通信网理论基础

第五章 通信网的可靠性

第二节 通信网的可靠性

北京邮电大学 通信网络综合技术研究所
授课教师：武穆清
电子信箱：wumuqing@bupt.edu.cn

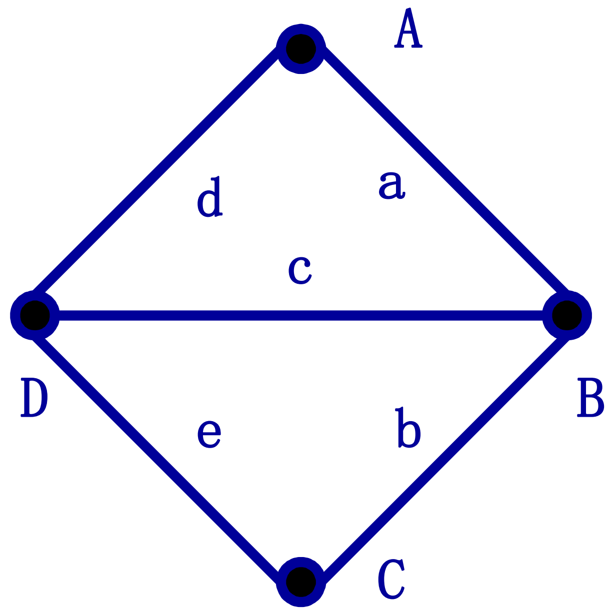
本章内容

5.2 通信网的可靠性

– 概述
  - 前面已经阐述了可靠性的基本概念、可靠度的计算方法和可靠性设计的大意
  - 本节要把这些理论与通信网的可靠性结合起来
  - 要分析通信网的可靠性，首先
    - 必须明确规定什么叫可靠，什么叫通信网的可靠
    - 用数学语言说，就是要定义可靠集和失效集
      - Δ 定义不同，结果也不同
    - 对于不同的问题，可靠和失效的定义可以是不同的
  - 对于通信网来说，可靠集大致有三种定义
    - 各有其特定的含义和特定的应用场合

- 着眼于全网的可靠集定义
  - 通信网可以用点和线组成的图来表示
    - 成网的必要条件是：形成一个联结图
    - 即任何两个端之间至少有一条径
      - 使它们之间可互送信息
  - 由于某些故障，图可能成为不联结的，而分成几个部分
    - 如由于边失效，或由于端失效，都可能造成不联结
    - 那么，从整体来说，网络已不能起到任何两端间均可通信的作用
    - 所以，就说此网络已失效
      - 当然，并不排除此时某些端之间还可通信
  - 在这种失效的概念下
    - 可靠集定义为：**U = {**任何未失效的端之间均有径**} (5-47)**
    - 失效集定义为：**V = {**某两个未失效的端之间无径**} (5-48)**

- 例
  - 如图所示的网络
    - 有四个端：A，B，C，D
    - 有五条边：a，b，c，d，e
  - 如果去掉图中的任一端或任一边
    - 则图仍然是联结的
    - 留下的端之间仍能通信
  - 如果边a和d同时失效
    - 则未失效的端A已无法与其它端通信
    - 就认为网络失效
  - 如果端A，B和C同时失效
    - 留下的只有一个端D，也无法与别的端通信
    - 也认为网络失效



≡所以，网络的可靠集为

$$U = \{\varphi;\ A;\ B;\ C;\ D;\ a;\ b;\ c;\ d;\ e;\ Ac;\ Ab;\ Ae;\ Ca;\ Cb;\ Cc;\ AC\}$$

Δφ：表示所有边和端均正常

- 即没有网络元素失效

ΔA：表示端A失效，这隐含边a和d也失效

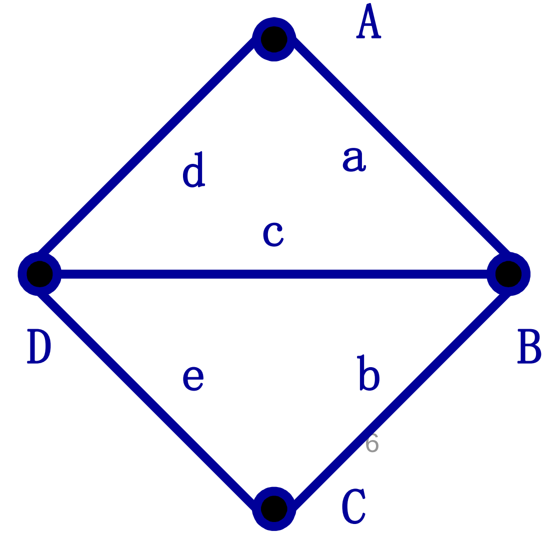
- 一个端失效，与之关联的边也不起作用

Δa：表示边a失效

ΔAc：表示端A和边c同时失效

- 当然也隐含边a和边d失效

ΔAC：表示端A和C同时失效



- 对于大型通信网来说
  - 尤其是等级制转接网
    - 如我国的五级电路交换电话网
  - 可考虑高层次的交换点之间能正常交换信息，就算网络正常
  - 当然，此时网络的结构和联结性都只考虑到相应的层次
  - 这对于大型网络来说，似乎更合理一些

- 着眼于网内某些端的可靠集定义
  - 如果只考虑网内比较重要的某些端点
    - 这些端点之间能相互通信，就算网络正常
  - 在这种意义下，可靠集定义为：
    - U = {某些端之间能正常通信}
    - 正常通信意味着
      - 有相互联通的径
      - 也还可以包括某些性能指标
      - 如电路转接中的呼损应小于规定值
        - 转接次数应小于某定值
        - 信息转接中的时延应小于规定值

- 着眼于随机图的可靠集定义
  - 当网的边数和端数都很大时
    - 用确定型的算法来计算可靠度几乎成为不可能
    - 因为不论可靠集还是不可靠集
    - 其元素数将按边和端的指数增加
  - 另一方面，当发生大灾害时（如战争，地震等）
    - 要达到前面两种定义的要求，是十分困难的
  - 随机图
    - 已给定端集，但端间的边是随机的
    - 即：代表网络的图并不是确定的
      - 任两端之间是否有边是以概率规定的
    - 当网络很大时，即边数和端数都很大时
      - 依这种概念计算的结果是有一定意义的
      - 至少是实际值的估值

- 随机图概念下的可靠集定义为
  - U = {尚联结的端数大于某规定值}
  - 这就是说，在大灾害情况下，网内仍有一定数量的交换节点之间能够通信
  - 它表征了抗灾害的能力
  - 这样定义下的可靠集越大，它们抗灾害的能力就越大
  - 在设计某些重要网络时，用它来规定参量是有价值的

- 着眼于其它方面的可靠集定义
  - 着眼于最短径的可靠集定义
    - 某些端最短径长大于规定值时，就认为网络失效
  - 着眼于其它指标
  - 我们从全网的观点出发

5.3 通信网的联结性

5.3.1 联结度和结合度

- 通信网可靠性最常用的定义——联结性定义
  - 人们经常根据图论中的联结性来研究可靠性
  - 联结性越好，（如全联结图）可靠度就越高
- 图的联结度α（connectivity）
  - 是去掉某些端使图成为不联结的度量
  - 定义： $\alpha = \min |X|$
    - X：是图的割端集
      - Δ 即去掉X中的端后，图就失去联结性，图被分为若干部分
    - 绝对值符号 |·|：代表这个集中的元素数
  - 所有割端集中，最小集的元素数称为**图的联结度**
    - 显然α越大，联结性越好

– 图的结合度β（cohesion）
  – 是去掉某些边使图成为不联结的度量
  – 定义： $\beta = \min |Y|$
    – Y：是图的割边集
      – 即去掉Y中的边后，图就失去联结性
      – 图就被分为若干部分
  – 即所有割边集中，最小集的元素数称为图的结合度
    – 同样β越大，联结性越好

– 混合联结度γ
  – 是去掉某些边和端使图成为不联结的度量
  – 定义： $\gamma = \min |Z|$
    – Z：是混合割集，Z中可以有边，也可以有端
      – 去掉Z中的边和端，就使图分成若干部分

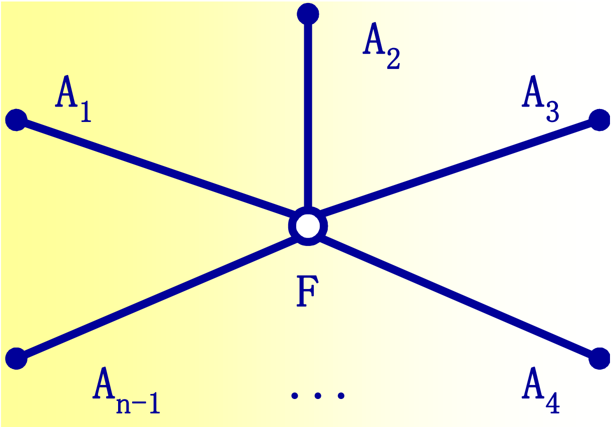
– 三个参数之间的关系

  = 对于无向图，这三个参量之间有如下关系：

$$\gamma = \alpha \le \beta \le \frac{2m}{n}$$

一些结论

- 图的联结性好或通信网可靠，常希望2m/n大一些
  - 这意味着：边越多，任两端之间的径就越多
- 边数最少的联结图是树
  - 此时，$m = n - 1$
  - 则：$2m/n = 2 - 2/n < 2$
  - $\alpha = \beta = \gamma = 1$
  - 这意味着：
    - 树中任两端之间只有一条径
    - 去掉一条边或一个端，必使图成为不联结的
  - 树具有一种星形的结构
    - 通常设有一个转接交换点（F）
    - 其度数为：$n - 1$
    - 其它端度数均为1
    - 这种结构的联结性最差
    - 可靠性也就最差
    - 要提高可靠性，就必须增大α和β，即增大边数m



- ≡ 2m/n 称为网的冗余度
  - Δ 它越大，径就越多，超过联结的最低要求就越大
- ≡ 在网络冗余度2m/n一定的情况下，提高α和β的方法：
  - Δ 这引出了一个网络结构的问题
  - Δ 冗余度一定，意味着网络代价（或成本）已定
    - 此时，α和β越大越好，网络可靠性越高
    - 也就是希望不等式 $\gamma = \alpha \le \beta \le \dfrac{2m}{n}$ 中的等式成立
    - 即α = β = 2m/n
  - Δ 只有各端的度数δ_i相等时，其平均值也会等于最小值
    - 才可能使β等于2m/n
    - 但这只是必要条件，并不是充分条件
    - 因为β可能小于 $\min\limits_i \delta_i$
  - Δ 这就需要去研究各端度数相等的网络

– 联结度α和结合度β的计算方法

= 直接计算法

≡ 先求｜X_{st}｜和｜Y_{st}｜，要遍历所有s和t

≡ 再求各｜X_{st}｜和｜Y_{st}｜的最小值，便得到α和β

≡ 求｜X_{st}｜的方法

Δ 先把图改成有向图

- 即每条无向边用两条方向相反的有向边来代替
- 再把每个端也分裂为出和入两个端，中间用有向边联结

Δ 对所有有向边，置其容量为∞，置入端和出端间的边容量为1

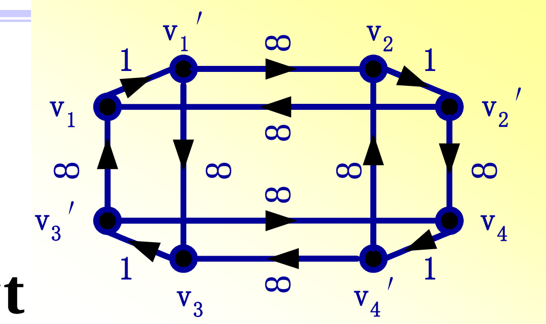
Δ 用M算法求最大流量 f_{st}

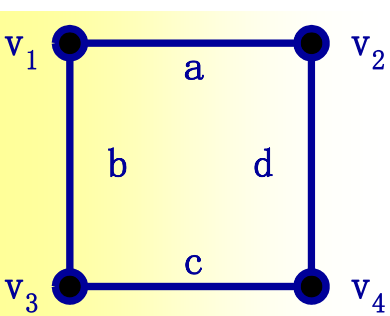
Δ 则 f_{st}就是｜X_{st}｜

- 因为根据最大流量最小割集定理
- 此f_{st}就是入端和出端之间的边作为最小割集的流量
- 因而就是这些端组成最小割端集
- 由于置端容量为1，所以：f_{st} = ｜X_{st}｜

Δ 若v_s和v_t之间有直通边，则f_{st} = ∞

- 这意味着，没有一个割端集能把v_s和v_t分割开



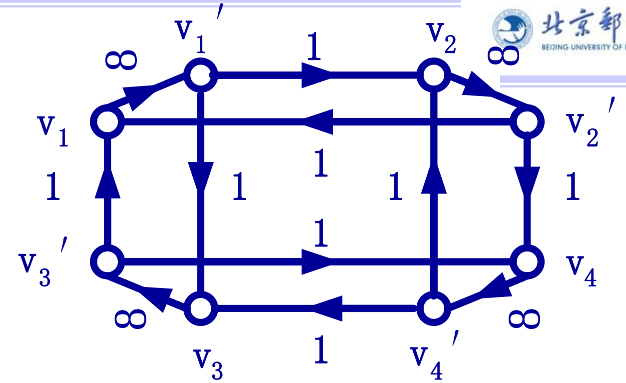


≡求｜Y_{st}｜的方法是一样的

Δ只是要把端容量置为∞

- 而把边容量置为1

Δ求得的最大流量就是｜Y_{st}｜

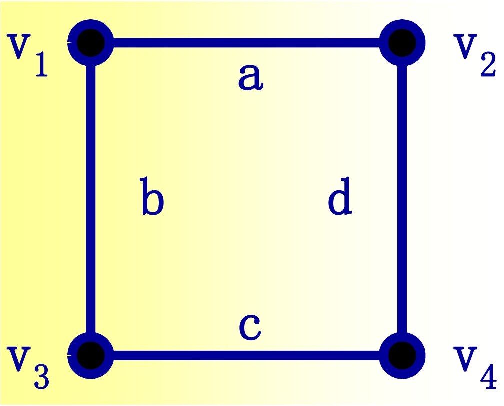


≡例：如图（a）所示的一个图

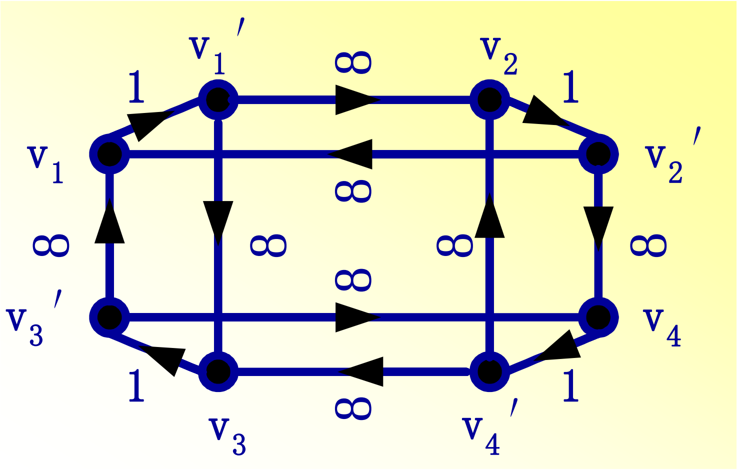
Δ共有四个端和四条边

Δ改画成有向图（b）

- 并置边容量为∞，端容量为1
- 每个端分裂为：入端v_i和出端v_i'



图（a）



图（b）

Δ可以求得：

- $|X_{12}| = |X_{13}| = |X_{24}| = |X_{34}|$

  $=\infty$

- $|X_{14}| = |X_{23}| = 2$

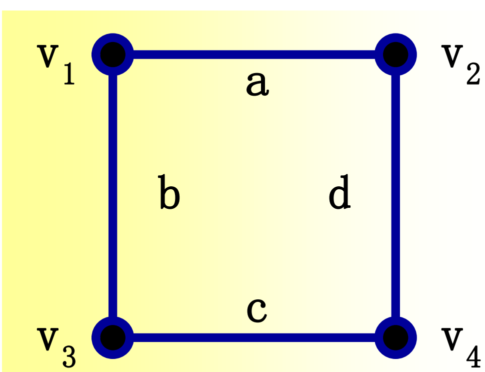
- $\alpha = 2$

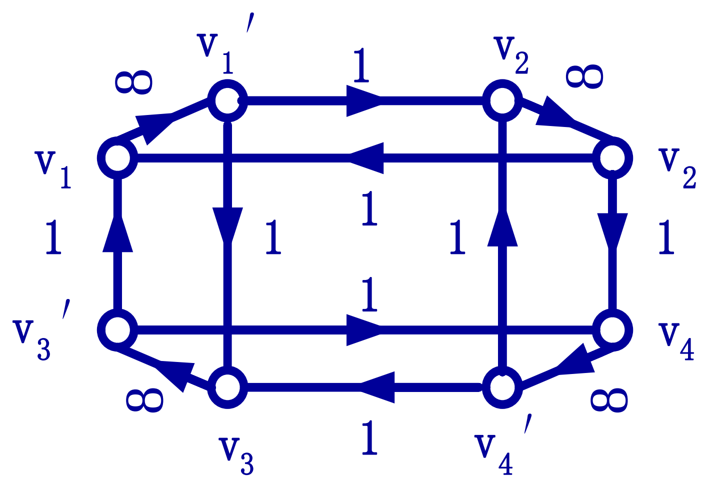
△ 改画成有向图（c）

- 并置边容量为1，端容量为∞
- 可以求得:
- $|Y_{12}| = |Y_{13}| = |Y_{14}| = |Y_{23}|$
  $= |X_{24}| = |X_{34}| = 2$
- $\beta = 2$

△ 这是直接计算，需要计算n^2次f_{st}

△ 重复计算较多，还可以简化一些





图（c）

5.3.2 局和线路分别故障下网的联结概率

- 概述
  - 通信网可以用端和边组成的图来表示
    - 图的联结性就是网的可靠性
    - 端是交换局，边是传输线
    - 两者的故障会破坏网的联结性
  - 如果这些故障的发生和排除都是独立的
    - 则可从这些局和线路的可靠度来计算网的联结概率
      - 即网的可靠度
    - 从原理上说，这种计算是直截了当的
      - 把各端和边都规定有正常和失效两种状态
      - 可得到一个2^{n+m}个元的状态空间
        - **n:** 端数
        - **m:** 边数

- Δ 把这些元分成两个集
  - 可靠集：图仍是联结的，即正常端之间仍是相互联通的
  - 失效集：图已成为不联结的几个部分，
    或只剩下一个正常端
- Δ 任选一个集来计算其概率，就得到网的可靠度或不可靠度

≡ 但是当m和n很大时

- Δ 这样的计算会变得很繁复，甚至不好计算

= 下面讨论在各边的可靠度相等、各端的可靠度也相等的情况下

≡ 通信网可靠度的近似计算公式

≡ 可作为实际网的可靠度的参考

– 只有边会出故障时的网络可靠度${}_eR$

= 设端总是正常的，边的不可靠度为F_e

= 网的可靠度为

$$ {}_eR = \sum_{i=n-1}^{m} A_i \left(1 - F_e\right)^i \cdot F_e^{\,m-i} \qquad (5-56) $$

≡ A_i：是覆盖n个端的具有 i 条边的联结子图的数目

Δ 由于最小联结子图是树，其边数是$n-1$

Δ 所以，$i \geq n-1$

Δ A_{n-1}：是主树的棵数

≡ 只要n个端都联结，网络就正常运行，网络就可靠

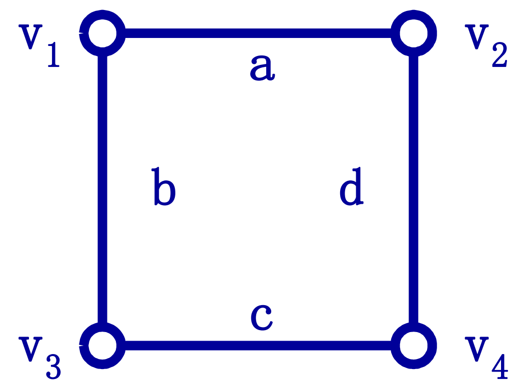
≡ 通常：$F_e \ll 1$

Δ 作近似计算时，可忽略F_e的高次幂

- 从网的不可靠度出发来求网的可靠度，更容易得到较简的近似式
  - $$_{e}R = 1 - \sum_{i=\beta}^{m} B_i \cdot F_e^{i} \cdot (1 - F_e)^{m-i} \qquad （5－57）$$
  - **B**$_i$：是具有 i 条边的割边集的数目
    - 由于结合度β是最小的割边集的元素数
    - 所以，$i \geq \beta$
  - 当割边集中的边均出故障时，网络就不联结了，认为网失效
  - 此式可近似为:
    - $$_{e}R \approx 1 - B_{\beta} \cdot F_e^{\beta} \qquad （5－58）$$
  - 即只取（5－57）式中的F_e最低幂项

例：计算网的可靠度$_eR$

- $n = m = 4$
- 边的不可靠度为F_e
- 解：



  - $A_3 = C_4^3 = 4$
  - $A_4 = C_4^4 = 1$
  - $B_2 = C_4^2 = 6$
  - $B_3 = C_4^3 = 4$
  - $B_4 = C_4^4 = 1$

$$_eR = \sum_{i=n-1}^{m} A_i (1 - F_e)^i \cdot F_e^{m-i}$$

  - 由（5－56）式可得：

$$
\begin{aligned}
_eR &= A_3(1-F_e)^3 \cdot F_e + A_4(1-F_e)^4 \\
&= 4(1-F_e)^3 \cdot F_e + (1-F_e)^4 \\
&= (1-F_e)^3 \cdot (1+3F_e) \\
&= 1 - 6F_e^{\,2} + 8F_e^{\,3} - 3F_e^{\,4}
\end{aligned}
$$

Δ 由（5－57）式也可得到此式 $$ {}_{e}R = 1 - \sum_{i=\beta}^{m} B_i \cdot F_e^{\,i} \cdot \left(1 - F_e\right)^{m-i} $$

$$ \begin{aligned} {}_{e}R &= 1 - B_2 F_e^{\,2}\left(1 - F_e\right)^{4-2} - B_3 F_e^{\,3}\left(1 - F_e\right)^{4-3} - B_4 F_e^{\,4}\left(1 - F_e\right)^{4-4} \\ &= 1 - 6F_e^{\,2}\left(1 - F_e\right)^2 - 4F_e^{\,3}\left(1 - F_e\right) - F_e^{\,4} \\ &= \left(1 - F_e\right)^3 \cdot \left(1 + 3F_e\right) \\ &= 1 - 6F_e^{\,2} + 8F_e^{\,3} - 3F_e^{\,4} \end{aligned} $$

Δ $B_2 = C_4^2 = 6$

Δ $B_3 = C_4^3 = 4$

Δ $B_4 = C_4^4 = 1$

Δ 由（5－58）式得: $$ {}_{e}R \approx 1 - 6 \cdot F_e^{\,2} $$

≡一般情况下，（5－58）式已足够精确

– 只有端会出故障时的网络可靠度 $_nR$

- 设边总是正常的，端的不可靠度都是 F_n
- 网的可靠度为：

$$_nR = 1 - \sum_{i=\alpha}^{n} C_i \cdot F_n^i \cdot (1 - F_n)^{n-i} \qquad (5-59)$$

  - C_i：是具有 i 个端的割端集的数目
  - α：是图的联结度
  - 所以：$i \geq \alpha$

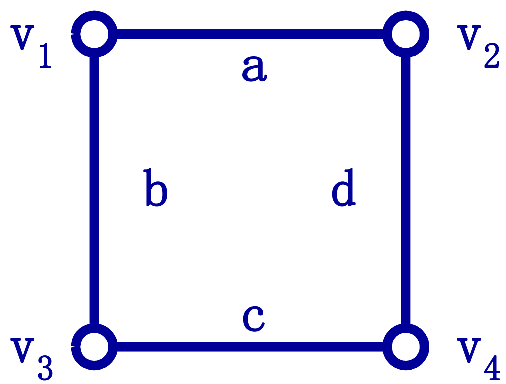
- 若 $F_n << 1$，则此式可近似为：

$$_nR \approx 1 - C_\alpha \cdot F_n^\alpha$$

- 例：计算网络可靠度 $_nR$
  - $C_2 = 2$ 即：$\{v_1, v_4\}$，$\{v_2, v_3\}$
  - $C_3 = C_4^3 = 4$
  - $C_4 = C_4^4 = 1$
  - $_nR = 1 - 2 \cdot F_n^2 \cdot (1 - F_n)^2 - 4 \cdot F_n^3 \cdot (1 - F_n) - F_n^4$

    $= 1 - 2 \cdot F_n^2 + F_n^4$

    $= (1 - F_n^2)^2$
  - $_nR \approx 1 - 2 \cdot F_n^2$



- 从近似式中可以看出
  - α和β越大，网络可靠度也越大
  - 因为网被断开的概率主要由 F_e^{β}或F_n^{α} 所决定
  - 当然要增大α和β，就要增加网的冗余度，这需要代价

- 边和端都可能出故障时的网络可靠度
  - 方法一：直接求解
    - 把各种状态列出来
    - 直接计算可靠度R
  - 方法二：利用$_eR$和$_nR$求解R的上界R_u和下界R_L
    - 只有边故障而使图不联结的概率是$1-{}_eR$
    - 只有端故障而使图不联结的概率是$1-{}_nR$
    - 实际不联结的概率必大于$1-{}_eR$或$1-{}_nR$
      - Δ 因为还有两种故障同时发生而使图不联结的情况
    - 所以，可靠度R将小于这两种情况的小者
    - 即可靠度的上界为：$R_u=\min({}_eR, {}_nR)$

≡ 另一方面，在边和端都可能出故障的情况下

Δ 图仍然联结的状态集可分为四类：

Δ 第一类是：边和端都无故障

- 这个集出现的概率是： $(1-F_n)^n(1-F_e)^m$

Δ 第二类是：边至少有一个故障，端无故障。这个集出现的概率是

$$ {}_eR(1-F_n)^n-(1-F_n)^n\cdot(1-F_e)^m=\left[{}_eR-(1-F_e)^m\right]\cdot(1-F_n)^n $$

- ${}_eR-(1-F_e)^m$：边至少有一个故障的概率
- 即：边可能故障下，网仍然联结的概率${}_e$**R**减去m条边均无故障的概率

Δ 第三类是：端至少有一个故障，边无故障。这个集出现的概率是

$$ {}_nR(1-F_e)^m-(1-F_n)^n\cdot(1-F_e)^m=\left[{}_nR-(1-F_n)^n\right]\cdot(1-F_e)^m $$

- ${}_nR-(1-F_n)^n$：端至少有一个故障的概率
- 即：端可能故障下，网仍然联结的概率${}_n$**R**减去n个端均无故障的概率

Δ 第四类是：端和边都有故障

- 这个集出现的概率必为非负

- 由于这四种集合是互斥的，所以图仍联结的概率应为这四种集的概率之和
  - Δ 所以，**可靠度R的下界**应为：

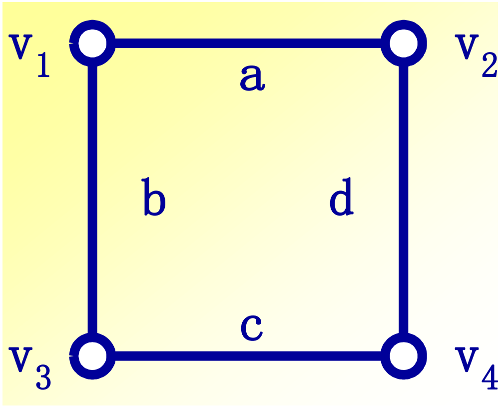
$$R_L = {}_eR(1-F_n)^n + {}_nR(1-F_e)^m - (1-F_e)^m \cdot (1-F_n)^n \qquad (5-61)$$

  - Δ 亦即： $\mathbf{R_L \le R \le R_u}$ （5－62）

例：计算两种故障都可能发生时网的可靠度

- 解：
  - Δ 先求上界，利用前例的结果有：

$$R_u = \min\left[(1-F_e)^3(1+3F_e)\ ;\ (1-F_n^2)^2\right]$$



Δ 再求下界，也需利用前例结果:

$$R_L = {}_eR(1-F_n)^n + {}_nR(1-F_e)^m - (1-F_e)^m \cdot (1-F_n)^n$$
$$= (1-F_e)^3 \cdot (1+3F_e) \cdot (1-F_n)^4 + (1-F_n^2)^2 \cdot (1-F_e)^4 - (1-F_e)^4 (1-F_n)^4$$
$$= (1-F_e)^3 \cdot (1-F_n)^2 \cdot \left[(1+F_n)^2 + F_e(3-F_n) \cdot (1-3F_n)\right]$$
$$= (1-F_e)^3 \cdot (1-F_n)^2 \cdot \left[1+2F_n+F_n^2+3F_e-10F_eF_n+3F_eF_n^2\right]$$

≡ 若 $F_n <<1$，$F_e <<1$，则上下界可近似为:

$$R_u = \min\left[(1-6F_e^2)\ ;\ (1-2F_n^2)\right]$$
$$R_L \approx 1-6F_e^2-2F_n^2-16F_eF_n$$

≡ 若 $F_n << F_e$ 或 $F_e << F_n$，则上下界将重合

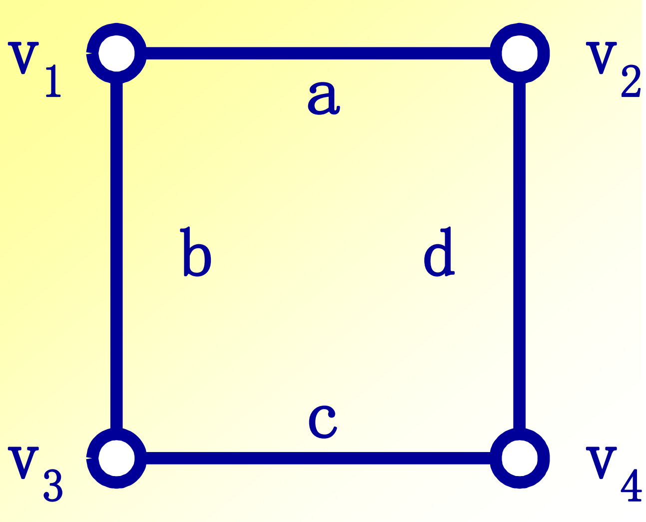
≡ 若 $F_e = F_n = F$，则上下界有一定差别:

$$R_u \approx 1-6F^2$$
$$R_L \approx 1-24F^2$$

≡精确解法

Δ 本题中m = 4，n = 4，共有$2^{m+n}=2^8=256$种状态

Δ 其中仍为联结图的情况及其概率如下:



- 四端四边均正常: $(1-F_n)^4(1-F_e)^4$
- 四端三边正常: $4(1-F_n)^4 F_e(1-F_e)^3$
- 三端四边正常: $4F_n(1-F_n)^3(1-F_e)^4$
- 三端三边正常: $8F_n(1-F_n)^3 F_e(1-F_e)^3$
- 三端二边正常: $4F_n(1-F_n)^3 F_e^2(1-F_e)^2$
- 二端四边正常: $4F_n^2(1-F_n)^2(1-F_e)^4$
- 二端三边正常: $12F_n^2(1-F_n)^2 F_e(1-F_e)^3$
- 二端二边正常: $12F_n^2(1-F_n)^2 F_e^2(1-F_e)^2$
- 二端一边正常: $4F_n^2(1-F_n)^2 F_e^3(1-F_e)$

Δ 把它们都加起来，就是网的可靠度：

$$R=(1-F_n)^2\cdot(1-F_e)\cdot\left[(1-F_e)^2(1+3F_e)+2F_n(1-F_e)\cdot\left(1-2F_e+3F_e^2\right)+F_n^2\left(1+5F_e-5F_e^2+3F_e^3\right)\right]$$

Δ 若F_n <<1，F_e <<1，可得近似式：

$$R\approx1-2F_n^2-6F_e^2-8F_eF_n$$

Δ 若F_n = F_e = F，则：

$$R\approx1-16F^2$$

- 它正好在上下界之间

≡通过本例可以看出

Δ 精确计算是比较困难的，当m和n很大时，几乎成为不可能

Δ 计算上下界比较简单，而且一般足够用来估计可靠度

5.3.3 可靠网的设计

- 引言
  - 设计一个可靠网，就是预先给定一个（按联结与否定义的）可靠度
    - 在满足这个可靠度的条件下
    - 去寻找最便宜的网络结构
    - 或最便宜地改造已有网的方案
  - 这是前面网络可靠度计算的逆问题
    - 我们要先找到网络所有可能的结构
    - 计算其可靠度
    - 其中肯定有满足要求的和不满足要求的
    - 在满足要求的结构中，找一个最便宜的

- 这种穷举法在理论上是可行的
  - 但是，当网络很大时，端点很多时，将成为不可实现的方法
  - 在计算技术中，称为**NP问题**
    - 即计算量将随端点数的指数增大
- 在实践中，一般求助于**经验解**，或称为**准最佳解**
  - 本节介绍两种方法
    - 保证网法
    - 多径网法

– 保证网法

= 保证网

≡ 定义：网内任何两端之间至少有两条无共边的径

≡ 在实际的通信中

Δ 端局的维护力量较集中，可靠度较高

Δ 传输线路则常是薄弱环节

Δ 所以，对传输线路要加以备份

= 保证网需要环形结构

≡ 保证网等效于结合度$\beta \geq 2$的网

≡ 所以，它需要有相当于环形的结构来支持

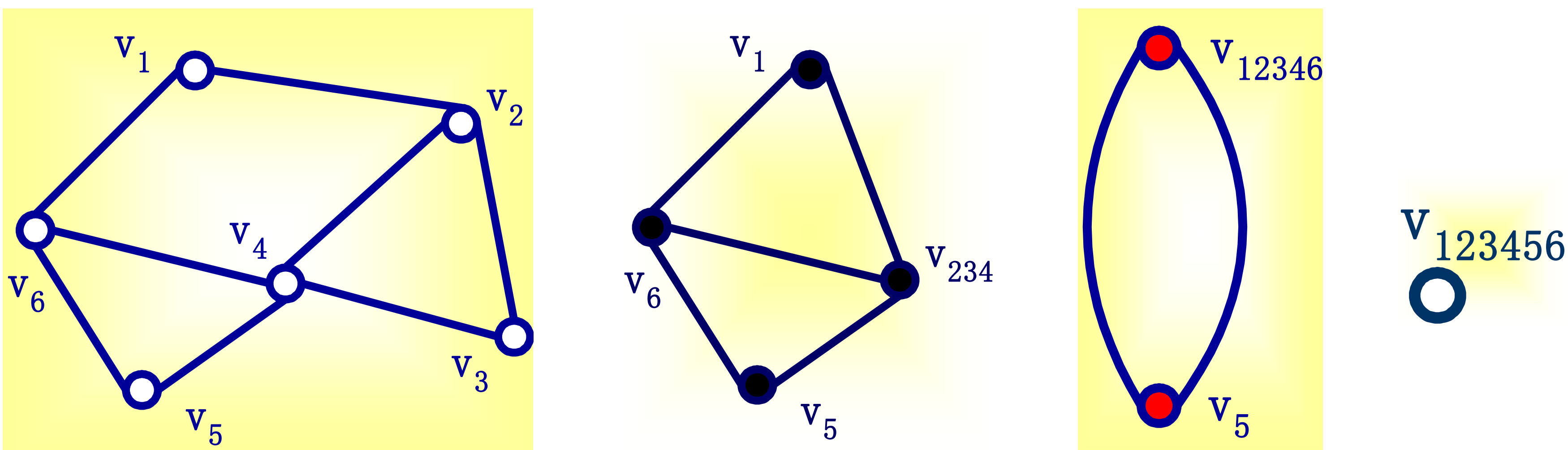
≡ 环形结构有很多种

Δ 但在价格便宜的条件下，可能的结构会少一些

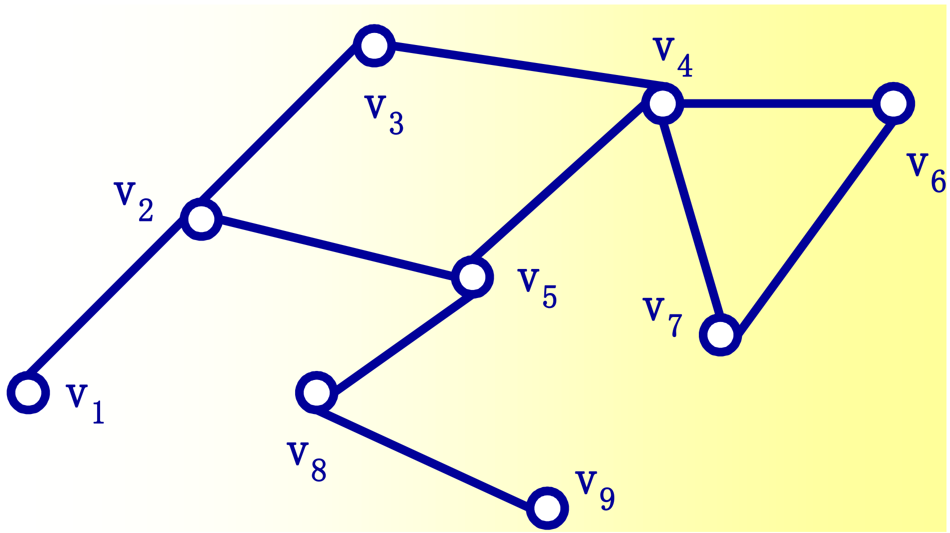
Δ 在限制转接次数的条件下，可能的结构会更少一些

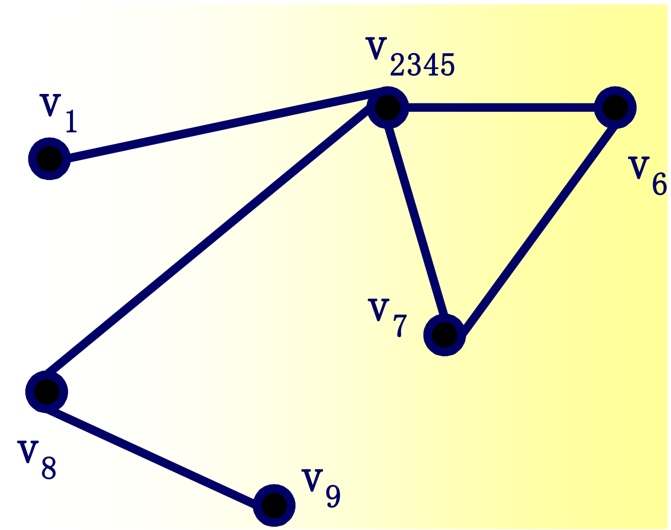
- 保证网的验证方法——退化法
  - 在图中任找一个环，把它收缩成一个端
  - 在得到的图中，再任找一个环，再收缩成一个端
  - 如此下去，直到没有环为止
  - 如果最后得到的图是
    - 一个单点图，则原来的图构成了保证网
    - 一棵树，则原来的图不是保证网

- 例1 如图所示网络
  - 先把2，3，4端缩成一点V_{234}
  - 再与1，6端缩成一点V_{12346}
  - 最后六个端均收缩成一点V_{123456}
  - 所以这种图可构成保证网



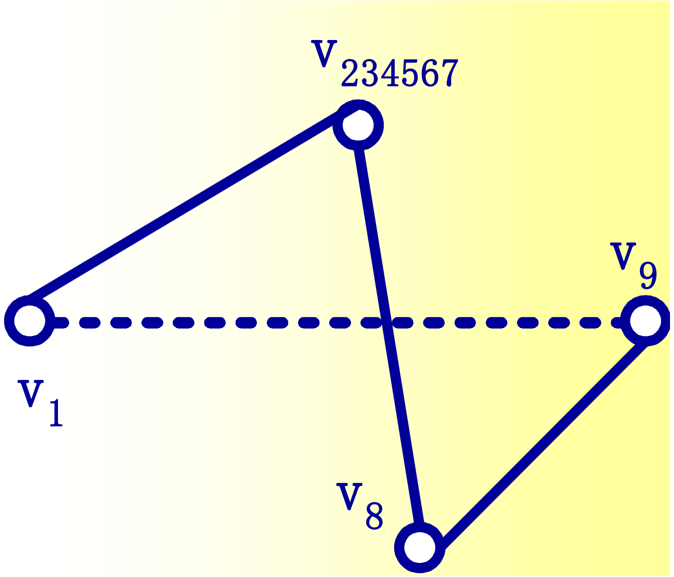
- 例2 如图所示网络
  - 先把2，3，4，5端缩成一点
  - 再与6，7端缩成一点
  - 得到一棵树，所以它不是保证网
  - 我们可以把它改造成一个保证网

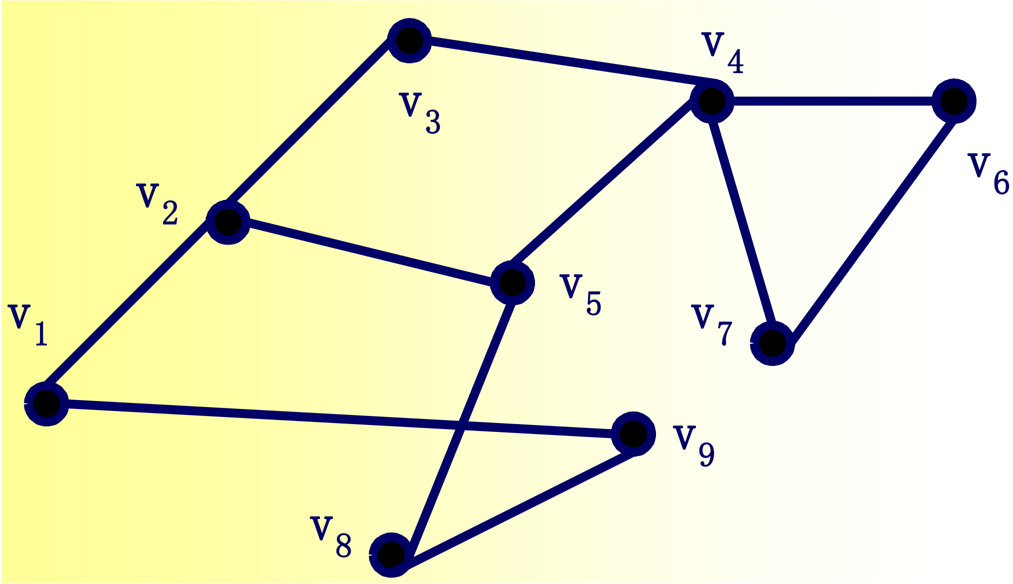




≡若想把它改造成一个保证网

- Δ 可以在树上加一些连枝
- Δ 如把1，9端连起来，就形成环
- Δ 再倒回去，即把缩合点恢复成环，便得到保证网
- Δ 如右下图





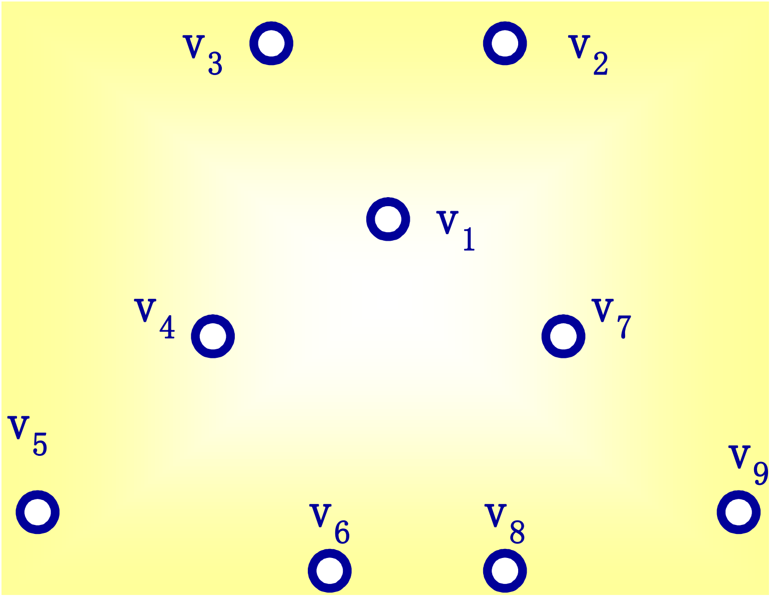
= 这两例说明了用退化法可以分析或改变已有网络，使之成为保证网

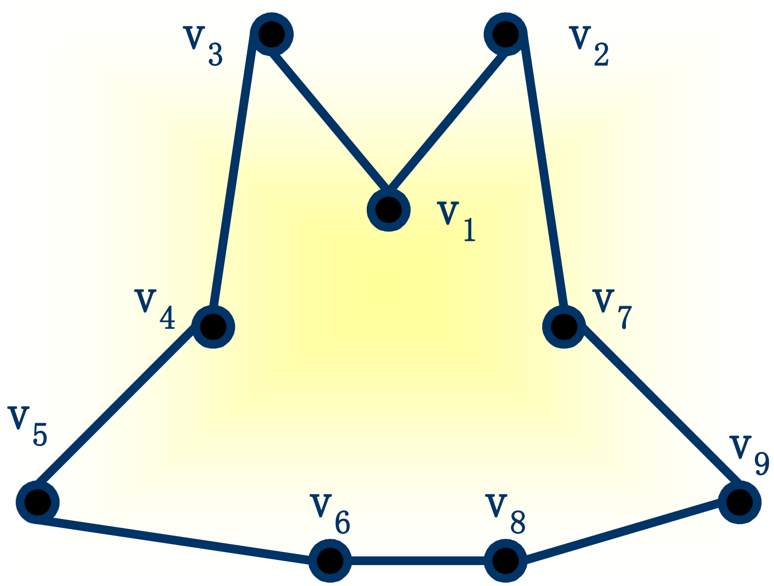
≡也可以用这种方法来设计一个待建网络

= 例3　有九个端，需要建成一个保证网

≡ 方法一：

- Δ 将九个端直接联成一个环
- Δ 如右下图
- Δ 此时转接次数较多





≡ 方法二:

△ 把九个端按位置分布情况分为三组

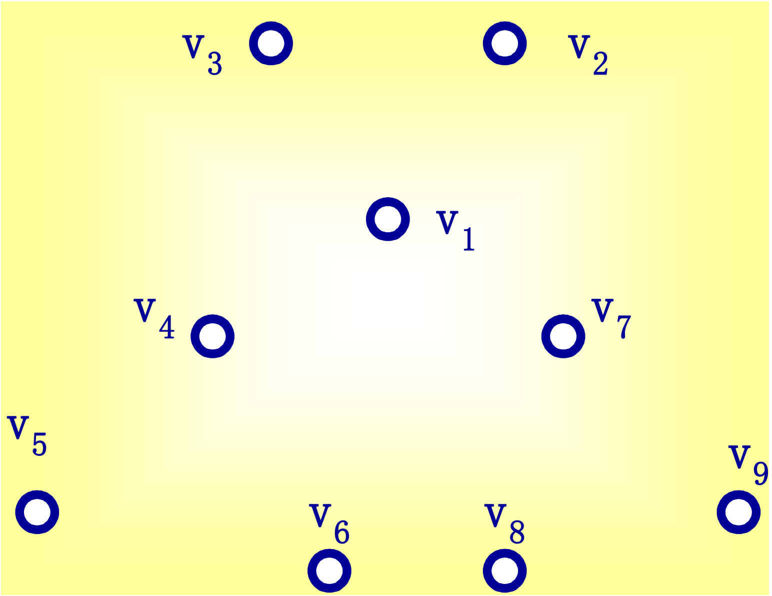
- 各组作为一个缩合点，即各自形成一个环

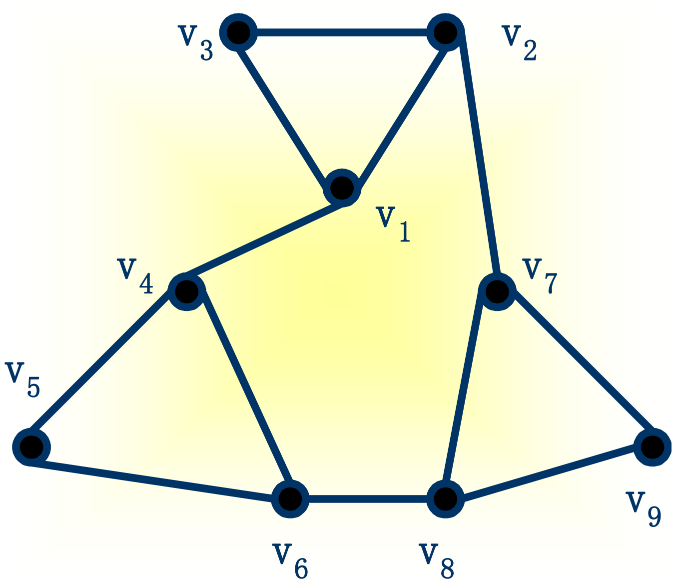
△ 如：端1，2，3为一组;

- 端4，5，6为一组;
- 端7，8，9为一组。

△ 然后把三组再形成一个环，如下图

- 这里的三条边是随意加的，只要能把三组联起来
- 如用了 v_6-v_8， v_1-v_4， v_2-v_7





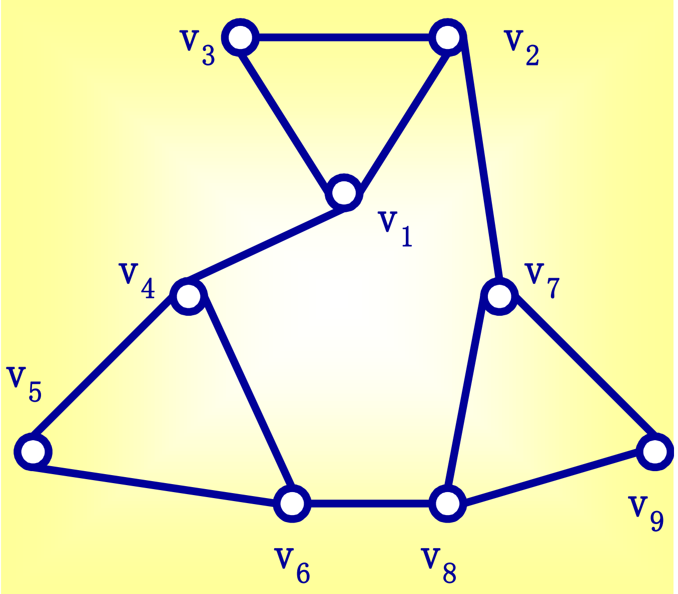
Δ 但是这样得到的保证网

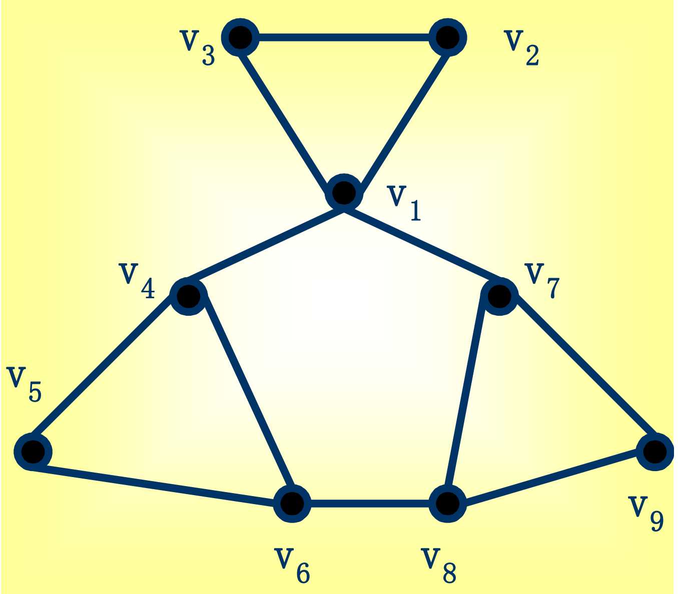
- 不一定能使任何两端间有两条以上的无共端径
- 即只能做到结合度 $\beta \geq 2$，不能保证联结度 $\alpha \geq 2$

Δ 若要求 $\alpha \geq 2$，则必须保证网中无割端

- 即去掉此端后，图将成为几个部分，不能有这样的端
- 这在加连枝时适当注意，就可做到这一点

Δ 如左下图中 $\alpha = 2$；　　而右下图中，v_1是割端，其 $\alpha = 1$





– 多径网法
  - 多径网
    - 在某些重要的网中，两条径对某些端之间仍显不够
      - 要按需为每个端分配不同的度数δ
    - 当所有端的最小度数δ_i（i = 1，2，…，n）已给定时
      - 可按下列方法来设计一个多径网
  - 多径网设计方法
    - 设已给定n个端
      - 各端的度数给定为：δ_i（i = 1，2，…，n）
      - 各端间的距离（或代价）为：r_{ij}（i, j = 1，2，…，n）
    - 任选一个度数最大的端，令为v_1
      - 即$\delta_1 \geq \delta_i$

- ≡ 寻找与v_1最近的端，令为v_2
  - Δ 即$r_{12} \le r_{1j}$
  - Δ 联结$v_1 - v_2$，令边为e_{12}
  - Δ 更新各端所要求的度数：$\delta_1 - 1$，$\delta_2 - 1$，δ_3，δ_4，…；δ_n
- ≡ 在新的度数中寻找一个度数最大的端，令为：v_3
- ≡ 寻找与它最近的端
  - Δ 联结构成边
  - Δ 更新各端所要求的度数
- ≡ 这样做下去
  - Δ 直到各端所需的度数均为零
  - Δ 所得到的图将满足要求

- 调整和比较
  - 按前述方法设计的网一般是代价较小的
    - 因为每次都选最短（代价最小）边
  - 但不能保证最好，往往需要调整
    - 比如可以换一个起始点
      - 因为一般给定度数相同的端都不止一个
      - 按上述步骤可以得到另外一个或几个图
    - 也可以在步骤进行中，换一个同等条件的其它端
      - 这样也可以得到多个图
  - 在多个图中作比较
    - 取代价最低的图
    - 通常可以得到准最佳的多径网

= 例：有七个端，如图

≡ 要求所有端的度数都大于等于3

Δ 各端间的距离已知，相当于图中的几何距离

≡ 选v_1作起始端　　　　端号：1 2 3 4 5 6 7

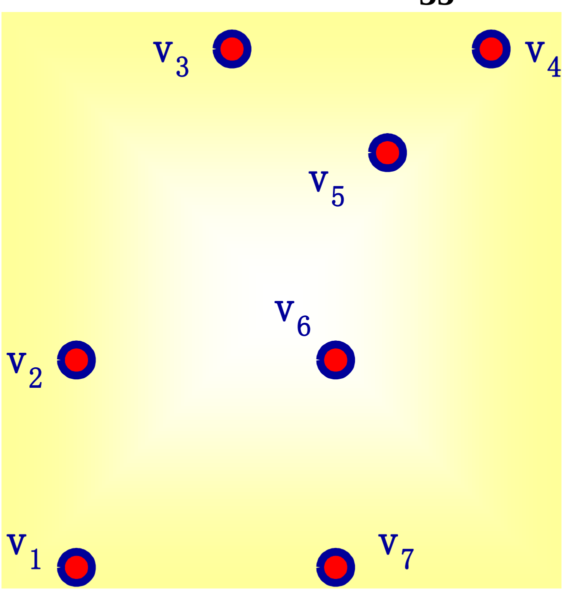
Δ 找其最近端v_2，连接v_1-v_2。原要求度数：3 3 3 3 3 3 3

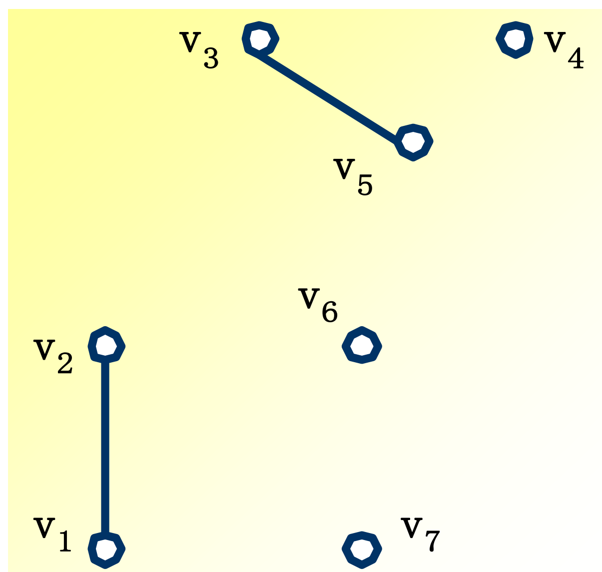
Δ 加e_{12}后，更新所要求度数：2 2 3 3 3 3 3

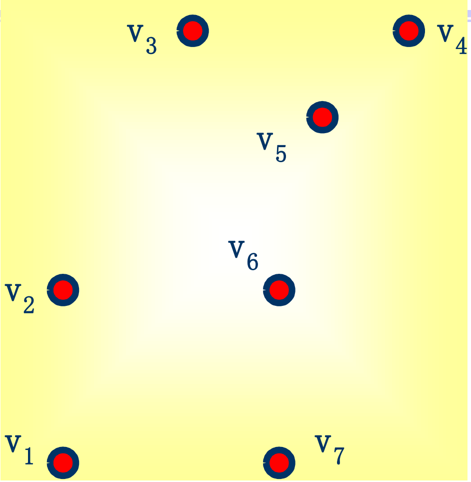
≡ 选v_3

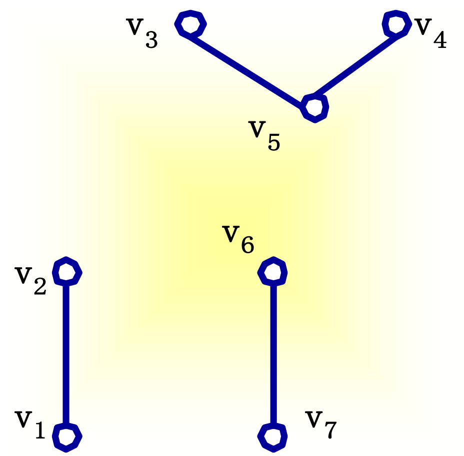
Δ 找v_5为其最近端，连接v_3-v_5

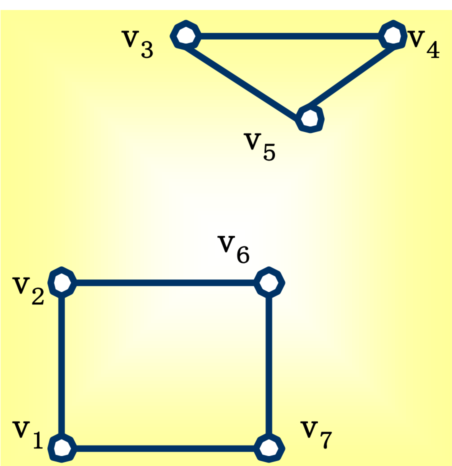
Δ 加e_{35}后，更新所要求度数：2 2 2 3 2 3 3











≡ **选v_4**

△ 找v_5为其最近端，连接$v_4 - v_5$

△ 加e_{45}后，更新所要求度数: 2 2 2 2 1 3 3

≡ **选v_7**

△ 找v_6为其最近端，连接$v_7 - v_6$

△ 加e_{67}后，更新所要求度数: 2 2 2 2 1 2 2

≡ **选v_1**

△ 找v_7为其最近端，连接$v_1 - v_7$

△ 加e_{17}后，更新所要求度数: 1 2 2 2 1 2 1

≡ **选v_2**

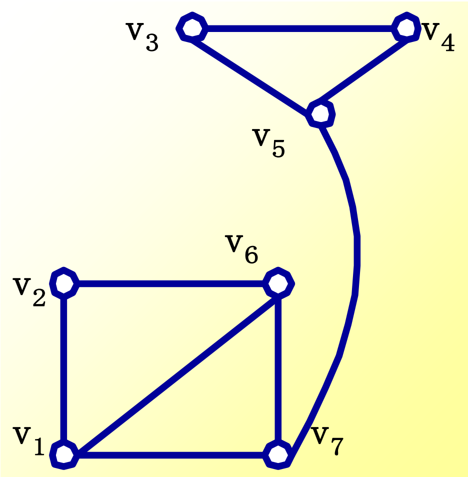
△ 找v_6为其最近端，连接$v_2 - v_6$

△ 加e_{26}后，更新所要求度数: 1 1 2 2 1 1 1

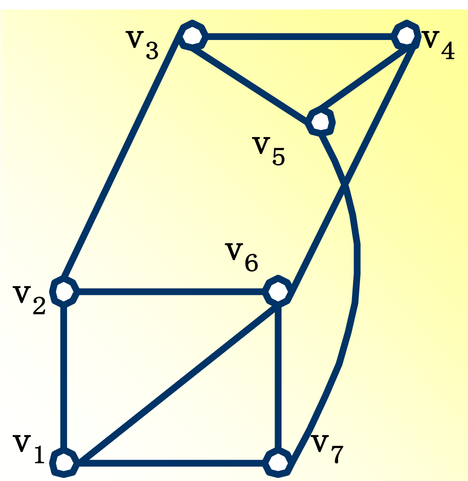
≡ **选v_3**

△ 找v_4为其最近端，连接$v_3 - v_4$

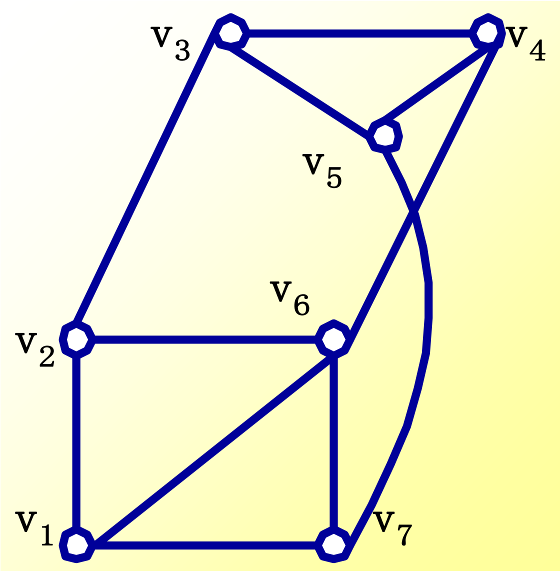
△ 加e_{34}后，更新所要求度数: 1 1 1 1 1 1 1



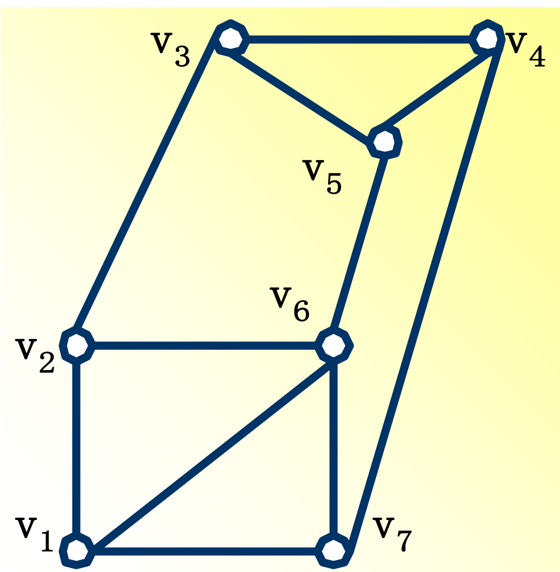
- 选v_1
  - 找v_6为其最近端，连接v_1-v_6
  - 加e_{16}后，更新所要求度数: 0 1 1 1 1 0 1
- 选v_5： 此时，距它最近的v_6的度数已经为0
  - 找v_7为其最近端，连接v_5-v_7
  - 加e_{57}后，更新所要求度数: 0 1 1 1 0 0 0
- 选v_2
  - 找v_3为其最近端
  - 加e_{23}后，更新所要求度数: 0 0 0 1 0 0 0
- 选v_4
  - 找v_6为其最近端
  - 加e_{46}后，更新所要求度数: 0 0 0 0 0 0 0



- 结果：如左图所示
  - v_6端的度数为4，其它各端的度数均为3
    - 可见满足了要求
  - 用上述方法求得的图，并不是最佳的



- ≡ 调整
  - Δ 把e_{46}和e_{57}换成e_{56}和e_{47}，如左下图所示
  - Δ 使总的边长还可略小一些
- ≡ 调整规则：成对置换
  - Δ 即当$r_{ij} + r_{kl} > r_{ik} + r_{jl}$时
  - Δ 就可把e_{ij}和e_{kl}换成e_{ik}和e_{jl}
  - Δ 这样置换不会影响各端的度数
    - 但边的代价减小了



- ≡ 其实，如果把选取端和边的顺序改为：
  - Δ v_1， e_{12}， v_3， e_{35}， v_4， e_{45}， v_6， e_{65}， v_1， e_{17}， v_2， e_{26}， v_3， e_{34}， v_7， e_{76}， v_4， e_{47}， v_1， e_{16}
  - Δ 这仍是按最大度数和最小距离的原则选取的
  - Δ 但所得结果已不是左上图，而是左下图了
  - Δ 只要多做几次选择，就能得到最佳解
  - Δ 左下图已是本题的最佳解了

- 作业:
  - 周先生教材
  - 第336页：　5.8

  - 张琳教材
  - 第214页：　6.9

(本节结束)