离散数学 B

重点知识：

1. 支配集的概念，极小和最小支配集的概念

- 无向图 $G=<V,E>$,
- 支配集: $V^* \subseteq V$ 使得 $\forall u \in V-V^*$, $\exists v \in V^*$, uEv

一个顶点子集，使得不属于这个子集的顶点都与子集内的某顶点相邻。（通过事物【顶点】的一部分【子集】，利用特定关系【边】控制事物的全体）

- 极小支配集: V^*是支配集, 其真子集都不是
- 最小支配集: $|V^*|$最小的支配集
- 支配数: $\gamma_0(G)=|V^*|$, V^*是最小支配集

定理1

- 定理1: 无向图G无孤立点, V_1^*是极小支配集, 则存在V_2^*也是极小支配集, 且$V_1^* \cap V_2^* = \varnothing$.
- 证明: V_1^*是极小支配集, 则$V-V_1^*$也是支配集.(反证: 否则, $\exists u \in V_1^*$, $\forall v \in V-V_1^*$, $(u,v) \notin E$, 又因为u 不是孤立点, 所以$V_1^*-\{u\}$还是支配集, 与V_1^*是极小支配集矛盾.)

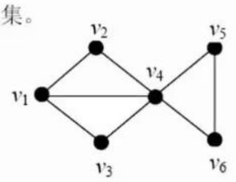
$V-V_1^*$是支配集, 则$V-V_1^*$中有子集是极小支配集, 设为V_2^*, 则$V_1^* \cap V_2^* = \varnothing$.

2. 求极小和最小支配集的算法

$$\psi(v_1, v_2, \ldots, v_n) = \prod_{i=1}^{n}\left(v_i + \sum_{u \in N(v_i)} u\right)$$

N(v)为v的邻域，求左式的最简化析取式，每一项对应一个极小支配集

例：求下图的所有极小支配集。



(a+b)(a+c)=a+bc

a(a+b)=a+ab=a

解：
$$\psi(v_1, v_2, \cdots, v_v) = (v_1+v_2+v_3+v_4)(v_2+v_1+v_4)(v_3+v_1+v_4)$$
$$\cdot(v_4+v_1+v_2+v_3+v_5+v_6)(v_5+v_4+v_6)(v_6+v_4+v_5)$$
$$= v_1v_5 + v_1v_6 + v_4 + v_2v_3v_5 + v_2v_3v_6$$

故 G 的所有极小支配集为：$\{v_1,v_5\}$，$\{v_1,v_6\}$，$\{v_4\}$，$\{v_2,v_3,v_5\}$，$\{v_2,v_3,v_6\}$。

3. 点覆盖的概念，极小和最小点覆盖的概念

- 无向图G=<V,E>
- 点覆盖: $V^* \subseteq V$ $\forall e \in E$, $\exists v \in V^*$, v关联e

一个顶点的子集，使得所有的边都与子集中的某顶点关联

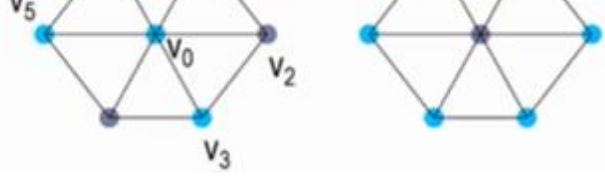
(找出一组事物【顶点】，使得由这些事物可以找到所有的关系【边】)

- 极小点覆盖: V^*是点覆盖, 其真子集都不是
- 最小点覆盖: $|V^*|$最小的点覆盖
- 点覆盖数: $\alpha_0(G)=|V^*|$, V^*是最小点覆盖

- 连通图中，点覆盖一定是支配集？
  YES!
- 极小点覆盖一定是极小支配集？
  NO！反例: $\{v_0,v_1,v_3,v_5\}$是极小点覆盖, $\{v_1,v_3,v_5\}$是极小支配集
- 支配集一定是点覆盖？
  NO！反例: $\{v_1,v_4\}$是支配集,不是点覆盖

点覆盖(例)

- $\{v_0,v_1,v_3,v_5\}$, $\{v_1,v_2,v_3,v_4,v_5,v_6\}$, $\alpha_0=4$



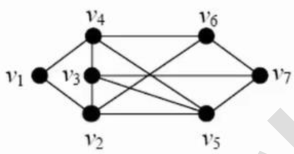
4. 求极小和最小点覆盖的算法

求最（极）小点覆盖集

$$\psi(v_1,v_2,\dots,v_n)=\prod_{i=1}^{n}\left(v_1+\prod_{u\in N(v_1)}u\right)$$

求左式的最简化析取式，每一项对应一个极小点覆盖集

例：求下图的所有极小点覆盖和极大点独立集。



解：$\varphi(v_1,v_2,\cdots,v_7)=(v_1+v_2v_4)(v_2+v_1v_3v_5v_6)(v_3+v_2v_4v_5v_7)(v_4+v_1v_3v_5v_6)$
$\cdot(v_5+v_2v_3v_4v_7)(v_6+v_2v_4v_7)(v_7+v_3v_5v_6)$

$=v_1v_3v_5v_6+v_2v_3v_4v_5v_6+v_2v_4v_5v_7+v_2v_3v_4v_7$。

故极小点覆盖有：

$C_1=\{v_1,v_3,v_5,v_6\}$，$C_2=\{v_2,v_3,v_4,v_5,v_6\}$，$C_3=\{v_2,v_4,v_5,v_7\}$，$C_4=\{v_2,v_3,v_4,v_7\}$。

5. 独立集的概念，极大和最大独立集的概念

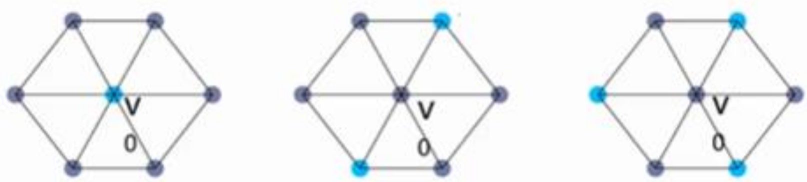
- 无向图G=<V,E>
- 独立集: $V^*\subseteq V$ $\forall u,v\in V^*$, $(u,v)\notin E$

一个顶点的子集，其中任意两个顶点之间没有边相连（根据给定关系【边】，找出一个互不干扰的事物集合）

- 极大独立集: V*是独立集, 其真母集都不是
- 最大独立集: |V*|最大的独立集
- 独立数: $\beta_0(G)=|V^*|$, V*是最大独立集

独立集(例)

- $\{v_0\}$, $\{v_1,v_4\}$, $\{v_1,v_3,v_5\}$, $\beta_0=3$



定理2

- 定理2: 无向图G无孤立点,V*是极大独立集,则V*也是极小支配集.

- 证明: V*是极大独立集,则V*也是支配集.(反证: 否则, $\exists u\in V-V^*$, $\forall v\in V^*$, $(u,v)\notin E$, $V^*\cup\{u\}$还是独立集,矛盾.)

V*是极小支配集(反证: 否则, $\exists u\in V^*$, $V^*-\{u\}$是支配集, 则$\exists v\in V^*$, $(u,v)\in E$, 与V*是独立集相矛盾.)

逆命题不成立:
极小支配集不一定是极大独立集

定理3

- 定理3: 无向图G无孤立点, $V^*\subset V$,

V^*是点覆盖 $\Leftrightarrow$ $V-V^*$是独立集.

极小 极大

证明:

($\Rightarrow$) (反证) 若$V-V^*$不是独立集, 则$\exists u,v\in V-V^*$, 且$(u,v)\in E$, 则V^*不是点覆盖, 与V^*是点覆盖矛盾.

($\Leftarrow$) $V-V^*$是独立集, $\forall(u,v)\in E$, $u\notin V-V^* \vee v\notin V-V^*$, 即 $u\in V^* \vee v\in V^*$, 故 V^*是点覆盖.

推论: 无向图G无孤立点, V^*是极(最)小点覆盖$\Leftrightarrow$$V-V^*$是极(最)大独立集. $\alpha_0+\beta_0=n$.

总结:

V_1^*是极小支配集,则存在V_2^*也是极小支配集,且$V_1^*\cap V_2^*=\varnothing$

V*是点覆盖，则V*也是支配集

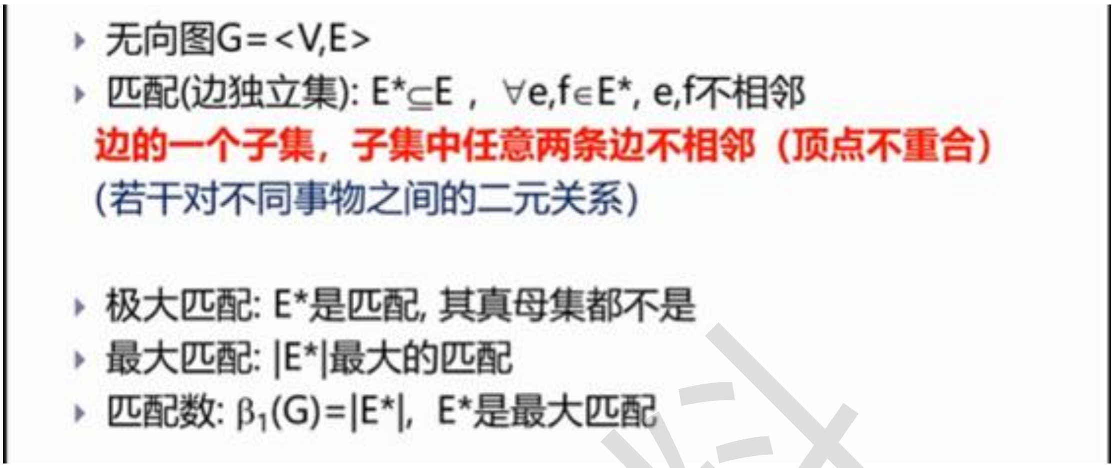
V*是极大独立集, 则V*也是极小支配集

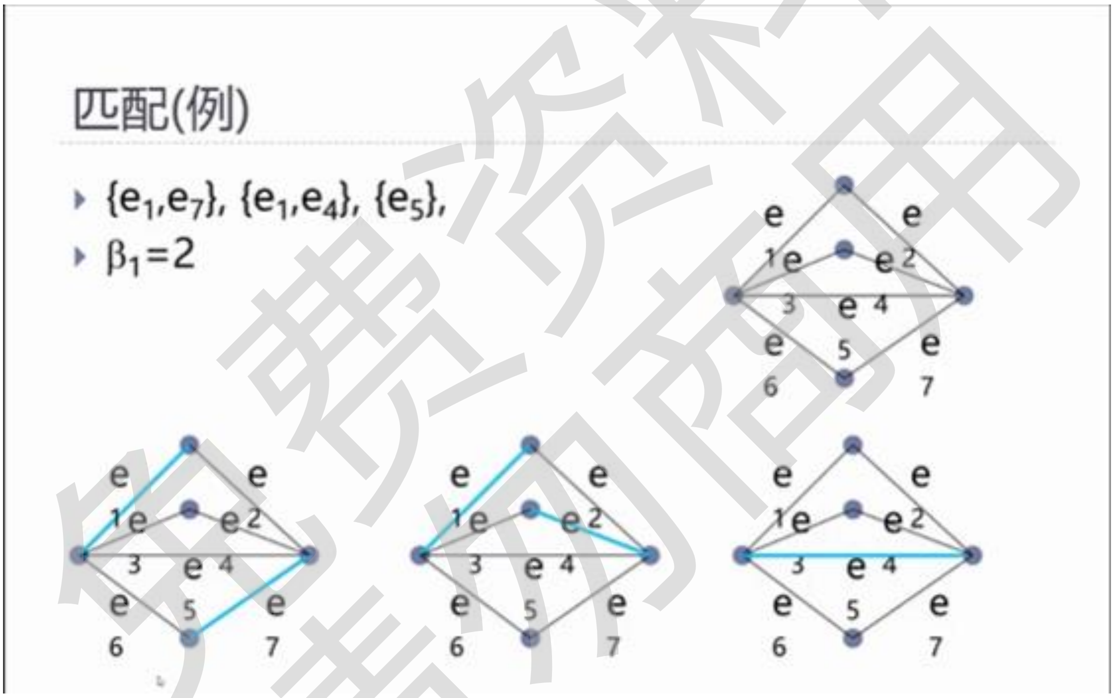
V*是点覆盖 $\Leftrightarrow$ V-V*是独立集.

V*是G的团 $\Leftrightarrow$ V*是G补的独立集.

求解：极小支配集，极小点覆盖，极大独立集

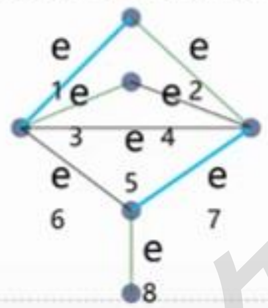
6. 匹配的概念，可增广的交错路径的概念，求二部图最大匹配的匈牙利算法

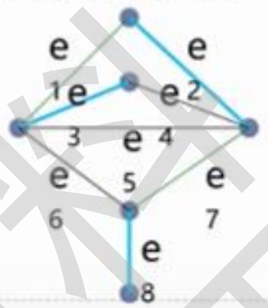




饱和点,交错路径,增广路径

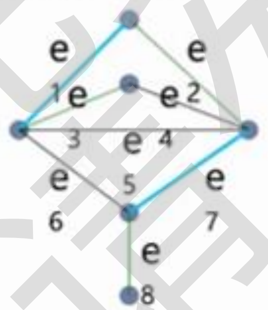
- 设M是G中匹配
- **饱和点:** v与M中边关联
- **非饱和点:** v不与M中边关联
- **交错路径:** 在M和E-M中交替取边的路径
- **可增广交错路径:** 两端都是非饱和点的交错路径

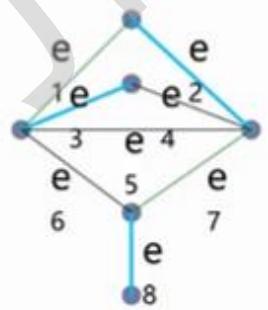




求解最大匹配

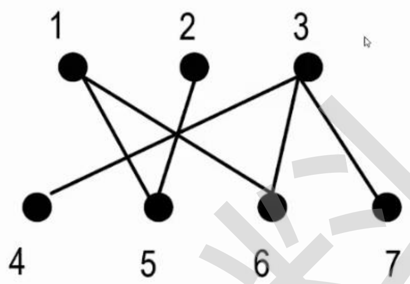
- 从一个匹配开始
- 逐一检查不饱和点，对每一个不饱和点尝试寻找增广路径。（DFS/BFS）
- 得到更大的匹配
- 递归直到没有不饱和点或没有增广路径

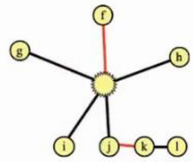


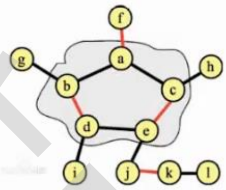


```
int Path(结点v)
{
  for (每个与v邻接的结点u)
  {
    if(u不饱和 || path(与u的匹配的那个点))
    {
      v和u构成新匹配;
      u原有的匹配取消;
      return 1;
    }
  }
  return 0;
}
```

```
for(每个不饱和点v)
{
   path (v)
}
```







7. 网络流的概念

一些符号和定义

- V表示整个图中的所有结点的集合。
- E表示整个图中所有边的集合。
- G = (V, E)，表示整个图。
- s表示网络的源点， t表示网络的汇点。
- 对于每条边(u, v)，有一个容量c(u, v)，(c(u, v)>=0)。
- 如果c(u, v)=0，则表示边(u, v)不存在。 c(u, v)>0，表示u, v之间有边
- 对于每条边(u, v)，有一个实际流量f(u, v)。

s, t, c(u,v), f(u,v)

网络流的三个性质

1、**容量限制**：f[u,v]<=c[u,v]

2、**反对称性**：f[u,v] = - f[v,u]

3、**流量平衡**：对于不是源点也不是汇点的任意结点,流入该结点的流量和等于流出该结点的流量和。结合反对称性,流量平衡也可以写成,对于任意结点v，有：

$$\sum_{u \in V} f(v,u) = 0$$

只要满足这三个性质,就是一个合法的网络流,也称为可行流。可行流至少有一个：零流。

最大流问题

- 定义一个网络的流量 F= $\sum_{v \in V} f(s,v)$ (即从源点流出的总流量)
- **最大流**问题，就是求在满足网络流3条性质的情况下,求F的最大值。

弧的分类

给定一个可行流F:

零流弧：f(u,v)=0

非零流弧：f(u,v)>0

饱和弧：f(u,v)=c(u,v)

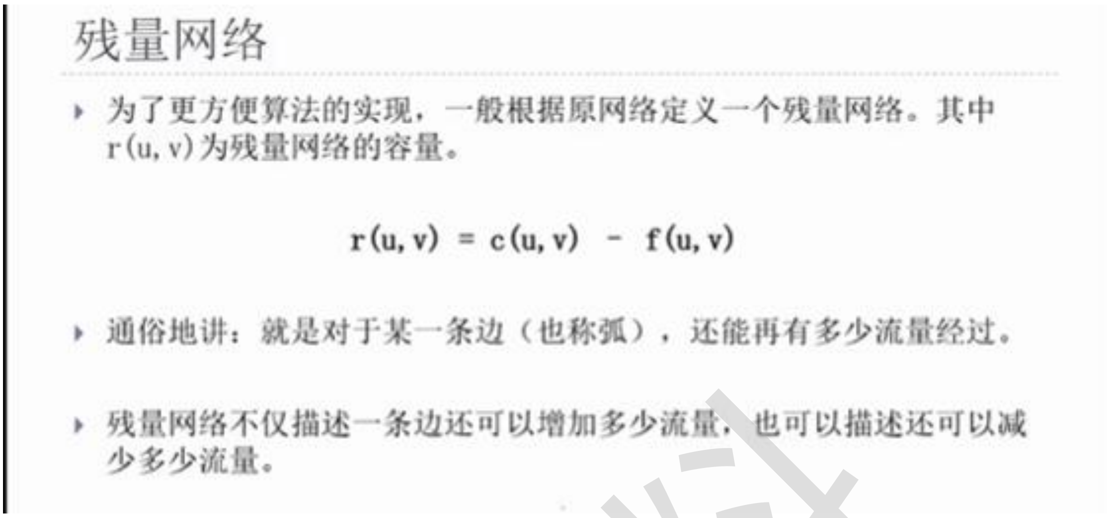
非饱和弧：f(u,v)<c(u,v)

若P是网络中联结源点s和汇点t的的一条路(不考虑边的方向性)，若称路的方向是从V_s到V_t，则路上的弧被分为两类：

前向弧：与路的方向一致的边。

后向弧：与路的方向相反的边。

8. 残量网络与可改进路


残量网络
为了更方便算法的实现，一般根据原网络定义一个残量网络。其中r(u, v)为残量网络的容量。
r(u, v) = c(u, v) - f(u, v)
通俗地讲：就是对于某一条边（也称弧），还能再有多少流量经过。
残量网络不仅描述一条边还可以增加多少流量，也可以描述还可以减少多少流量。

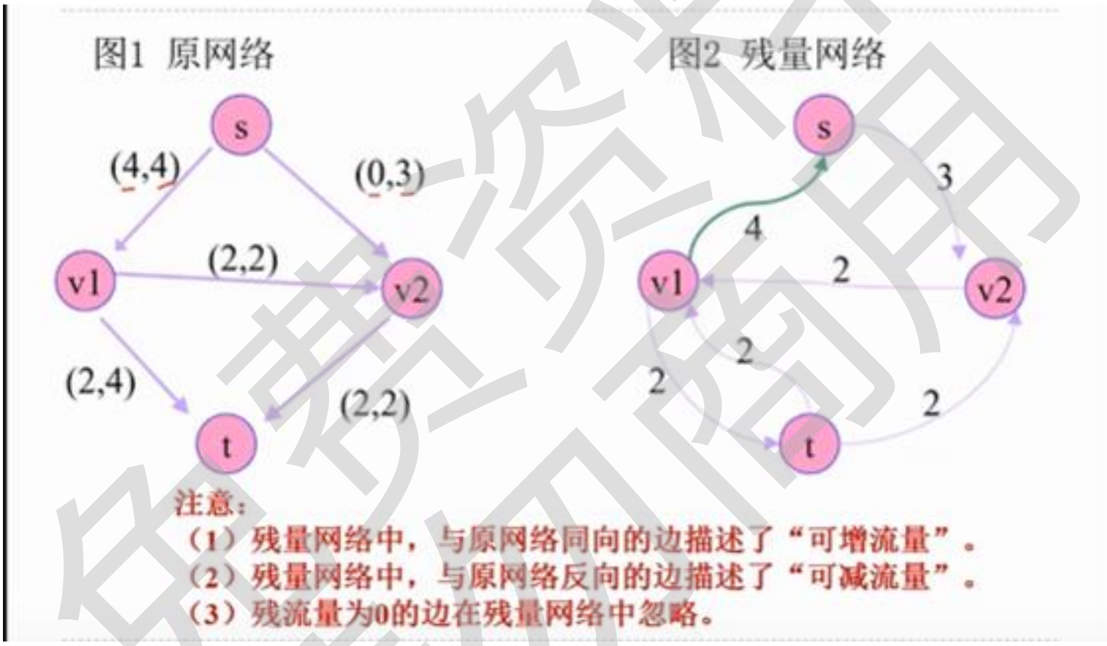


图1 原网络
图2 残量网络
s
v1
v2
t
(4,4)
(0,3)
(2,2)
(2,4)
(2,2)
3
4
2
2
2
2
注意：
（1）残量网络中，与原网络同向的边描述了“可增流量”。
（2）残量网络中，与原网络反向的边描述了“可减流量”。
（3）残流量为0的边在残量网络中忽略。

9. 标号法求最大流

标号法寻求可改进路（Ford-Fulkerson算法）

标号过程：

- 标号过程开始，总先给起点V_s标上$(0, +\infty)$，这V_s时是标号而未检查的顶点，其余都是未标号点。
- 取一个标号而未检查的标号V_i 进行检查，考察与他相邻的每个未标号点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V_j成为标号未检查的顶点。

  (2) 若在弧(V_j, V_i)上$F_{ij}>0$,则给V_j标号$(-V_i, L(V_j))$，这里$L(V_j)=\min[L(V_i), F_{ij}]$。

  这时V_j成为标号未检查的顶点。

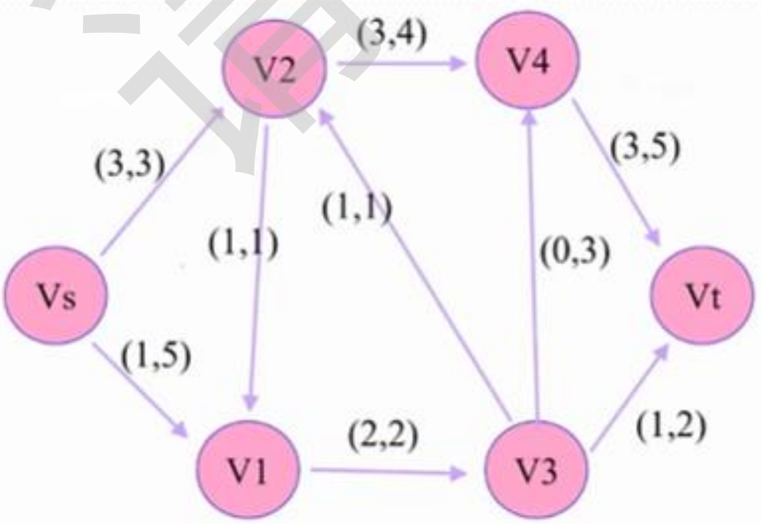
- 在V_i的全部可标号的相邻顶点都已标号后，V_i成为标号且已检查过的顶点。不断重复第二步，检查新产生的标号但未检查的顶点。一旦终点V_t被标上号，表明得到一条从V_s到V_t的可改进路P，转入调整过程。

调整过程：

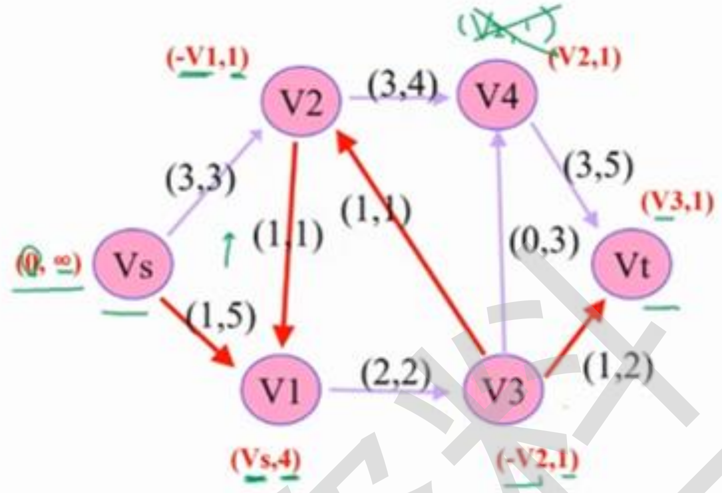
采用“倒向追踪”的方法，从终点Vt开始，利用标号的前半部分Vx找出可改进路P，并以所有标号的后半部分L(Vx)的最小值a作为改进量，改进P上的流量：

$$F'_{ij}=\begin{cases} F_{ij}+a & (V_i,V_j)\in P^+ \quad \text{前向弧（标号中顶点为+）} \\ F_{ij}-a & (V_i,V_j)\in P^- \quad \text{后向弧（标号中顶点为-）} \\ F_{ij} & (V_i,V_j)\notin P \quad \text{不在改进路中的弧} \end{cases}$$

去掉所有的标号，对新的可行流重新进入标号过程。直到标号过程无法继续。



例1 用标号法求如下网络的最大流



1.标号过程

(1) 首先给V_s标上$(0,+\infty)$

(2) 检查V_s。弧(V_s,V_2)上，$F_{s2}=C_{s2}=3$，不满足标号条件；弧(V_s,V_1)上，$F_{s1}=1<C_{s1}=5$，则V_1的标号为$(V_s,L(V_1))$，其中$L(V_1)=\min[L(V_s),(C_{s1}-F_{s1})]=\min[+\infty,5-1]=4$

(3) 检查V_1。弧(V_1,V_3)上，$F_{13}=C_{13}=2$，不满足标号条件；弧(V_2,V_1)上，$F_{21}=1>0$，则给V_2记下标号为$(-V_1,L(V_2))$，其中$L(V_2)=\min[L(V_1),F_{21}]=\min[4,1]=1$

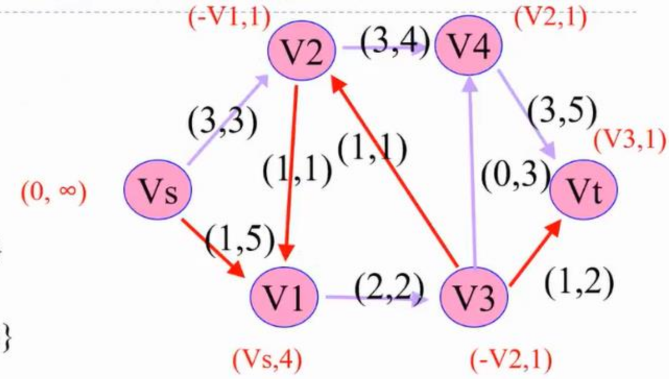
(4) 检查V_2。弧(V_2,V_4)上，$F_{24}=3<C_{24}=4$，则给V_4标号$(V_2,L(V_4))$，其中$L(V_4)=\min[L(V_2),C_{24}-F_{24}]=\min[1,1]=1$
同理，标注V_3为$(-V_2,1)$.

(5) 在V_3，V_4中任选一个进行检查，如V_3。弧(V_3,V_t)上，$F_{3t}=1<C_{3t}=2$，则给V_t标号$(V_3,L(V_t))$，其中$L(V_t)=\min[L(V_3),C_{3t}-F_{3t}]=\min[1,1]=1$

(6) V_t有了标号，转入调整过程。

2.调整过程

(1) 如图为按照每个标号中前面的顶点找到的一个可改进路



(2) 前向弧集合

$$P^{+}=\{(V_s,V_1),(V_3,V_t)\}$$

(3) 后向弧集合

$$P^{-}=\{(V_2,V_1),(V_3,V_2)\}$$

(4) 按a＝1在P上调整

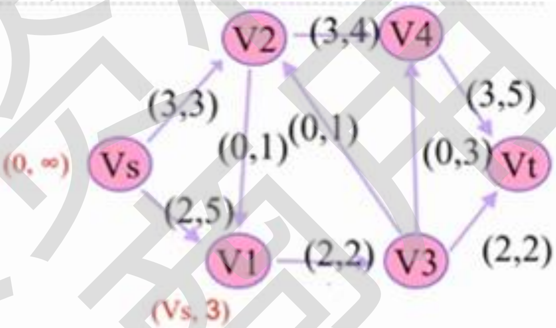
$$P^{+}:F_{s1}^{'}=F_{s1}+a=2,F_{3t}^{'}=F_{3t}+a=2$$

$$P^{-}:F_{21}^{'}=F_{21}-a=0,F_{32}^{'}=F_{32}-a=0$$

(5) 对调整后的图重新进入标号过程

3. 重新开始标号过程

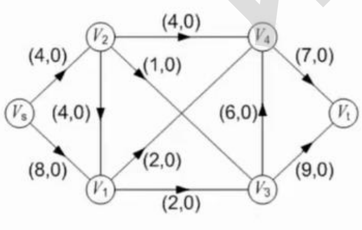
当算法进行到检查V_1时$F_{21}=0$，$F_{13}=C_{13}$均不符合条件，标号无法继续，这时的可行流为最大流5。



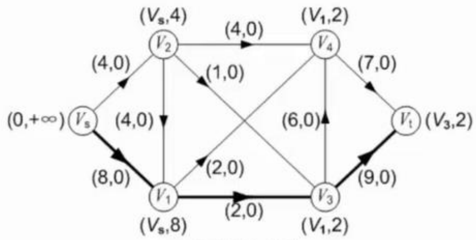
以上是寻求最大流的算法，

也可以通过最大流最小割定理来对最大流加以计算和验证

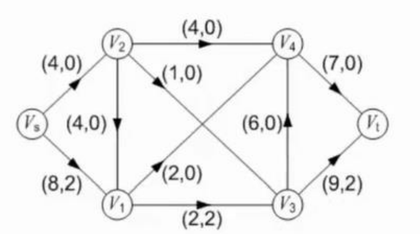
求解最大流 – 例



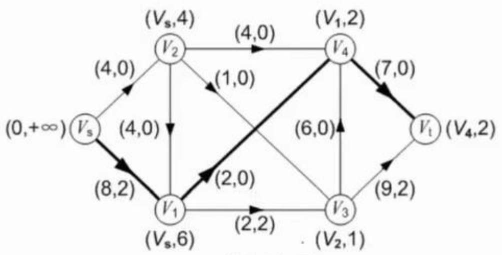
(a) 初始流为零流



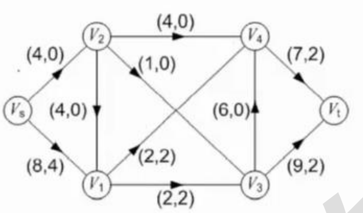
(b) 第1次标号



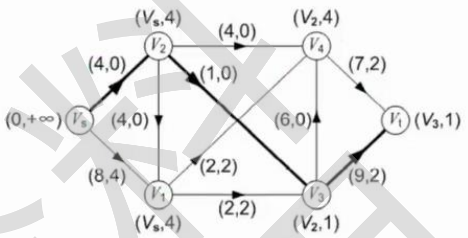
(c) 第1次调整后的网络流



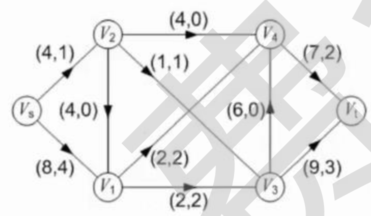
(d) 第2次标号



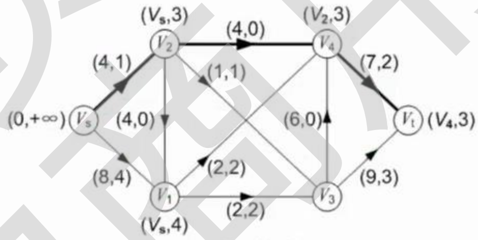
(e) 第2次调整后的网络流



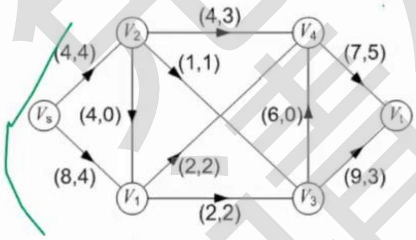
(f) 第3次标号



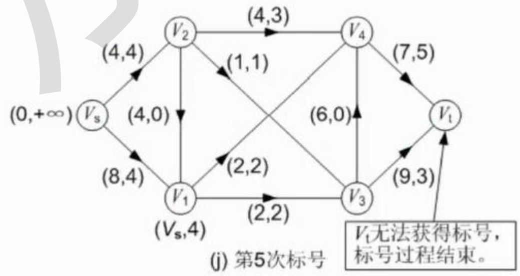
(g) 第3次调整后的网络流



(h) 第4次标号



(i) 第4次调整后的网络流



(j) 第5次标号

10. 面向高维随机变量的链式法则和贝叶斯法则

对于高维随机变量，两个基本法则分别对应这两个基本问题：

◆ 加法法则——边缘概率：

$$p(X)=\sum_{Y}\mathrm{p}(X,Y)$$

X为性别，Y职业

男性的概率=所有职业男性的概率和

◆ 乘法法则——条件概率：

$$p(\mathrm{X},Y)=p(X|Y)p(Y)=\mathrm{p}(Y|X)p(X)$$

根据两个基本法则可以推出两个常用的法则：

◆ 链式法则：

$$p(x_1,x_2,\ldots,x_p)=p(x_1)\prod_{i=2}^{p}p(x_i|x_1,x_2,\ldots,x_{i-1})$$

◆ 贝叶斯法则：

$$p(x_2|x_1)=\frac{p(x_1,x_2)}{p(x_1)}=\frac{p(x_1,x_2)}{\sum_{x_2}p(x_1,x_2)}=\frac{p(x_1|x_2)p(x_2)}{\sum_{x_2}p(x_1,x_2)}$$

11. 朴素贝叶斯算法、马尔科夫链、条件独立性的概念与假设

◆ 假设1：假设维度之间是相互独立，彼此之间互不相干

➢ 典型的算法：朴素贝叶斯算法，即假设各个维度相互独立

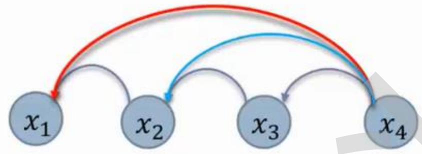
$$\mathrm{p}(x_1,x_2,\ldots,x_p)=\prod_{i=1}^{p}p(x_i)$$

➢ 缺陷：过于简化

◆ 假设2：假设**当前状态只与前一个状态有关，其他状态无关**，即马尔可夫假设

$$X_j \perp X_{i+1} | X_i \quad 其中j<i$$

➢ 缺陷：一个状态往往跟前面多个状态有关



◆ 假设3：将所有变量分为三个互不相交的集合A，B，C，得到**条件独立性假设**：

$$X_A \perp X_B | X_C$$

◆ **概率图研究问题：高维随机变量**

◆ **解决办法：条件独立性假设**

12. 贝叶斯网络的概念

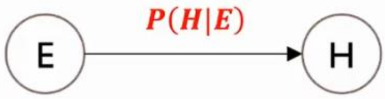
贝叶斯网络

◆ 节点：表示随机变量$\{X_1, X_2, \ldots, X_n\}$，即可观察到的变量，或隐变量、未知参数等

◆ 边：节点间有因果关系或非条件独立的变量或命题则用箭头来连接

◆ 权值：两个节点间以单箭头连接，父节点是“因”，子节点是“果”，两节点产生条件概率值

◆ 规范化表示：假设**节点**E直接影响到节点H，即E->H，则用从E指向H的箭头建立节点E到节点H的**有向弧**（E，H），**权值**为条件概率$P(\mathrm{H}|\mathrm{E})$



令G=（L,E）表示一个有向无环图（DAG），

- L代表图形中所有的结点的集合
- E代表有向连接线段的集合
- X_i （i∈L）为有向无环图中的某一结点i所代表的随机变量
- 若所有结点x={X_i }的联合概率可以表示成：

$$P(x)=\prod_{i\in I}p\left(x_i|x_{pa(i)}\right)$$

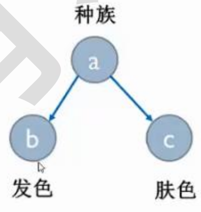
则称X为相对于**一有向无环图G的贝叶斯网络**，其中pa(i)表示结点i的父节点集合。

13. 基于贝叶斯网络的高维随机变量因子分解

◆ 链式法则： $p(x_1,x_2,\dots,x_k)=p(x_1)p(x_2|x_1)\dots p(x_k|x_1,\dots,x_{k-1})$

◆ 因子分解： $p(x_1,x_2,\dots,x_p)=\prod_{i=1}^{p}p\left(x_i|x_{pa(i)}\right)$

◆ 如下图所示，一个简单的贝叶斯网络：



链式法则：

$$p(a,b,c)=p(a)p(b|a)p(c|a,b)$$

因子分解：

$$p(a,b,c)=p(a)p(b|a)p(c|a)$$

◆ 概率图就是将图赋予了概率定义，可以直观的根据图结构寻找到概率之间的独立性。

◆ 根据条件独立性假设，我们可以将复杂的计算简化。

条件独立性：

$$X_A \perp X_B \mid X_C$$

因子分解：

$$p(x_1, x_2, \dots, x_p) = \prod_{i=1}^{p} p(x_i \mid x_{pa(i)})$$

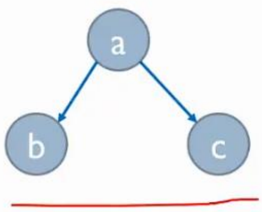
其中：$x_{pa(i)}$表示x_i的父节点。

◆ 根据概率图模型可以直观写出因子分解
◆ 图结构作用：表达条件独立性。箭头符号定义如下：

贝叶斯网络 (1) tail-tail

◆ 根据因子分解：

$$p(a, b, c) = p(a)p(b|a)p(c|a)$$



◆ 根据链式法则：

$$p(a, b, c) = p(a)p(b|a)p(c|a, b)$$

◆ 两个公式对比：

$$p(c|a) = p(c|a, b)$$

即在**给定a的前提下，b和c相互独立。**

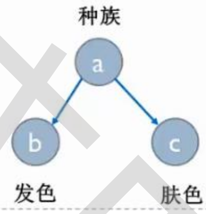
继续推导：

$$p(c|a)p(b|a) = p(c|a,b)p(b|a) = p(c,b|a)$$

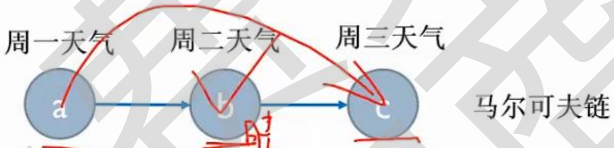
即得到了条件独立的定义：

$$p(\mathrm{c},b|a) = p(c|a)p(b|a)$$

即在tail-tail的拓扑结构中，若a被观察，则路径阻塞，即b和c相互独立。

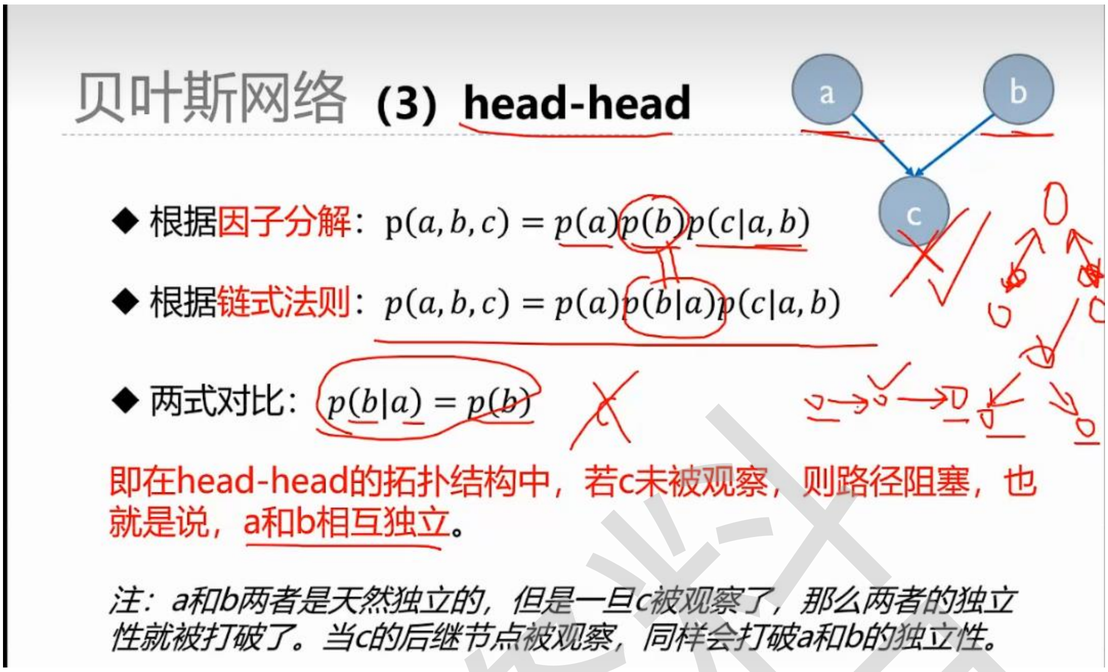


贝叶斯网络 **(2) head-tail**

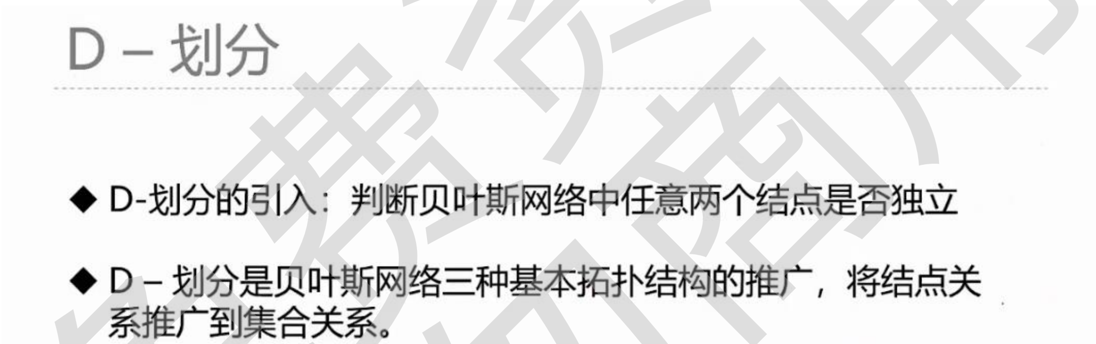


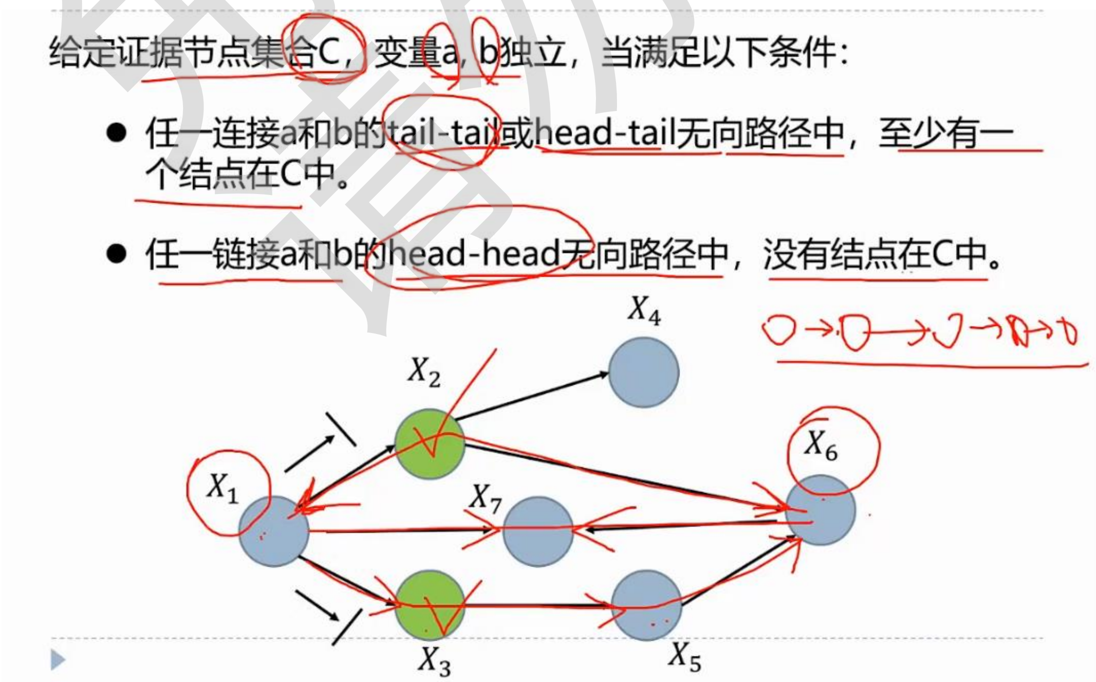
- 根据因子分解：$\mathrm{p}(a,b,c) = p(a)p(b|a)p(c|b)$
- 根据链式法则：$p(a,b,c) = p(a)p(b|a)p(c|a,b)$
- 两式对比：$p(c|b) = p(c|a,b)$

即在head-tail的拓扑结构中，若b被观察，则路径阻塞，也就是说，a和c相互独立。



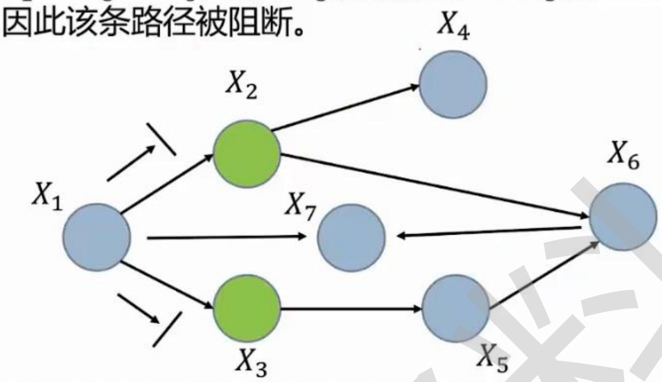
14. 贝叶斯网络的 D-划分





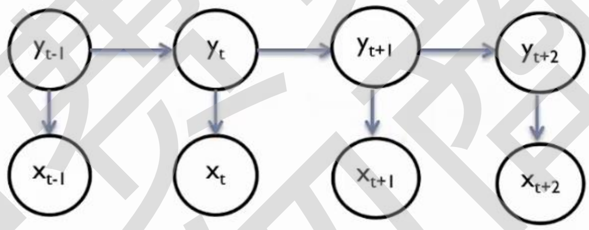
如下图所示，在给定X_2和X_3的情况下，X_1和X_6是独立的，即$X_1 \perp X_6|(X_2, X_3)$。具体来说，从X_1到X_6有两条路径：

- X_1-> X_2-> X_6，其中X_2被观测到，且X_2是head-tail结构，因此该条路径被阻断。
- X_1-> X_3-> X_6，其中X_3被观测到，且X_3是head-tail结构，因此该条路径被阻断。



15. 隐马尔科夫模型的构成及其三个问题（评估、预测、学习）

隐马尔可夫链：一个隐状态序列产生一个观察值序列。每个隐状态依赖于前一个隐状态



HMM的模型参数：

- 状态种类数量 = K=3, 观察值种类数量= M=3
- π: 每种状态做为初始状态的概率 (K dimensional vector)
- A: 状态之间的转移概率 Transition probabilities (K×K matrix)
- B: 状态到观察值的发射概率 Emission probabilities (K×M matrix)

1. 评估问题：给定观察序列, $x_1, x_2, \cdots x_n$ 和模型参数 (π, A, B), 观察序列的概率有多大？（根据语言模型判断一句话是不是人话） **Evaluation**
2. 解码问题：给定观察序列 $x_1, x_2, \cdots, x_n$ 和模型 (π, A, B), 最可能的状态序列是什么？（根据语言模型给一句话的每个词进行类别标注） **Decoding**
3. 学习问题：给定观察序列 $x_1, x_2, \cdots, x_n$，最佳的模型参数 (π, A, B) 是什么？（训练获得语言模型） **Learning**

补充知识：

1. 边覆盖的概念，极小和最小边覆盖的概念

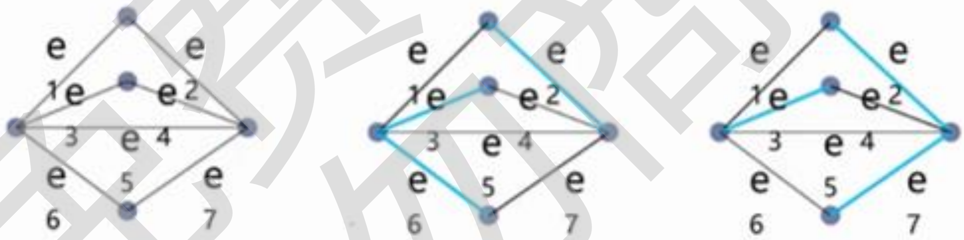
- 无向图$G=<V,E>$
- 边覆盖: $E^* \subseteq E$ ，$\forall v \in E, \exists e \in E^*$, e关联v

所有边的一个子集，任意顶点均与子集中的某条边关联(一个能包含所有事物的关系集合)

- 极小边覆盖: E^*是边覆盖, 其真子集都不是
- 最小边覆盖: $|E^*|$最小的边覆盖
- 边覆盖数: $\alpha_1(G)=|E^*|$, E^*是最小边覆盖

边覆盖(例)

- $\{e_2,e_3,e_6\}$, $\{e_2,e_3,e_7\}$, $\alpha_1=3$

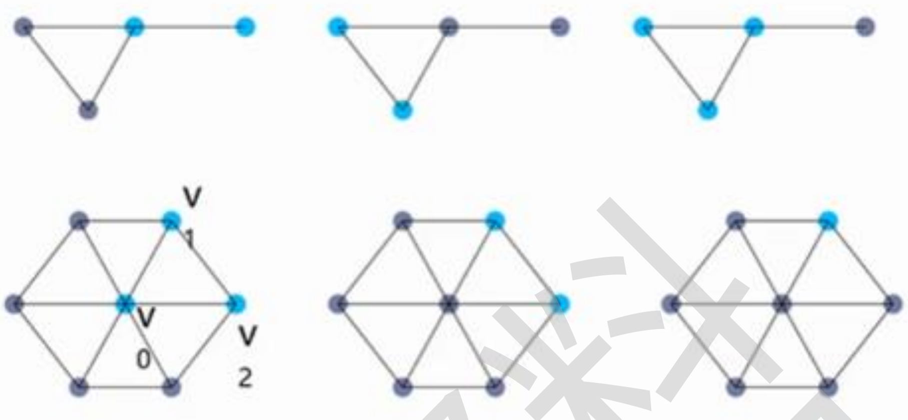


2. 团的概念，极大和最大团的概念

- 无向图$G=<V,E>$, $V^* \subseteq V$
- 团: $G[V^*]$是完全子图
- 极大团: V^*是团, 其真母集都不是
- 最大团: $|V^*|$最大的团
- 团数: $\nu_0(G)=|V^*|$, V^*是最大团

团(例)

- $\{v_0,v_1,v_2\}$, $\{v_1,v_2\}$, $\{v_1\}$, $\nu_0(G)=3$



定理4

- 定理4: 无向图G,

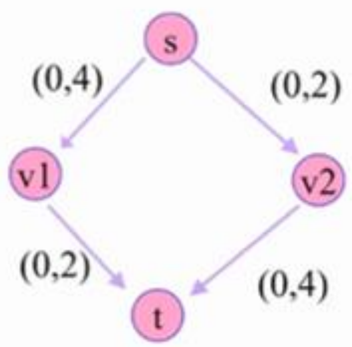
V*是G的团 ⇔ V*是G补的独立集.

- 推论: 无向图G, V*是G的极(最)大团⇔V*是G补的极(最)大独立集. $\nu_0(G)=\beta_0(G补)$.

3. 割集的概念，最大流最小割定理，构成最小割的边的求解

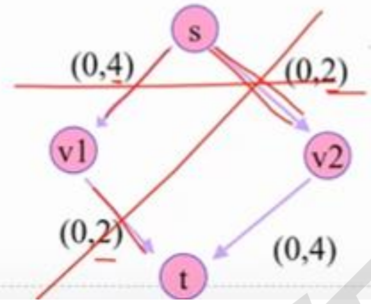
割集的定义

- 一个网络流中，一个割集(S,T)由两个点集S,T组成.
- S+T = V
- s 属于 S.
- t 属于 T.



最大流最小割定理

- 割集间的容量: $C(S,T)=\sum_{x\in S}\sum_{y\in T}c(x,y)$
- 即:所有由S中的一点指向T中的一点的边上的容量之和。
- **任一个网络D中从V_s到V_t的最大流的流量等于分离Vs和Vt的割集中容量最小的割集的容量。**



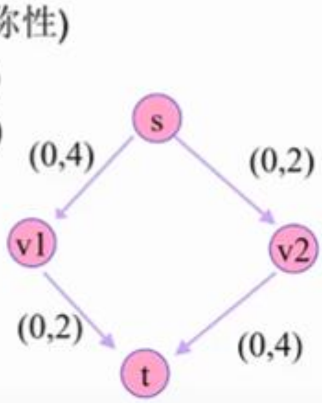
最大流最小割定理 – 证明

- **割集间的容量:** $c(S,T)=\sum_{x\in S}\sum_{y\in T}c(x,y)$
- 即:所有由S中的一点指向T中一点的边上的容量之和。
- **割集间的流量:** $f(S,T)=\sum_{x\in S}\sum_{y\in T}f(x,y)$
- 即:所有由S中的一点与T中的一点组成的边上的流量之和。（注意，与容量不同，流量既有正也有负。）

定理1：割流量的运算法则

X，Y，Z为顶点子集，且没有交集。

- $f(X,X)=0$ （由流量反对称性)
- $f(X,Y)=-f(Y,X)$ （由流量反对称性)
- $f(X\cup Y, Z)=f(X,Z)+f(Y,Z)$ （显然)
- $f(X, Y\cup Z)=f(X,Y)+f(X,Z)$ （显然)



定理2

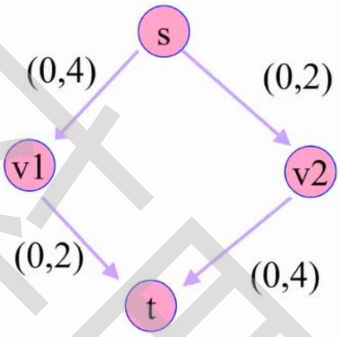
- 任意不包含s和t的顶点集X,与它相关联的边上的流量之和为0。

(因为，由流量平衡，对于X中每个结点x，与之相关联的边的流量之和为0)，如下式（V为所有结点的集合）：

- $$f(X,V) = \sum_{x\in X}[\sum_{v\in V} f(x,v)]$$

$$= \sum_{x\in X}[0] \quad \text{(由流量平衡)}$$

$$= 0$$



定理3

- 整个网络的流量等于任意割的流量
- 证明:
- $f(S,T) = f(S,V) - \cancel{f(S,S)}$ (由运算法则)

$V-S$ $= f(S,V)$ (由运算法则)

$= f(s,V) + \cancel{f(S-s,V)}$ (由运算法则) 0

$= f(s,V)$ (由定理2)

$= |f|$ (由|f|的定义)

- $f(X,X) = 0$ (由流量反对称性)
- $f(X,Y) = -f(Y,X)$ (由流量反对称性)
- $f(X \cup Y, Z) = f(X,Z) + f(Y,Z)$ (显然)
- $f(X, Y \cup Z) = f(X,Y) + f(X,Z)$ (显然)
- $$f(X,V) = \sum_{x\in X}[\sum_{v\in V} f(x,v)]$$

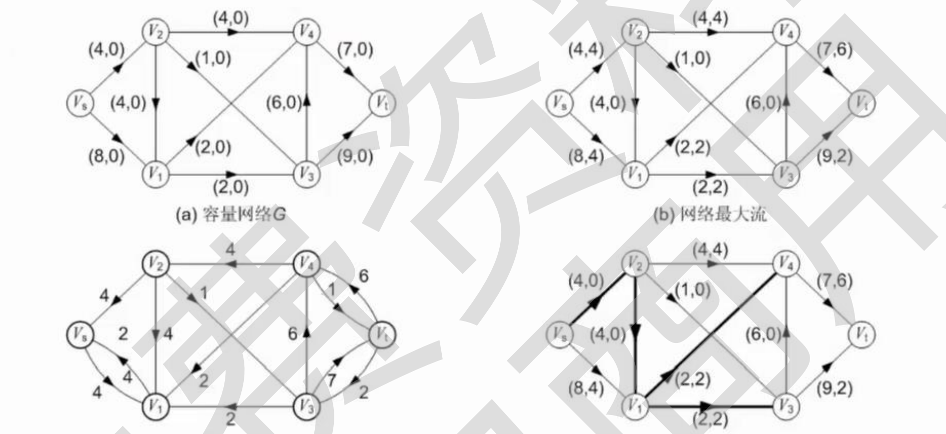
$$= \sum_{x\in X}[0] \quad \text{(由流量平衡)}$$

$$= 0$$

由于任意割流量相同且等于整个网络的流量，所以整个网络的流量就不可能超过容量最小的割的容量。
（最短板原理）

构成最小割的边的求解

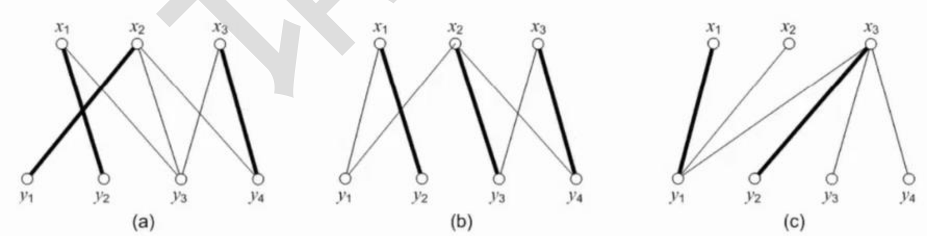
- 求最大流
- 求最大流的残量网络
- 在残量网络中，遍历所有可以由源点出发到达的节点，其集合为S，图中除S之外的其余节点为T，则（S，T为最小割）
- 所有连接S和T的边为最小割的边



(a) 容量网络G　(b) 网络最大流　(c) 残留网络G'　(d) 最小割

4. 利用网络流求二部图的最大匹配

二部图（*bipartite graph*）：设无向图为 $G(V, E)$，它的顶点集合 V 包含两个没有公共元素的子集：$X=\{x_1, x_2, \dots, x_s\}$和 $Y=\{y_1, y_2, \dots, y_t\}$，元素个数分别为 s 和 t；并且 x_i 与 x_j 之间$(1 \le i, j \le s)$、y_l 与 y_r 之间$(1 \le l, r \le t)$之间没有边连接，则称 G 为二部图，有的文献也称为**二分图**。



(a)　(b)　(c)

利用网络流求二部图最大匹配

(1) 从二部图 G 出发构造一个容量网络 G′，步骤如下：

a) 增加一个源点 S 和汇点 T；

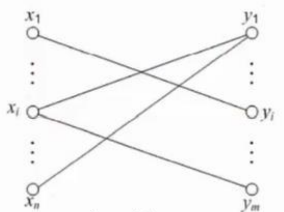
b) 从 S 向 X 的每一个顶点都画一条有向弧，从 Y 的每一个顶点都向 T 画一条有向弧；

c) 原来 G 中的边都改成有向弧，方向是从 X 的顶点指向 Y 的顶点；

d) 令所有弧的容量都等于 1。

(2) 求容量网络 G′ 的最大流 F。

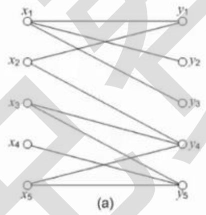
(3) 最大流 F 求解完毕后，从 X 的顶点指向 Y 的顶点的弧集合中，弧流量为 1 的弧对应二部图最大匹配中的边，最大流 F 的流量对应二部图的最大匹配的边数。



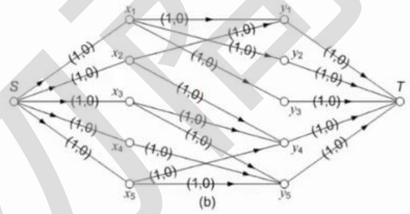
(a)

(b)

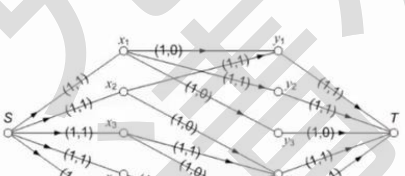
设有 5 位待业者，用 x1, x2, …, x5表示；另外有 5 项工作，用 y1, y2, …, y5表示；如果 xi能胜任yi工作，则在他们之间连一条边。现在要求设计一个就业方案，使尽量多的人能就业。



(a)



(b)



(c)

(d)