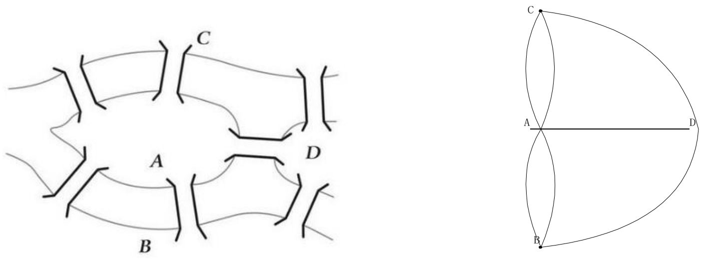
第五章 图与网络优化

教学内容	第一节 图论
教学目标	**知识目标：** 理解图的基本概念及其矩阵表示。 **能力目标：**通过实例演示、问题驱动等方式激发学生学习的积极性，通过观察对比、学生交流、师生交流、小组探究等多种形式， 培养学生的逻辑思维能力、分析判断能力、和解决问题能力。 **素质目标：** 培养学生敢于质疑、善于分析、勇于创新的精神；培养学生的自主学习意识和团队协作精神；培养学生脚踏实地、不畏艰辛、锲而不舍的精神。
授课学时	2 学时
教学重点	图的基本概念及矩阵表示
教学难点	图的矩阵表示。
教学方法	采用问题驱动法、案例演示法、启发式讲授法及自主学习法相结合，以教师的讲解为主，学生的课堂报告、分组讨论为辅，充分调动学生学习的主动性和思考问题的积极性。
教学手段	以课堂讲授为主，主要是多媒体课件和板书相结合的形式。同时借助在线资源，如慕课、雨课堂等平台。
教学过程	**（一）通过回顾“七桥问题”，引出图的概念（约 20 分钟）** 【教师活动】通过有趣的“七桥问题”，引入图的定义。 回顾“七桥问题”。在18 世纪，东普鲁士哥尼斯堡有一条大河，河中有两个小岛。全城被大河分割成四块陆地，河上架有七座桥， 把四块陆地联系起来。当时许多市民都在思索一个问题：一个散步者能否从某一陆地出发，不重复地经过每座桥一次，最后回到原来的出发地。这就是历史上有名的哥尼斯堡七桥问题。这个问题似乎不难解决，所以吸引了许多人来尝试，但是日复一日谁也

没有得出肯定的答案。于是有人便写信求教当时著名的数学家欧拉(1707~1783)。欧拉毕竟是数学家，他并没有去重复人们已失败了多次的试验，而是产生了一种直觉的猜想：人们千百次的失败，也许意味着这样的走法根本就不存在。于是欧拉把七桥问题进行了数学的抽象。用 A、B、C、D 四个点表示四块陆地（图5.1(a)）用两点间的一条线表示连接两块陆地之间的一座桥，就得到如下图所示的一个由四个点和七条线组成的图形（图5.1（b）。



(a) (b)

图5.1 尼斯堡七桥问题

【学生活动】自由讨论如何将“七桥问题”转化为数学问题，并随机选择学生讲解自己的想法。

【设计意图】由学生熟悉的内容引入新课，并且适当的分配一部分课堂时间给学生，有利于提高学生的参与感，引发学生的认知热情，提高探究新知的兴趣。

（二）在前期引入的基础上，介绍图的基本概念及其矩阵表示（约 40 分钟）

【教师活动】通过总结与分析课堂引入的内容，给出图的概念。

知识点 1: 图的定义。

构成一个图有两个关键要素，即顶点和连接顶点之间的边，记为 $G=(V,E)$，V 是以上述点为元素的顶点集，E 是以上述连线为元素的边集。各条边都加上方向的图称为有向图，否则称为无向图。如果有的边有方向，有的边无方向，则称为混合图。

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，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； 非空集合 $E=\{e_1,e_2,\cdots,e_m\}$为 G 的边集， E 中的元素称G 的边，其元素的个数为图G 的边数。

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、曲线或任意形状的线）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称为邻边。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。

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，则称这些边为平行边或重边。没有环也没有平行边的图称为简单图。

带有方向的边称为有向边，又称为弧。如果给图的每条边规定一个方向，我们就得到有向图。有向图通常记为 $D=(V,A)$ 。其中 A 是图中弧的集合，每一条弧与一个有序的顶点对相对应。与无向图类似,也记为$a_k=(v_i,v_j)$ 表示边的方向自顶点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。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。平面上，有向图的图形表示方法与无向图基本一样，只需在每一条弧的末端加一个箭头方向就可以了。

若在图G 中的各边e_k 都赋值一个实数$w(e_k)$ ，称为边e_k 的权，把每条边赋以权值的图称为赋权图， 也称为一个网络，记为$G=(V,E,W)$ ，其中W 为所有边的权集合。若在有向图上给每条弧赋以权值，称其为有向赋权图，也叫有向网络。

知识点 2: 关联矩阵。

	对无向图G，其关联矩阵$M=(m_{ij})_{d\times\varepsilon}$，其中 $$m_{ij}=\begin{cases}1, & 若v_i与e_j相关联\\0, & 若v_i与e_j不关联\end{cases}$$ 对有向图G，其关联矩阵$M=(m_{ij})_{d\times\varepsilon}$，其中 $$m_{ij}=\begin{cases}1, & 若v_i是e_j的起点\\-1, & 若v_i是e_j的终点\\0, & 若v_i与e_j不关联\end{cases}$$ 邻接矩阵。 对无向图G，其邻接矩阵$A=(a_{ij})_{d\times d}$，其中 $$a_{ij}=\begin{cases}1, & 若v_i与v_j相邻\\0, & 若v_i与v_j不相邻\end{cases}$$ 对有向图$G=(V,E)$，其邻接矩阵$A=(a_{ij})_{d\times d}$，其中 $$a_{ij}=\begin{cases}1, & 若(v_i,v_j)\in E\\0, & 若(v_i,v_j)\notin E\end{cases}$$ 对有向赋权图G，其邻接矩阵$A=(a_{ij})_{d\times d}$，其中 $$a_{ij}=\begin{cases}w_{ij}, & 若(v_i,v_j)\in E, 且w_{ij}为其权\\0, & 若i=j\\\infty, & 若(v_i,v_j)\notin E\end{cases}$$ 无向赋权图的邻接矩阵可类似定义。 【学生活动】听讲，有问题可发言提问。 **（三）采用讲授法，问题驱动法，介绍 Matlab 软件的相关命令。（约 25 分钟）** **（四）课堂总结（5 分钟）** 总结本次课的教学内容： （1）：图的定义及其矩阵表示； （2）：Matlab 软件的相关命令。
课后作业	课下寻找与图有关的数学建模案例，并进行分析。

教学内容	第三节 最短路问题
教学目标	**知识目标：** 了解最短路的概念；掌握最短路的算法。 **能力目标：**通过实例演示、问题驱动等方式激发学生学习的积极性，通过观察对比、学生交流、师生交流、小组探究等多种形式， 培养学生的逻辑思维能力、分析判断能力、和解决问题能力。 **素质目标：** 培养学生敢于质疑、善于分析、勇于创新的精神；培养学生的自主学习意识和团队协作精神；培养学生脚踏实地、不畏艰辛、锲而不舍的精神。
授课学时	2 学时
教学重点	最短路的算法
教学难点	最短路的步骤
教学方法	采用问题驱动法、案例演示法、启发式讲授法及自主学习法相结合，以教师的讲解为主，学生的课堂报告、分组讨论为辅，充分调动学生学习的主动性和思考问题的积极性。
教学手段	以课堂讲授为主，主要是多媒体课件和板书相结合的形式。同时借助在线资源，如慕课、雨课堂等平台。
教学过程	**（一）采用讲授法，介绍最短路的概念（约 15 分钟）** 最短路问题（short-path problem）是网络理论解决的典型问题之一，旨在寻找图中两顶点之间的最短距离。实际应用中的许多优化问题，如管路铺设、线路安排、厂区布局和设备更新等，都可以被归结为最短路问题来解决。 知识点 1：设图G 是赋权图， Γ 为G 中的一条路，则称Γ 的各边权之和为Γ 的长度。 知识点 2：对于图$G=(V,E)$ ，任意两点均有路径的图称为连通图；起点与终点重合的路径称为圈；连通而无圈的图称为树；只有顶点没有边的图称为空图。 知识点 3：设$P(u,v)$ 是赋权图 G 从u 到v 的路径，则称

	$w(P)=\sum_{e\in E(P)} w(e)$ 为路径 P 的权。在赋权图 G 中，从顶点 u 到 v 的具有最小权的路 $P^*(u,v)$，称为 u 到 v 的最短路。最短路的长度称为从 u 到 v 的距离，记为 $d(u,v)$。 **（二）采用案例启发法、讲授法，介绍固定起点的最短路算法。（约 30 分钟）** 寻求从固定起点 u_0 到其余各点的最短路的最有效的算法之一是 E. W. Dijkstra 于1959 年提出的 Dijkstra（迪杰斯特拉）算法。该算法可以用于求解图中某一特定点到其他各顶点的最短路问题 （单源最短路问题），也可以求解任意两个指定顶点间的最短路问题。 知识点 4：设 G 为赋权有向图或无向图，G 边上的权均非负。Dijkstra 算法的基本思想是按其距离固定起点 u_0 从近到远为顺序，依次求得 u_0 到 G 的各顶点的最短路和距离，直至 v_0（或直至 G 的所有顶点）。 用 S 表示具有永久标号的顶点集。对每个顶点，定义两个标号 $(l(v),z(v))$，其中， $l(v)$：顶点 v 的标号，表示从起点 u_0 到 v 当前路的长度。 $z(v)$：顶点 v 的父节点标号，用以确定最短路的路线。 算法的过程就是在每一步改进这两个标记，使最终 $l(v)$ 为从顶点 u_0 到 v 得到的最短路的权，输入为带权邻接矩阵 W。Dijkstra 算法的计算步骤如下： （1）赋初值：令 $S=\{u_0\}, l(u_0)=0. \forall v\in \bar{S}=V\setminus S$，令 $l(v)=W(u_0,v), z(v)=u_0, u\leftarrow u_0$。 （2）更新 $l(v), z(v): \forall v\in \bar{S}=V\setminus S$，若 $l(v)>l(u)+W(u,v)$，则令 $l(v)=l(u)+W(u,v)$，$z(v)=u$　　$\cup$ 3(　设 v^* 是使 $l(v)$ 取最小值的 $\bar{S}$ 中的顶点，则令 $S=S\{v^*\}, u\leftarrow v^*$。 4(　若 $\bar{S}\neq\varnothing$，转步骤（2）；否则，停止。

案例 1：求图 5.2 中从顶点u_0到其余顶点的最短路.

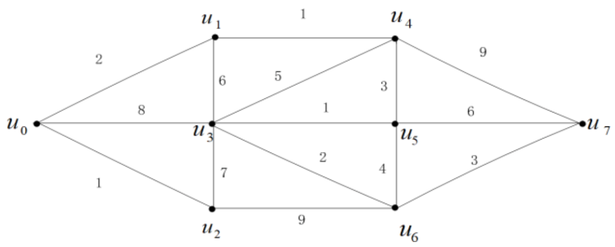


图5.2 某赋权图

解： 先写出带权邻接矩阵

$$W=\begin{bmatrix} 0 & 2 & 1 & 8 & \infty & \infty & \infty & \infty \\ & 0 & \infty & 6 & 1 & \infty & \infty & \infty \\ & & 0 & 7 & \infty & \infty & 9 & \infty \\ & & & 0 & 5 & 1 & 2 & \infty \\ & & & & 0 & 3 & \infty & 9 \\ & & & & & 0 & 4 & 6 \\ & & & & & & 0 & 3 \\ & & & & & & & 0 \end{bmatrix}$$

因G是无向图，故W是对称矩阵。

Dijkstra 算法步骤如下：

迭代次数	$l(u_i)$							
	u_0	u_1	u_2	u_3	u_4	u_5	u_6	u_7
1	$\boxed{0}$	∞	∞	∞	∞	∞	∞	∞
2		2	$\boxed{1}$	8	∞	∞	∞	∞
3		$\boxed{2}$		8	∞	∞	10	∞
4				8	$\boxed{3}$	∞	10	∞
5				8		$\boxed{6}$	10	12
6				$\boxed{7}$			10	12
7							$\boxed{9}$	12
8								$\boxed{12}$
最后标记 $l(v)$	0	2	1	7	3	6	9	12

$z(v)$	u_0	u_0	u_0	u_5	u_1	u_4	u_3	u_4

从第二个标记 $z(v)$ 向前追溯，即得以u_0为根的树（图 5.3）

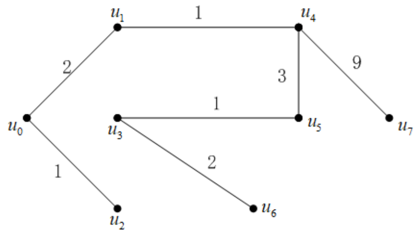


图 5.3 以u_0为根的树

（三）结合案例，讲授每对顶点之间的最短路的思想和算法（约 40 分钟）

【教师活动】结合多媒体课件和板书，由问题出发，采用启发式教学，建立模型。

知识点 5：求任意两顶点之间最短路的 Floyd 算法的基本思想为

直接在图的带权邻接矩阵中用插入顶点的方法依次构造出v 个矩阵$D^{(1)},D^{(2)},...,D^{(v)}$，使最后得到的矩阵 $D^{(v)}$ 成为图的距离矩阵，同时也求出插入点矩阵以便得到两点间的最短路径。

知识点 6：Floyd 算法原理。

（1）求距离矩阵的方法

把带权邻接矩阵W 作为距离矩阵的初值，即 $D^{(0)}=(d_{ij}^{(0)})_{v\times v}=W$ 。

1）$D^{(1)}=(d_{ij}^{(1)})_{v\times v}$，其中$d_{ij}^{(1)}=\min\left\{d_{ij}^{(0)},d_{i1}^{(0)}+d_{1j}^{(0)}\right\}$，$d_{ij}^{(1)}$ 是从v_i 到 v_j 的只允许以v_1 作为中间点的路径中最短路的长度。

2）$D^{(2)}=(d_{ij}^{(2)})_{v\times v}$，其中$d_{ij}^{(2)}=\min\left\{d_{ij}^{(1)},d_{i2}^{(1)}+d_{2j}^{(1)}\right\}$，$d_{ij}^{(2)}$ 是从v_i 到 v_j 的只允许v_1, v_2 作为中间点的路径中最短路的长度。

……

v）$D^{(v)}=(d_{ij}^{(v)})_{v\times v}$，其中$d_{ij}^{(v)}=\min\left\{d_{ij}^{(v-1)},d_{iv}^{(v-1)}+d_{vj}^{(v-1)}\right\}$，$d_{ij}^{(v)}$ 是从v_i 到 v_j 的只允许$v_1,v_2,...,v_v$ 作为中间点的路径中最短路的长度，即是从 v_i 到 v_j 中间可插入任何顶点的路径中最短路的长度，因此 $D^{(v)}$ 即是距离矩阵。

（2）求路径矩阵的方法

在建立距离矩阵的同时可建立路径矩阵 R ，$R=(r_{ij})_{v\times v}$，r_{ij} 的含义是从 v_i 到 v_j 的最短路要经过点号为 r_{ij} 的点，

$$R^{(0)}=(r_{ij}^{(0)})_{v\times v}, \quad r_{ij}^{(0)}=j$$

每求得一个 $D^{(k)}$ 时，按下列方式产生相应的新的 $R^{(k)}$：

$$r_{ij}^{(k)}=\begin{cases}k, & \text{若} d_{ij}^{(k-1)}>d_{ik}^{(k-1)}+d_{kj}^{(k-1)}\\ r_{ij}^{(k-1)}, & \text{否则}\end{cases}$$

即当 v_k 被插入任何两点间的最短路径时，被记录在 $R^{(k)}$ 中，依次求 $D^{(v)}$ 时求得 $R^{(v)}$，可由 $R^{(v)}$ 来查找任何点对之间的最短路的路径。

（3）查找最短路径的方法

若 $r_{ij}^{(v)}=p_1$，则点 p_1 是点 i 到点 j 的最短路的中间点，然后用同样的方法再分头查找若 ①向点 i 追溯得：$r_{ip_1}^{(v)}=p_2, r_{ip_2}^{(v)}=p_3, \ldots, r_{ip_k}^{(v)}=i$

②向点 j 追溯得：$r_{p_1 j}^{(v)}=q_1, r_{q_1 j}^{(v)}=q_2, \ldots, r_{q_m j}^{(v)}=j$

则由点 i 到 j 的最短路的路径为：$i, p_k, \ldots, p_2, p_1, q_1, q_2, \ldots, q_m, j$.

知识点 7：算法步骤。

$Floyd$ 算法：求任意两点间的最短路。

$D(i,j)$：i 到 j 的距离。

$R(i,j)$：i 到 j 之间的插入点。

输入带权邻接矩阵 W，

（1） 赋初值：对所有 i,j，$d(i,j)\leftarrow w(i,j)$，$r(i,j)\leftarrow j$，$k\leftarrow 1$。

（2） 更新 $d(i,j)$，$r(i,j)$：对所有 i,j，若 $d(i,k)+d(k,j)<d(i,j)$，则

$$d(i,j)\leftarrow d(i,k)+d(k,j), \quad r(i,j)\leftarrow k$$

（3） 若 $k=v$，停止；否则 $k\leftarrow k+1$，转（2）。

案例 2:（设备更新问题） 某种工程设备的役龄为 4 年，每年年初都面临着是否更新的问题：若卖旧买新，就要支付一定的购置费用；若继续使用，则要支付更多的维护费用，且使用年限越长维护费用越多。若役龄期内每年的年初购置价格、当年维护费用

及年末剩余净值如表 5.1 所示。请为该设备制定一个 4 年役龄期内的更新计划，使总的支付费用最少。

表5.1　相关费用数据

年份	1	2	3	4
年初购置价格（万元）	25	26	28	31
当年维护费用（万元）	10	14	18	26
年末剩余净值（万元）	20	16	13	11

解　可以把这个问题化为图论中的最短路问题。

构造赋权有向图$D=(V,A,\mathbf{W})$，其中顶点集 $V=\{v_1,v_2,\cdots,v_5\}$，这里 v_i（$i=1,2,3,4$）表示第i 年年初的时刻，v_5 表示第 4 年年末的时刻，A 为弧的集合，邻接矩阵$\mathbf{W}=(w_{ij})_{5\times 5}$，这里$w_{ij}$ 为第i 年年初至第 j 年年初（或$j-1$ 年年末）期间所支付的费用，计算公式为

$$w_{ij}=p_i+\sum_{k=1}^{j-i}a_k-r_{j-i},$$

其中 p_i 为第i 年年初的购置价格，a_k 为使用到第k 年当年的维护费用，r_i 为第i 年年末旧设备的出售价格（残值），则邻接矩阵

$$\mathbf{W}=\begin{bmatrix} 0 & 15 & 33 & 54 & 82 \\ \infty & 0 & 16 & 34 & 55 \\ \infty & \infty & 0 & 18 & 36 \\ \infty & \infty & \infty & 0 & 21 \\ \infty & \infty & \infty & \infty & 0 \end{bmatrix}.$$

则制定总的支付费用最小的设备更新计划，就是在有向图 D 中求从 v_1 到 v_5 的费用最短路。

利用Dijkstra算法，求得v_1 到 v_5 的最短路径为$v_1\to v_2\to v_3\to v_5$，最短路径的长度为67。设备更新最小费用路径如图5.10中的粗线所示，即设备更新计划为第1年年初买进新设备，使用到第1年年底，第2年年初购进新设备，使用到第2年年底，第3年年初再购进新设备，使用到第4年年底。

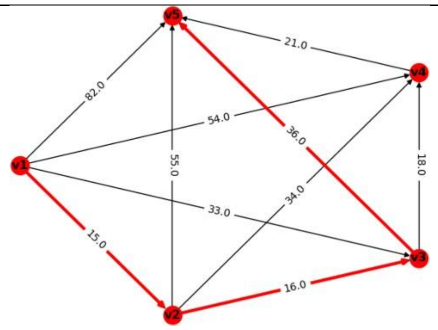
<table>
<tr>
<td></td>
<td>


图5.4　设备更新最小费用示意图
（四）课堂总结（5 分钟）总结本次课的教学内容：
1(　：固定起点的最短路算法思想及其计算步骤；
2(　：每对顶点之间的最短路算法思想及其计算步骤。
</td>
</tr>
<tr>
<td>课后作业</td>
<td>本章课后题第 1、2、3 题。</td>
</tr>
</table>

教学内容	第四节 最小生成树问题 第五节 最大流问题
教学目标	**知识目标：** 理解最小生成树问题的算法；理解最大流问题的算法。 **能力目标：**通过实例演示、问题驱动等方式激发学生学习的积极性，通过观察对比、学生交流、师生交流、小组探究等多种形式， 培养学生的逻辑思维能力、分析判断能力、和解决问题能力。 **素质目标：** 培养学生敢于质疑、善于分析、勇于创新的精神；培养学生的自主学习意识和团队协作精神；培养学生脚踏实地、不畏艰辛、锲而不舍的精神。
授课学时	2 学时
教学重点	最小生成树问题和最大流问题。
教学难点	最大流问题的算法。
教学方法	采用问题驱动法、案例演示法、启发式讲授法及自主学习法相结合，以教师的讲解为主，学生的课堂报告、分组讨论为辅，充分调动学生学习的主动性和思考问题的积极性。
教学手段	以课堂讲授为主，主要是多媒体课件和板书相结合的形式。同时借助在线资源，如慕课、雨课堂等平台。
教学过程	**（一）采用讲授法，介绍最小生成树问题及其算法（约 25 分钟）** 树是图论中非常重要的一类图，它结构简单、应用广泛，最小生成树问题是其中的经典问题之一。 知识点1： 若图 $G=(V(G),E(G))$ 和树 $T=(V(T),E(T))$ 满足 $V(G)=V(T)$，$E(T)\subset E(G)$，则称 T 是 G 的生成树。图 G 为连通的充要条件为 G 有生成树。一个连通图的生成树的个数很多。赋权图的具有最小权的生成树叫做最小生成树。 知识点 2：树有下面常用的五个充要条件： （1）$G=(V,E)$ 是树当且仅当 G 中任两顶点之间有且仅一条轨

道。

（2）G 是树当且仅当G 中无圈，且$|E|=|V|-1$。

（3）G 是树当且仅当G 连通，且$|E|=|V|-1$。

（4）G 是树当且仅当G 连通，且$\forall e \in E(G)$，$G-e$ 不连通。

（5）G 是树当且仅当G 无圈，且$\forall e \notin E(G)$，$G+e$ 恰有一个圈。

知识点 3：求连通图最小生成树的 Kruskal（克鲁斯卡尔）算法如下：

（1）选择边$e_1 \in E(G)$，使得e_1是权值最小的边。

（2）若 $e_1,e_2,\cdots,e_i$ 已选好，则从 $E(G)-\{e_1,e_2,\cdots,e_i\}$ 中选取e_{i+1}使得

①$\{e_1,e_2,\cdots,e_i,e_{i+1}\}$ 中无圈，

② e_{i+1}是 $E(G)-\{e_1,e_2,\cdots,e_i\}$ 中权值最小的边。

（3）直到选得到$e_{|V|-1}$为止。

知识点 4：求连通图最小生成树的 Prim（普里姆）算法。

构造连通赋权图$G=(V,E,W)$ 的最小生成树，设置两个集合 P 和 Q，其中P 用于存放G 的最小生成树中的顶点，集合Q 存放G 的最小生成树中的边。令集合 P 的初值为$P=\{v_1\}$（假设构造最小生成树时，从顶点v_1出发），集合Q 的初值为$Q=\varnothing$（空集）。Prim 算法的思想是，从所有 $p \in P, v \in V-P$ 的边中，选取具有最小权值的边pv，将顶点v 加入集合 P 中，将pv 加入集合Q 中，如此不断重复，直到$P=V$ 时，最小生成树构造完毕，这时集合Q 中包含了最小生成树的所有边。

Prim 算法构造最小生成树算法如下：

（1）$P=\{v_1\}$，$Q=\varnothing$。

（2）while $P \neq V$

找最小边 pv，其中 $p \in P, v \in V-P$；

$P=P+\{v\}$；

$Q=Q+\{pv\}$；

end

（二）结合案例，介绍最小生成树问题的 Matlab 软件求解方法（约 20 分钟）

（三）采用讲授法，介绍最大流问题及其算法。（约 25 分钟）

知识点 5：设 $G=(V,E)$ 为有向图，在 V 中指定一点称为发点或源（记为 v_s），另一点称为收点或汇（记为 v_t），其余点称为中间点。对每一条边 $v_iv_j\in E$，对应一个非负实数 C_{ij}，称为它的容量。这样的 G 称为容量网络，简称网络，记为 $G=(V,E,C)$。

知识点 6：网络 $G=(V,E,C)$ 中任一条边 v_iv_j 有流量 f_{ij}，称集合

$f=\{f_{ij}\}$ 为网络 G 上的一个流。满足下述条件的流 f 称为可行流：

⑴ 容量限制条件：对每一边 v_iv_j，有 $0\le f_{ij}\le C_{ij}$。

⑵ 平衡条件：对于中间点 v_k，有 $\sum f_{ik}=\sum f_{kj}$，即中间点 v_k 的输入量等于输出量。

如果 f 是可行流，则对收、发点 v_t，v_s 有 $\sum f_{si}=\sum f_{jk}=W_f$，即从 v_s 点发出的物质总量等于 v_t 点输入的量。W_f 称为网络流 f 的总流量。

知识点 7：最大流问题的数学模型表示如下:

$$\max z=v_f$$

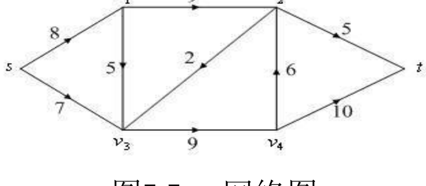
$$s.t.\begin{cases}\sum\limits_{\substack{j\in V\\ v_iv_j\in E}} f_{ij}-\sum\limits_{\substack{j\in V\\ v_iv_j\in E}} f_{ji}=\begin{cases}v_f, i=s\\ -v_f, i=t\\ 0, i\neq s,t\end{cases}\\ 0\le f_{ij}\le C_{ij}\quad v_iv_j\in E\end{cases}$$

知识点 9：最大流的标号算法（Ford-Fulkerson 标号法）

Ford-Fulkerson 标号法的基本思想就是，从一个可行流开始，寻找从 s 到 t 的增广链，然而沿增广链增加流量，反复这样，直到找不出增广链为止。

从 v_s 到 v_t 的一个可行流出发（若网络中没有给定 f，则可以设 f 是零流）经过标号过程和调整过程，即可求得从 v_s 到 v_t 的最大流。最大流的 Ford-Fulkerson 标号算法如下：

（1）标号过程.

	①初始化，给发点v_s标号$(0,+\infty)$。 ②选择一个已标号的点 x，对于x 的所有未给标号的邻接点 y，按下列规则处理： 当$yx\in E$ 且$f_{yx}>0$时，令$\delta_y=\min\left\{f_{yx},\delta_x\right\}$，并给 y 以标号$\left(x-,\delta_y\right)$； 当$xy\in E$ 且 $f_{xy}<C_{xy}$ 时，令 $\delta_y=\min\left\{C_{xy}-f_{xy},\delta_x\right\}$，并给 y 以标号 $\left(x+,\delta_y\right)$。 ③重复②直到收点v_t被标号或不再有点可标号时为止。若v_t得到标号，说明存在一条可增广链，转(2)调整过程；若v_t未得到标号，标号过程已无法进行时，说明 f 已经是最大流。 （2）调整过程。 ④决定调整量$\delta=\delta_{vt}$，令$u=v_t$。 ⑤若 u 点标号为$(v+,\delta_u)$，则以$f_{vu}+\delta$ 代替 f_{vu}；若 u 点标号为 $(v-,\delta_u)$，则以 $f_{vu}-\delta$ 代替 f_{vu}。 ⑥若$v=v_s$，则去掉所有标号转（1）重新标号；否则令$u=v$，转⑤。 **（四）结合案例，介绍最大流问题的 Matlab 软件求解方法（约 18 分钟）** 案例 2：现需要将城市s的石油通过管道运送到城市t，中间有 4 个中转站v_1,v_2,v_3 和 v_4，城市与中转站的连接以及管道的容量如图 5.5 所示，求从城市s 到城市t 的最大流。  图5.5　网络图 **（五）课堂总结（2 分钟）** 总结本次课的教学内容：

	1(：最小生成树问题机器算法； 2(：最大流问题及其算法。

课后作业	利用软件求解本章课后题 1、2、3。

教学内容	第六节 最小费用流问题 第七节 建模案例—锁具装箱问题
教学目标	**知识目标：** 理解最小费用流问题及其算法；掌握复杂的图论模型的建模。 **能力目标：**通过实例演示、问题驱动等方式激发学生学习的积极性，通过观察对比、学生交流、师生交流、小组探究等多种形式，培养学生的逻辑思维能力、分析判断能力、和解决问题能力。 **素质目标：** 培养学生敢于质疑、善于分析、勇于创新的精神；培养学生的自主学习意识和团队协作精神；培养学生脚踏实地、不畏艰辛、锲而不舍的精神。
授课学时	2 学时
教学重点	最小费用流问题
教学难点	复杂案例的建模
教学方法	采用问题驱动法、案例演示法、启发式讲授法及自主学习法相结合，以教师的讲解为主，学生的课堂报告、分组讨论为辅，充分调动学生学习的主动性和思考问题的积极性。
教学手段	以课堂讲授为主，主要是多媒体课件和板书相结合的形式。同时借助在线资源，如慕课、雨课堂等平台。
教学过程	**（一）采用讲授法，介绍最小费用流问题的模型及其算法（约 20 分钟）** 最小费用流问题是一种组合最优化问题，也是网络流理论研究的一个重要问题。这里我们要进一步探讨不仅要使网上的流量达到最大，或者达到要求的预定值，而且还要使运输流的费用是最小的，这就是最小费用流问题。 知识点 1：最小费用流问题的一般提法：已知网络$G=(V,E,C)$，每条边 $v_iv_j\in E$ 除了已给容量 C_{ij} 外，还给出了单位流量的费用 $b_{ij}(\geq 0)$。所谓最小费用流问题就是求一个总流量已知的可行流

$f=\{f_{ij}\}$ 使得总费用 $b(f)=\sum_{v_iv_j\in E} b_{ij}f_{ij}$ 最小。

特别地，当要求 f 为最大流时，此问题即为最小费用最大流问题。知识点2：最小费用流问题的数学表示如下：

$$\min z=\sum_{v_iv_j\in E} b_{ij}f_{ij}$$

$$s.t.\quad \begin{cases}\sum\limits_{\substack{j\in V\\ v_iv_j\in E}} f_{ij}-\sum\limits_{\substack{j\in V\\ v_jv_i\in E}} f_{ji}=\begin{cases}v_f, i=s\\ -v_f, i=t\\ 0, i\neq s,t\end{cases}\\ 0\le f_{ij}\le C_{ij}, v_iv_j\in E\end{cases}$$

（1）设网络 $G=(V,E,C)$，取初始可行流 f 为零流，求解最小费用流问题的迭代步骤如下。

①构造有向赋权图 $G_f=(V,E_f,f)$，对于任意的 $v_iv_j\in E, E_f$，F 的定义如下：

当 $f_{ij}=0$ 时，$v_iv_j\in E_f, F(v_iv_j)=b_{ij}$；

当 $f_{ij}=C_{ij}$ 时，$v_iv_j\in E_f, F(v_iv_j)=$
$-b_{ij}$；

当 $0<f_{ij}<C_{ij}$ 时，$v_iv_j\in E_f, F(v_iv_j)=b_{ij}$，$v_iv_j\notin E_f, F(v_iv_j)=-b_{ij}$。然后转向②。

②求出有向赋权图 $G_f=(V,E_f,f)$ 中发点 v_s，到收点 v_t 的最短路 μ，

若最短路 μ 存在，转向③；否则 f 是所求的最小费用最大流，停止。

③增流。同求最大流的方法一样，重述如下：

令 $\delta_{ij}=\begin{cases}C_{ij}-f_{ij}, v_iv_j\in\mu^+\\ f_{ij}, \quad v_iv_j\in\mu^-\end{cases}$，$\delta=\min\{\delta_{ij}|v_iv_j\in\mu\}$，重新定义流 $f=\{f_{ij}\}$ 为

$$f_{ij}=\begin{cases}f_{ij}+\delta, v_iv_j\in\mu^+\\ f_{ij}-\delta, v_iv_j\in\mu^-\\ f_{ij}, \quad \text{其他}\end{cases}$$

如果 W_f 大于或等于预定的流量值，则适当减少 δ 值，使 W_f 等于预定的流量值，那么 f 是所求的最小费用流，停止；否则转向①。

（3）求解含有负权的有向赋权图 $G=(V,E,C)$ 中某一点到其他各

点最短路的 Ford 算法。

当 $v_iv_j \in E$ 时，记$w_{ij}=F(v_iv_j)$，否则取$w_{ij}=0$，$w_{ij}=+\infty(i\neq j)$。v_1到v_i的最短路长记为$\pi(i)$，v_1到 v_i 的最短路中v_i的前一个点记为$\theta(i)$。Ford 算法的迭代步骤：

①赋初值 $\pi(1)=0$，$\pi(i)=+\infty,\theta(i)=i(i=2,3,\cdots,n)$。

②更新$\pi(i)$，$\theta(i)$。对于 $i=2,3,\cdots,n$ 和 $j=1,2,\cdots,n$，如果

$\pi(i)<\pi(j)+w_{ji}$，则令$\pi(i)=\pi(j)$，$\theta(i)=j$。

③终止判断。若所有的$\pi(i)$ 都无变化，停止；否则转向②。

在算法的每一步中，$\pi(i)$ 都是从v_1到 v_i 的最短路长度的上界。若不存在负长回路，则从v_1到 v_i 的最短路长度是$\pi(i)$ 的下界，经过$n-1$次迭代后$\pi(i)$将保持不变。若在第 n 次迭代后$\pi(i)$仍在变化时，说明存在负长回路。

（二）结合案例，介绍利用 Matlab 软件求解最小费用流问题。（约 15 分钟）

案例 1：求图 5. 6 所示网络从v_s到 v_t的最小费用最大流，其中边上的权重的第 1 个数字表示网络的容量，第 2 个数字表示网络的单位流量的费用。

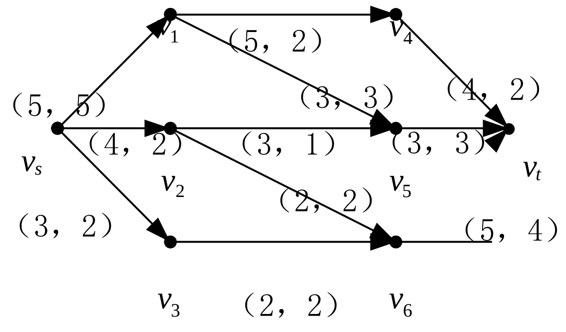


图5. 6　网络流

（三）给出案例，引导学生分组讨论。（约 10 分钟）

【教师活动】展示历年竞赛真题，启发学生进行讨论。

案例2：1994年全国大学生数学建模竞赛B题（锁具装箱）中关

	于锁具总数的问题可叙述如下：某厂生产一种弹子锁具，每个锁具的钥匙有5个槽，每个槽的高度从{1，2，3，4，5，6}中任取一数。由于工艺及其它原因，制造锁具时对5个槽的高度还有两个限制：（1）至少有3个不同的数；（2）相邻两槽的高度之差不能为5。满足以上条件制造出来的所有互不相同的锁具称为一批。我们的问题是如何确定每一批锁具的个数？从顾客的利益出发，自然希望在每批锁具中“一把钥匙开一把锁”。但是在当前工艺条件下，对于同一批中两个锁是否能够互开，有以下试验结果：若二者相对应的5个槽的高度中有4个相同，另一个槽的高度差为1，则能互开；在其它情形下，不可能互开。原来销售部门在一批锁具中随意的取60个装一箱出售。团体顾客往往购买几箱到几十箱，他们抱怨购得的锁具会出现互开的情形。现聘你为顾问，回答并解决以下的问题： 1） 每一批锁具有多少个，装多少箱。 2） 为销售部门提出一种方案，包括如何装箱（仍是 60 个锁具一箱），如何给箱子以标志，出售时如何利用这些标志，使团体顾客不再或减少抱怨。 3） 采取你提出的方案，团体顾客的购买量不超过多少箱，就可以保证一定不会出现互开的情形。 4） 按照原来的装箱办法，如何定量地衡量团体顾客抱怨互开的程度（试对购买一、二箱者给出具体结果）。 【学生活动】分组讨论，寻找建模方案。 【设计意图】通过分组讨论，让学生能具体的投入到教学过程中来。 **（四）采用案例分析法，结合所给问题及学生讨论情况，讲解建模方案。（约 40 分钟）** （1）每一批锁具数量求解。 首先求满足条件的锁具的个数，可以使用计算机枚举的方法或排列组合计数的方法，也可以使用图论的方法解决。已知每把锁都

有5 个槽，每个槽有 6 个高度，至少有三个不同高度的槽，且相邻槽高差不为 5。我们先求出无相邻高差为 5 的锁具数量，再减去仅有一个、两个槽高的锁具数目。先计算由 1，2，3，4，5， 6 构成无 1，6 相邻的情况的数目。为此，构造一个 6 节点的图： 将 1，2，3，4，5，6 这6 个数作为 6 个节点，当两个数字可以相邻时，这两个节点之间加一条边，每个节点有自己到自己的一条边，我们得到了锁具各槽之间的关系示意图（图5.14）

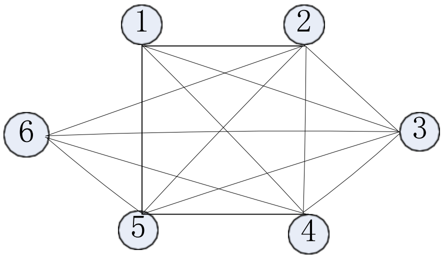


图 5.7 锁具各槽之间的关系示意

图该图的邻接矩阵为

$$A=\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

邻接矩阵 A 的所有元素之和表示两个槽高无 1、6 相邻的锁具的个数。每个无 1、6 相邻的 5 位数与图 5.14 中长度为 4 的一条链一一对应，如 12345，11111，22335 等。A 的 k 次方 A^k 中各元素之和就是长度为 k 的链的个数。事实上，从这个具体问题可以看出，A^2 中第 i 行第 j 列的元素指从 i 开始经过两条边到达 j 的链数，即从 i 开始经过一条边到 k，再从 k 经过一条边到达 j，i 和 j 就决定了中间顶点 k 的数目。

于是，得到

$$A^4=\begin{pmatrix} 141 & 165 & 165 & 165 & 165 & 140 \\ 165 & 194 & 194 & 194 & 194 & 165 \\ 165 & 194 & 194 & 194 & 194 & 165 \\ 165 & 194 & 194 & 194 & 194 & 165 \\ 165 & 194 & 194 & 194 & 194 & 165 \\ 140 & 165 & 165 & 165 & 165 & 141 \end{pmatrix}$$

将A^4中元素求和可得相邻高差不为 5 的锁具数为 6 306 把。但这 6306 把锁具中包含了仅有一个、两个槽高的锁具，需要从其中减去，需减去的锁具的个数为

$$6+\left(C_6^2-1\right)\left(C_5^1+C_5^2+C_5^3+C_5^4\right)=426$$

其中，第一个 6 为仅有 1 个槽高的锁具；C_6^2为$1, 2, 3, 4, 5, 6$ 这6个数中取两个的取法，但扣除 1、6 这一种取法；C_5^1,C_5^2,C_5^3,C_5^4是 5 个槽中有 1,2, 3, 4 个槽用所取出的这两个数中的一个。

最后得到一批锁具的个数为 6306 -426=5880。这样，就用图论的知识成功地解决了一批锁具的数量问题，这个方法比用别的方法简便，且容易推广。

（2）最大不能互开锁具数求解。

销售时每 60 个锁具装一箱，求出最大不能互开的锁具数。首先，由 5.7.2 知满足要求的合格锁具共有 5880 种组合，可令$(x_{i1},x_{i2},x_{i3},x_{i4},x_{i5})$表示五个槽槽高分别为 $x_{i1},x_{i2},x_{i3},x_{i4},x_{i5}$的一个锁具。记 H_1是所有槽高之和为奇数的锁具集合， H_2是所有槽高之和为偶数的锁具集合，在有奇数个槽，每个槽有偶数高度的情况下，集合H_1和H_2之间存在双射$\varnothing: H_1\rightarrow H_2$，其对应关系为

$$(x_{i1},x_{i2},x_{i3},x_{i4},x_{i5})\mapsto(x'_{i1},x'_{i2},x'_{i3},x'_{i4},x'_{i5})$$

其中$x'_{ij}=7-x_{ij}\ (j=1,2,\dots,5)$，所以$H_1$和$H_2$各有 2940 个元素。然后，

将每个合格锁具都看作一个顶点，能够互开的两个顶点连一条边。可计算出互开总对数为 22778，所以共有 22778 条边。因为能互开的锁具奇偶性不同，所以 H_1和H_2之间有边相连，而 H_1和H_2内部无边相连，即构成一个二分图。

最大不能互开锁具数，即为此图的最大独立顶点数。即

	“最大独立顶点数=顶点总数–最大对集的边数” 由于此图的邻接关系比较复杂，所以从理论上求出最大对集是很困难的，在这里我们可以用“匈牙利算法”来求解二分图的最大独立顶点数，即最大不能互开锁具数. 首先求出 H_1 、H_2 和它们之间的邻接矩阵W ，再用 Matlab 软件求解可得到最大对集e 及其边数 2940。所以最大独立顶点数=5880-2940 =2940，即为最大不能互开锁具数。 **（五）课堂总结（5 分钟）** 总结本次课的教学内容： 1(：最小费用流问题及其算法； 2(：复杂案例的建模方案。
课后作业	继续思考锁具装箱问题，形成完整的竞赛论文。