

本章研究管理科学或运筹学中的决策分析问题,内容主要包括图与网络分析、网络计划、排队论、存储论、对策论和决策分析等。在工作表中运用公式及函数、图形等功能,可以分解、演算、可视化复杂的决策问题,既可以提高学习效率,又可以解决实际问题。

5.1 图与网络分析

图论(Graph Theory)是运筹学的重要分支,它起源于 18 世纪著名的 Königsberg 桥梁问题。Königsberg 是坐落在普雷格尔(Pregolya)河畔的一个小镇,由 4 个独立的社区组成,其中有两个社区位于河两岸,另两个社区位于河中小岛上,共有 7 座桥梁将这些社区相连。当时,人们企图破解沿一条线路穿过 7 座桥(每桥只能穿行一次)并返回原点问题。1936 年,欧拉(Euler)用数据图(Graphs)证明了人们关于一条线穿过 Königsberg 七桥问题是不可能的。因为 Königsberg 的 4 个社区由奇数座桥连接,所以不可能画出想要的路线。解答此问题的同时,欧拉开创了图论研究并奠定了理论基础。如今,图论被用于解决许多学科和领域的实际问题,如博弈论、人工智能、资源优化及分类、迷宫设计与求解等。

图论中常用的基本概念或定义简述如下。

图(Graphs)是端点和边的集合,通常记为 $G=(V,E)$,其中, V 是端点集合, E 是边集合。

端点或顶点(Vertexes)是图中连线的端点或交点(Node),常用于表示位置,如城市、行政区划、十字路口、车站、港口和机场等。

边(Edges)是两端点之间的连线,常用于表示端点间的通道或运输设施。带箭头的连线是有向边,没有箭头的连线则假定连接是双向的。

次(Degree)与奇、偶点(Odd and Even Vertex):端点上边的数量称为次;次为奇数的端点称为奇点,反之称为偶点。

路径(Path)、回路(Cycles)与轨迹(Trail):沿着边经过端点的“行程”称为路径;起点与终点相同的路径称为回路;端点之间的行程称为轨迹,返回起点的轨迹称为圈(Circuit)或圈形轨迹(Circular Trail)。

连通图(Connectedness):端点之间存在连接边或路径的图称为连通图。在连通图中,与端点或边有关的数量指标称为权(Weight),权可以表示距离、费用、容量等。这种带有数量指标的赋权连通图称为网络。

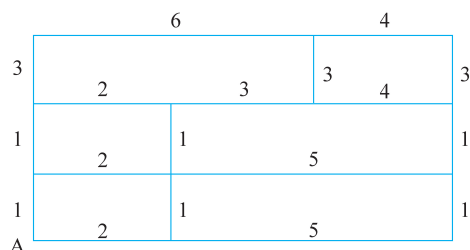
5.1.1 欧拉回路与中国邮递员问题

在连通图中,经过每边一次的道路或路径可称为欧拉道路或路径。从某端点出发穿过每条边一次返回端点的环形轨迹称为欧拉回路,具有欧拉回路的图称为欧拉图。根据欧拉

定理,如果一个网络连通图中的奇点超过两个,则该图没有欧拉路径;如果连通图中正好有两个奇点,则该图的欧拉路径从一个奇点开始到另一个奇点结束;如果连通图中没有奇点,则该图至少有一条欧拉回路。

如果把邮递员每天沿街道投递的路线看作一个连通图,那么,邮递员从邮局出发去辖区内每条街投递后返回的轨迹就是一个回路。显然,邮递员希望找到最短回路。这个问题是山东师范大学管梅谷教授于 1960 年首先提出的并给出了“奇偶点图上作业”的解法,因此,国际上称之为中国邮递员问题(Chinese Postman Problem, CPP)。在图论中,连通的赋权欧拉回路被称为中国邮递员问题。下面用示例讲解中国邮递员问题奇偶点图上作业法。

示例 5-1(胡运权主编《运筹学教程》第 5 版 8.4 题,清华大学出版社) 求解如图 5-1 所示的中国邮递员问题,A 点是邮局。



5.1.2 生成树

在图论中,无圈的连通图称为树(Tree)。树状网络图类似多源头河流,其基本特性是:任意两个顶点之间有且仅有一条链,树的顶点数减1等于边数。生成树又名支撑树(Spanning Trees),是无向连通图的子图,它包括最少边连接的所有顶点。从已知连通图求解出生成树的方法可分为以下两种。

一是避圈法。有深探法和广探法两种避圈求解生成树的方法。深探法就是在连通图中任一点开始标号,并沿着该点的边对另一端点继续标号,每步选择要标号的边与已有标号的边不能构成圈,遇到已标号端点时退回并从其他边继续标号,直到所有端点均得到标号。广探法就是在连通图中任取一个边,并按不成圈的原则继续取与之相连的下一个边,重复这个步骤直到取得生成树。

二是破圈法。就是在连通图中任取一个圈并从圈中去掉一个边,重复这个步骤,直到剩下不含圈的支撑树。

示例 5-2(胡运权主编《运筹学教程》第5版 8.5 题,清华大学出版社) 分别用深探法、广探法、破圈法找出如图 5-3①所示网络图的一个生成树。

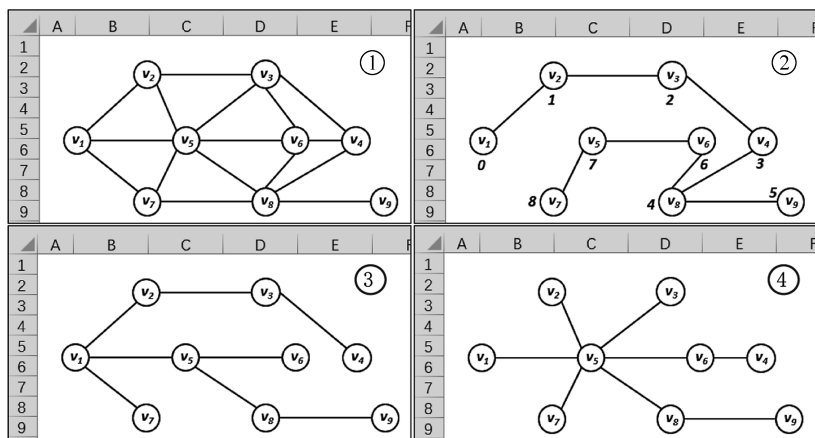


图 5-3 深探法、广探法、破圈法

解 在工作表中选择“插入”→“形状”绘制连通图,然后用深探法、广探法、破圈法求解出生成树,具体方法和步骤如下。

(1) 深探法,如图 5-3②所示。

① 以端点 v_1 为起点并编标号“0”。然后按照与已编号的端点不成圈或避圈的原则沿 v_1 的一个边往下一端点(v_2)给予标号“1”。

② 沿 v_3, v_4, v_8, v_9 点分别给予标号“2, 3, 4, 5”。 v_9 为悬挂点给予标号后返回上一端点 v_8 , 然后沿 v_6, v_5, v_7 继续标号,直到标完所有端点。

(2) 广探法,如图 5-3③所示。

① 取 $u_0 = (v_1, v_2)$ 为生成树的一个边,然后按照与已取边不成圈或避圈的原则继续取相连的边 $u_2 = (v_2, v_3), u_3 = (v_3, v_4)$ 。

② 继续取边 $u_4 = (v_1, v_5), u_5 = (v_5, v_6), u_6 = (v_5, v_8), u_7 = (v_8, v_9), u_8 = (v_1, v_7)$ 。

(3) 破圈法,如图 5-3④所示。

- ① 取圈 (v_1, v_2, v_5, v_1) ,从中去掉边 (v_1, v_2) ,即破掉所取的圈。
- ② 按上述方法逐个破掉其他圈便得到生成树。

5.1.3 最小生成树

通常,在一个连通图可以找出多个不相同的生成树。对于赋权的连通图来说,其生成树所有边的权之和称为生成树的权。最小生成树就是赋权连通图中具有最小权的生成树,简称最小树。在许多网络优化问题中可以运用最小树方法,例如,道路、供电、供水、供气、通信等网络的建造、使用及维护。求解最小树的方法有避圈法和破圈法。

避圈法(Kruskal):首先选取赋权图最小权的边,然后从相连且与已取边不构成圈的未选取边中选择一条权最小的边,直到取得生成树。

破圈法:从赋权图中任取一个圈并去掉其中权最大的边,然后重复取圈去其最大权边的操作,直到赋权图不含圈为止。

例 5-3(胡运权主编《运筹学教程》第 5 版 8.6 题,清华大学出版社) 设计如图 5-4①所示的锅炉房到各座楼铺设暖气管道的路线,使管道总长度最小(单位: m)。

解 先将暖气管道连通图输入工作表并将各节点连线的权(距离)输入相应单元格,然后可用避圈法或破圈法求解,具体方法和步骤如下。

(1) 避圈法(Kruskal),如图 5-4②所示。

① 用最小值函数 MIN 找到最小权值“90”并选取所对应的边 (v_6, v_8) ,然后,按与已选取边不成圈的原则,选取与之相连的最小权边 (v_6, v_7) 。

② 按此依次选取 (v_0, v_6) , (v_0, v_1) , (v_1, v_2) , (v_2, v_3) , (v_3, v_4) , (v_4, v_9) , (v_9, v_{10}) , (v_4, v_5) 。然后,在 S4 输入计算最短管线长度公式,结果为 1540(m)。

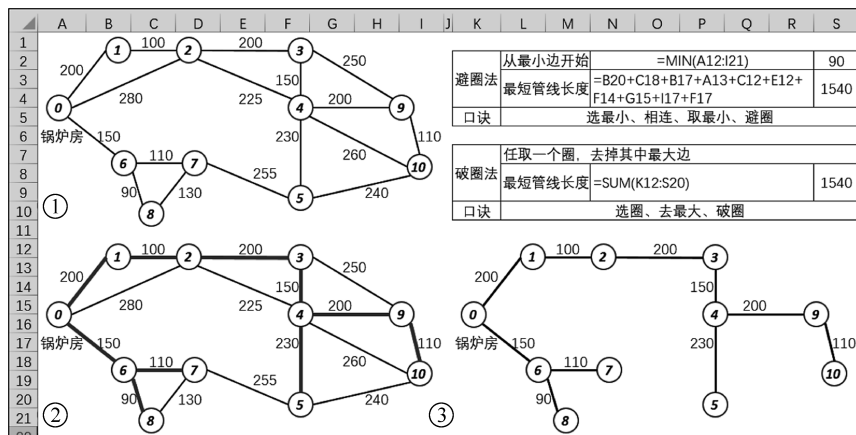


图 5-4 用最小生成树求暖气管道最短路线

(2) 破圈法,如图 5-4③所示。

- ① 选取圈 (v_0, v_1, v_2, v_0) ,去掉其中最大权边 (v_0, v_2) 。
- ② 重复取圈,依次去掉 (v_2, v_4) , (v_3, v_9) , (v_4, v_{10}) , (v_5, v_{10}) , (v_5, v_7) , (v_7, v_8) 。然后,在 S8 输入计算最短管线长度公式,结果为 1540(m)。

5.1.4 最短路问题及 Dijkstra、Floyd 算法

在图论中,最短路指的是两个端点或顶点之间路程(如距离、权等)最短。与最小树不同的是,最短路分析的是连通图中指定的两个端点之间的路程,而不需要经过所有端点。许多优化问题可以使用最短路网络模型,如设备更新、各种管网和建筑布局等。常用的求解最短路方法有 Dijkstra 与 Floyd 算法。

Dijkstra 算法是求解连通图端点间最短路径的方法。该算法是由荷兰阿姆斯特丹大学 Edsger W.Dijkstra 博士于 1959 年提出的。其基本思路是:从源点开始使用贪心方法(Greedy Approach)迭代寻找最短路,在每次更新时忽略较长的距离,最终得到问题的一个最优解。

Floyd 算法是用于求解加权连通图点与点之间最短路径的算法。1962 年,计算机科学家 Robert Floyd 和 Stephen Warshall 分别独立发表了该算法,而在 1959 年,Bernard Roy 曾提出过此算法,因此该算法也称为 Floyd-Warshall 算法、Roy-Warshall 算法、Roy-Floyd 算法或 WFI 算法。Floyd 算法通过比较连通图每对顶点之间所有可能的路径,并在逐步迭代计算中用更短的路径替代已比选的路径,直到求出最优解。

Dijkstra 算法可用于计算连通图中两点之间的最短路径,Floyd 算法则用于求解连通图中所有点之间的最短路径。下面用示例来讲解两种算法。

示例 5-4(胡运权主编《运筹学教程》第 5 版 8.10 题,清华大学出版社) 如图 5-5①所示, v_0 是一仓库, v_9 是商店,用 Dijkstra 算法求解一条从 v_0 到 v_9 的最短路。

解 在工作表中选择“插入”→“形状”绘制连通图,并将每条边的权(距离)输入相应单元格,然后建立计算表并按各端点与 v_0 的层次关系输入端点表头,如图 5-5②中 F1:R1 单元格所示。

(1) 令 $P(v_0)=0$,其余各点均标为未访问,即 $T(v_i)=\infty(i=1,2,\cdots,9)$ 。

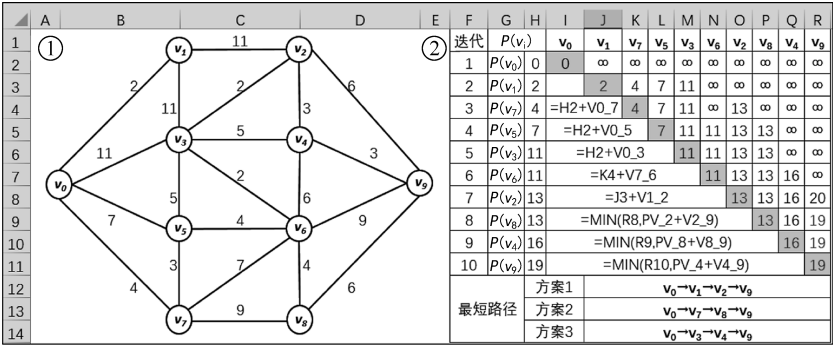


图 5-5 最短路 Dijkstra 算法

(2) 从 v_0 出发,访问与之相连的点,计算 $T(v_1)=\min[T(v_1),P(v_0)+u_{01}]=2$ 。式中, u_{01} 为 v_0 至 v_1 的距离。同理得 $T(v_3)=11,T(v_5)=7,T(v_7)=4$ 。然后比较所有 $T(v_i)$ 找出最小值并赋予 P 标号,即 $P(v_1)=2$ 。

(3) 从新得到 P 标号的 v_1 出发,访问相连的点,计算 $T(v_2)=\min[T(v_2),P(v_1)+u_{12}]=\min(\infty,2+11)=13$, $T(v_3)=\min[T(v_3),P(v_1)+u_{13}]=\min(11,2+11)=11$ 。再次比较所有 $T(v_i)$ 找出最小值并赋予 P 标号,得 $P(v_7)=4$ 。

(4) 重复上述步骤,直到计算出所有端点的 P 标号,如图 5-5②所示。计算得到三条最短路经: $v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow v_9$; $v_0 \rightarrow v_7 \rightarrow v_8 \rightarrow v_9$; $v_0 \rightarrow v_3 \rightarrow v_4 \rightarrow v_9$ 。其最小值为 19。图中给出了各端点 P 标号计算公式。

示例 5-5(胡运权主编《运筹学教程》第 5 版 8.11 题,清华大学出版社) 用 Floyd 算法求图 5-6 中任意两点间的最短路。

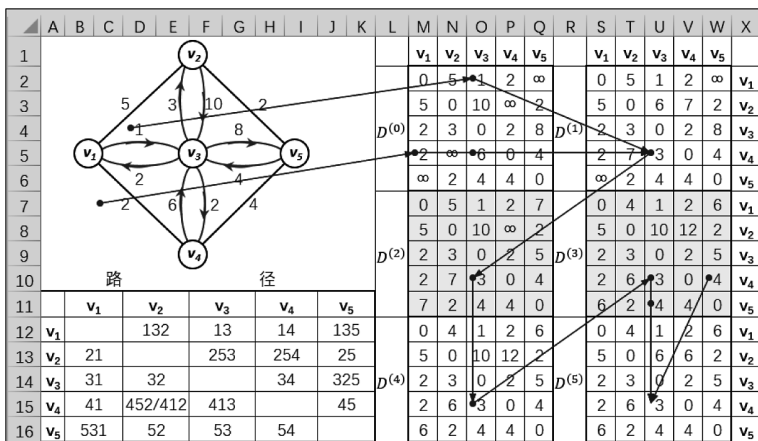


图 5-6 最短路 Floyd 算法

解 本题给出的是有向与无向混合的赋权(距离)连通图,其中,无向连接表示该路径为双向均连通。将各路径权值输入相关单元格后,按如下方法和步骤操作。

(1) 建立权矩阵计算表并输入初始矩阵。将连通图中两端点间可通行路径权值的引用输入矩阵 $D^{(0)}$,并假设不直接相连的端点间路径尚未开通,暂赋值 ∞ ,如图 5-6 中 L1:Q6 单元格所示。

(2) 计算 $D^{(1)}$ 。把 v_1 点看作中转站,其他所有点可以按连通路经到此点中转至另一端点,如果新中转路径赋权之和小于 $D^{(0)}$ 中原路径,则用新路径替换之。例如, $v_2 \rightarrow v_3$ 从 v_1 中转路径为 $v_2 \rightarrow v_1 \rightarrow v_3$,故在 U3 单元格输入公式“=MIN(O3,M3+O2)”,返回值为从 v_1 中转的新路径合计值 6(MIN(10,5+1));将 $v_4 \rightarrow v_3$ 中转路径 $v_4 \rightarrow v_1 \rightarrow v_3$ 的计算公式“=MIN(O5,M5+O2)”输入 U5 单元格,得到返回值为 3,如图 5-6 中 S1:X6 单元格所示。

(3) 按上述(2)分别计算 $D^{(2)}$, $D^{(3)}$, $D^{(4)}$, $D^{(5)}$ 。也就是把 v_i 分别作为中转点,计算每个端点经该点中转到另一端点路径值之和,并与 $D^{(i-1)}$ 中原值比较,取最小值。计算结果如图 5-6 中 L7:X16 单元格所示。每次计算中,只考虑经 v_i 点的中转,无须考虑经 v_i 点后再次中转或其他点的中转。因为,其他端点的中转要么在已计算的成果中被现在采用,要么将在后续迭代点中计算。

(4) 求路径矩阵。在计算 $D^{(i)}$ 时,可以同步记录路径矩阵 $P^{(i)}$ 。在工作表中可使用“公式审核”中“追踪引用单元格”功能追踪路径,如图 5-6 所示,图中折线为 $v_4 \rightarrow v_3$ 最短路经计算公式中引用单元格的追踪线。据此求得的最短路经矩阵如图 5-6 中 A10:K16 单元格所示。

5.1.5 最大流问题

网络或赋权连通图上的最大流量问题(Max.Flow Problem)是网络流最优规划问题。

在现实社会,此类问题普遍存在。例如,交通运输网、能源供应网、通信网和排污管网等。设网络 $G=(V,E,C)$ 的出发点为 v_s , 收点为 v_t , x_{ij} 与 c_{ij} 分别为 i 点到 j 点的流量和容量, P 为出发点 v_s 至收点 v_t 的路径, 则可将最大流问题写成线性规划形式:

$$\begin{aligned} \max \quad & f = \sum_{ij \in P} x_{ij} \\ \text{s.t.} \quad & \begin{cases} x_{ij} \leq c_{ij} \\ \sum_j x_{ij} = \sum_i x_{ji} \\ x_{ij} \geq 0 \end{cases} \end{aligned}$$

1956 年, L.R.Ford Jr. 与 D.R.Fulkerson 发表了网络最大流的标号算法, 因此, 也称为 Ford-Fulkerson 方法。其基本思路是通过对端点标号, 分析路径的可行流, 从而找到增广链 (Augmenting Path), 也就是找出还有通行余量的路径, 然后利用增广链尽可能增加网络流量, 从而求解出网络的最大流。

示例 5-6 (胡运权主编《运筹学教程》第 5 版 8.15 题, 清华大学出版社) 求如图 5-7①所示网络图中最大流, 边上数为 (c_{ij}, f_{ij}) 。要求: ①将最大流问题列出线性规划模型; ②用标号算法求解。

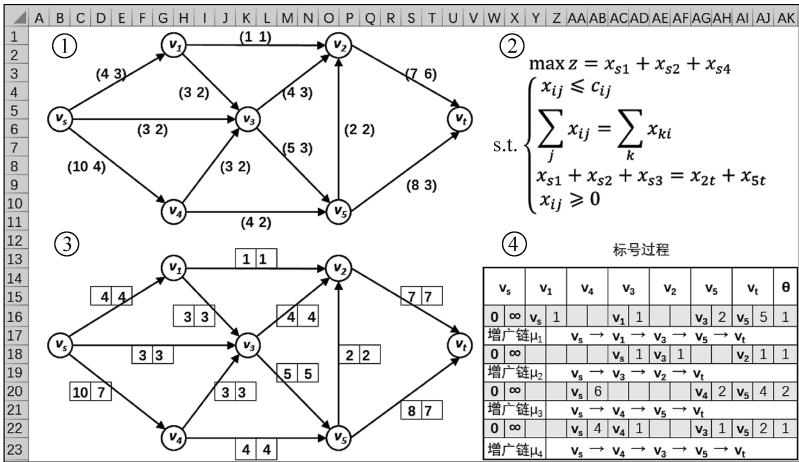


图 5-7 最大流 Ford-Fulkerson 算法 (标号法)

解 先分析问题的目标和约束并按题目要求写出 LP 模型, 然后用标号法 (Ford-Fulkerson) 求解, 具体方法和步骤如下。

(1) 根据题意, 该问题的目标函数为网络上的最大流量, 其约束条件有: 端点间 (边) 的流量小于或等于容量; 中间点的输入量与输出量相等; 发点与收点的流量相等; 流量应大于或等于零。据此, 可写出问题的线性规划模型, 如图 5-7②所示。

$$\begin{aligned} \max \quad & z = x_{s1} + x_{s2} + x_{s3} \\ \text{s.t.} \quad & \begin{cases} x_{ij} \leq c_{ij} \\ \sum_j x_{ij} = \sum_k x_{ki} \\ x_{s1} + x_{s2} + x_{s3} = x_{2t} + x_{5t} \\ x_{ij} \geq 0 \end{cases} \end{aligned}$$

式中, x_{ij} 为 i 到 j 的流量。

(2) 给 v_s 标号 $(0, \infty)$, 检查弧 (v_s, v_1) , 可为 v_1 点标号 $(v_s, 4-3)$ 。

(3) 检查 (v_1, v_2) 和 (v_1, v_3) , 前者已饱和, 不能标号, v_3 可标号 $(v_1, 3-2)$ 。

(4) 检查 (v_3, v_2) 和 (v_3, v_5) 均可标号, 给 v_5 可标号 $(v_3, 5-3)$ 。

(5) 检查 (v_5, v_t) , 可为 v_t 标号 $(v_5, 8-3)$ 。

(6) 求出已标号点容量与流量之差的最小值 $\theta = \min(1, 1, 2, 5) = 1$ 。这样, 就得到一条增广链 $(v_s, v_1, v_3, v_5, v_t)$, 其可调增流量为 1。

(7) 将求出的增广链上所有节点的流量加 θ 。

(8) 重复上述(2), 新找到 3 条增广链: (v_s, v_3, v_2, v_t) 、 (v_s, v_4, v_5, v_t) 、 $(v_s, v_4, v_3, v_5, v_t)$; 重复上述(3), 分别为新增广链调增流量, 如图 5-7③④所示。

(9) 最大流为 $\max z = x_{s1} + x_{s2} + x_{s3} = 4 + 3 + 7 = 14$ 。

5.1.6 最小费用流问题

最小费用流与最大流是密切相关的。在许多实际网络流量问题中, 流是需要费用的, 如交通物流、通信信息流等。如果一个网络包含单位流量费用 d_{ij} , 即 $G = (V, E, C, d)$, 则该网络上一个可行流的最小费用为

$$\min d(f) = \sum_{(v_i, v_j) \in E} d_{ij} f_{ij}$$

式中, f_{ij} 为 i 点到 j 点的流量; v_i, v_j 为端点。

当 f 为最大流时, 便是最小费用最大流问题。网络最小费用流的常用算法有以下两种。

一是原始算法。其基本方法是先算出网络的最大流, 然后通过构建增流网, 找出费用的负回路, 也就是找出可减少费用的路径, 之后, 在不减少给定总流量的情况下, 调整优化网络负回路上的流量分配, 直到网络上不存在负回路, 或者是找不出可减少费用的路径, 则已得到最小费用流, 结束算法。

二是对偶算法。就是从零流 ($f = \{0\}$) 开始构建增流网, 并用最短路方法从中找出一条最小费用增广链, 然后充分利用找出的最小费用流增广链安排流量, 直到找不出最小费用流增广链或已满足流量需要时结束算法。

示例 5-7 (胡运权主编《运筹学教程》第 5 版 8.20 题, 清华大学出版社) 如图 5-8①所示网络中, 有向边旁数字为 (c_{ij}, d_{ij}) , c_{ij} 表示容量, d_{ij} 表示单位流量费用, 试求从 v_s 到 v_t 流值为 6 的最小费用流。

解 网络图中给出的是有向边的容量和单位流量费用, 可用最小费用流对偶算法, 具体方法和步骤如下。

(1) 令 $f^{(0)} = \{0\}$, 即将各边的初始流量均设为 0。

(2) 用 Dijkstra 算法, 求出单位流量费用 d_{ij} 的最短路 (v_s, v_3, v_4, v_t) , 如图 5-8①②所示。将此路径作为最小费用流增广链, 通过比较此链上各边的容量, 求出其最大通行流量 $\theta = 3$, 然后将此链上各边流量由 0 改为 3, 如图 5-8③所示。

(3) 构造有向赋权图 $f^{(1)}$ 。流量为 0 的均画前向弧; (v_s, v_3) 为非饱和边, 需添加前向弧和反向弧; (v_3, v_4) 和 (v_4, v_t) 为饱和弧, 均画为后向弧, 如图 5-8④所示。其中, 前向弧表

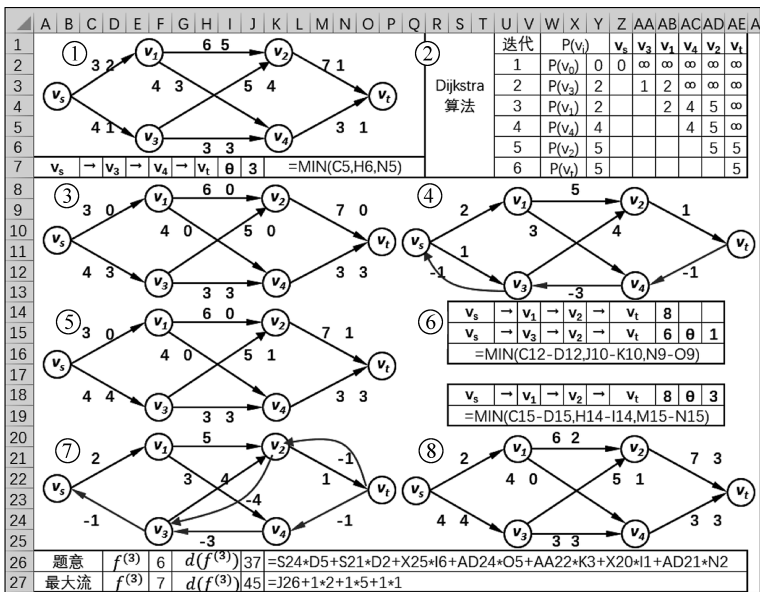


图 5-8 最小费用最大流

示可增加流量,后向弧表示可减少流量。

(4) 求出 $f^{(1)}$ 的最短路。由于网络剩下的可行流路径较少,可直接用枚举法找最短路。找出的最短路径为 (v_s, v_3, v_2, v_t) ,其可调增流量受 (v_s, v_3) 容量限制,只能调增 1 个流量,如图 5-8⑤⑥所示。

(5) 构造有向赋权图 $f^{(2)}$ 。 $(v_s, v_3), (v_3, v_4), (v_4, v_t)$ 为饱和弧,均画为后向弧; (v_3, v_2) 和 (v_2, v_t) 为非饱和弧,均画前向弧和反向弧;其余零流量边均画前向弧,如图 5-8⑦所示。

(6) 求出 $f^{(2)}$ 的最短路。找出的最短路径为 (v_s, v_1, v_2, v_t) ,其可调增流量为 $\theta=3$ 。题目要求“从 v_s 到 v_t 流值为 6 的最小费用流”,故可在此链上调增两个流量,如图 5-8⑧所示。

(7) 经分析,已找不出最小费用增广链。因此,计算得流量值为 6 的最小费用流为 37。网络的最大流为 7,其最小费用为 45。计算公式如图 5-8 中 A26:A27 单元格所示。

5.2 网络计划

网络计划(Network Planning Technology)指的是运用网络图的统筹分析和形象展示功能,编制大型工程或复杂工作进度计划的技术方法,也是计划管理的有效工具。早在 1917 年, Henry Laurence Gantt 统筹工序与持续时间,提出了横道图工程计划管理方法。1950 年,计划评审技术(Program/Project Evaluation and Review Technique, PERT)在大型工程项目中开始得到应用。1956 年, Du Pont Company 在建筑工程项目管理中运用网络计划技术提出了关键线路法(The Critical Path Method, CPM)。1966 年, A. Pritsker 首次提出了基于网络技术的图解评审法(Graphical Evaluation and Review Technique)。至今,网络计划方法在工业、农业、国防和科研等领域得到了广泛应用。本节介绍了网络计划图规则及绘制方法与技巧,网络计划时间参数计算,概率型网络图时间参数计算,网络计划的优化和图解评审法等内容。

5.2.1 网络计划图规则及绘制方法与技巧

网络图是展示工程项目或工作任务全过程的流程图,它包括开工或起始节点、工序或阶段工作及节点、工序或工作持续时间、竣工或结束节点等。绘制网络图需要遵循如下基本规则。

(1) 工序或活动。用一条弧(带指向箭头的直线)连接两个编号节点表示一个完成的工序或活动。不占时间和资源仅表示逻辑关系的虚工序应采用虚线箭头。在弧线的下边(上边)注明工序或活动的名称,上边(下边)则是工序或活动所需的时间。例如,某工程中 A 工序可为



(2) 整项工程或工作任务,从开工到竣工(结束)全过程中的各项工序要按先后顺序和逻辑关系由左至右排列并链接成网络图。图中各工序的节点应由小到大统一编号,最小编号为开工或开始,最大编号为竣工或结束。网络中只能有一个开工或开始的起点编号,一个竣工或结束的终点编号。任一工序首尾编号(i, j)应符合从小至大的原则,即 $j > i$ 。

(3) 网络为从起点至终点的有向图,不能有缺口和回路。

(4) 工序或活动首尾节点 i, j 之间不能有两项和两项以上工序。

(5) 一项工序的紧前工序与紧后工序必须符合逻辑关系。对于两项或多项紧前(紧后)工序,可以使用虚工序正确表示前、后工序关系。

在工作表中可以运用插图与绘图工具绘制网络计划节点图,具体操作方法和技巧如下。

(1) 插入节点圆圈并调整大小、添加数字或文本。

① 单击“插入”→“形状”→“椭圆”,如图 5-9①所示。

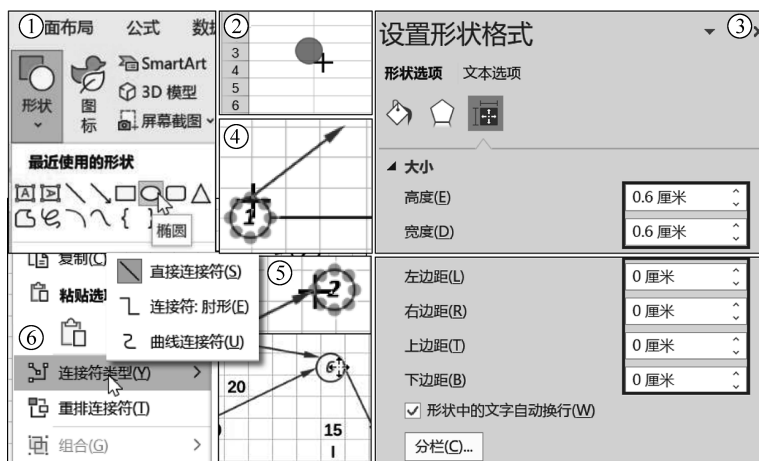


图 5-9 绘制网络图节点及连线

② 将鼠标移动到要绘制节点的位置,按住鼠标左键并拖曳画适度大小圆圈后松开鼠标,如图 5-9②所示。

③ 用鼠标右击画好的圆圈,从鼠标快捷菜单中选择“大小和属性”,打开“设置形状格式”对话框“文本”选项卡。在“高度”与“宽度”框中输入圆圈直径,如 0.6(厘米),并在“文本