# 6 中国邮递员问题和旅行商问题

程龚 (gcheng@nju.edu.cn)

# 本节课的主要内容

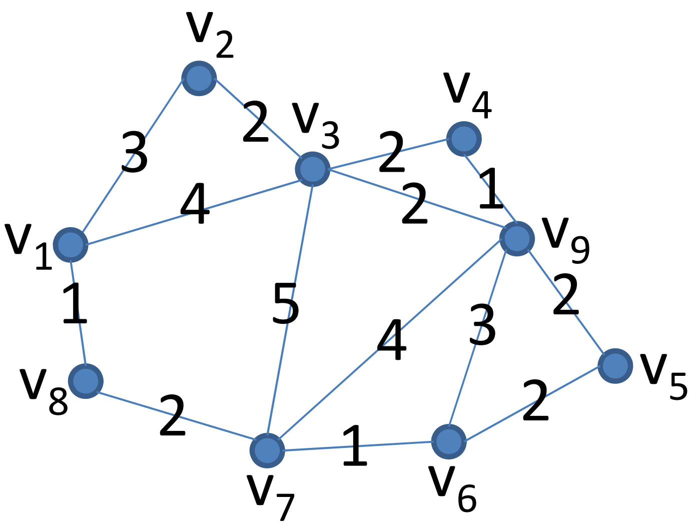
4.2 中国邮递员问题

4.4 旅行商问题

宁海路派出所
剑阁路
南阴阳营
汉口西路
南秀大厦
北东瓜市
宁海路
南东瓜市
城中派出所
苏宁环球套房饭店
五环大厦
南京无为高尔夫球俱乐部
百步坡1号民国建筑
五台山体育场
拉萨路
振道路拳道
丰汇大厦
永庆巷
星月大厦
南京大学(西门)
南京大学幼儿园
平仓巷
陶谷新村
江苏省档案馆
随园大厦
上海路
海苑大厦
广州路
青岛路
南京汉口路小学
南京儿童医院
鼓楼区一中心小学
豆菜桥
华侨路
春风大厦
平家巷
华荣大厦
管家桥
图书馆
榴园
南京大学
金陵大学堂
南京大学图书馆
南京大学(南门)
汉口路
中山楼
南京大学软件学院
南京大学
天津路
新纪元大酒店
小粉桥
君临国际广场
中山大厦宾馆
干河沿前街
金陵中学
南京规划局
管家桥
中央路
中山路
双龙巷
尖角营
南京第九中学
大石桥街
汇杰广场
四方商务
吉兆营
地铁大厦
同仁西街
同仁街
金鹰珠江壹号国际酒店
新月大厦
汇文里
人民中学
韩家巷
估衣廊
江苏省实验小学
南京财政局
长江路
国信大酒店
苏苑大厦
虹飞大厦
丹凤街
北门桥
北门桥路
东南大学东区
斯霞纪念馆
大石桥街
新安里
进香河世纪缘大酒店
卫巷
居安里
新街口派出所街责任区
佳汇大厦
珠江路
新世界百货
珠江路
鸡鹅巷
新宁商务
石婆婆庵
洪武北


# 中国邮递员问题

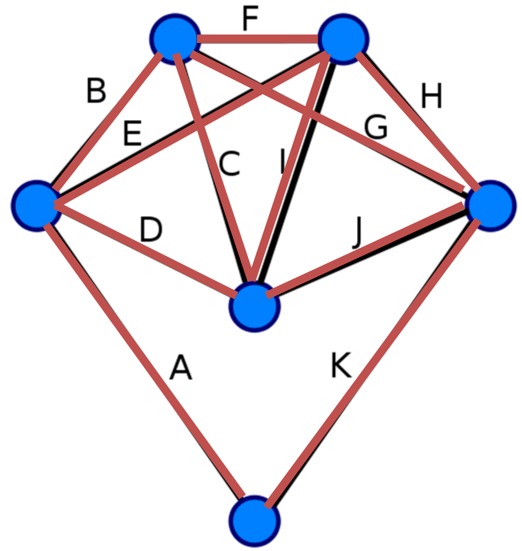
- 求赋权连通图中含所有边且权和最小的闭途径（称作最优邮路）。



# Euler图 (复习)

- Euler迹：经过每条边恰好一次的迹。
- Euler闭迹：经过每条边恰好一次的闭迹。
- Euler图：有Euler闭迹的图。

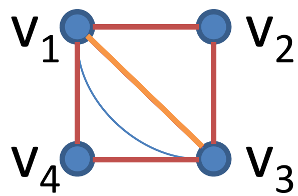
- Euler图的最优邮路
  - Euler闭迹




http://upload.wikimedia.org/wikipedia/commons/thumb/7/72/Labelled_Eulergraph.svg/500px-Labelled_Eulergraph.svg.png

# 中国邮递员问题的转换

- 非Euler图的最优邮路
  - 必然要重复经过一些边。
  - 将重复走过的边作为重边添加到图中 ⇒ 原图中的最优邮路等价于新图中的Euler闭迹
- 中国邮递员问题等价于
  1. 添加重边成为Euler图（如果本来不是的话）。
  2. 使添加的边权和最小。
  3. 找Euler闭迹。



# 1. 添加重边成为Euler图

- Euler图的充要条件：一个非空连通图是Euler图当且仅当它没有奇度顶点。

- ⇒ 添加重边的目标
  - 消除奇度顶点。

# 1. 添加重边成为Euler图 (续)

- Euler图的充要条件：一个非空连通图是Euler图当且仅当它没有奇度顶点。

证明：⇒

Euler图G有Euler闭迹C ⇒ ∀v∈V(G)

- C经过v关联的每条边恰一次。
- C每次经过v，都要使用2条边。

⇒ d(v)是偶数

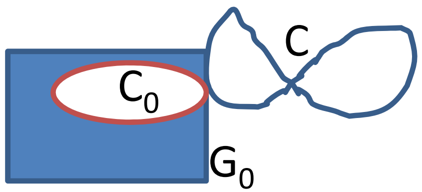
# 1. 添加重边成为Euler图 (续)

- Euler图的充要条件：一个非空连通图是Euler图当且仅当它没有奇度顶点。

证明：$\Leftarrow$

反证法：

1. 令S={非空、无奇度顶点的、连通的非Euler图}，则$S\neq\emptyset$。
2. 取S中边数最少的一个图，称作G $\Rightarrow \delta(G)\geq 2 \Rightarrow$ G有圈 $\Rightarrow$ G有闭迹
3. G中最长的闭迹称作C $\Rightarrow$ C不是Euler闭迹 $\Rightarrow$ C不经过G的至少一条边 $\Rightarrow$ G-E(C)有一个非空连通分支$G_0$且$\varepsilon(G_0)<\varepsilon(G)$
4. C是闭迹 $\Rightarrow$ G-E(C)后顶点度的奇偶性不变 $\Rightarrow$ $G_0$无奇度顶点
5. 假设$G_0\in S \Rightarrow \varepsilon(G)\leq\varepsilon(G_0) \Rightarrow$ 与$\varepsilon(G_0)<\varepsilon(G)$矛盾 $\Rightarrow G_0\notin S \Rightarrow G_0$是Euler图 $\Rightarrow G_0$有Euler闭迹$C_0$
6. G是连通图 $\Rightarrow$ C经过$G_0$的至少一个顶点 $\Rightarrow$ C与$C_0$有公共顶点无公共边 $\Rightarrow$ C与$C_0$可构成G中的闭迹且其边数>$\varepsilon(C)$ $\Rightarrow$ C不是G中最长的闭迹 $\Rightarrow$ 矛盾



# 2. 使添加的边权和最小

- 定理4.2.2 设G是赋权连通图，G中有2k个奇度顶点。G*是G的最优邮路对应的Euler图，令E’=E(G*)\E(G)。则H=G[E’]是以G的奇度顶点为起点和终点的k条无公共边的最短路之并。

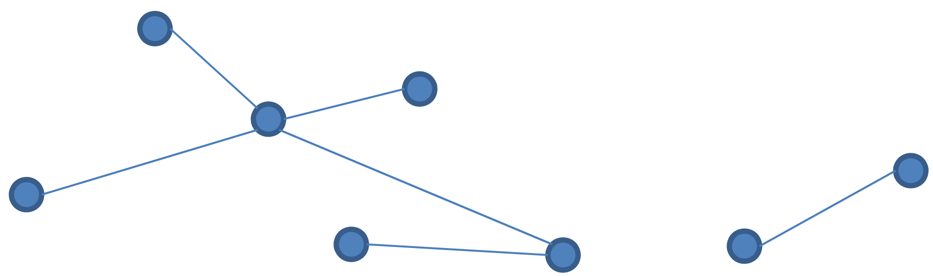
- ⇒ 重边的添法
  - 连接k对奇度顶点的k条无公共边的最短路。
  - 且边权和最小。

# 2. 使添加的边权和最小 (续)

- 定理4.2.2 设G是赋权连通图，G中有2k个奇度顶点。G*是G的最优邮路对应的Euler图，令E'=E(G*)\E(G)。则H=G[E']是以G的奇度顶点为起点和终点的k条无公共边的最短路之并。

证明：

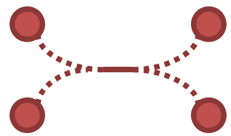
1. G*是Euler图 ⇒ G*无奇度顶点 ⇒ E'中关联到G的每个奇/偶度顶点的边有奇/偶数条 ⇒ G中的奇度顶点在H中仍是奇度顶点，G中的偶度顶点在H中仍是偶度顶点或不出现 ⇒ H仍有2k个奇度顶点
2. 假设H中有圈C ⇒ C在G*中 ⇒ G*中删去C后无奇度顶点 ⇒ 是Euler图且边权和比G*小 ⇒ G*不是最优邮路对应的Euler图 ⇒ 矛盾 ⇒ H中没有圈
3. 从H的任一个奇度顶点出发沿未走过的边前行直到无法继续，形成迹$P_1$ ⇒
   - $P_1$的终点是奇度顶点。
   - $P_1$是路。
   - $P_1$是最短路：否则 ⇒ 在G*中用更短的路替换P1 ⇒ 得到边权和更小的Euler图 ⇒ G*不是最优邮路对应的Euler图 ⇒ 矛盾
4. 从H去掉$P_1$，剩余2k-2个奇度顶点，同理可找到$P_2$、$P_2$、……、$P_k$。
5. 剩余图中无圈且无奇度顶点 ⇒ 无边 ⇒ H是以G的奇度顶点为起点和终点的k条无公共边的最短路之并

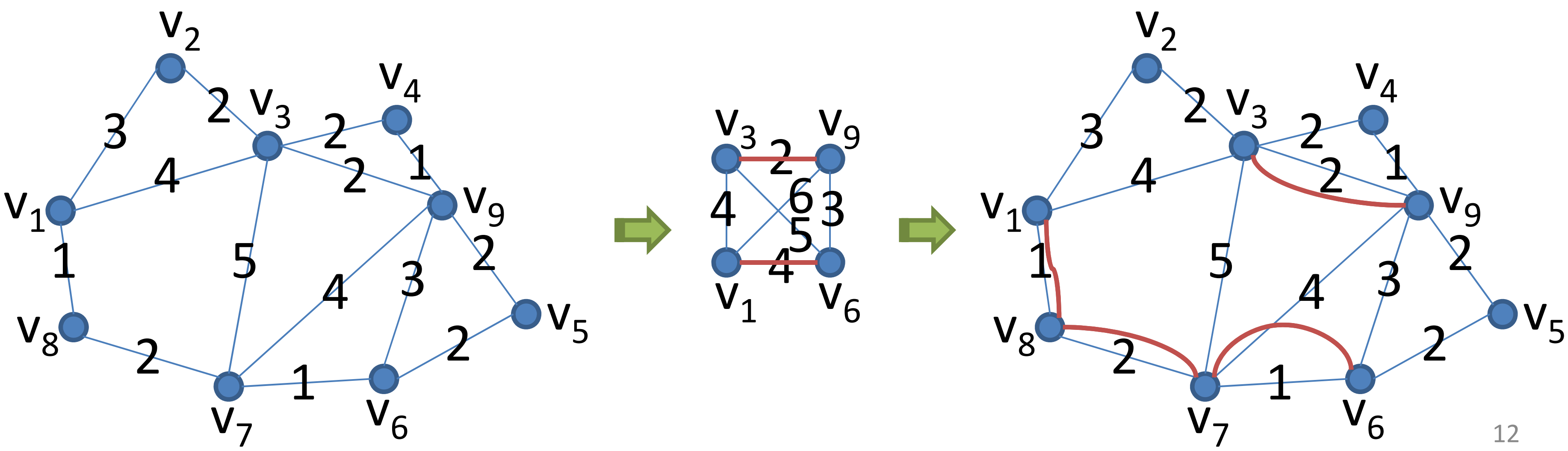


# 2. 使添加的边权和最小 (续)

- Edmonds-Johnson算法
  1. 找到所有2k个奇度顶点间的最短路。
  2. 构造一个完全图$K_{2k}$：顶点为2k个奇度顶点，边权为最短路的边权和。
  3. 找$K_{2k}$的最小权完美匹配M。
  4. 沿M对应的最短路添加重边。

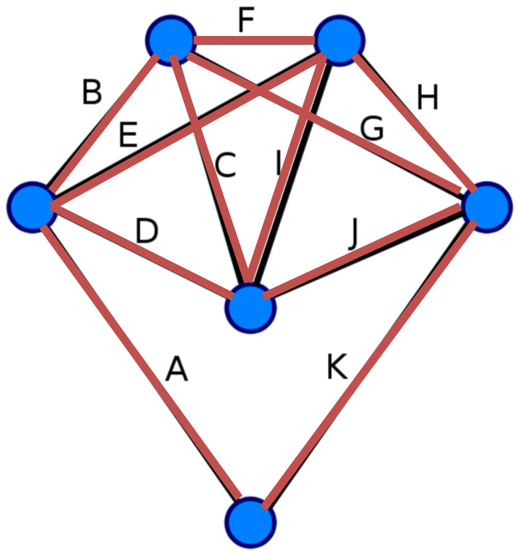
为什么这k条路无公共边？
删去公共边后，可得到权更小的完美匹配 ⇒ 矛盾





# 3. 找Euler闭迹

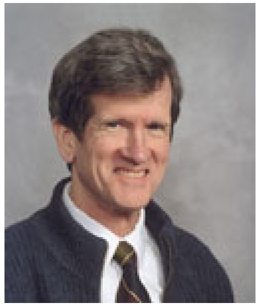
- Fleury算法
  - 基本思路：尽可能沿剩余图的非割边前行。



# Edmonds-Johnson算法的时间复杂度

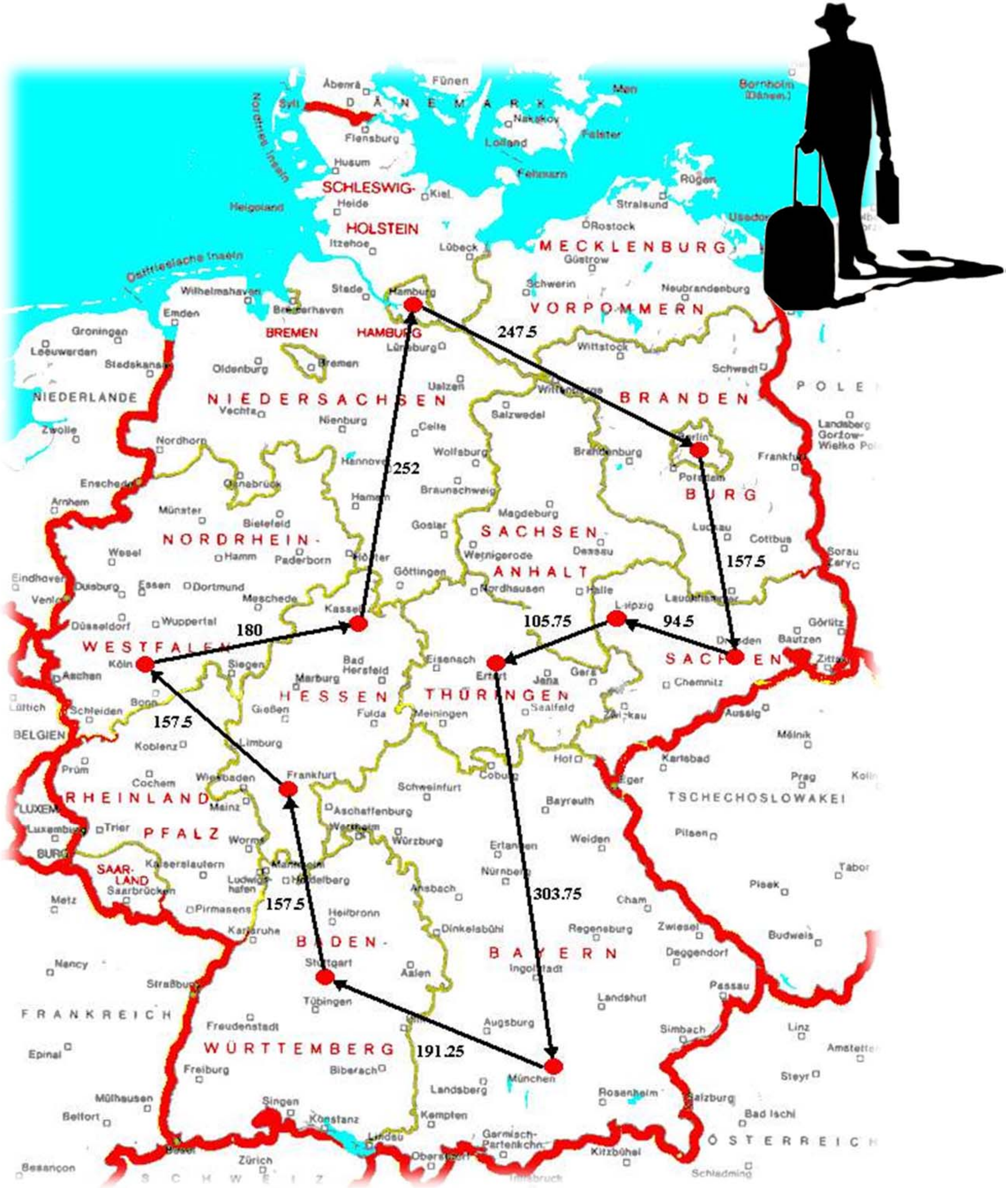
1. 找两两最短路
   – Floyd–Warshall算法：$O(v^3)$
2. 构造完全图：$O(v^2)$
3. 找最小权完美匹配
   – Edmonds算法：$O(v^3)$
   – 另有 $O\left(\sqrt{v}\varepsilon\right)$ 算法
4. 添加重边：$O(\varepsilon)$
5. 找Euler闭迹
   – Fleury算法：$O(\varepsilon^2)$
   – 另有$O(\varepsilon)$算法

Jack R. Edmonds, 加拿大, 1934--



Ellis L. Johnson, 美国, 1937--

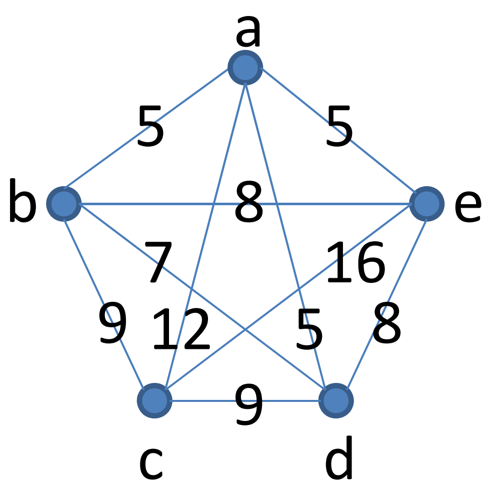
http://www.iasi.cnr.it/jack/edmonds_big.jpg
http://www.research.gatech.edu/sites/research.gatech.edu/files/styles/79_109/public/Ellis%20L%20Johnson.png?itok=pUxz6qFU

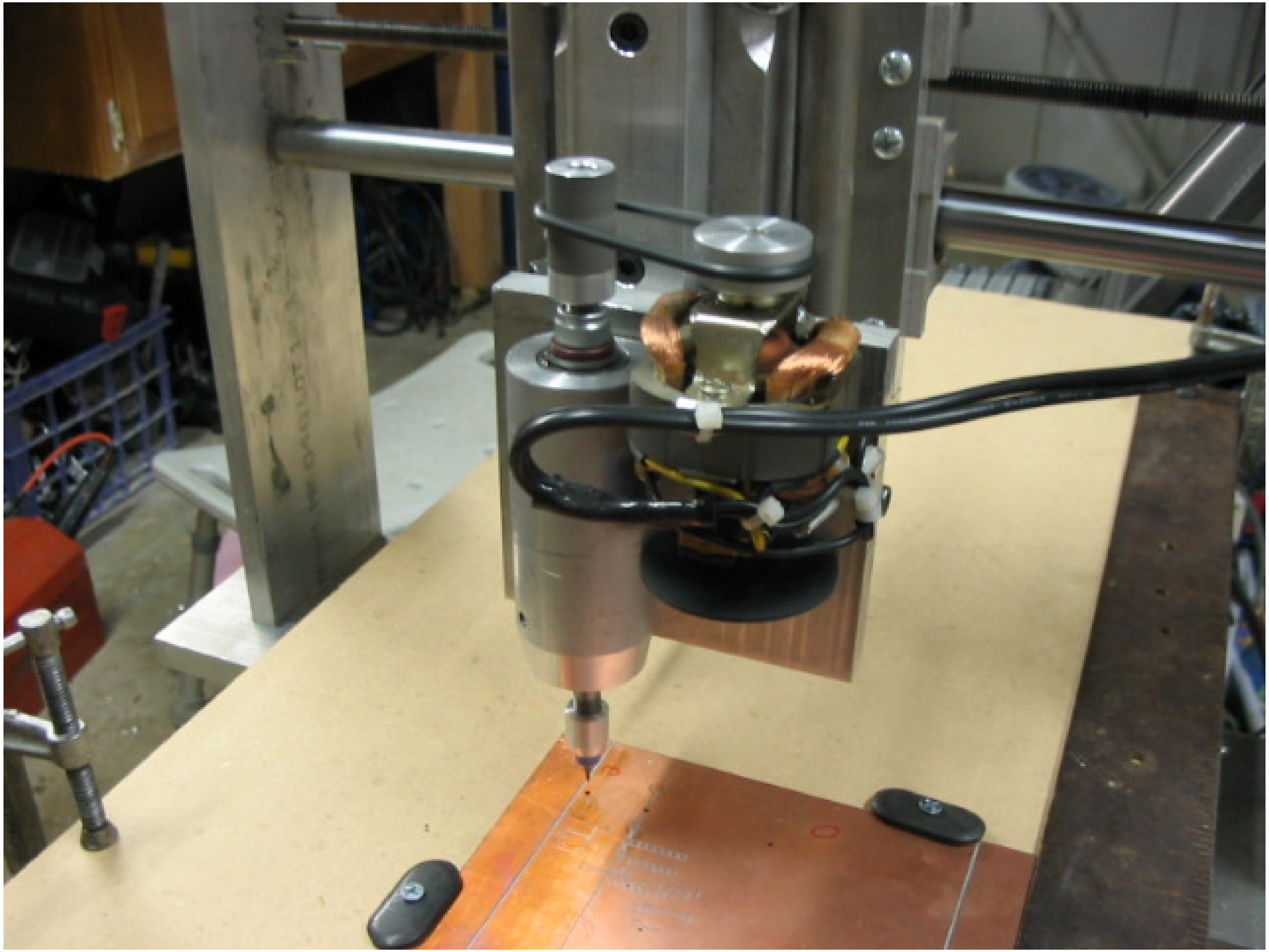



http://blog.electricbricks.com/wp-content/uploads/TSP.jpg

# 旅行商问题

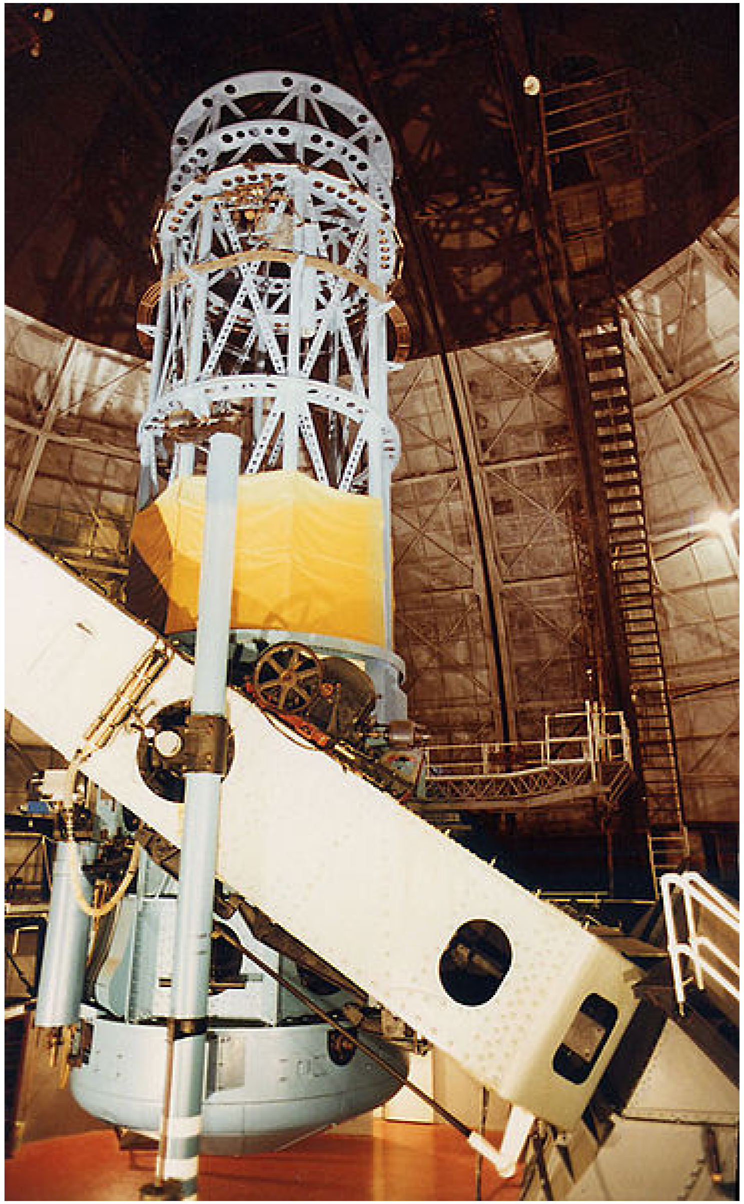
- 求赋权连通图中经过每个顶点恰一次且权和最小的圈。
  - 通常假设：$\forall v_i, v_j, v_k \in V(G)$, $w(v_i, v_j)+w(v_j, v_k) \geq w(v_i, v_k)$
  - 通常只讨论边权为正数的完全图$K_n$（缺失的边可以赋权∞）







http://www.buildyouridea.com/cnc/pcb_drill/pcb_drill_4.jpg



http://en.wikipedia.org/wiki/File:100inchHooker.jpg
http://en.wikipedia.org/wiki/File:Constellation_Fornax,_EXtreme_Deep_Field.jpg

# Hamilton图 (复习)

- Hamilton路：经过每个顶点恰一次的路。
- Hamilton圈：经过每个顶点恰一次的圈。
- Hamilton图：有Hamilton圈的图。

# 旅行商问题的难度

- 找Hamilton圈： NP-complete
- 找权和最小的Hamilton圈： NP-hard

- 因此，通常采用近似算法
  - 高效地找出较优解

# 旅行商问题的近似算法

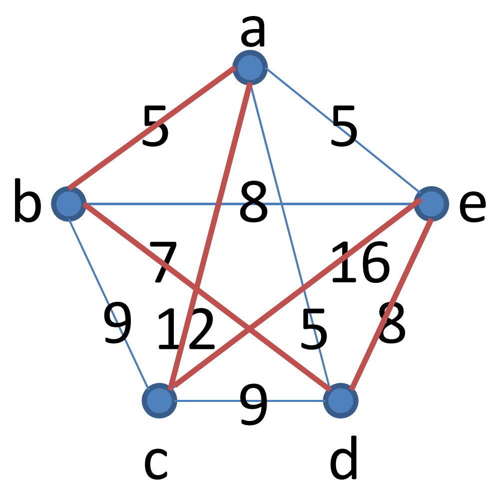
- 邻近点法
- 最小生成树法
- 最小权匹配法
- Kernighan-Lin

# 邻近点法

- 基本思路
  - 总是贪心地选择最近的未访问邻点前行。

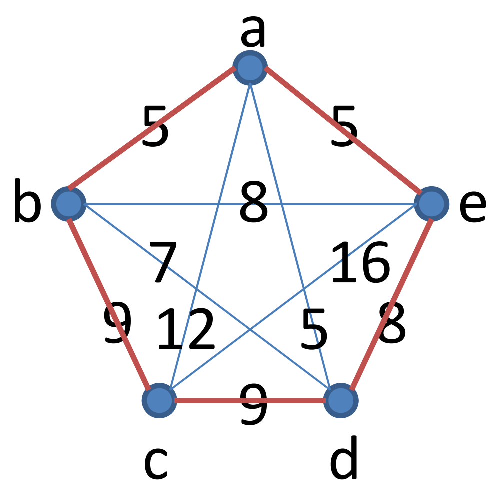
# 邻近点法 (续)

- 举例（从a出发）
  - 5+7+8+16+12=48



# 邻近点法 (续)

- 举例（从b出发）
  - 5+5+8+9+9=36



# 邻近点法 (续)

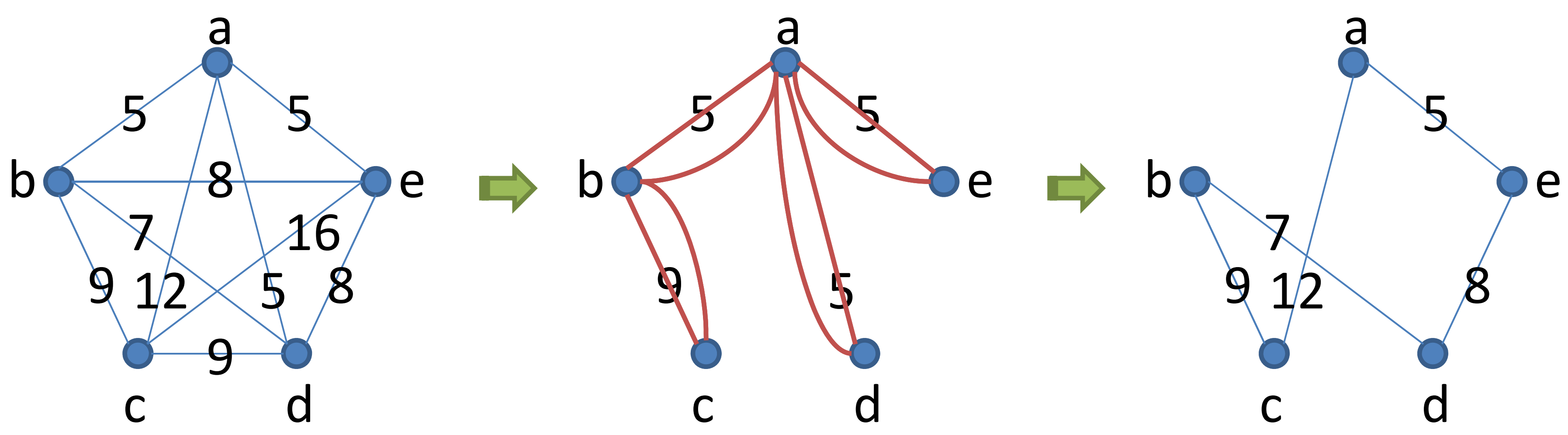
- 近似比w(H)/w(H*) $\leq \frac{1}{2}(\lceil \log_2 \nu \rceil)$
- 最终结果与以下因素有关
  - 初始点
  - 邻点

# 邻近点法 (续)

- 时间复杂度：$O(v^2)$

# 最小生成树法

1. 找$K_n$的一棵最小生成树T。
2. 为T中的每条边添加重边成为T*。
3. 找T*的一条Euler闭迹C。
4. 沿C前行，跳过已访问过的顶点，直至访问完所有顶点。



# 最小生成树法 (续)

- 近似比$w(H)/w(H^*)<2$

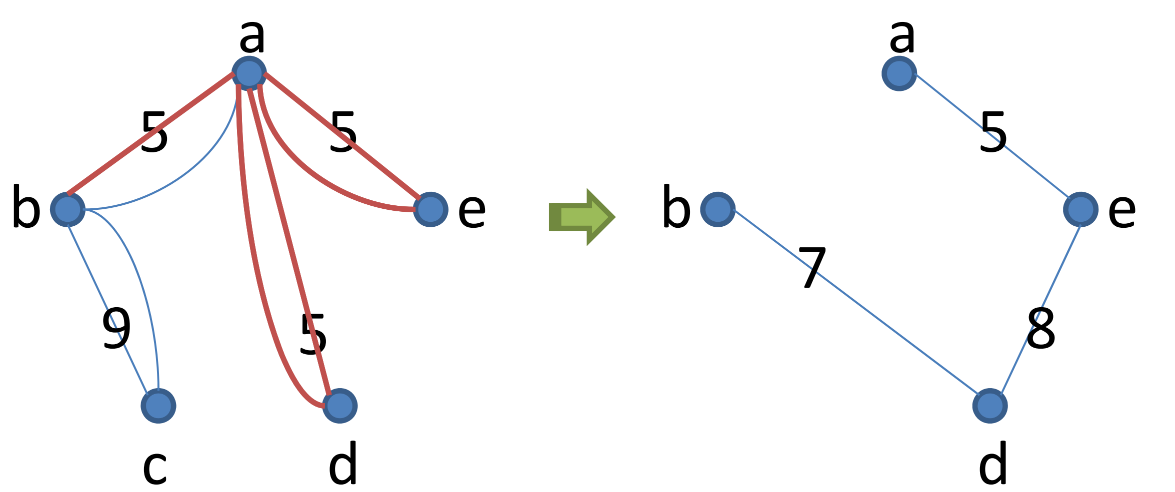
- 最终结果与以下因素有关
  - 最小生成树
  - 闭迹
  - 闭迹的初始点

# 最小生成树法 (续)

- 近似比w(H)/w(H*)<2

证明：

1. 三角不等式 ⇒ w(H)≤w(C)=w(T*)=2w(T)
2. H*比生成树多一条边 ⇒ w(H*)>w(T)
3. w(H)/w(H*)<2

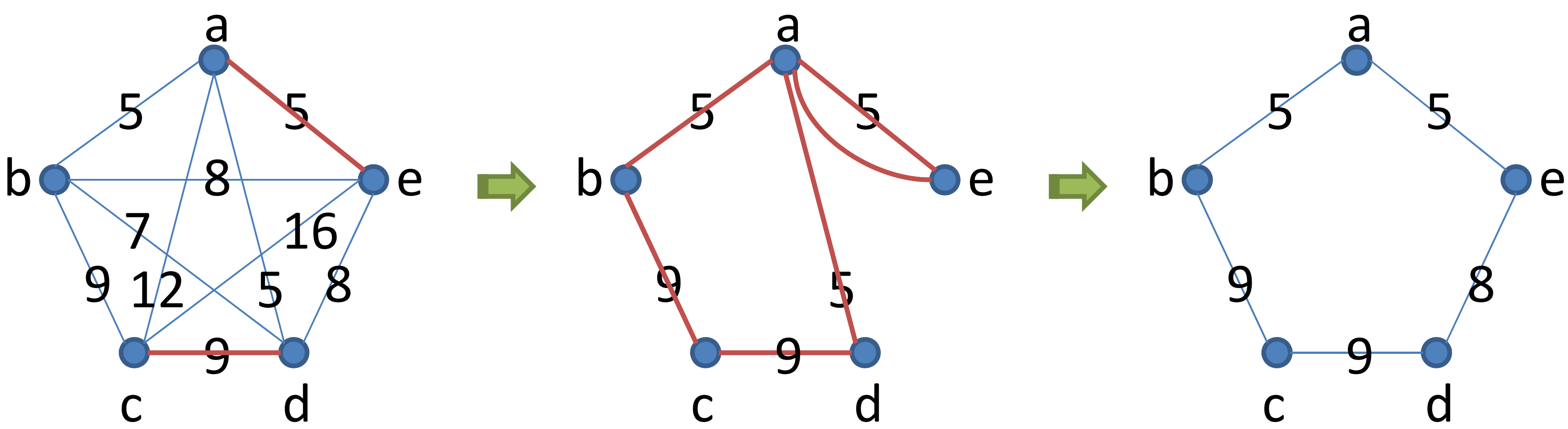


# 最小生成树法 (续)

- 时间复杂度
  1. 找最小生成树
     - Prim算法：O(ε+vlogv)
  2. 添加重边：O(v) //生成树的边数为v-1
  3. 找Euler闭迹
     - Fleury算法：$O(v^2)$
     - 另有O(v)算法
  4. 沿Euler闭迹前行：O(v)

# 最小权匹配法

1. 找$K_n$的一棵最小生成树T。
2. 找T中奇度顶点在$K_n$中导出子图G′的最小权完美匹配M。
3. 将M添加到T中成为T*。
4. 找T*的一条Euler闭迹C。
5. 沿C前行，跳过已访问过的顶点，直至访问完所有顶点。



# 最小权匹配法 (续)

- 近似比$w(H)/w(H^*)<3/2$

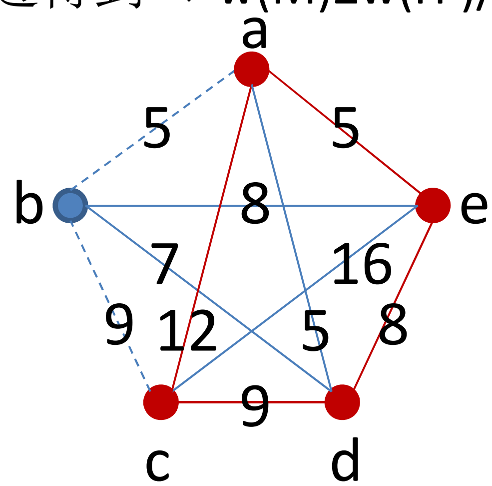
- 最终结果与以下因素有关
  - 最小生成树
  - 最小权完美匹配
  - 闭迹
  - 闭迹的初始点

# 最小权匹配法 (续)

- 近似比w(H)/w(H*)<3/2

证明：

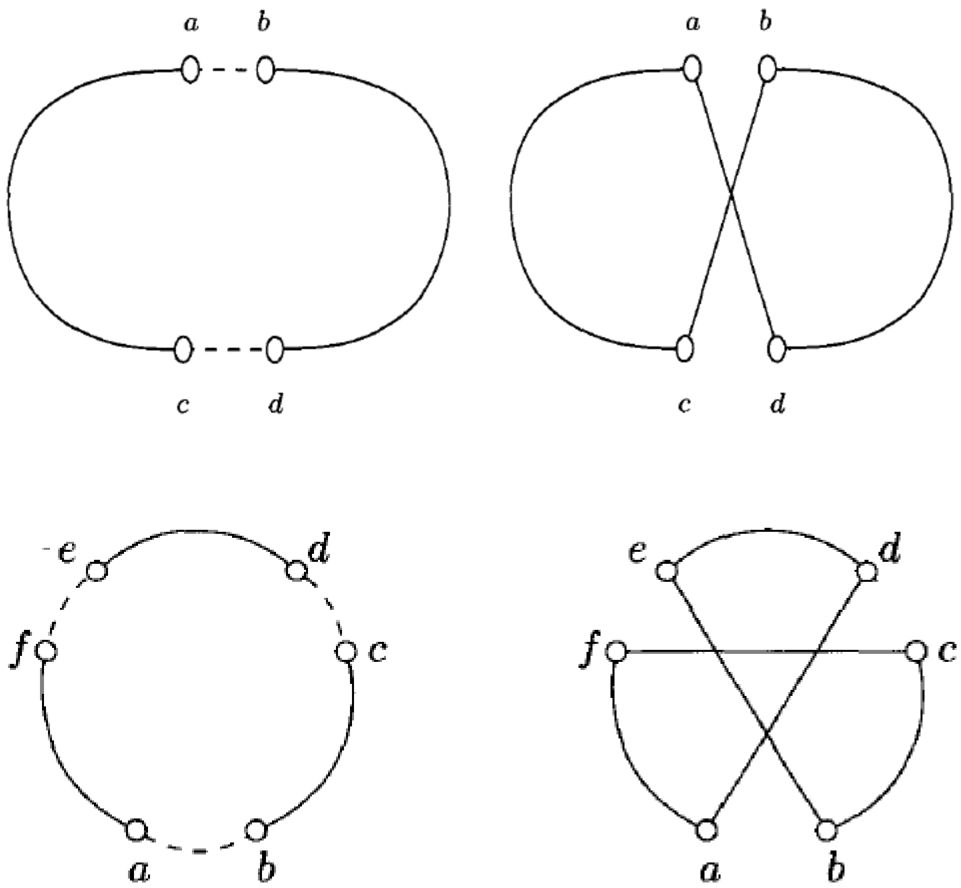
1. 三角不等式 ⇒ w(H)≤w(C)=w(T*)=w(T)+w(M)
2. H*比生成树多一条边 ⇒ w(H*)>w(T)
3. G’中的一个Hamilton圈H’可由G中权和最小的Hamilton圈“去二添一”得到
   - 三角不等式 ⇒ w(H’)≤w(H*)
4. G’的完美匹配可由H’中交替取边得到 ⇒ w(M)≤w(H’)/2
5. w(H)/w(H*)<3/2



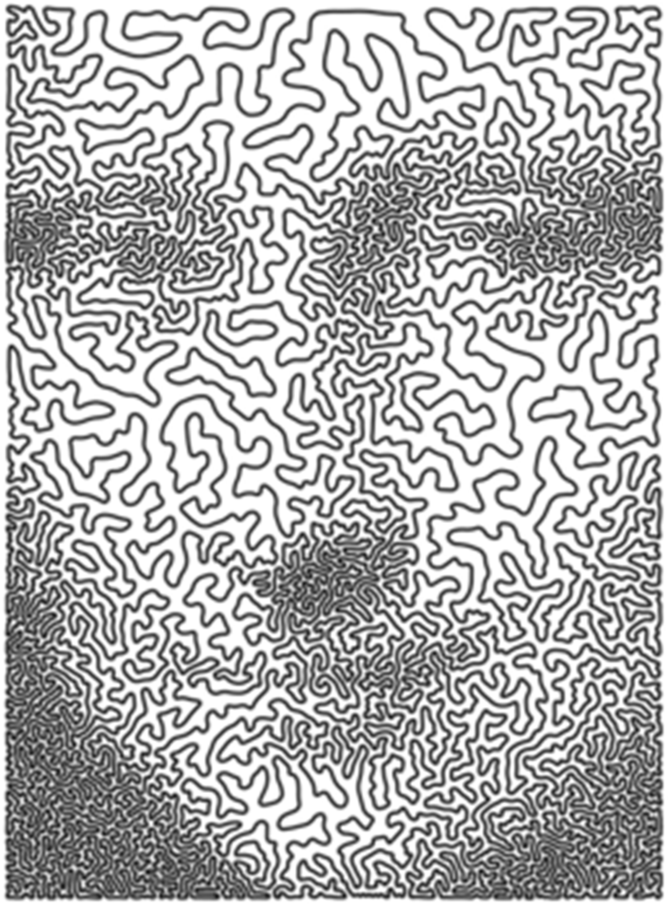
# 最小权匹配法 (续)

- 时间复杂度
  1. 找最小生成树
     - Prim算法：O(ε+vlogv)
  2. 找最小权完美匹配
     - Edmonds算法：$O(v^3)$
     - 另有$O\left(\sqrt{\nu}\varepsilon\right)$算法
  3. 添加边：O(v) //M的边数最多为v/2
  4. 找Euler闭迹
     - Fleury算法：$O(v^2)$
     - 另有O(v)算法
  5. 沿Euler闭迹前行：O(v)

# Kernighan-Lin




理论近似比较差
但实际效果较好



http://www.sciencebuzz.org/sites/default/files/imagecache/kiosk_page_full/kiosk_page_images/monalisa_0.png

# 作业

- 4.12 (要写出过程) //Edmonds-Johnson算法