中山大學
SUN YAT-SEN UNIVERSITY

《图论及其应用》2024

潘嵘

计算机学院

第一章 图的基本概念

本次课主要内容

最短路算法、图的代数表示

(一)、最短路算法

(二)、图的代数表示

1、图的邻接矩阵

2、图的关联矩阵

(一)、最短路算法

1、相关概念

（1）赋权图

在图G的每条边上标上一个实数$w\ (e)$后，称G为边赋权图。被标上的实数称为边的权值。

若H是赋权图G的一个子图，称$W(H)=\sum_{e\in E(H)}w(e)$为子图$H$的权值。

权值的意义是广泛的。可以表示距离，可以表示交通运费，可以表示网络流量，在朋友关系图甚至可以表示友谊深度。但都可以抽象为距离。

（2）赋权图中的最短路

设G为边赋权图。u与v是G中两点，在连接u与v的所有路中，路中各边权值之和最小的路，称为u与v间的最短路。

（3）算法

解决某类问题的一组有穷规则，称为算法。

1）好算法

算法总运算量是问题规模的多项式函数时，称该算法为好算法。（问题规模：描述或表示问题需要的信息量）

算法中的运算包括算术运算、比较运算等。运算量用运算次数表示。

2）算法分析

对算法进行分析，主要对时间复杂性进行分析。分析运算量和问题规模之间的关系。

2、最短路算法

1959年，旦捷希(Dantzig）发现了在赋权图中求由点a到点b的最短路好算法，称为顶点标号法。

$t(a_n)$: 点a_n的标号值, 表示点$a_1 = a$到a_n的最短路长度

$A_i = \{a_1, a_2, \dots, a_i\}$: 已经标号的顶点集合。

T_i：a_1到a_i的最短路上的边集合

算法叙述如下：

（1）记 $a=a_1$，$t(a_1)\ =0$，$A_1=\ \{a_1\}$，$T_1=\ \emptyset$；

（2） 若 已 经 得 到 $A_i\ =\{a_1,a_2,\dots,a_i\}$，且对于 $1\le n\le i$，已知$t(a_n)$，对每一个$a_n\in\ A_i$，求一点：

$$b_n{}^{(i)}\in N(a_n)-A_i=B_n{}^{(i)},$$

使得：$l(a_nb_n{}^{(i)})=\min\limits_{v\in B_n{}^{(i)}}l(a_nv)$

（3）设有m_i，$1\le m_i\le i$，而$b_{m_i}{}^{(i)}$是使$t(a_{m_i})+l(a_{m_i}b_{m_i}{}^{(i)})$取最小值，令：$a_{i+1}=b_{m_i}{}^{(i)}$，$t(a_{i+1})=t\left(a_{m_i}\right)+l\left(a_{m_i}a_{i+1}\right)$，$T_{i+1}=T_i\cup\left\{a_{m_i}a_{i+1}\right\}$.

（4）若$a_{i+1}=b$，停止；否则，记$A_{i+1}=A_i\cup\{a_{i+1}\}$，转(2).

时间复杂性分析：

对第i次循环：步骤(2)要进行i次比较运算，步骤(3)要进行i次加法与i次比较运算。所以，该次循环运算量为$3i$. 所以，在最坏的情况下，运算量为n^2级，是好算法。

算法证明：

定理1：算法中的函数$t(a_i)$给出了a与a_i的距离。

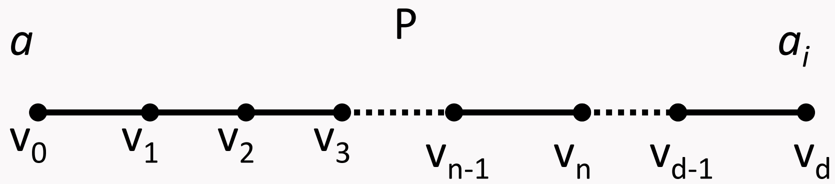
证明：对i作数学归纳法。

(1) $i = 1$时结论显然成立。

(2) 设对所有的j，$1 \leq j < i$时，$t(a_j) = d(a, a_j)$.

(3) 考虑$j = i$

令$P = v_0 v_1 \ldots v_d, v_0 = a, v_d = a_i$是连接$a$与$a_i$的一条最短路.



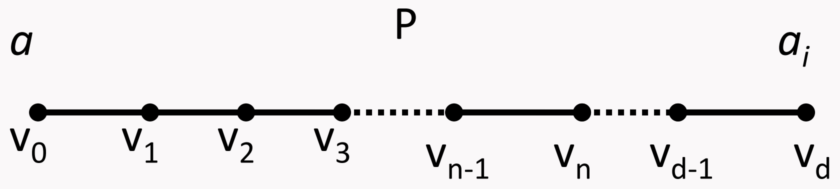
于是$d = d(a, a_i)$.

又令v_n是P中第一个不在A_{i-1}中的点。由于$a_i \notin A_{i-1}$, 所以这样的点存在。

因为$v_0 \in A_{i-1}$，所以：$n \geq 1$.

记P中a到v_n一段长为l, 而a到v_{n-1}的一段长为 l_1.

由归纳假设有：$l_1 \geq t(v_{n-1})$.



显然有：

$$d = d(a, a_i) \geq l = l_1 + l(v_{n-1}v_n) \\ \geq t(v_{n-1}) + l(v_{n-1}v_n)$$

由算法：当已知$A_{i-1} = \{a_1, a_2, \cdots, a_{i-1}\}$, 而要给$a_i$标号时，其中要选择${b_{n-1}}^{(i-1)}$，由选择知：$l\left(v_{n-1}{b_{n-1}}^{(i-1)}\right) \leq l(v_{n-1}v_n)$,

所以：

$$d = d(a, a_i) \geq l = l_1 + l(v_{n-1}v_n) \\ \geq t(v_{n-1}) + l(v_{n-1}v_n) \\ \geq t(v_{n-1}) + l(v_{n-1}{b_{n-1}}^{(i-1)})$$

又由算法最终对点a_i的标号值的选择方法知：

$$\begin{aligned} d &\geq t(v_{n-1}) + l\left(v_{n-1}, {b_{n-1}}^{(i-1)}\right) \\ &\geq t\left(a_{m_{i-1}}\right) + l\left(a_{m_{i-1}} a_i\right) \geq t(a_i). \end{aligned}$$

另一方面：由算法知，存在一条长度为$t(a_i)$的联结a与a_i的路，所以

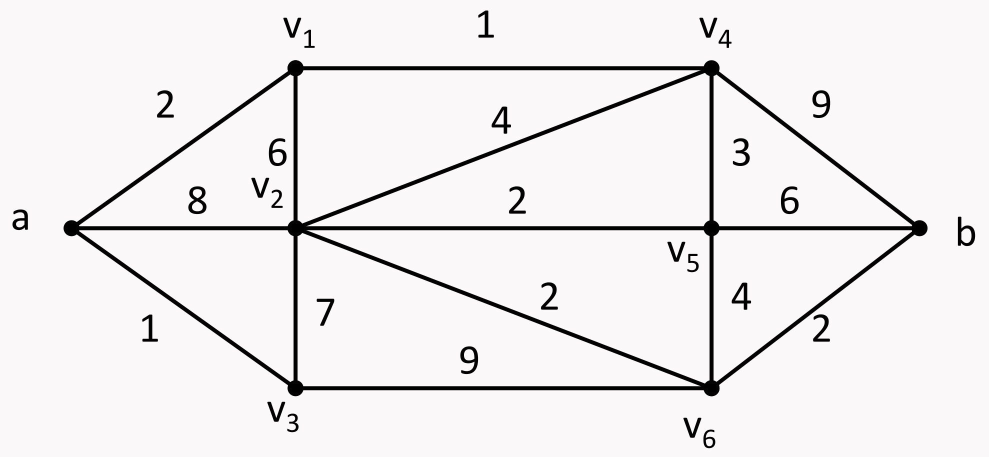
$$t(a_i) \geq d(a, a_i).$$

这样，我们证明了：

$$t(a_i) = d(a, a_i).$$

故，算法是正确的。

例1　如图所示，求点a到点b的距离。



解：

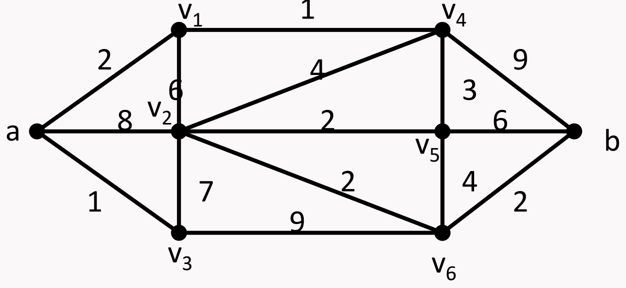
1. $A_1 = \{a\}$，$t(a) = 0$，$T_1 = \Phi$;

2. $b_1^{(1)} = v_3$;

3. $m_1 = 1, a_2 = v_3$,

$t(v_3) = t(a) + l(av_3) = 1$ (最小),

$T_2 = \{av_3\}$;



2. $A_2=\{a,v_3\}$, $b_1^{(2)}=v_1,b_2^{(2)}=v_2$;

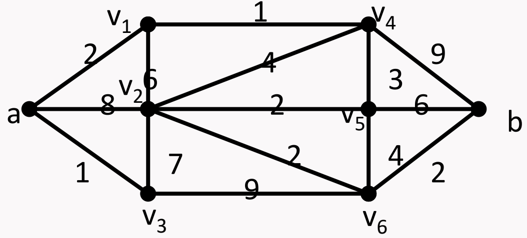
3. $m_2=1,a_3=v_1,t(v_1)=t(a)+l(av_1)=2$（最小），$T_3=\{av_3,av_1\}$;

2. $A_3=\{a,v_3,v_1\},b_1^{(3)}=v_2,b_2^{(3)}=v_2,b_3^{(3)}=v_4$;

3. $m_3=3,a_4=v_4,t(v_4)=t(v_1)+l(v_1v_4)=3$（最小），$T_4=\{av_3,av_1,v_1v_4\}$;

2. $A_4=\{a,v_3,v_1,v_4\},b_1^{(4)}=v_2,b_2^{(4)}=v_2,b_3^{(4)}=v_2$, $b_4^{(4)}=v_5$;

3. $m_4=4,a_5=v_5,t(v_5)=t(v_4)+l(v_4v_5)=6$（最小），$T_5=\{av_3,av_1,v_1v_4,v_4v_5\}$;



2\. $A_5 = \{a, v_3, v_1, v_4, v_5\}, b_1^{(5)} = v_2, b_2^{(5)} = v_2,$
$b_3^{(5)} = v_2, b_4^{(5)} = v_2, b_5^{(5)} = v_2$;

3\. $m_5 = 4, a_6 = v_2, t(v_2) = t(v_4) + l(v_4v_2) = 7$（最小）, $T_6 = \{av_3, av_1, v_1v_4, v_4v_5, v_4v_2\}$;

2\. $A_6 = \{a, v_3, v_1, v_4, v_5, v_2\}, b_2^{(6)} = v_6, b_4^{(6)} = b, b_5^{(6)} = v_6, b_6^{(6)} = v_6$;

3\. $m_6 = 6, a_7 = v_6, t(v_6) = t(v_2) + l(v_2v_6) = 9$（最小）, $T_7 = \{av_3, av_1, v_1v_4, v_4v_5, v_4v_2, v_2v_6\}$;

2\. $A_7 = \{a, v_3, v_1, v_4, v_5, v_2, v_6\}, b_4^{(7)} = b, b_5^{(7)} = b, b_7^{(7)} = b$;

3\. $m_7 = 7, a_8 = b, t(b) = t(v_6) + l(v_6b) = 11$（最小）, $T_8 = \{av_3, av_1, v_1v_4, v_4v_5, v_4v_2, v_2v_6, v_6b\}$;

于是知a与b的距离

$$d\,(a,b)=t(b)=11.$$

由T_8导出的a到b的最短路为：$av_1v_4v_2v_6b$.

课外作业

某公司在六个城市C_1,C_2,C_3,C_4,C_5,C_6中有分公司，从C_i到C_j的直接航程票价记在下述矩阵的(i,j)位置上，∞表示没有直接航程。制作一张任意两城市间的最便宜的路线表。

$$\begin{pmatrix} 0 & 50 & \infty & 40 & 25 & 10 \\ 50 & 0 & 15 & 20 & \infty & 25 \\ \infty & 15 & 0 & 10 & 20 & \infty \\ 40 & 20 & 10 & 0 & 10 & 25 \\ 25 & \infty & 20 & 10 & 0 & 55 \\ 10 & 25 & \infty & 25 & 55 & 0 \end{pmatrix}$$

例2 某两人有一只8升的酒壶装满了酒，还有两只空壶，分别为5升和3升。求最少的操作次数能均分酒。

解：设$x1, x2, x3$分别表示8, 5, 3升酒壶中的酒量。则

$$x_1 + x_2 + x_3 = 8, x_1 \le 8, x_2 \le 5, x_3 \le 3.$$

容易算出(x_1, x_2, x_3) 的组合形式共24种。

每种组合用一个点表示，两点连线，当且仅当可通过倒酒的方式相互变换。

若各边赋权为1，则问题转化为在该图中求(8,0,0)到(4,4,0)的一条最短路。结果如下：

$$(8,0,0) \to (3,5,0) \to (3,2,3) \to (6,2,0) \to (6,0,2) \to (1,5,2) \to (1,4,3) \to (4,4,0).$$

例3 在一河岸有狼，羊和卷心菜。摆渡人要将它们渡过河去，由于船太小，每次只能载一样东西。由于狼羊，羊卷心菜不能单独相处。问摆渡人至少要多少次才能将其渡过河？

分析：人，狼，羊，菜所有组合形式为：

$$C_4^0+C_4^1+C_4^2+C_4^3+C_4^4=2^4=16.$$

但是以下组合不能允许出现：

狼羊菜，羊菜，狼羊，人，人狼，人菜，共6种。

岸上只能允许出现10种组合：

人狼羊菜，人狼羊，人狼菜，人羊，空，菜，羊，狼，狼菜，人羊菜

每种情况用点表示；

两点连线，当且仅当两种情况可用载人(或加一物)的渡船相互转变。

每条边赋权为1

于是，问题转化为求由顶点“人狼羊菜”到顶点“空”的一条最短路

结果为：

人狼羊菜→狼菜→人狼菜→狼→人狼羊→羊→人羊→空；

(2) 人狼羊菜→狼菜→人狼菜→菜→人羊菜→羊→人羊→空。

(二)、图的代数表示

用邻接矩阵或关联矩阵表示图，称为图的代数表示。用矩阵表示图，主要有两个优点：

（1）能够把图输入到计算机中；

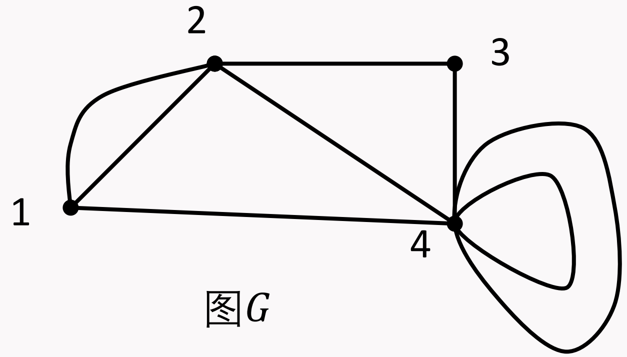
（2）可以用代数方法研究图论。

1、图的邻接矩阵

定义1 设G为n阶图，$V=\{v_1, v_2, \dots, v_n\}$, **邻接矩阵**$A(G)=\left(a_{ij}\right)$, 其中:

$$a_{ij}=\begin{cases} l, v_i\text{与}v_j\text{间边数} \\ 0, v_i\text{和}v_j\text{不邻接} \end{cases}.$$

例如：写出下图G的邻接矩阵：



图G

解：邻接矩阵为：

$$A(G)=\begin{pmatrix} 0 & 2 & 0 & 1 \\ 2 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 2 \end{pmatrix}$$

图G的邻接矩阵的性质

(1)非负性与对称性

由邻接矩阵定义知a_{ij}是非负整数，即邻接矩阵是非负整数矩阵；

在图中点v_i与v_j邻接，有v_i与v_j邻接，即$a_{ij} = a_{ji}$.所以，邻接矩阵是对称矩阵。

(2) 同一图的不同形式的邻接矩阵是相似矩阵。

这是因为，同图的两种不同形式矩阵可以通过交换行和相应的列变成一致。

(3)如果G为简单图，则$A(G)$为布尔矩阵；行和(列和)等于对应顶点的度数；矩阵元素总和为图的总度数，也就是G的边数的2倍。

（4）G连通的充分必要条件是：$A(G)$不能与如下矩阵相似

$$\begin{pmatrix} A_{11} & 0 \\ 0 & A_{22} \end{pmatrix}.$$

证明：1）必要性

若不然：设A_{11}对应的顶点是$v_1, v_2, \dots, v_k$，A_{22}对应的顶点为$v_{k+1}, v_{k+2}, \dots, v_n$.

显然，$v_i\ (1 \le i \le k)$与$v_j\ (k+1 \le j \le n)$不邻接，即G是非连通图。矛盾！

2）充分性

若不然：设G_1与G_2是G的两个不连通的部分，并且设

$$V(G_1) = \{v_1, v_2, \dots, v_k\}, V(G_2) = \{v_{k+1}, v_{k+2}, \dots, v_n\},$$

如果在写G的邻接矩阵时，先排$V(G_1)$中点，再排$V(G_2)$中点，则G的邻接矩阵形式必为：

$$\begin{pmatrix} A_{11} & 0 \\ 0 & A_{22} \end{pmatrix}.$$

（5）定理：设$A^k(G)=\left(a_{ij}^{(k)}\right)$，则$a_{ij}^{(k)}$表示顶点$v_i$到顶点$v_j$的途径长度为$k$的途径条数。

证明：对k作数学归纳法证明。

当$k=1$时，由邻接矩阵的定义，结论成立；

设结论对$k-1$时成立。当为k时：

一方面：先计算A^k；

$$A^k = A^{k-1} \cdot A = \left(a_{i1}^{(k-1)} a_{j1} + a_{i2}^{(k-1)} a_{j2} + \cdots + a_{in}^{(k-1)} a_{jn}\right)_{n \times n}$$

另一方面：考虑v_i到v_j的长度为k的途径

设v_m是v_i到v_j的途径中点，且该点和v_j邻接。则v_i到v_j的经过v_m且长度为k的途径数目应该为：

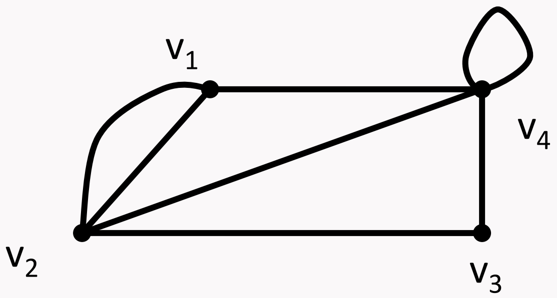
$$a_{im}^{(k-1)} a_{mj}.$$

所以，v_i到v_j的长度为k的途径数目为：

$$a_{i1}^{(k-1)} a_{j1} + a_{i2}^{(k-1)} a_{j2} + \cdots + a_{in}^{(k-1)} a_{jn}.$$

即为$a_{ij}^{(k)}$.

例4 求下图中v_1到v_3的途径长度为2和3的条数。



解：由于：

$$A(G)=\begin{pmatrix}0&2&0&1\\2&0&1&1\\0&1&0&1\\1&1&1&1\end{pmatrix}\quad A^2(G)=\begin{pmatrix}5&1&3&3\\1&6&1&4\\3&1&2&2\\3&4&2&4\end{pmatrix}\quad A^3(G)=\begin{pmatrix}5&16&4&12\\16&7&10&12\\4&10&3&8\\12&12&8&13\end{pmatrix}$$

所以，v_1到v_3的途径长度为2和3的条数分别为：3和4。

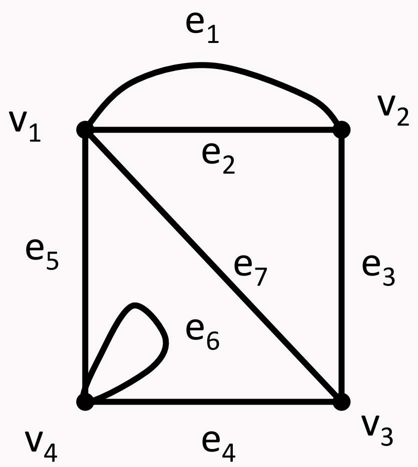
推论：(1)设G是简单图， A^2的元素$a_{ii}^{(2)}$是v_i的度数， A^3的元素$a_{ii}^{(3)}$是含v_i的三角形个数的2倍.

(2) 若G是连通的，对于$i \neq j, v_i$和v_j间距离是使A^n的$a_{ij}^{(n)} \neq 0$的最小整数。

2、图的关联矩阵

（1）若G是 (n,m)图。定义G的关联矩阵：$M(G) = (a_{ij})_{n\times m}$, 其中：$a_{ij} = l, v_i$与$e_j$关联的次数(0, 1, 或2(环）).

例如：



$$M(G)=\begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 2 & 0 \end{pmatrix}$$

(2) 关联矩阵的性质

1)关联矩阵的元素为0，1或2;

2)关联矩阵的每列和为2；每行的和为对应顶点度数;

3) 无环图G连通的充分必要条件是$R(M)=n-1$.

证明："$\Rightarrow$"

令：

$$\boldsymbol{M}=\begin{pmatrix} \boldsymbol{m}_1 \\ \boldsymbol{m}_2 \\ \vdots \\ \boldsymbol{m}_n \end{pmatrix}$$

由于M中每列恰有两个“1”元素，所有行向量的和为0(模2加法运算)。所以：

$$R(M)\le n-1.$$

另一方面：在M中任意去掉一行m_k, 由于G是连通的，因此，m_k中存在元素“1”， 这样，M中去掉行m_k，后的行按模2相加不等于零，即它们是线性无关的。所以：$R(M) \geq n-1.$

因此，$R(M) = n-1.$

$" \Leftarrow "$

若G不连通，假设G有两个连通分支G_1与G_2, 又设G_1与G_2的关联矩阵分别为M_1与M_2, 则G的关联矩阵可以写为:

$$M(G)=\begin{pmatrix} M_1 & \\ & M_2 \end{pmatrix}$$

于是，$R(M) = V(G_1) - 1 + V(G_2) - 1 = n - 2$, 矛盾!

定义：在G的关联矩阵中删掉任意一行后得到的矩阵可以完全决定G，该矩阵称为G的基本关联矩阵。删掉的行对应的顶点称为该基本关联矩阵的参考点。

说明:（1）图的关联矩阵及其性质是网络图论的基础，在电路分析中有重要应用；在第二章中再作一些相关介绍。

（2）图的关联矩阵比邻接矩阵大得多，不便于计算机存储。但二者都有各自的应用特点。

谢 谢！