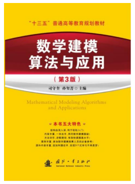


数学建模算法与应用

第4章

4.1图与网络的基本理论

4.2MATLAB工具箱

数学建模算法与应用

目录 CONTENTS

数学建模算法与应用

01 图与网络的基本理论

4.1.1 图与网络的基本概念

1.无向图和有向图

定义 4.1 一个无向图G是由非空顶点集V和边集E按一定的对应关系构成的连接结构，记为$G=(V,E)$。其中非空集合$V=\{v_1,v_2,\cdots,v_n\}$为G的顶点集，V中的元素称为G的顶点，其元素的个数为顶点数；集合$E=\{e_1,e_2,\cdots,e_m\}$为G的边集，E中的元素称为G的边，其元素的个数为图G的边数。

以下用$|V|$表示图$G=(V,E)$中顶点的个数，$|E|$表示边的条数。

图G的每一条边是由连接G中两个顶点而得的一条线(可以是直线或曲线)，因此与G的顶点对相对应，通常记作$e_k=(v_i,v_j)$，其中，顶点v_i,v_j称为边e_k的两个端点，有时也说边e_k与顶点v_i,v_j关联。

对无向图来说，对应一条边的顶点对表示是无序的，即(v_i,v_j)和(v_j,v_i)表示同一条边e_k。

有公共端点的两条边，或称邻边。同样，同一条边e_k的两个端点（v_i和v_j）称为是相邻的顶点。

带有方向的边称为有向边，又称为弧。

定义 4.2 有向图通常记为$D=(V,A)$，其中非空集合$V=\{v_1,v_2,\cdots,v_n\}$为D的顶点集，$A=\{a_1,a_2,\cdots,a_m\}$为D的弧集合，每一条弧与一个有序的顶点对相对应弧$a_k=(v_i,v_j)$表示弧的方向自顶点v_i指向v_j，v_i称为弧a_k的始端，v_j称为弧a_k的末端或终端，其中a_k称为v_i的出弧，称为v_j的入弧。

与无向图不同，在有向图情形下，(v_i,v_j)与(v_j,v_i)表示不同的弧。

把有向图$D=(V,A)$中所有弧的方向都去掉，得到的边集用E表示，就得到与有向图D对应的无向图$G=(V,E)$，称G为有向图D的基本图，称D为G的定向图。

例 4.1 设$V=\{v_1,v_2,v_3,v_4,v_5\}$，$E=\{e_1,e_2,e_3,e_4,e_5\}$，$e_1=(v_1,v_2)$，$e_2=(v_2,v_3)$，$e_3=(v_2,v_3)$，$e_4=(v_3,v_4)$，$e_5=(v_4,v_4)$. 则$G=(V,E)$是一个图，其图形如图 4.1 所示。

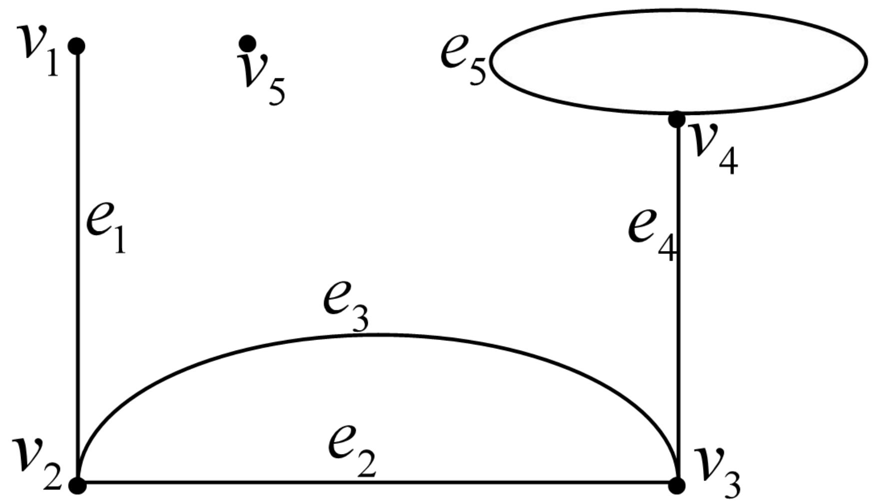


图 4.1 非简单图示例

2.简单图、完全图、赋权图

定义 4.3 如果一条边的两个端点是同一个顶点，则称这条边为环。如果有两条边或多条边的端点是同一对顶点，则称这些边为重边或平行边。称不与任何边相关联的顶点为孤立点。

图 4.1 中，边e_2和e_3为重边，e_5为环，顶点v_5为孤立点。

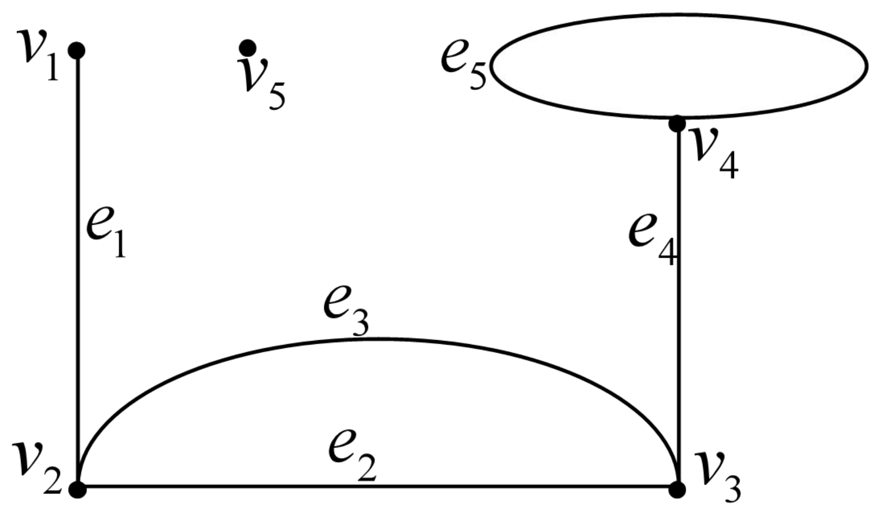


图 4.1 非简单图示例

定义 4.4 无环且无重边的图称为简单图。

如果不特别申明，一般的图均指简单图。

定义 4.5 任意两顶点均相邻的简单图称为完全图。含n个顶点的完全图记为K_n。

定义 4.6 如果图G的每条边e都附有一个实数$w(e)$，则称图G为赋权图，实数$w(e)$称为边e的权。

赋权图也称为网络。赋权图中的权可以是距离、费用、时间、效益、成本等。赋权图G一般记作$G=(V,E,W)$，其中W为权重的邻接矩阵。赋权图也可以记作$N=(V,E,W)$。

如果有向图D的每条弧都被赋予了权，则称D为有向赋权图。以后对于无向图、有向图或网络都可以用G表示，从下文中就能够区分出无向的还是有向的，赋权的还是非赋权的。

3.定点的度

定义 4.7 （1）在无向图中，与顶点v关联的边的数目（环算两次）称为v的度，记为$d(v)$。

（2）在有向图中，从顶点v引出的弧的数目称为v的出度，记为$d^+(v)$，从顶点v引入的弧的数目称为v的入度，记为$d^-(v)$，$d(v)=d^+(v)+d^-(v)$称为v的度。

度为奇数的顶点称为奇顶点，度为偶数的顶点称为偶顶点。

定理 4.1 给定图$G=(V,E)$，所有顶点的度数之和是边数的 2 倍，即

$$\sum_{v\in V} d(v)=2|E|.$$

推论 4.1 任何图中奇顶点的总数必为偶数。

4.子图与图的连通性

定义 4.8 设$G_1=(V_1,E_1)$与$G_2=(V_2,E_2)$是两个图并且满足$V_1\subset V_2$，$E_1\subset E_2$，则称G_1是G_2的子图，G_2称为G_1的母图。如G_1是G_2的子图，且$V_1=V_2$，则称G_1是G_2的生成子图（支撑子图）。

定义 4.9 设$W = v_0e_1v_1e_2\cdots e_kv_k$，其中

$e_i \in E(i=1,2,\cdots,k)$，$v_j \in V(j=0,1,\cdots,k)$，

e_i与v_{i-1}和v_i关联，称W是图G的一条**道路**（walk），简称**路**，k为**路长**，v_0为起点，v_k为终点；

各边相异的道路称为**迹**（trail）；

各顶点相异的道路称为**轨道**（path），记为$P(v_0,v_k)$

起点和终点重合的道路称为**回路**；

起点和终点重合的轨道称为**圈**，即对轨道$P(v_0,v_k)$当$v_0=v_k$时成为一个圈。

称以两顶点u,v分别为起点和终点的最短轨道之长为顶点u,v的**距离**。

定义 4.10 在无向图G中，如果从顶点u到顶点v存在道路，则称顶点u和v是连通的。如果图G中的任意两个顶点u和v都是连通的，则称图G是连通图，否则称为非连通图。非连通图中的连通子图，称为连通分支。

在有向图D中，如果对于任意两个顶点u和v，从u到v和从v到u都存在道路，则称图D是强连通图。

4.1.2图的矩阵表示

设图的顶点个数为n，边（或弧）的条数为m。

对于无向图$G=(V,E)$，其中$V=\{v_1,v_2,\cdots,v_n\}$，$E=\{e_1,e_2,\cdots,e_m\}$。

对于有向图$D=(V,A)$，其中$V=\{v_1,v_2,\cdots,v_n\}$，$A=\{a_1,a_2,\cdots,a_m\}$。

1.关联矩阵

对于无向图G，其关联矩阵$M=(m_{ij})_{n\times m}$，其中

$$m_{ij}=\begin{cases}1, & \text{顶点}v_i\text{与边}e_j\text{关联},\\ 0, & \text{顶点}v_i\text{与边}e_j\text{不关联},\end{cases}$$

$i=1,2,\cdots,n$，$j=1,2,\cdots,m$.

对有向图G，其关联矩阵$M=(m_{ij})_{n\times m}$，其中

$$m_{ij}=\begin{cases}1, & \text{顶点}v_i\text{是弧}a_j\text{的始端},\\ -1, & \text{顶点}v_i\text{是弧}a_j\text{的末端},\\ 0, & \text{顶点}v_i\text{与弧}a_j\text{不关联},\end{cases}$$

$i=1,2,\cdots,n$，$j=1,2,\cdots,m$.

2.邻接矩阵

对无向非赋权图G，其邻接矩阵$W=(w_{ij})_{n\times n}$，其中

$$w_{ij}=\begin{cases}1, & \text{顶点}v_i\text{与}v_j\text{相邻},\\ 0, & i=j\text{或顶点}v_i\text{与}v_j\text{不相邻},\end{cases}\quad i,j=1,2,\cdots,n.$$

对有向非赋权图D，其邻接矩阵$W=(w_{ij})_{n\times n}$，其中

$$w_{ij}=\begin{cases}1, & 弧(v_i,v_j)\in A,\\ 0, & i=j或顶点v_i到v_j无弧,\end{cases}\quad i,j=1,2,\cdots,n.$$

对无向赋权图G，其邻接矩阵$W=(w_{ij})_{n\times n}$，其中

$$w_{ij}=\begin{cases}\text{顶点}v_i\text{与}v_j\text{之间边的权，}(v_i,v_j)\in E,\\ 0(\text{或}\infty), \qquad v_i\text{与}v_j\text{之间无边,}\end{cases}$$

$i,j=1,2,\cdots,n.$

例 4.2 写出图 4.2 所示的无向图的邻接矩阵。

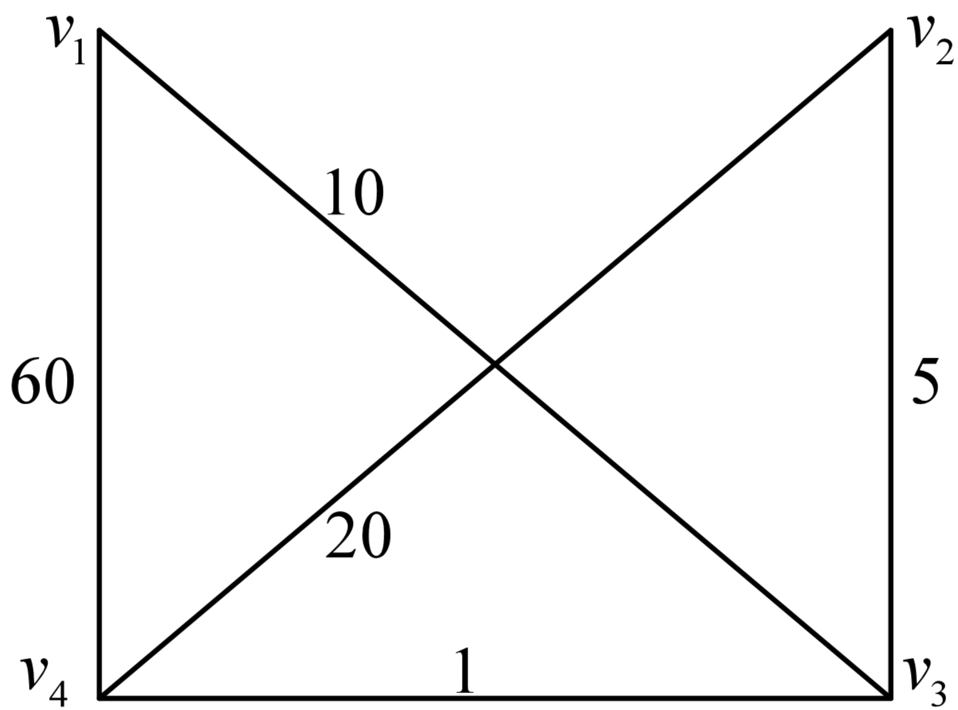


图 4.2 赋权无向图

图 4.2 所示的无向图，其邻接矩阵

$$W=\begin{bmatrix} 0 & 0 & 10 & 60 \\ 0 & 0 & 5 & 20 \\ 10 & 5 & 0 & 1 \\ 60 & 20 & 1 & 0 \end{bmatrix}.$$

写出图 4.3 所示的有向图的邻接矩阵。

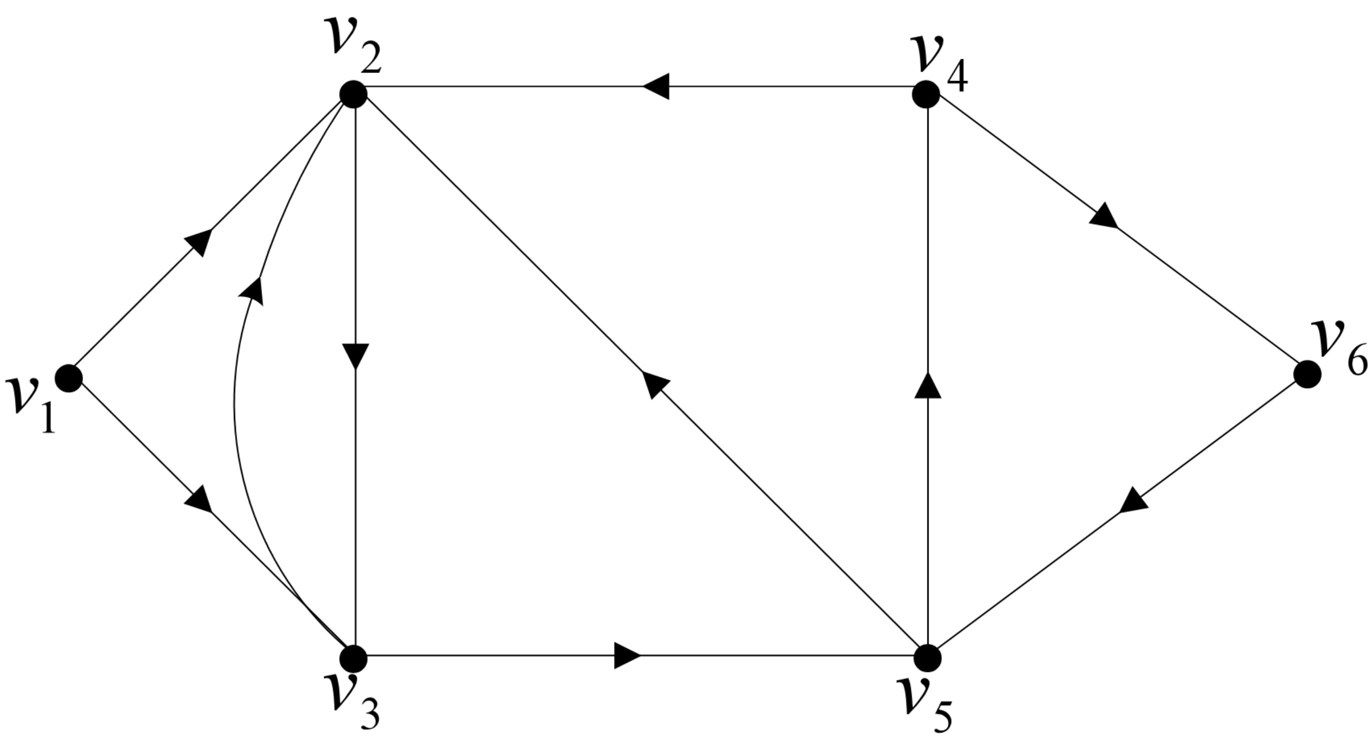
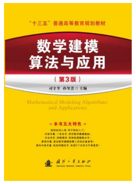


图 4.3 非赋权有向图

图 4.3 所示的有向图的邻接矩阵

$$W=\begin{bmatrix}0&1&1&0&0&0\\0&0&1&0&0&0\\0&1&0&0&1&0\\0&1&0&0&0&1\\0&1&0&1&0&0\\0&0&0&0&1&0\end{bmatrix}.$$



02 MATLAB 工具箱简介

1.图的生成

Graph：无向图（undirected Graph)；

Digraph：有向图（directed Graph)；

G = graph——创建空的无向图对象。

G = graph (A)——使用邻接矩阵 A 创建赋权无向图。

G = graph(A,nodes) ——使用邻接矩阵 A 和节点名称 nodes 创建赋权无向图。

G = graph(s,t) ——使用节点对组 s,t 创建无向图。

G = graph(s,t,weights) ——使用节点对组 s,t 和权重向量 weights 创建赋权无向图。

G = graph(s,t,wei ghts,nodes)——使用字符向量元胞数组 nodes 指定节点名称。

G = graph(s,t,weights,num) ——使用数值标量 num 指定图中的节点数。

G = graph(A[,nodes],type) ——仅使用 A 的上或下三角形阵构造赋权图，type 可以是'upper' 或 'lower'。

例 4.3 画出一个 60 个顶点的 3 正则图（每个顶点的度都为 3）。

```
clc, close all
G = graph(bucky); plot(G)
```

所画的图形如图 4.4 所示。

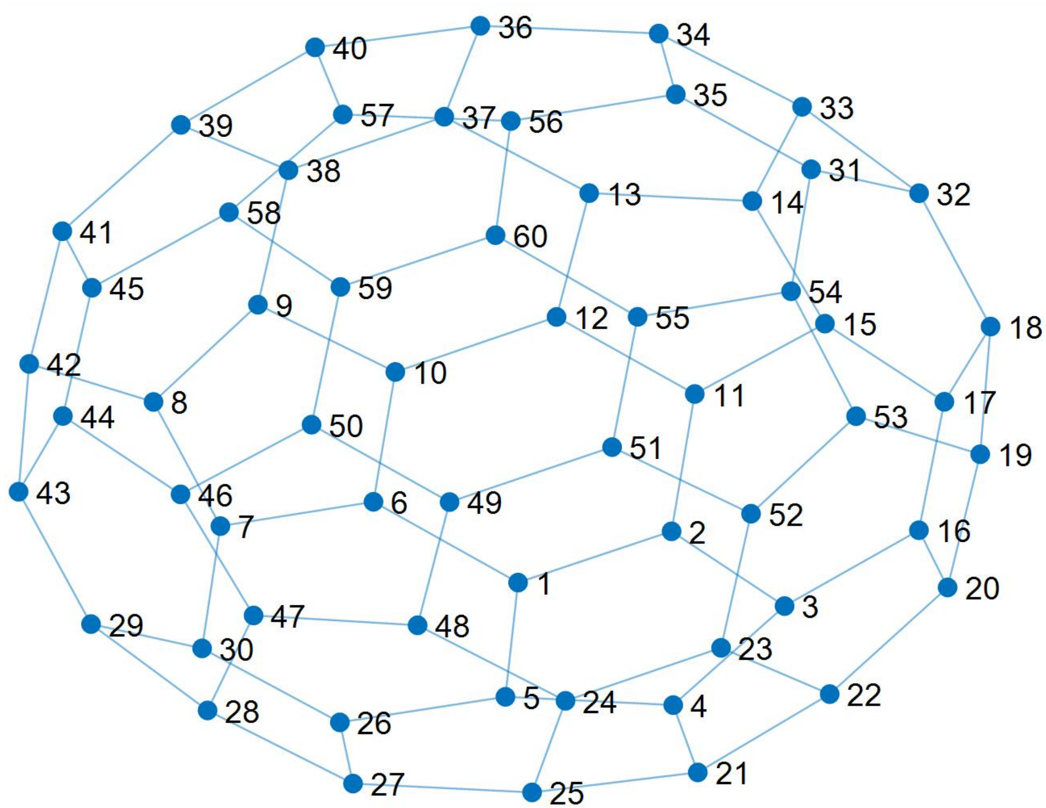


图 4.4 3 正则图

2.数据存储结构

MATLAB 存储网络的相关数据时，使用了稀疏矩阵，这有利于在存储大规模稀疏网络时节省存储空间。

例 4.4（续例 4.2） 画出图 4.3 的非赋权有向图并导出邻接矩阵和关联矩阵。

```
clc, clear, close all
E=[1,2;1,3;2,3;3,2;3,5;4,2;4,6;5,2;5,4;6,5]
s = E(:,1); t = E(:,2);
nodes =cellstr(strcat('v',int2str([1:6]')))
G = digraph(s, t, [], nodes);
plot(G,'LineWidth',1.5,'Layout','circle','NodeFontSize',15)
W1 = adjacency(G)  %导出邻接矩阵的稀疏矩阵
W2 = incidence(G)  %导出关联矩阵的稀疏矩阵
```

所画的图如图 4.5 所示。

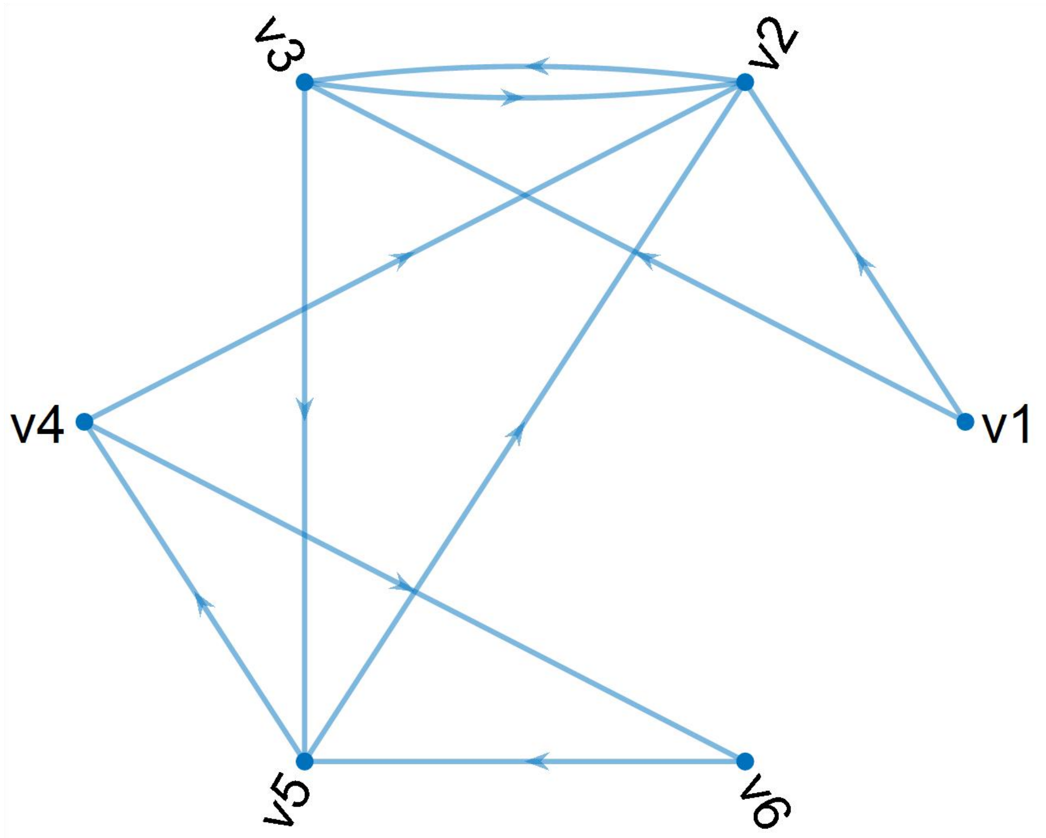


图 4.5 MATLAB 画的有向图

例 4.5（续例 4.2）导出图 4.2 所示赋权图的邻接矩阵和关联矩阵，并重新画图。

```
clc, clear, close all
E = [1, 3, 10; 1, 4, 60; 2, 3, 5; 2, 4, 20; 3, 4, 1];
G = graph(E(:,1), E(:,2), E(:,3));
W1 = adjacency(G,'weighted'), W2 = incidence(G)
plot(G,'Layout','force','EdgeLabel',G.Edges.Weight)
```