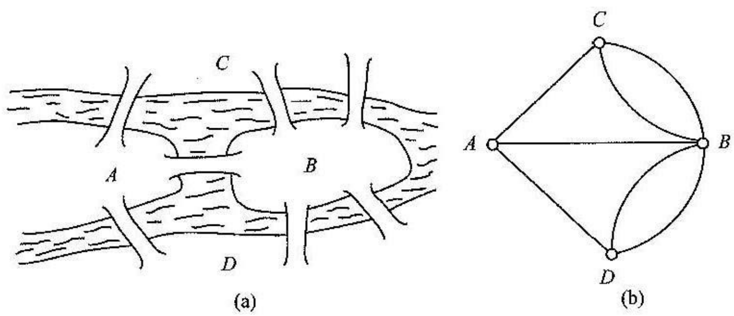
图与网络模型

一、哥尼斯堡七桥问题

普瑞格尔河从哥尼斯堡市中心流过，河中有两座小岛，筑有七座桥，如图下图(a)。1736 年有市民向 Euler 提出所谓的“七桥问题”：从家里出发，七座桥各桥恰好通过一次，再回到家里，是否可能？

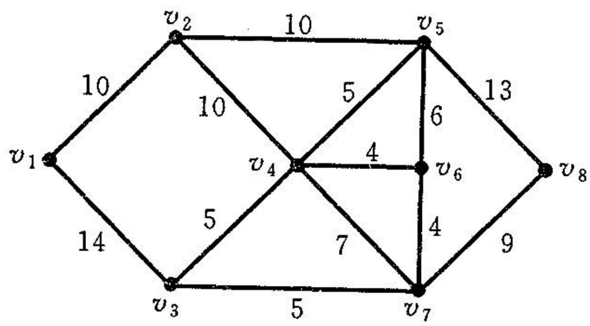


事实上，人们此前已多次试验，均未成功，但又不能严格证明该问题的答案是否定的。Euler 把两岛分别用 A 与 B 两点来表示，两岸分别用 C 与 D 两点来表示。A,B,C,D 各点的位置无关紧要，仅当两块陆地之间有桥时，在相应的两点间连一曲线段，此曲线段的曲直长短也无关紧要，得上图(b)，Euler 称之为图（graph）。他指出，若家在 C 岸，C 点处有三座桥，游人通过其中之一离家出游，不久又经另一座桥回到家，因为要求每座桥恰过一次，他只能经第三座与 C 相连的桥离家游行，此时他已走过与他家相连的三座桥各一次，无法过桥回家了。家在别处同理。故该问题的答案是否定的。Euler 对七桥问题的抽象和论证思想开创了图论（一维拓扑）的研究

二、最短路径问题

1、问题 在加权图$G=(V,E,W)$中求u,v两点之间的路径$P=P(u,v)$，使该路径上的边权之和最小，其中V为（顶）点集，E为边集，$w(e)$为边e的权，$W=\{w(e)|e\in E\}$为权集。（顶点表示研究对象，边表示研究对象之间的某种二元关系）

例如，求下图中从v_1到其它顶点的最短路径。



2、模型

（1）抽象模型 $\min W(P)=\sum_{e\in E(P)} w(e)$。

（2）形象模型 用w_{ij}表示边$v_i\ v_j$的权，令$x_{ij}=\begin{cases}1, & v_iv_j\text{在最短路径上}\\ 0, & \text{否则}\end{cases}$，则$v_1$到$v_m(m=2,3,\cdots,8)$最短路径问题的 0—1 线性规划模型为

$$
\begin{aligned}
&\min z=\sum_{v_iv_j\in E} w_{ij}x_{ij}\\
&s.t.\ \sum_{v_1v_j\in E} x_{1j}=1\\
&\qquad \sum_{v_iv_k\in E} x_{ik}=\sum_{v_kv_j\in E} x_{kj},k\neq 1,m\\
&\qquad \sum_{v_iv_m\in E} x_{im}=1\\
&\qquad x_{ij}=0\text{或}1
\end{aligned}
$$

可用分枝定界法求解该模型。但当点数较多时，LINGO 程序运行时间会很长，运行效率会很差。

3、图论算法

（1）**求给定两点之间最短路径的 Dijkstra 算法**，计算复杂性为

$O(n^2)$，其中$n=|V|$。

例如，对于上图，当v_i与v_j不相邻，即边$v_iv_j\notin E$时，取$w(v_iv_j)=\infty$。

1）标号$l(v_1)=0$，$l(v_2)=10$，$l(v_3)=14$，$l(v_4)=\infty$，$l(v_5)=\infty$，$l(v_6)=\infty$，$l(v_7)=\infty$，$l(v_8)=\infty$。

取固定标号顶点集合为$S=\{v_1\}$，临时标号顶点集合为$T=\{v_2,v_3,v_4,v_5,v_6,v_7,v_8\}$。

由$l(v_2)=\min\limits_{v_i\in V-S}\{l(v_i)\}$可知，$v_1$到$v_2$的最短路径为$v_1v_2$，权为 10。

令$S=\{v_1,v_2\}$，$T=\{v_3,v_4,v_5,v_6,v_7,v_8\}$。

2）令$l(v_3)=\min\{l(v_3),l(v_2)+w(v_2v_3)\}=\min\{14,10+\infty\}=14$

$l(v_4)=\min\{l(v_4),l(v_2)+w(v_2v_4)\}=\min\{\infty,10+10\}=20$

$l(v_5)=\min\{l(v_5),l(v_2)+w(v_2v_5)\}=\min\{\infty,10+10\}=20$

$l(v_6)=\min\{l(v_6),l(v_2)+w(v_2v_6)\}=\min\{\infty,10+\infty\}=\infty$

$l(v_7)=\min\{l(v_7),l(v_2)+w(v_2v_7)\}=\min\{\infty,10+\infty\}=\infty$

$l(v_8)=\min\{l(v_8),l(v_2)+w(v_2v_8)\}=\min\{\infty,10+\infty\}=\infty$

由$l(v_3)=\min\limits_{v_i\in V-S}\{l(v_i)\}$可知，$v_1$到$v_3$的最短路径为$v_1v_3$，权为 14。

令$S=\{v_1,v_2,v_3\}$，$T=\{v_4,v_5,v_6,v_7,v_8\}$。

3）令$l(v_4)=\min\{l(v_4),l(v_3)+w(v_3v_4)\}=\min\{20,14+5\}=19$

$l(v_5)=\min\{l(v_5),l(v_3)+w(v_3v_5)\}=\min\{20,14+\infty\}=20$

$l(v_6)=\min\{l(v_6),l(v_3)+w(v_3v_6)\}=\min\{\infty,14+\infty\}=\infty$

$l(v_7)=\min\{l(v_7),l(v_3)+w(v_3v_7)\}=\min\{\infty,14+5\}=19$

$l(v_8)=\min\{l(v_8),l(v_3)+w(v_3v_8)\}=\min\{\infty,14+\infty\}=\infty$

由$l(v_7)=\min\limits_{v_i\in V-S}\{l(v_i)\}$可知，$v_1$到$v_7$的最短路径为$v_1v_3v_7$，权为 19。

令 $S=\{\ v_1,v_2,v_3,v_7\}$ ， $T=\{v_4,v_5,v_6,v_8\}$ 。

4）令 $l(v_4)=\min\{l(v_4),l(v_7)+w(v_7v_4)\}=\min\{19,19+7\}=19$

$l(v_5)=\min\{l(v_5),l(v_7)+w(v_7v_5)\}=\min\{20,19+\infty\}=20$

$l(v_6)=\min\{l(v_6),l(v_7)+w(v_7v_6)\}=\min\{\infty,19+4\}=23$

$l(v_8)=\min\{l(v_8),l(v_7)+w(v_7v_8)\}=\min\{\infty,19+9\}=28$

由 $l(v_4)=\min\limits_{v_i\in V-S}\{l(v_i)\}$ 可知， v_1 到 v_4 的最短路径为 $v_1v_3v_4$，权为 19。

令 $S=\{\ v_1,v_2,v_3,v_4,v_7\}$ ， $T=\{v_5,v_6,v_8\}$ 。

5）令 $l(v_5)=\min\{l(v_5),l(v_4)+w(v_4v_5)\}=\min\{20,19+5\}=20$

$l(v_6)=\min\{l(v_6),l(v_4)+w(v_4v_6)\}=\min\{23,19+4\}=23$

$l(v_8)=\min\{l(v_8),l(v_4)+w(v_4v_8)\}=\min\{28,19+9\}=28$

由 $l(v_5)=\min\limits_{v_i\in V-S}\{l(v_i)\}$ 可知， v_1 到 v_5 的最短路径为 $v_1v_2v_5$，权为 20。

令 $S=\{\ v_1,v_2,v_3,v_4,v_5,v_7\}$ ， $T=\{v_6,v_8\}$ 。

6）令 $l(v_6)=\min\{l(v_6),l(v_5)+w(v_5v_6)\}=\min\{23,20+6\}=23$

$l(v_8)=\min\{l(v_8),l(v_5)+w(v_5v_8)\}=\min\{28,20+13\}=28$

由 $l(v_6)=\min\limits_{v_i\in V-S}\{l(v_i)\}$ 可知， v_1 到 v_6 的最短路径为 $v_1v_3v_4v_6$ 或 $v_1v_3v_7v_6$，权为 23。

令 $S=\{\ v_1,v_2,v_3,v_4,v_5,v_6,v_7\}$ ， $T=\{v_8\}$ 。

7）令 $l(v_8)=\min\{l(v_8),l(v_6)+w(v_6v_8)\}=\min\{28,23+\infty\}=28$

由此可知， v_1 到 v_8 的最短路径为 $v_1v_3v_7v_8$，权为 28。

令 $S=\{\ v_1,v_2,v_3,v_4,v_5,v_6,v_7,v_8\}$ ， $T=\Phi$，停止计算。

Dijkstra 算法也适用于加权有向图。

(2)求任意两点之间最短路径的 Floyd 算法，计算复杂性为 $O(n^3)$。

以权矩阵为初始矩阵$D^{(0)}=\left(d_{ij}^{(0)}\right)_{n\times n}$，其中$d_{ij}^{(0)}=w_{ij}=w(v_i v_j)$，$w_{ii}=0$。利用$D^{(k-1)}=\left(d_{ij}^{(k-1)}\right)_{n\times n}$迭代计算$D^{(k)}=\left(d_{ij}^{(k)}\right)_{n\times n}$，其中$d_{ij}^{(k)}=\min\left\{d_{ij}^{(k-1)},\ d_{ik}^{(k-1)}+d_{kj}^{(k-1)}\right\}$，$k=1,2,\cdots,n$，$d_{ij}^{(n)}$即为$v_i$到$v_j$的最短路径的权。若$d_{ij}^{(n)}=\infty$，则不存在$v_i$到$v_j$的路径。迭代过程中记录$v_i$到$v_j$的路径。

例如，对于上图，

$$D^{(0)}=\begin{bmatrix} 0 & 10 & 14 & \infty & \infty & \infty & \infty & \infty \\ 10 & 0 & \infty & 10 & 10 & \infty & \infty & \infty \\ 14 & \infty & 0 & 5 & \infty & \infty & 5 & \infty \\ \infty & 10 & 5 & 0 & 5 & 4 & 7 & \infty \\ \infty & 10 & \infty & 5 & 0 & 6 & \infty & 13 \\ \infty & \infty & \infty & 4 & 6 & 0 & 4 & \infty \\ \infty & \infty & 5 & 7 & \infty & 4 & 0 & 9 \\ \infty & \infty & \infty & \infty & 13 & \infty & 9 & 0 \end{bmatrix} \qquad D^{(1)}=\begin{bmatrix} 0 & 10 & 14 & \infty & \infty & \infty & \infty & \infty \\ 10 & 0 & 24 & 10 & 10 & \infty & \infty & \infty \\ 14 & 24 & 0 & 5 & \infty & \infty & 5 & \infty \\ \infty & 10 & 5 & 0 & 5 & 4 & 7 & \infty \\ \infty & 10 & \infty & 5 & 0 & 6 & \infty & 13 \\ \infty & \infty & \infty & 4 & 6 & 0 & 4 & \infty \\ \infty & \infty & 5 & 7 & \infty & 4 & 0 & 9 \\ \infty & \infty & \infty & \infty & 13 & \infty & 9 & 0 \end{bmatrix}$$

$$D^{(2)}=\begin{bmatrix} 0 & 10 & 14 & 20 & 20 & \infty & \infty & \infty \\ 10 & 0 & 24 & 10 & 10 & \infty & \infty & \infty \\ 14 & 24 & 0 & 5 & 34 & \infty & 5 & \infty \\ 20 & 10 & 5 & 0 & 5 & 4 & 7 & \infty \\ 20 & 10 & 34 & 5 & 0 & 6 & \infty & 13 \\ \infty & \infty & \infty & 4 & 6 & 0 & 4 & \infty \\ \infty & \infty & 5 & 7 & \infty & 4 & 0 & 9 \\ \infty & \infty & \infty & \infty & 13 & \infty & 9 & 0 \end{bmatrix} \qquad D^{(3)}=\begin{bmatrix} 0 & 10 & 14 & 19 & 20 & \infty & 19 & \infty \\ 10 & 0 & 24 & 10 & 10 & \infty & 29 & \infty \\ 14 & 24 & 0 & 5 & 34 & \infty & 5 & \infty \\ 19 & 10 & 5 & 0 & 5 & 4 & 7 & \infty \\ 20 & 10 & 34 & 5 & 0 & 6 & 39 & 13 \\ \infty & \infty & \infty & 4 & 6 & 0 & 4 & \infty \\ 19 & 29 & 5 & 7 & 39 & 4 & 0 & 9 \\ \infty & \infty & \infty & \infty & 13 & \infty & 9 & 0 \end{bmatrix}$$

$$D^{(4)}=\begin{bmatrix} 0 & 10 & 14 & 19 & 20 & 23 & 19 & \infty \\ 10 & 0 & 15 & 10 & 10 & 14 & 17 & \infty \\ 14 & 15 & 0 & 5 & 10 & 9 & 5 & \infty \\ 19 & 10 & 5 & 0 & 5 & 4 & 7 & \infty \\ 20 & 10 & 10 & 5 & 0 & 6 & 12 & 13 \\ 23 & 14 & 9 & 4 & 6 & 0 & 4 & \infty \\ 19 & 17 & 5 & 7 & 12 & 4 & 0 & 9 \\ \infty & \infty & \infty & \infty & 13 & \infty & 9 & 0 \end{bmatrix} \qquad D^{(5)}=\begin{bmatrix} 0 & 10 & 14 & 19 & 20 & 23 & 19 & 33 \\ 10 & 0 & 15 & 10 & 10 & 14 & 17 & 23 \\ 14 & 15 & 0 & 5 & 10 & 9 & 5 & 23 \\ 19 & 10 & 5 & 0 & 5 & 4 & 7 & 18 \\ 20 & 10 & 10 & 5 & 0 & 6 & 12 & 13 \\ 23 & 14 & 9 & 4 & 6 & 0 & 4 & 19 \\ 19 & 17 & 5 & 7 & 12 & 4 & 0 & 9 \\ 33 & 23 & 23 & 18 & 13 & 19 & 9 & 0 \end{bmatrix}$$

$$D^{(6)}=\begin{bmatrix} 0 & 10 & 14 & 19 & 20 & 23 & 19 & 33 \\ 10 & 0 & 15 & 10 & 10 & 14 & 17 & 23 \\ 14 & 15 & 0 & 5 & 10 & 9 & 5 & 23 \\ 19 & 10 & 5 & 0 & 5 & 4 & 7 & 18 \\ 20 & 10 & 10 & 5 & 0 & 6 & 10 & 13 \\ 23 & 14 & 9 & 4 & 6 & 0 & 4 & 19 \\ 19 & 17 & 5 & 7 & 10 & 4 & 0 & 9 \\ 33 & 23 & 23 & 18 & 13 & 19 & 9 & 0 \end{bmatrix} \quad D^{(7)}=\begin{bmatrix} 0 & 10 & 14 & 19 & 20 & 23 & 19 & 28 \\ 10 & 0 & 15 & 10 & 10 & 14 & 17 & 23 \\ 14 & 15 & 0 & 5 & 10 & 9 & 5 & 14 \\ 19 & 10 & 5 & 0 & 5 & 4 & 7 & 16 \\ 20 & 10 & 10 & 5 & 0 & 6 & 10 & 13 \\ 23 & 14 & 9 & 4 & 6 & 0 & 4 & 13 \\ 19 & 17 & 5 & 7 & 10 & 4 & 0 & 9 \\ 28 & 23 & 14 & 16 & 13 & 13 & 9 & 0 \end{bmatrix}$$

$D^{(8)}=D^{(7)}$。停止计算，得到任意两点之间的最短路径的权。

对于加权有向图，$D^{(0)}$一般不是对称阵。

4、应用

（1）最可靠路径问题：转化为最短路径问题。

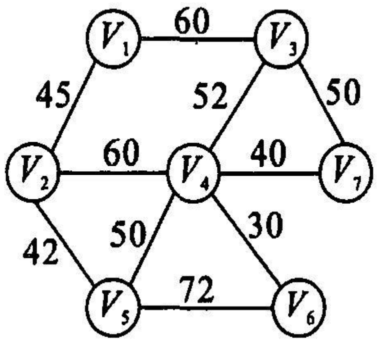
（2）工序 PT 图与 PERT 图的关键路径问题：仿照 Dijkstra 算法求最长路径。（不可仿照 Dijkstra 算法求一般加权图的最长路径）

（3）选址问题（应急服务设施的中心点，非应急服务设施的重心点）：以最短路径为基础。

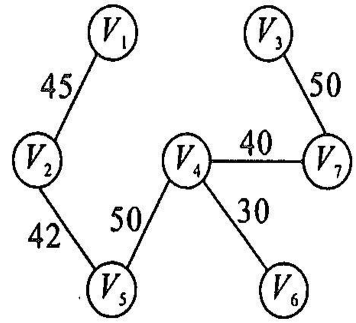
（4）点带权的加权图最短路径问题：重构图，将点权转化为边权。

三、最小生成树问题

1、问题　在加权连通图中求生成树T，使该树上的边权之和最小。（存在路径的两点之间称为是连通的；任意两点均存在路径的图称为是连通图；$|E|=|V|-1$的连通图称为树；含有图的所有顶点的树形子图称为该图的生成树）



加权连通图



最小生成树

例如，上图中，已知v_i城与v_j城间的铁路造价为w_{ij}，设计一个连接7个城市的总造价最低的铁路筑路图的问题可归结为最小生成树问题。

2、模型

（1）抽象模型　$\min W(T)=\sum\limits_{e\in E(T)} w(e)$。

（2）形象模型　不妨设v_1为树根节点，w_{ij}表示边v_iv_j的权，令

$$x_{ij}=\begin{cases}1，\text{弧}v_iv_j\text{在生成树上}\\0，\qquad\text{否则}\end{cases}$$

目标函数：生成树的权之和最小，即$\min \sum\limits_{v_iv_j\in E} w_{ij}x_{ij}$；

约束条件：由树的定义知，$\sum\limits_{v_iv_j\in E} x_{ij}=n-1$；

但只采用上述约束条件可能会出现圈$v_4\to v_5\to v_6\to v_4$。为此，不妨以v_1为生成树的树根，给点v_i分配一个级别值u_i，其中树根v_1的级别值$u_1=0$。级别值可理解为生成树中从树根v_1到v_i的唯一路径的边数，显然$u_i\le n-1$。则有

$$u_j\ge u_i+1-n+nx_{ij}，\ v_iv_j\in E$$

若该模型的解为生成树，则从树根开始遍历此树求出的各级别值均

可满足此约束条件。若 $x_{ij}=1$，即弧 v_iv_j 在生成树上，且 $u_j=u_i+1$，则此式成立；若 $x_{ij}=0$，此式显然成立。

假设该模型的某个解中包含一个圈 $v_i \to v_j \to v_k \to v_i$，对于此圈，由上述约束条件可得

$$u_j \ge u_i+1，\quad u_k \ge u_j+1，\quad u_i \ge u_k+1$$

逐项相加得矛盾结果：0≥3。故包含 3 条以上边构成的圈的解不满足上述约束条件。

根据上述分析，可建立最小生成树问题的混合整数规划模型

$$\begin{aligned}
&\min \sum_{v_iv_j\in E} w_{ij}x_{ij} \\
&s.t. \sum_{v_iv_j\in E} x_{ij}=n-1 \\
&\qquad u_j \ge u_i+1-n+nx_{ij}，\ v_iv_j\in E \\
&\qquad x_{ij}\in\{0,1\}，\ v_iv_j\in E \\
&\qquad u_1=0，\ u_i\in N，\ v_i\in V
\end{aligned}$$

3、图论算法

（1）Kruskal 算法（避圈法）；

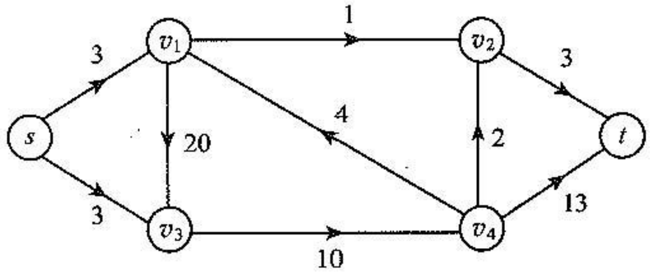
（2）Prim 算法；（首选）

（3）破圈法。

计算复杂性均为 $O(n^2)$。不可利用最小生成树求最短路径。

四、网络最大流问题

1、问题 在单源单汇具有容量上限的网络 $N=(V,E,C)$ 中求从源到汇的流量最大的可行流，其中 $C=\{c_{ij}\,|\,v_iv_j\in E\}$，$c_{ij}$ 为 v_iv_j 上单位时间流量的上限，称为 v_iv_j 的容量。



例如，把一种商品从产地s通过上图所示铁路或公路网络运往市场t，交通网络中每一路段的单位时间运输能力有一定限度，边权即为限度值，单位时间内运输量最大的运输方案问题可归结为最大流问题。

2、模型 设f_{ij}为$v_i v_j$上单位时间的流量，记$f=\{f_{ij}|v_i v_j \in E\}$，称为网络$N$中的流。设流量守恒，满足下述约束条件的流称为可行流，可建立最大流问题的线性规划模型

$$
\begin{aligned}
&\max \quad v(f) \\
&s.t. \sum_j f_{ij} = \sum_k f_{ki}, v_i \neq v_s, v_t \\
&\quad \sum_j f_{sj} = \sum_k f_{kt} = v(f) \\
&\quad 0 \le f_{ij} \le c_{ij}, v_i v_j \in E
\end{aligned}
$$

其中$v(f)$表示网络N的单位时间流量。

可利用优化软件求解该模型。多源多汇网络易化为单源单汇网络。

3、图论算法

（1）Ford-Fulkerson 算法，计算复杂性与容量有关，而与点数和边数无关，适用于一般网络最大流的求解；

（2）Edmonds-Karp 算法，计算复杂性为$O(m^2 n)$,其中$n=|V|$，$m=|E|$，主要用于求解某些特殊网络最大流。

4、应用

（1）最小割集：Ford-Fulkerson 算法停止时某些弧构成割集，割集中边的容量总和称为割量，割量最小的割集称为最小割集，即“瓶颈边”构成的集合。著名的双最定理：网络的最大流量等于最小割量。

（2）最小流。

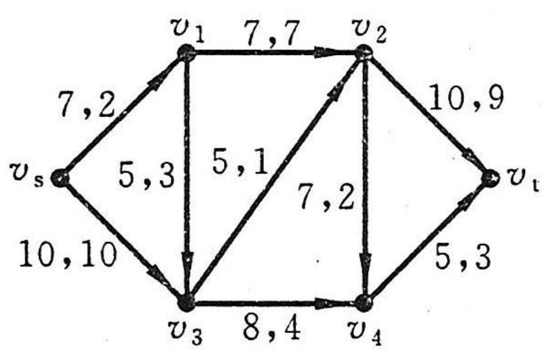
（3）多端最大流。

（4）增益流。

（5）点具有容量的最大流。

（6）最小费用流，只需在最大流模型中将目标改为 $\min \sum_{v_i v_j \in E} b_{ij} f_{ij}$ 即可得最小费用流的线性规划模型，其中 b_{ij} 为 $v_i v_j$ 上单位流量的费用。还可进一步考虑目标函数非线性的情况。

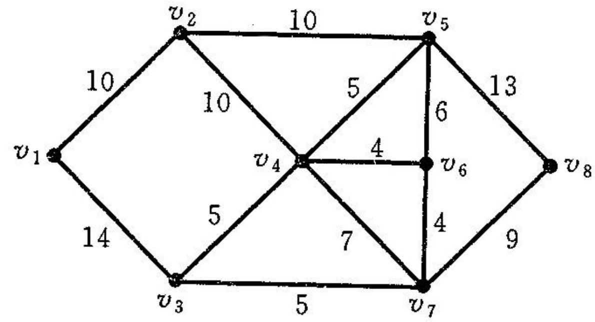
如下图所示，图中弧旁的数字分别为容量和单位流量费用。



五、图的独立集、覆盖集与支配集问题

1、匹配（边独立集）

（1）问题　求图中边数最多的不相邻的边集，即最大匹配。最大匹配的边数称为匹配数。



例如，设上图中$v_i(i=1,2,\cdots,8)$为 8 名工作人员，$v_iv_j\in E$表示v_i与v_j可组队工作，$v_iv_j\notin E$表示v_i与v_j不可组队工作，一名工作人员与他人最多组一队，最多队数问题即可归结为最大匹配问题。$\{v_2v_5,v_3v_7,v_4v_6\}$为匹配，$\{v_1v_2,v_3v_4,v_5v_6,v_7v_8\}$为最大匹配。

（2）算法

1）求非偶图最大匹配的“开花”算法，计算复杂性为$O(n^3)$。或引入 0-1 变量x_{ij}，当v_i与v_j配对时，$x_{ij}=1$；否则，$x_{ij}=0$，建立非偶图最大匹配问题的 0-1 线性规划模型

$$\max\sum_{v_iv_j\in E}x_{ij}$$

$$s.t.\begin{cases}\sum_{v_j\in N(v_i)}x_{ij}\le 1,\ v_i\in V,\ i=1,2,\cdots,n\\ x_{ij}=x_{ji}=0\text{或}1,\ v_iv_j\in E,\ i\ne j\end{cases}$$

其中$N(v_i)=\{$所有与v_i相邻的点$\}$，称为点v_i的邻集。

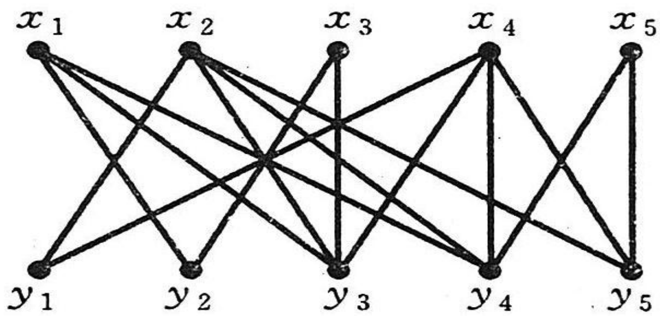
2）求偶图$G=(X,Y,E)$最大匹配的匈牙利算法，计算复杂性为$O(mn)$。或引入 0-1 变量x_{ij}，当$x_i\in X$与$y_j\in Y$配对时，$x_{ij}=1$；否则，$x_{ij}=0$，建立偶图最大匹配问题的 0-1 线性规划模型

$$\max\sum_{x_iy_j\in E}x_{ij}$$

$$s.t.\ \sum_{y_j \in N(x_i)} x_{ij} \le 1,\ \ x_i \in X$$

$$\sum_{x_i \in N(y_j)} x_{ij} \le 1,\ \ y_j \in Y$$

$$x_{ij} = 0或1,\ \ x_i y_j \in E,\ \ x_i \in X,\ \ y_j \in Y$$



例如，设上图中x_1, x_2, x_3, x_4, x_5表示 5 名工作人员，y_1, y_2, y_3, y_4, y_5表示 5 项工作，工作人员x_i能胜任工作y_j就在x_i与y_j之间连边$x_i y_j$，假设一名工作人员最多承担一项工作，一项工作最多由一名工作人员承担。

（3）**应用**

1）**问题**　求加权偶图中权和最大或最小的最大匹配，即最优匹配。

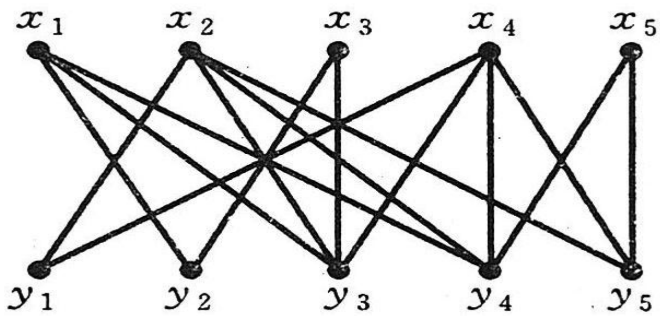
2）**模型**　若求权和最大的最优匹配，则只要在上述模型中将目标改为$\max \sum_{x_i y_j \in E} w_{ij} x_{ij}$即可得其线性 0-1 规划模型，其中$w_{ij} = w(x_i y_j)$，$x_i y_j \in E$；若求权和最小的最优匹配，则只要在上述模型中将目标改为$\max \sum_{x_i y_j \in E} w'_{ij} x_{ij}$即可得其线性 0-1 规划模型，其中$w'_{ij} = w - w_{ij}$，$w = \max_{x_i y_j \in E} \{w_{ij}\}$，$x_i y_j \in E$。

3）**图论算法**　kuhn-Munkras 算法。

（4）**推广**　一人承担多项工作或一项工作由多人承担的指派问题，只要将上述模型中的约束条件适当修改即可。

2、边覆盖集

（1）**问题**　求图中边数最少的覆盖所有点的边集，即最小边覆盖集。最小边覆盖集的边数称为边覆盖数。



例如，设上图中 x_1,x_2,x_3,x_4,x_5 表示 5 个地区，y_1,y_2,y_3,y_4,y_5 表示 5 个民族，边 $x_iy_j\in E$ 表示来自第 i 个地区和第 j 个民族的候选人，从这 15 个候选人中选出能覆盖所有地区和民族的最少人数组成委员会，即可归结为最小边覆盖集问题。

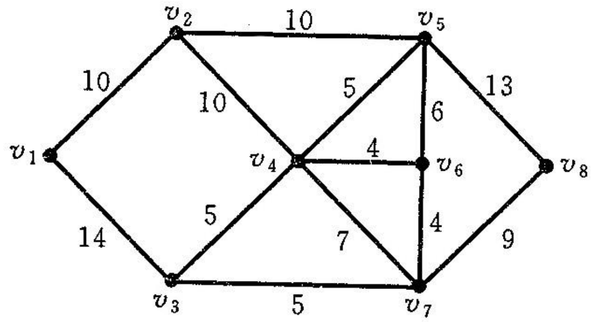
（2）**算法**　通过最大匹配求得。或引入 0-1 变量 x_{ij}，当 v_iv_j 属于边覆盖集时，$x_{ij}=1$；否则，$x_{ij}=0$，建立最小边覆盖集问题的 0-1 线性规划模型

$$\min \sum_{v_iv_j\in E} x_{ij}$$

$$s.t.\begin{cases}\displaystyle\sum_{v_j\in N(v_i)} x_{ij}\ge 1,\ v_i\in V,\ i=1,2,\cdots,n\\ x_{ij}=x_{ji}=0或1,\ v_iv_j\in E,\ i\ne j\end{cases}$$

3、**覆盖集**

（1）**问题**　求图中点数最少的覆盖所有边的点集，即最小（点）覆盖集。最小覆盖集的点数称为覆盖数。



例如，在上图所示的居民区，假设点表示道路交叉处，边表示道路，假设在任一点安装的监控设备可监控与该点关联的所有边，选择最少的点安装设备以监控所有边，即可归结为最小覆盖集问题。

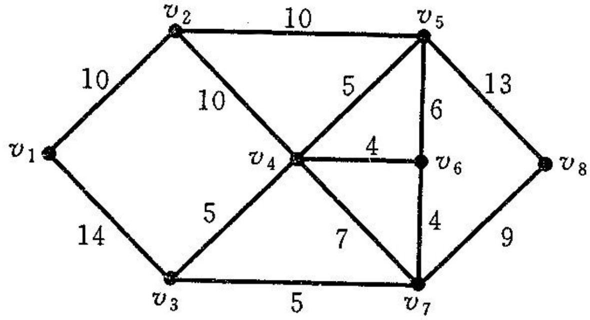
（2）**算法**　求所有极小覆盖集的逻辑算法：$\prod_{i=1}^{n}\left(v_i+\prod_{v_j\in N(v_i)}v_j\right)$。或引入 0-1 变量 x_i，当 v_i 属于覆盖集时，$x_i=1$，否则，$x_i=0$，建立最小覆盖集问题的 0-1 线性规划模型

$$\min\sum_{i=1}^{n}x_i$$

$$s.t.\begin{cases}x_i+x_j\geq 1, v_iv_j\in E, 1\leq i<j\leq n\\ x_i=0\text{或}1,\ \ i=1,2,\cdots,n,\end{cases}$$

4、独立集

（1）**问题**　求图中点数最多的不相邻的点集，即最大（点）独立集。最大独立集的点数称为独立数。



例如，设上图中点表示需要传输的 8 个基本信号，两个易于发生错乱的基本信号之间连一条边，最大的无错乱基本信号集即为最大独立集。

（2）算法　求出最小覆盖集，其补集即为最大独立集。或引入 0-1 变量 x_i，当 v_i 属于独立集时，$x_i = 1$；否则，$x_i = 0$，建立最大独立集问题的 0-1 线性规划模型

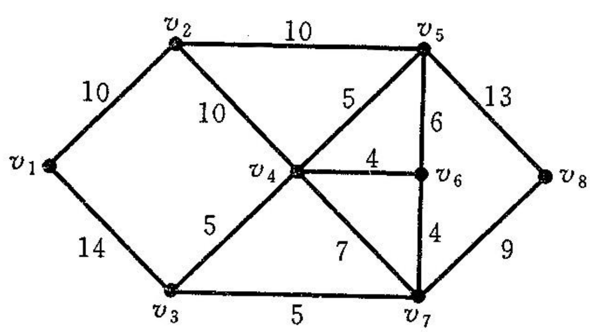
$$\max \sum_{i=1}^{n} x_i$$

$$s.t. \begin{cases} x_i + x_j \le 1, & v_i v_j \in E, 1 \le i < j \le n \\ x_i = 0 \text{或} 1, & i = 1, 2, \cdots, n \end{cases}$$

考虑到一般连通图的边数多于顶点数，为减少约束条件数目，将约束条件换成 $|N(v_i)| x_i + \sum_{v_j \in N(v_i)} x_j \le |N(v_i)|$，$v_i \in V$，$i = 1, 2, \ldots, n$，可减少计算量。

5、支配集

（1）问题　求点数最少的点集，使图中每个点或属于该点集，或与该点集中至少一点相邻，即最小支配集。最小支配集的点数称为支配数。



例如，设上图中点表示通讯网络中的台站，边表示直通线路，选定最少的台站建成中心台站，使其与其它所有台站有直通线路，即可归结为最小支配集问题。

（2）**算法**　求所有极小支配集的逻辑算法：$\prod_{i=1}^{n}\left(v_i+\sum_{v_j\in N(v_i)}v_j\right)$。或引入 0-1 变量$x_i$，当$v_i$属于支配集时，$x_i=1$；否则，$x_i=0$，建立最小支配集问题的 0-1 线性规划模型

$$\min\sum_{i=1}^{n}x_i$$

$$s.t.\begin{cases}x_i+\sum\limits_{v_j\in N(v_i)}x_j\geq 1, i=1,2,\cdots,n\\ x_i=0或1,\ i=1,2,\cdots,n,\end{cases}$$

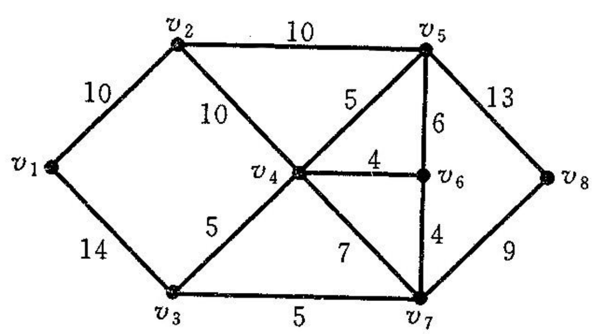
六、中国邮路问题（CPP）

1、**问题**　求 Euler 图中的 Euler 回路，即经过每条边恰好一次的回路。

2、**算法**　Fleury 算法，计算复杂性为$O(m)$。

3、**应用**

（1）**问题**　在加权连通图中求经过每条边至少一次且权和最小的回路，即中国邮路。



例如，一位邮递员从邮局v_4出发投递邮件，必须经过由他负责投递的每条街道（边）至少一次，最后回到邮局，行程最短的投递线路问题可归结为中国邮路问题。

（2）**模型**　对于无向加权连通图，设x_{ij}为从v_i到v_j经过的次数，建立 CPP 问题的整数线性规划模型

$$\min \sum_{v_i v_j \in E} w_{ij} x_{ij}$$

$$s.t. \sum_{v_j \in N(v_i)} x_{ij} = \sum_{v_j \in N(v_i)} x_{ji}, v_i \in V$$

$$x_{ij} + x_{ji} \geq 1, v_i v_j \in E$$

$$x_{ij} \in N, v_i v_j \in E$$

对于有向加权连通图，若所有顶点的入度和出度均大于零，则只要在上述模型中将第二个约束条件改为$x_{ij} \geq 1, v_i v_j \in E$，即可得其模型；否则，不存在有向中国邮路。

（3）**图论算法**

1）奇偶点图上作业法；

2）最小权匹配算法，计算复杂性为$O(n^3)$。

4、推广　多邮递员中国邮路问题（$k-CPP$）。

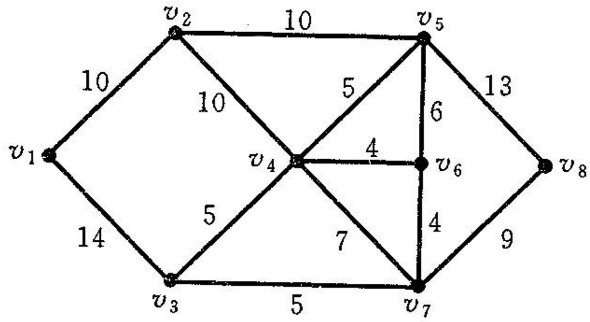
七、旅行推销商问题（TSP）

1、问题　求 Hamilton 图中的 Hamilton 回路，即经过每个点恰好一次的回路。

2、算法　DFS 法，计算复杂性为$O(n!)$。

3、应用

（1）**问题**　在加权连通图中求经过每个点至少一次的权和最小的回路，即旅行推销商回路（TSP 回路）。



例如，一位销售商从城市v_4出发投推销商品，必须经过由他负责投递的每座城市（点）至少一次，最后回到v_4，行程最短的旅行线路问题可归结为旅行推销商问题。

（2）**模型** 先将一般加权连通图转化成一个等价的加权完全图，设当从v_i到v_j时，$x_{ij}=1$，否则，$x_{ij}=0$，$i \neq j$，则可建立如下 0-1 线性规划模型，先求得该加权完全图权和最小的 Hamilton 回路，再转化为原加权连通图的 TSP 回路。

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}x_{ij}$$

$$s.t. \sum_{j=1}^{n} x_{ij}=1, i=1,\cdots,n$$

$$\sum_{i=1}^{n} x_{ij}=1, j=1,\cdots,n$$

$$x_{i_1i_2}+x_{i_2i_3}+\cdots+x_{i_ki_1} \le k-1,\ \ i_1,i_2,\cdots,i_k=1,2,\cdots,n,k=2,3,\cdots,n-1$$

$$x_{ij}=0 \text{ 或 } 1,\ \ i,j=1,\cdots,n,i\neq j$$

（3）近似算法

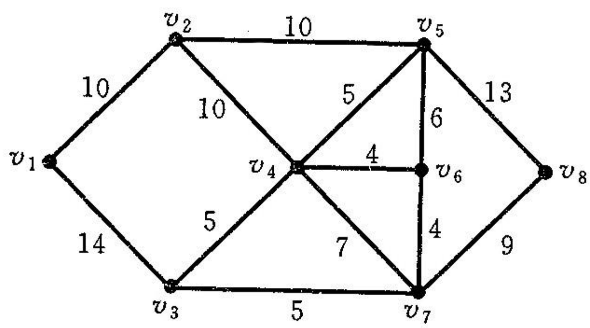
1）最小生成树法；2）代换法；3）插入法；4）最近邻法；5）神经网络法；6）模拟退火法；7）蚂蚁算法。

4、推广 多旅行推销商问题$(k-TSP)$。

七、图的着色问题

1、点着色

（1）问题 求给图的点着色，且使邻点异色的最少颜色数，即（点）色数。



例如，设上图中点表示需装箱的货物，装在同一箱里不安全的两种货物之间连一条边，准备最少数量的箱子装货即可归结为色数问题。

（2）模型 当v_i着第k种颜色时，令$x_{ik}=1$；否则，$x_{ik}=0$。设颜色种数为x，建立点着色问题的 0-1 线性规划模型

$$\min x$$

$$s.t. \sum_{k=1}^{\Delta+1} x_{ik}=1, v_i \in V$$

$$x_{ik}+x_{jk} \le 1, v_i v_j \in E, k=1,2,\cdots,\Delta+1$$

$$x \ge \sum_{k=1}^{\Delta+1} k x_{ik}, v_i \in V$$

$$x_{ik}=0\text{或}1, v_i \in V, k=1,2,\cdots,\Delta+1$$

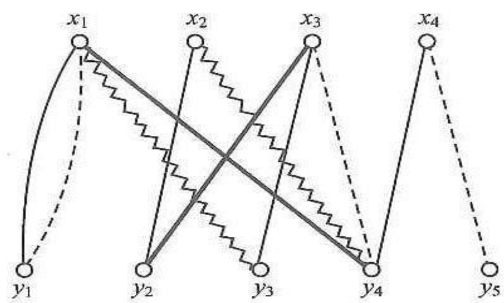
其中$\Delta=\max_{v\in V}\{d(v)\}$，$d(v)$为$v$的度数。可证最少颜色数不超过$\Delta+1$。

（3）图论算法

1）图收缩法，非多项式算法；

2）Welch-Powell 算法，为近似算法。

2、边着色 求给图的边着色，且使邻边异色的最少颜色数，即边色数。可证最少颜色数为Δ或$\Delta+1$。



例如，设上图中x_1,x_2,x_3,x_4表示 4 位教师，y_1,y_2,y_3,y_4,y_5表示 5 个班级，当x_i教师为y_j班级每天授p_{ij}节课时，在x_i与y_j之间连p_{ij}条边，制作学校上课时间最少的课表即可归结为边色数问题。

边着色可转化为点着色。或引入 0-1 变量x_{ijk}，当v_iv_j着第k种颜色时，$x_{ijk}=1$；否则；$x_{ijk}=0$。设颜色种数为x，建立边着色问题的 0-1 线性规划模型。

$$\min x$$

$$s.t.\sum_{k=1}^{\Delta+1}x_{ijk}=1, v_iv_j\in E$$

$$\sum_{v_j\in N(v_i)}x_{ijk}\le 1, v_i\in V, k=1,2,\cdots,\Delta+1$$

$$x\ge\sum_{k=1}^{\Delta+1}kx_{ijk}, v_iv_j\in E$$

$$x_{ijk}=0\text{或}1, v_iv_j\in E, k=1,2,\cdots,\Delta+1$$

3、面着色 求给平面图的面着色，且使邻面异色的最少颜色数，即面色数。面着色也可转化为点着色。