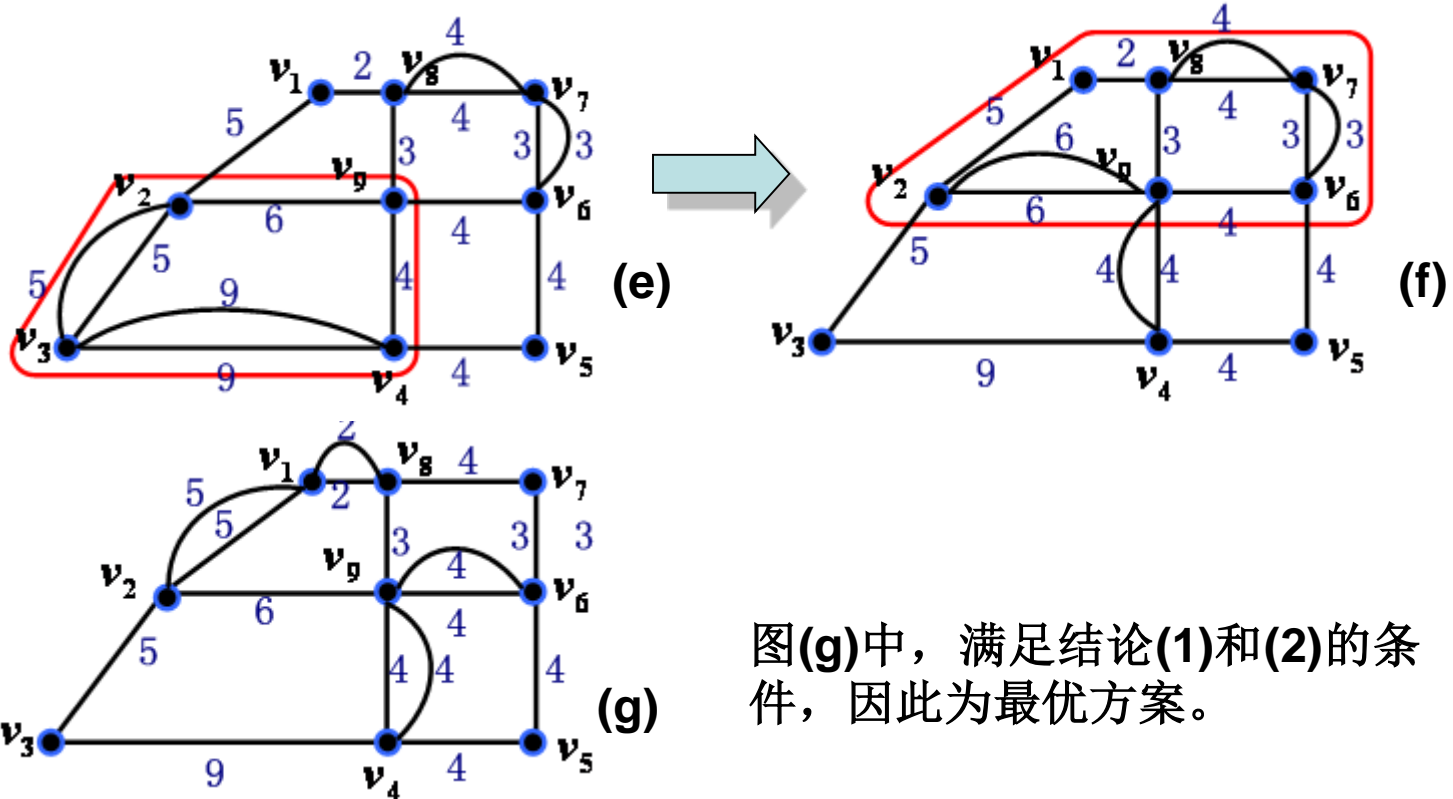


10.6 中国邮递员问题(cont.)

2014/5/4

56

- 圈 $v_2-v_3-v_4-v_9-v_2$ 中, $[v_2, v_3] + [v_3, v_4]$ 的权值大于所在圈 $v_2-v_3-v_4-v_9-v_2$ 的一半, 因此调整后得到新图(f)。
- 新图中圈 $v_2-v_1-v_8-v_7-v_6-v_9-v_2$ 中, $[v_2, v_9] + [v_6, v_7] + [v_7, v_8]$ 的权值大于所在圈 $v_2-v_1-v_8-v_7-v_6-v_9-v_2$ 的总权的一半, 因此调整后得到新图(g)



图(g)中, 满足结论(1)和(2)的条件, 因此为最优方案。

10.6 中国邮递员问题(cont.)

2014/5/4

57

- 以上所说的求最优邮递路线的方法，通常称为奇偶点图上作业法。注意：
 - 方法的主要困难在于检查条件（1），要求检查每一个圈。当图中点、边数比较多的时候，圈的个数将会很多。



2014/5/4

58

本章完
The end