在第 12 题～第 15 题中，已知图形所表示的图，找出权最小的哈密尔顿回路。

12．图 8.66。

13．图 8.67。

14．图 8.68。

15．图 8.69。

16．在图 8.66 所表示的图中，以 D 为始点和终点，找权最小的一条哈密尔顿回路。

17．在图 8.67 所表示的图中，以 F 为始点和终点，找权最小的一条哈密尔顿回路。

18．对于有 n 个顶点的完全图 K_n $(n\geq 3)$。

(a) 一条哈密尔顿回路一定要有多少条边？

(b) 以一个固定点为始点，存在多少条不同的哈密尔顿回路？

19．证明：有 $n(n\geq 3)$个顶点的完全图 K_n 有$(n-1)!$条哈密尔顿回路。

20．给出一个至少有 4 个顶点和一条既是欧拉回路又是哈密尔顿回路的图例。

21．给出一个有一条欧拉回路和一条哈密尔顿回路但两者不相同的图例。

22．设 $G=(V, E, \gamma)$是无多重边的一个图，$|V|=n$，由 G 定义的 V 上的关系 R 能用矩阵 $\boldsymbol{M}_R$ 表示。解释如何使用 M_{R^∞} (见 4.3 节)去判断 G 是否是连通的。

23．使用通常的顶点标号，即 0 和 1 的字符串，为 B_3 的哈塞图给出一条哈密尔顿回路。

24．使用通常的顶点标号，即 0 和 1 的字符串，为 B_4 的哈塞图给出一条哈密尔顿回路。

25．为 B_n 的哈塞图寻找哈密尔顿回路的问题等价于有关长度为 n 的 0 和 1 的字符串问题。请对此问题给予描述。

8.4 运输网络

前面已经考察了几种标号图的使用，本节将回到讨论有向图的应用。标号有向图的一个重要应用是作为通常所谓的运输网络模型。考虑图 8.70 所示的标号有向图，它也许表示城市里的部分供水系统中水从顶点 1 流到顶点 6 的管道，边上面的标号表示能通过该边的最大流量，称作边的**容量**。现实生活中有许多情况都能归纳为这种模型。例如，图 8.70 很可能表示石油管道系统、高速公路系统、通信网络或电力网等。网络的顶点通常称为节点，可以表示抽水站、装运站、转播站或高速公路的交叉路口等。

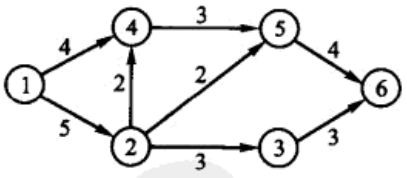

图 8.70

严格地说，**运输网络(或网络)**是具有如下性质的一个连通有向图 N：

(a) 存在惟一一个入度为 0 的节点，称为**源**。通常对源节点标号为 1。

(b) 存在惟一一个出度为 0 的节点，称为**汇**。如果 N 有 n 个节点，则通常对汇标号为 n。

(c) 图 N 已被标号，在边(i, j)上的标号 C_{ij} 是一个非负数，称为边的容量。

为简单起见，还假设所有的边都只以一个方向运送物资，即如果(i, j)属于 N，则(j, i)不属于 N。

流

网络的目的是实现水、油、电、交通或所设计的网络运送物资的流动。从数学上讲，网络 N 中的**流**就是一个函数，它使 N 的每条边(i, j)都指定一个不超过 C_{ij} 的非负数 F_{ij}。直观地讲，当流是 F 时，F_{ij} 表示通过边(i, j)的物资量。非正式地讲，F_{ij} 表示通过边(i, j) 的流。还要求除源和汇之外，对每个节点而言，进入节点 k 的边上的 F_{ik} 之和必须等于离开节点 k 的边上的 F_{kj} 之和。也就是说，除源和汇外，在任何节点物资不能积累、再生、消耗或丢失，这称作**流的守恒**。该项要求的结果便是离开源的流之和必须等于进入汇的流之和。该和称作**流量**，记为 $value(F)$。可以对每条边(i,j)标上数据对(C_{ij}, F_{ij})表示流 F。图 8.70 所表示的网络中的流 F 如图 8.71 所示。

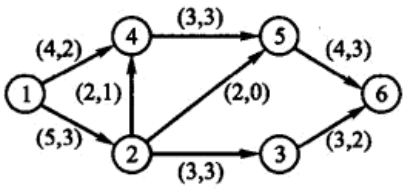

图 8.71

例 1 在图 8.71 中，节点 4 的流是守恒的，因为输入流的大小是 2 和 1，输出流的大小是 3。(验证其他节点的流也是守恒的。)这里 $value(F)=5$。 ■

最大流

对于任何一个网络，一个重要的问题是确定通过网络的最大流量，并且描述具有最大流量的流。由于这种显而易见的原因，该问题通常称作**最大流问题**。

例 2 图 8.72(a)给出了流量为 8 的流。5 条边中有 3 条边通过了它们的最大容量。这似乎是一个好的流函数，但是图 8.72(b)给出了同一网络流量为 10 的流。 ■

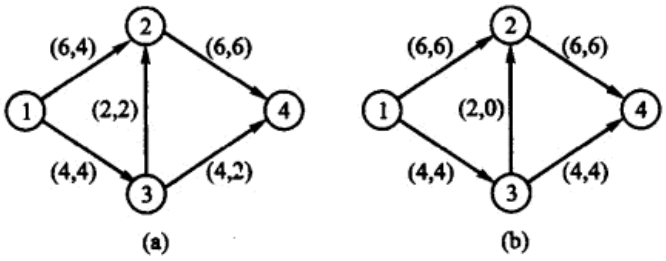

图 8.72

例 2 表明即使一个很小的网络，求最大流问题也需要一个系统化的处理过程。检查图 8.72(a)中的流表明刚才使用从节点 3 到节点 2 的边就是一个错误。应该在边(3，2)上减少流使得在其他边上能增加流。

假设在某个网络 N 中，一条边(i, j)上通过 5 个单位的流，如果希望该流减少到 3 个单位，那么可以推测它在相反的方向应有 2 个单位的流相配合。虽然边(j, i)不属于 N，但是只要它有减少实际边(i, j)上所存在的流的效果，考虑这种*虚拟流*是毫无害处的。图 8.73 显示了图 8.72(a)中所示的一部分流。

道路π：1，2，3，4 在此网络中不是一条实际道路，因为(2，3)不是一条实际边。然而，π是网络的对称闭包中的一条道路(参见 4.7 节中的对称闭包)。而且，如果考虑有 2 个单位的虚拟流通过π，那么对网络的影响是通过边(1，2)和(3，4)上的流要增加 2 个单位，通过边(3，2)上的流要减少 2 个单位。因此，图 8.72(a)中的流变成了图 8.72(b)中的流。

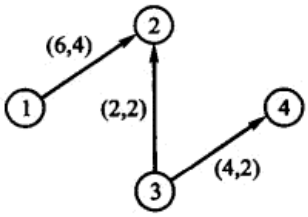

图 8.73

现在用通用术语来描述这种改进。设 N 是一个网络，G 是 N 的对称闭包，在 G 中选择一条道路和该道路上的一条边(i,j)。如果(i,j)属于 N，$e_{ij}=C_{ij}-F_{ij}>0$，则称该边有正的超容量。如果(i,j)不是 N 中的边，则以反方向通过该边，在这种情况下，如果 $F_{ji}>0$，则称(i,j)有超容量 $e_{ij}=F_{ji}$,于是通过边(i,j)的增流将产生减少 F_{ji} 的结果。下面给出求解最大流问题的具体步骤。

最大流算法

下面将提到的算法应归于 Ford 和 Fulkerson，常称为**标号算法**。这里所指的标号是节点的附加标记。为简单起见，容量使用整数，但 Ford 和 Fulkerson 已证明如果容量是有理数，此算法将在有限步骤内结束。

设 N 是有 n 个节点的一个网络，G 是 N 的对称闭包，要使用的所有边和道路都属于 G。开始的所有流都设为 0。当执行算法时，很方便跟踪边上的超容量以及它们的变化情况，而不是跟踪增流。当算法结束时，很容易从最终的超容量中找出最大流。

算法 标号算法

步骤 1：设 N_1 是通过一条具有正的超容量边与源相连接的所有节点的集合，N_1 中的每个 j 标以$[E_j, 1]$，其中 E_j 是边$(1，j)$上的超容量 $e_{1,j}$,标号中的 1 表示 j 与源节点 1 连接。

步骤 2：设 N_1 中的节点 j 是有最小节点号的节点，$N_2(j)$表示所有未标号的(除源外)与节点 j 相连并且有正的超容量的节点的集合。假设节点 k 属于 $N_2(j)$并且(j, k)是有正的超容量的边，则节点 k 标以$[E_k, j]$，其中 E_k 是 E_j 和边(j, k)上的超容量 $e_{j,k}$ 的最小值。当 $N_2(j)$中的所有节点都用这种方法标号时，对 N_1 中其他节点重复该过程，设 $N_2=\bigcup_{j\in N_1} N_2(j)$。

注意，在步骤 1 之后，对 N_1 中的每个节点已标号 E_j，它是通过一条边能从源流到 j 的物质量并且含有来自于节点 1 的流的信息。在步骤 2 中，从源通过道路π:1,j,k 到达先前未标号节点 k 是用$[E_k, j]$标号，其中 E_k 是能通过 π 的最大流，因为它是能到达 j 然后通过 k 的更小的量。因此，当步骤 2 结束时，对 N_2 中的所有节点都已构造了所谓的两步道路，每个节点标号记录了通过道路到达该节点和该道路中它的前一节点的总流。继续尝试这种构造，以增加道路的长度直至达到汇为止(如果可能的话)。于是增加了总流，并且能够回溯用于增流的这条道路。

步骤 3：重复步骤 2，给所有 N_3 中未标号的节点标号，其中 N_3 表示从 N_2 中的一个节点能通过一条有正的超容量的边到达的节点的集合。继续该过程形成集合 N_4，N_5，…，直到经过有

限步骤后出现下列情形为止。

(i) 汇没有得到标号并且没有其他任何节点可以标号，这也许会发生没有任何节点得到标号，注意源没有被标号，或者

(ii) 汇得到了标号。

步骤 4：在情形(i)，算法终止，则总流就是最大流(后面将给出证明)。

步骤 5：在情形(ii)，汇(即节点 n)得到标号$[E_n, m]$，其中 E_n 是通过道路 π 到达汇的额外流量。以相反的次序检查π，如果边$(i, j)\in N$，则在(i, j)上增流 E_n，并且减少相同量的超容量 $e_{i,j}$。同时给(虚)边(j, i)增加超容量 E_n，因为在(i, j)上与它的反向相比有更多的流。另一方面，如果$(i, j)\notin N$，则在(j, i)上减少流 E_n，并且增加它的超容量 E_n。同时在(i, j)上减少相同量的超容量，因为在(i, j)上与它的反向相比有更少的流。现在得到了一个新的比前面大 E_n 个单位的流，并且回到步骤 1。 ■

例 3 用标号算法求图 8.70 所示网络中的最大流。

解 图 8.74 给出了 G 中的所有边上带初始容量的网络。所有边上的初始流是 0。

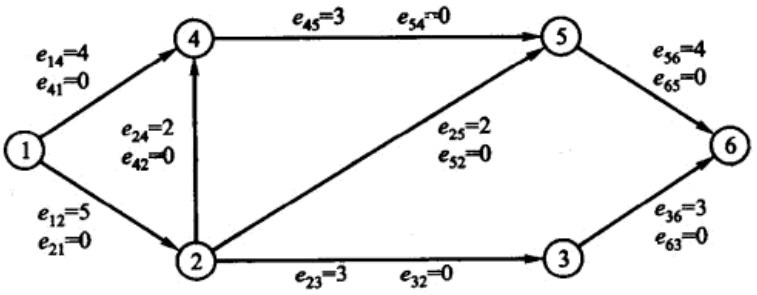

图 8.74

步骤 1：从源开始，通过有超容量的边能到达节点 2 和 4，所以 $N_1=\{2, 4\}$，给节点 2 和 4 分别以[5，1]和[4，1]标号，如图 8.75 所示。

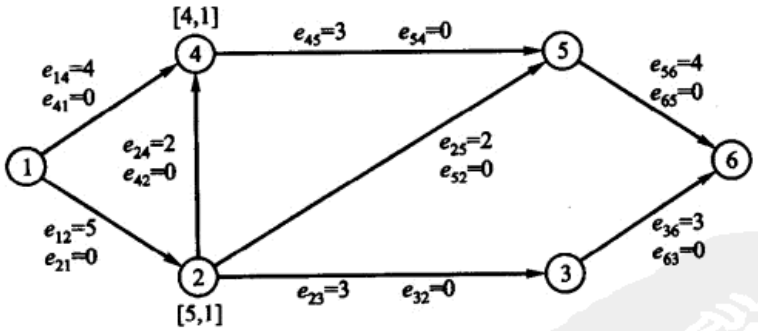

图 8.75

步骤 2：从节点 2 开始，通过具有正的超容量的边能到达节点 5 和 3。节点 5 用[2，2]标号，

因为只有 2 个额外的单位流能通过边(2，5)。节点 3 用[3，2]标号，因为只有 3 个额外的单位流能通过边(2，3)。这一步的结果如图 8.76 所示。不能通过某条边从节点 4 移到其他任何未标号的节点。因此，N_2={3，5}，步骤 2 结束。

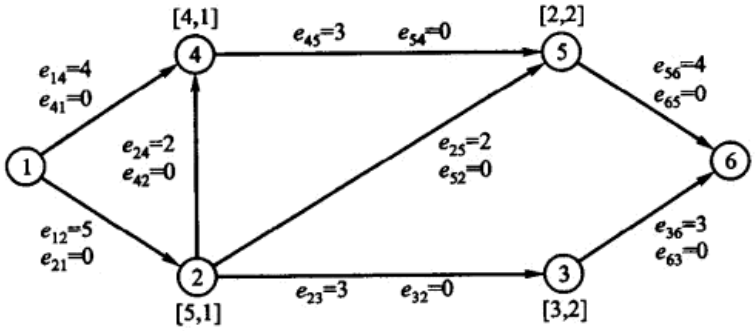

图 8.76

步骤 3：使用 N_2 重复步骤 2，能从节点 3 到达汇，并且有 3 个单位流能通过(3，6)，因此汇有标号[3，3]。

步骤 4：通过道路 1，2，3，6 进行回溯，把每条边的超容量减 3(表明给每条边增流)，并且把该(虚)边增加等值的超容量。然后用图 8.77 所示的情形回到步骤 1。

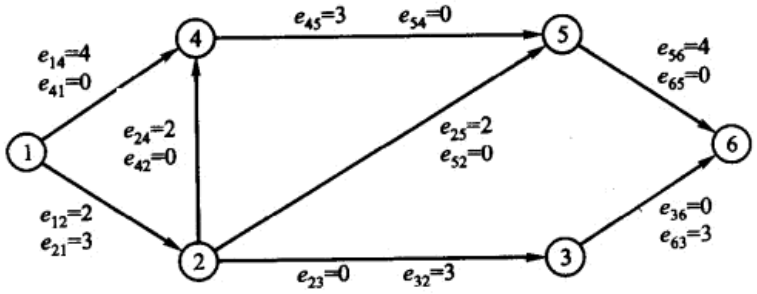

图 8.77

像前面那样继续进行标号，节点 2 和 4 分别得到标号[2，1]和[4，1]。注意 E_2 现在只是 2 个单位，即边(1，2)上新的超容量。节点 2 不能再用来给节点 3 标号，因为在边(2，3)上没有超容量。但节点 5 将得到标号[2，2]。再一次没有从节点 4 能到达的未标号的节点，所以转到步骤 3。于是能从节点 5 到达节点 6，所以节点 6 得到标号[2，5]。步骤 3 的最终结果如图 8.78 所示，把 2 个单位流已增加到总数为 5 个单位的流上。

再次转到步骤 4 并且沿道路 1，2，5，6 回溯，把这些边上的超容量减 2，对应的(虚)边增加相同的容量 2。用图 8.79 所示的情形回到步骤 1。这一次的步骤 1 和步骤 2 产生下列结果。只有节点 4 从节点 1 得到标号[4，1]，而节点 5 是从节点 4 得到的惟一的节点标号[3，4]。用图 8.80 开始步骤 3。此时节点 5 本可以用边(5，2)上的超容量给节点 2 标号(验证节点 2 标号是

[2，5])。然而，节点 5 同样能用来给汇标号，其标号是[2，5]并且总流增加到 7 个单位。步骤 4 沿道路 1，4，5，6 进行回溯，调整超容量。然后，如图 8.81 所示回到步骤 1。

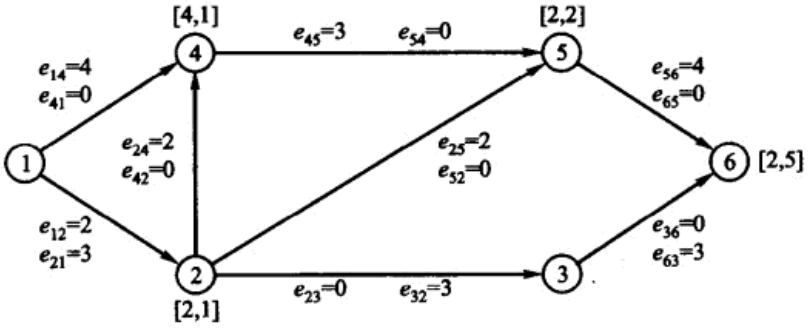

图 8.78

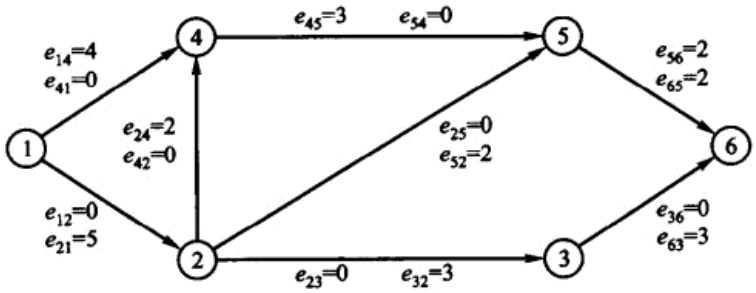

图 8.79

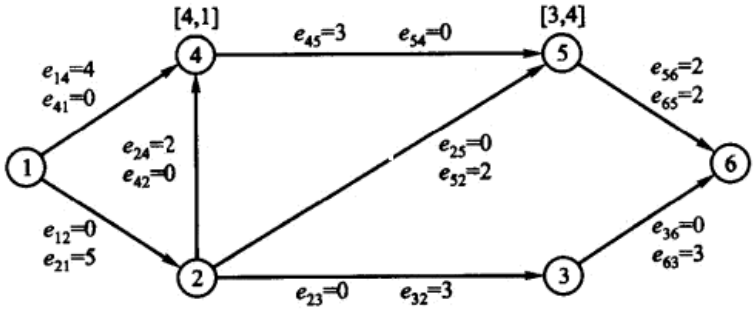

图 8.80

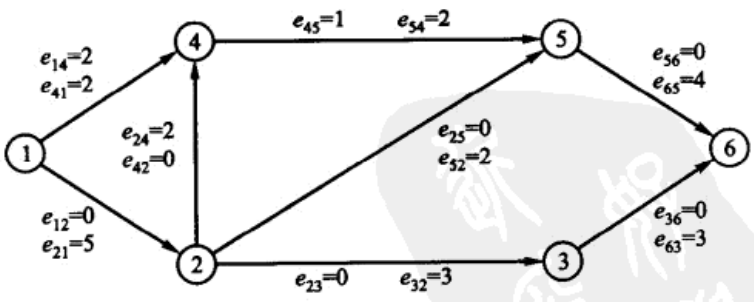

图 8.81

可以验证在经过步骤 1、步骤 2 和步骤 3 之后，节点 4、5 和 2 得到如图 8.82 所示的标号并且不可能还有别的标号。节点 2 的最终标号使用了虚边(5，2)。因此，最终的总流量是 7。通过从容量 C_{ij} 中减去 N 中每条边(i, j)上的最终超容量 e_{ij}，得到最大流量是 7 的流 F，如图 8.83 所示。 ■

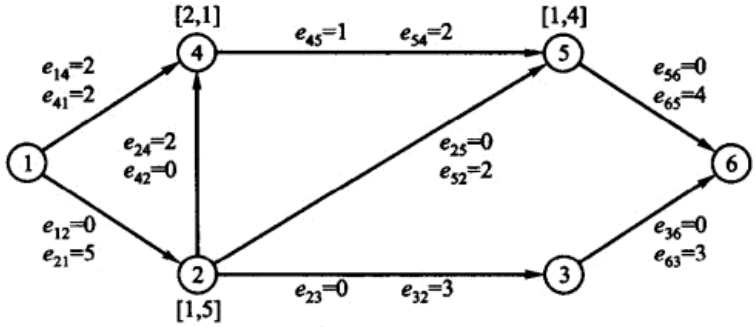

图 8.82

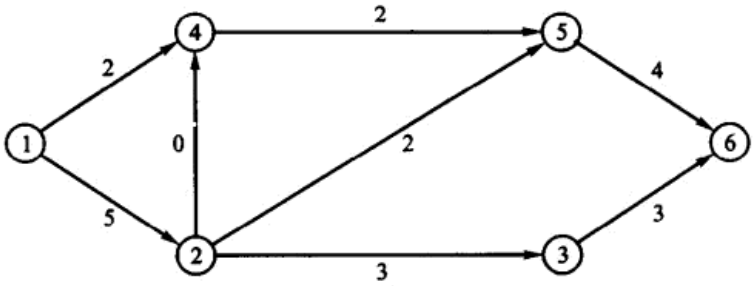

图 8.83

余下的问题是证明标号算法产生一个最大流。首先，把一个网络 N 中的**切割**定义为某些边的集合 K，它具有性质：从源到汇的每条道路至少包含 K 中的一条边。实际上，一个切割就是把一个有向图“切割”成两块，一块包含源，另一块包含汇。如果把切割中的边都删除，那么没有什么可以从源流到汇。一个**切割 K 的容量**，即 $c(K)$，是 K 中所有边的容量之和。

例 4 图 8.84 给出了图 8.70 中网络的两个切割，每个切割都用锯齿状的线标识，它是由锯齿状线所碰到的所有边组成的。可以验证 $c(K_1)=10$，$c(K_2)=7$。 ■

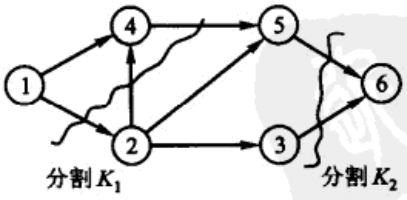

图 8.84

如果 F 是任意流，K 是任意切割，则 $value(F) \leqslant c(K)$。因为 F 中的所有流都必须通过 K 中的边，而 $c(K)$是能通过 K 中边的最大数值。现在假设对于某个流 F 和某个切割 K 有 $value(F) = c(K)$。换句话说，流 F 使用 K 中所有边的满容量。那么 F 是具有最大流量的一个流，因为没有流能够比 $c(K)$的值大。类似地，K 一定是一个最小容量切割，因为每个切割的容量一定至少等于 $value(F)$。于是得到下面的结论。

定理 1 (最大流—最小切割定理) 在一个网络中的最大流 F 与该网络中的最小切割容量相等。 ■

现在通过找一个容量与流量相等的最小切割来证明标号算法确实产生最大流。假设算法已执行并且在步骤 4 停止，那么汇没有得到标号。把节点分成两个集合 M_1 和 M_2，其中 M_1 包含源以及所有已标号的节点，M_2 包含所有未标号节点(源不在其中)。设 K 是由网络 N 中 M_1 里的一个节点与 M_2 里的一个节点相连接的所有边组成，则从源到汇的 N 中的任何道路 π 都是以 M_1 里的一个节点开始，以 M_2 里的一个节点结束。如果 i 是属于 M_1 的 π 中最后一个节点，j 是该道路中跟随 i 后的节点，那么 j 属于 M_2,于是由(i,j)的定义可知 j 属于 K。所以，K 是一个切割。

假设(i,j)是 K 中的一条边，则 $i \in M_1, j \in M_2$，由算法所产生的最终流 F 一定导致(i,j)传送满容量。否则，可以用节点 i 和超容量给 j 标号(由定义可知 j 没有标号)。因此算法的最终流量等于容量 $c(K)$，故 F 是一个最大流。

例 5 与例 3 中找到的最大流相对应的最小切割是 $K=\{(5,6),(3,6)\}$且 $c(K)=7=value(F)$。 ■

习 题 8.4

在第 1 题～第 4 题中(图 8.85～图 8.88)，已知用流标号的网络并且除源和汇之外每个节点的流守恒，则用它的最大容量为每条边标号。

1.

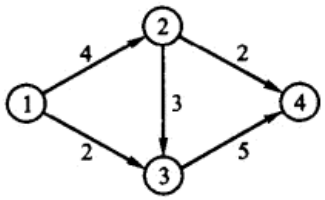

图 8.85

2.

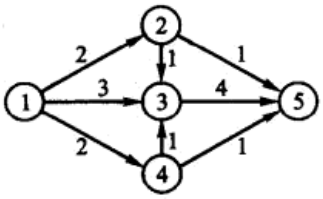

图 8.86

3.

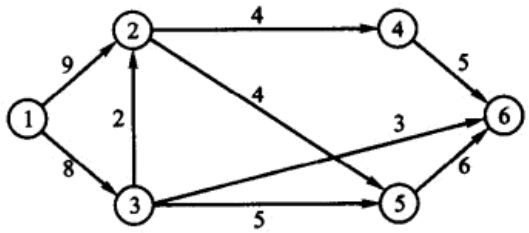

图 8.87

4.

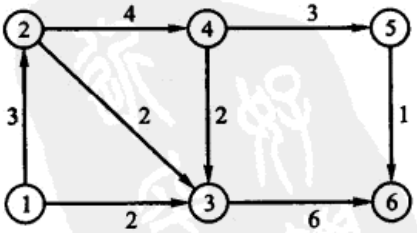

图 8.88

在第 5 题~第 11 题中，在已知的网络中用标号算法求最大流。

5．网络如图 8.85 所示。

6．网络如图 8.86 所示。

7．网络如图 8.87 所示。

8．网络如图 8.88 所示。

9．网络如图 8.89 所示。

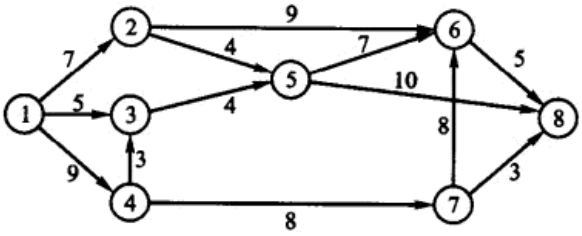

图 8.89

10．网络如图 8.90 所示。

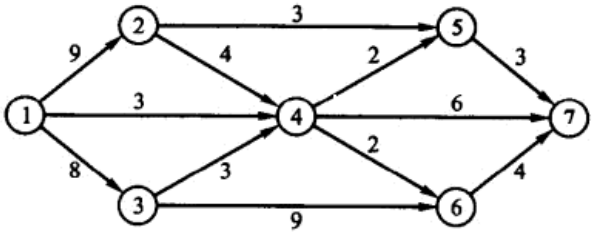

图 8.90

11．网络如图 8.91 所示。

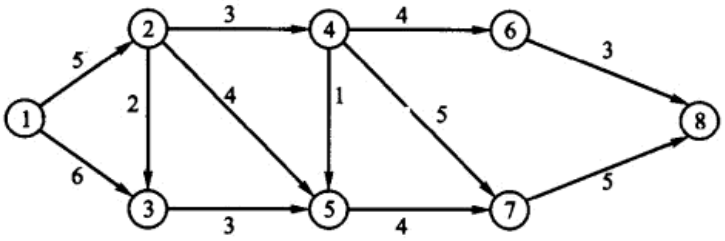

图 8.91

12．在图 8.87 所示的网络中，给出两个切割和它们的容量。

13．在图 8.89 所示的网络中，给出两个切割和它们的容量。

14．在图 8.91 所示的网络中，给出三个切割和它们的容量。

在第 15 题~第 21 题中，求与已知网络中最大流相对应的最小切割。

15．第 5 题中的网络。

16．第 6 题中的网络。