最短路问题

Multivac

January 16, 2025

问题 2.5.2： 一只狼、一头山羊和一箩卷心菜都在河的同一侧，摆渡人要将它们运到河对岸. 摆渡人每次只能载一样东西过河. 显然，不管是狼和山羊，还是山羊和卷心菜，都不能在无人监视的情况下留在一起. 问摆渡人怎样把它们运过河去？试为这位摆渡人设计一个最佳的运送方案.

1 最短路问题

给定 x_0 是铁路系统中的一个固定城市，最短路问题就是求从 x_0 到其它城市的最短路线，从图论角度来说，即为：在有正值权 w 的简单有向图（D,w）中，x_0 为 D 的固定点，求根在 x_0 且到每个点距离最小的支撑外向树。

注：相对于最小连接问题，最短路问题与其区别在于有一个固定的根 x_0，同时，考虑的是有向图。

2 Moore-Dijkstra 算法

2.1 历史介绍

Dijkstra 算法，是由荷兰计算机科学家 Edsger Wybe Dijkstra 在 1956 年发现的算法，戴克斯特拉算法使用类似广度优先搜索的方法解决赋权图的单源最短路径问题。Dijkstra 算法原始版本仅适用于找到两个顶点之间的最短路径，后来更常见的变体是固定了一个顶点作为源结点然后找到该顶点到图中所有其它结点的最短路径，产生一个最短路径树。本算法每次取出未访问结点中距离最小的，用该结点更新其他结点的距离。需要注意的是绝大多数的 Dijkstra 算法不能有效处理带有负权边的图。

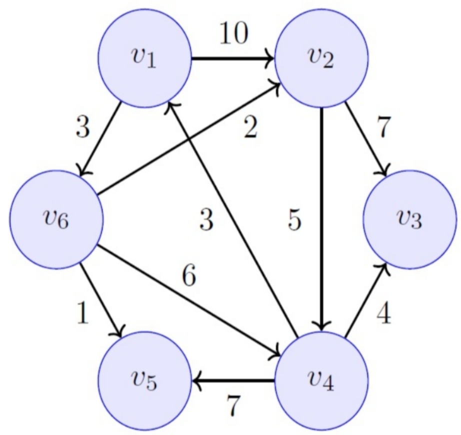

Figure 1: 图 2.2.1

2.2 具体算法

设赋权为 w 的简单有向图 D=（V,E,w）,w_{ij} 为边 e_{ij} 的权重，一般表示点 v_i 到 v_j 的距离。现在我们的任务是：找出从点 s 出发，到 V$\backslash\{s\}$ 中所有的节的最短路径。

算法步骤

- $l\left(x_0\right)=0, l(x)=\infty\left(x \neq x_0\right), S_0=x_0, T_0=x_0$ 且 $k=0$.
- $\left(S_k, \bar{S}_k\right) \neq \emptyset$, 对每个 $x \in N_D^{+}\left(x_k\right) \cap \bar{S}_k$, 用 $\min \left\{l(x), l\left(x_k\right)+w\left(x_k, x\right)\right\}$ 替代 $l(x)$. 取 $x_{k+1} \in N_D^{+}\left(S_k\right) \cap \bar{S}_k$ 和 $x_j(jk) \in S_k$, 使 $\left(x_j, x_{k+1}\right) \in E(D)$, 并且 $l\left(x_{k+1}\right)=\min \left\{l(x): x \in \bar{S}_k\right\}=l\left(x_j\right)+w\left(x_j, x_{k+1}\right)$. 令 $S_{k+1}=S_k \cup\left\{x_{k+1}\right\}, T_{k+1}=T_k+\left(x_j, x_{k+1}\right)$.
- 若 $k=v-1$, 则停止. 若 $k<v-1$ 且 $\left(S_{k+1}, \bar{S}_{k+1}\right) \neq \emptyset$, 则用 $k+1$ 替代 k 并转第 2 步; 否则停止, D 中不存在根在 x_0 的外向支撑树.

例 1：如图 2.2.1 所示，图 G 是有六个顶点的赋权有向图。

现选定 v_1 为起点，Moore-Dijkstra 算法执行如下：

1. 对每个顶点进行标号,取 $x_0=v_1, l(v_1)=0, l(v_i)=\infty,(i=2,3,4,5)$, 如图 a 所示，取 $k=0$, 令 $S_0=x_1, T_0=x_0$.

2. 需要执行五次，详述如下：

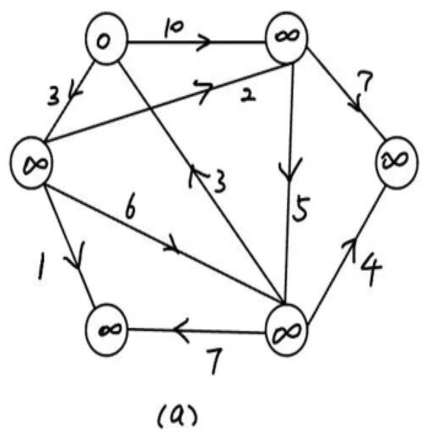

Figure 2: 图 a

(1).$N_D^+(x_0) \cap \bar{S}_0 = N_D^+(S_0) \cap \bar{S}_0 = \{v_2, v_6\}$,

$l(v_2) = \min\{\infty, l(v_0) + \boldsymbol{w}(v_0, v_2)\} = \min\{\infty, 0 + 10\} = 10$

$l(v_6) = \min\{\infty, l(v_0) + \boldsymbol{w}(v_0, v_6)\} = \min\{\infty, 0 + 3\} = 3 = l(x_0) + \boldsymbol{w}(v_0, v_6)$

将 $l(v_2)$ 和 $l(v_6)$ 中的 ∞ 分别修改为 10 和 3. 取 $x_1 = v_6, k = 1$. 令 $S_1 = S_0 \cup \{x_1\} = \{x_0, x_1\}, T_1 = T_0 + (x_0, x_1)$. (如图 b 所示).

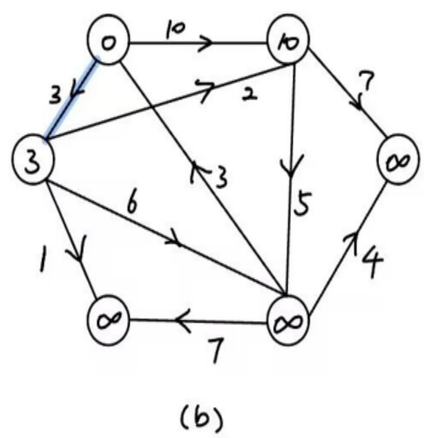

Figure 3: 图 b

(2).$N_D^+(x_1) \cap \bar{S}_1 = N_D^+(S_1) \cap \bar{S}_1 = \{v_2, v_4, v_5\}$,

$l(v_2) = \min\{10, l(v_6) + \boldsymbol{w}(v_6, v_2)\} = \min\{\infty, 3+2\} = 5$

$l(v_4) = \min\{\infty, l(v_6) + \boldsymbol{w}(v_6, v_4)\} = \min\{\infty, 6+3\} = 9 = l(x_1) + \boldsymbol{w}(v_6, v_4)$

$l(v_5) = \min\{\infty, l(v_6) + \boldsymbol{w}(v_6, v_5)\} = \min\{\infty, 1+3\} = 4 = l(x_1) + \boldsymbol{w}(v_6, v_5)$

将 $l(v_2), l(v_4)$ 和 $l(v_5)$ 分别修改为 5,4 和 9. 取 $x_2 = v_5, k = 2$. 令 $S_2 = S_1 \cup \{x_2\} = \{x_0, x_1, x_2\}, T_1 = T_0 + (x_1, x_2)$. (如图 c 所示).

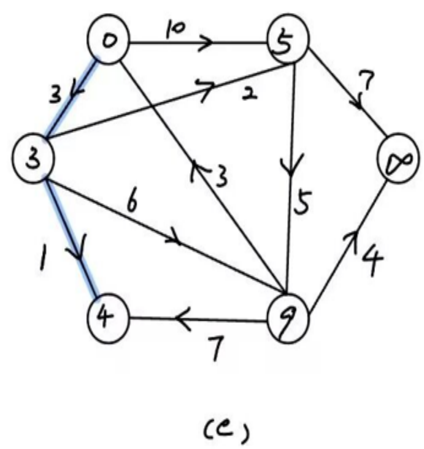

Figure 4: 图 c

(3).$N_D^+(x_2) \cap \bar{S}_2 = \emptyset, N_D^+(S_2) \cap \bar{S}_2 = \{v_2, v_4\}$,

$l(v_2) = 5, l(v_4) = 9$.

取 $x_3 = v_2, k = 3$. 令 $S_3 = S_2 \cup \{x_3\} = \{x_0, x_1, x_2, x_3\}, T_3 = T_2 + (x_1, x_3)$. (如图 d 所示).

(4).$N_D^+(x_3) \cap \bar{S}_3 = N_D^+(S_3) \cap \bar{S}_3 = \{v_3, v_4\}$,

$l(v_3) = \min\{\infty, l(v_2) + \boldsymbol{w}(v_2, v_3)\} = \min\{\infty, 5+7\} = 12$,

$l(v_4) = 9$.

将 $l(v_3)$ 分别修改为 12. 取 $x_4 = v_4, k = 4$. 令 $S_4 = S_3 \cup \{x_4\} = \{x_0, x_1, x_2, x_3, x_4\}, T_4 = T_3 + (x_1, x_4)$. (如图 e 所示).

(5).$N_D^+(x_4) \cap \bar{S}_4 = N_D^+(S_4) \cap \bar{S}_4 = \{v_3\}$,

$l(v_3) = 12$.

取 $x_5 = v_3, k = 5$. 令 $S_5 = S_4 \cup \{x_5\} = \{x_0, x_1, x_2, x_3, x_4, x_5\}, T_5 = T_4 + (x_4, x_5)$. (如图 f 所示).

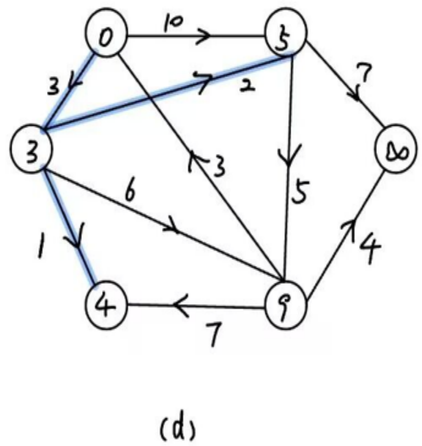

Figure 5: 图 d

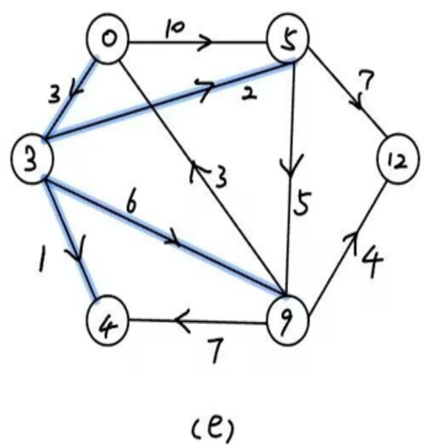

Figure 6: 图 e

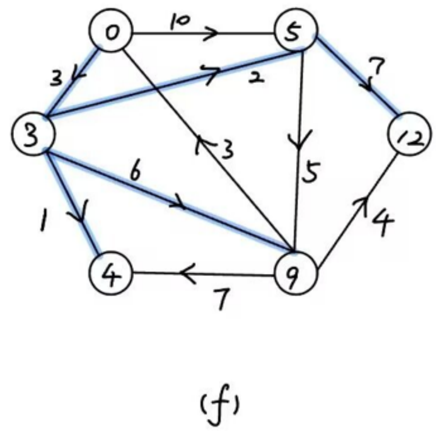

Figure 7: 图 f

3. 算法停止，因为 $k = 5 = 6 - 1$.

3 the solution of Problem 2.5.2

3.1 问题分析

用四维数组 (a, b, c, d) 表示狼,羊,菜,摆渡人的位置状态,其中 $a, b, c, d \in \{0, 1\}$, 0 表示还没过岸，1 表示已经到了河的另一边。

1. 根据狼羊无法共存和羊菜不能共存，从 $2^4 = 16$ 种情况中筛选出可能存在的状态，并视每一个状态为一个顶点，得到顶点集 V;

2. 根据摆渡人每次只能运输 1 件物品（严格来讲小于等于 1 件），推出所有可以直接相互转化的状态对 $(\mathrm{v_i}, \mathrm{v_j})$ ，得到边集 E；

3. 由于状态是可逆的, 所以每条边都是双向的, 因此我们得到了一个无向图(见图 3.1) $\mathrm{G} = (\mathrm{V}, \mathrm{E})$ (实际上也是一个二部图,顶点集可以分成人在河原岸的集合 $\mathrm{X} := \{(\mathrm{a}, \mathrm{b}, \mathrm{c}, 0)\}$ 和人在河对岸的集合 $\mathrm{Y} := \{(\mathrm{a}, \mathrm{b}, \mathrm{c}, 1)\}$ X, Y 中的任两顶点不相邻)；

4. 原问题等价于求从点 $(0, 0, 0, 0)$ 到点 $(1, 1, 1, 1)$ 的最短路径和相应的路径长。

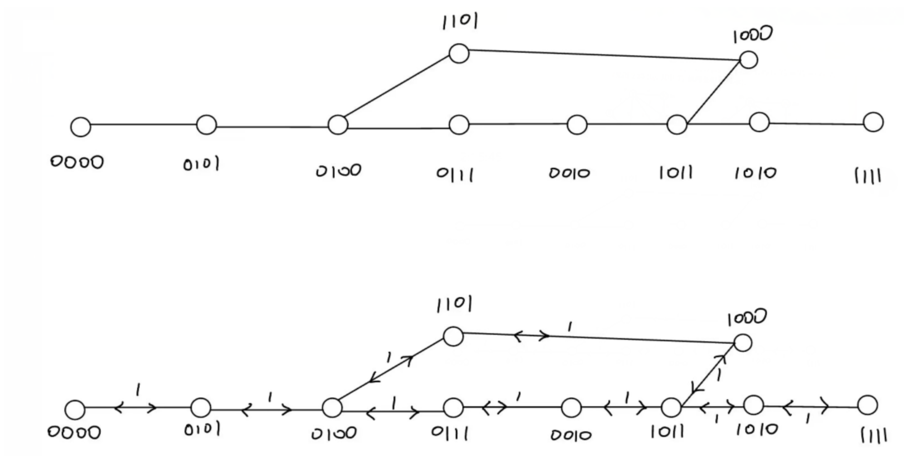

Figure 8: 图 3.1

3.2 solution

由于摆渡人来回路径相同，所以不妨对所有的边赋值为 1，从而得到加权简单有向图（见图 3.1），通过 Moore-Dijkstra 算法可得方案如图 3.2 所示。

代码如下

```
clc,clear %清空当前命令行窗口和工作区变量
a = [1 1 1 1;1 1 1 0;1 1 0 1;1 0 1 1;1 0 1 0;0 1 0 1;0
    1 0 0;0 0 1 0;0 0 0 1;0 0 0 0];%每一行是一个可行状
    态
b = [1 0 0 0;1 1 0 0;1 0 1 0;1 0 0 1]; %每一行是一个转移
    状态
w = zeros(10); %邻接矩阵初始化
for i = 1:9
    for j = i+1:10
        for k = 1:4
            if strfind(xor(a(i,:),b(k,:)),a(j,:)) %
                ok<STRIFCND>
                %strfind(s1,pattern), 因此其意思在 s1 中搜索
                    pattern
                %xor(a,b) 表示异或，当两者都是 0，或两者是非零
                    值时，xor(a,b) 结果为 0；否则，xor(a,b) 结果
```

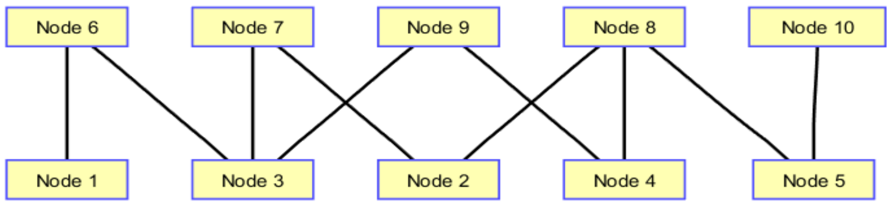

Figure 9: 图 3.2

```
                为 1;
                w(i,j) = 1;
            end
        end
    end
end
w = w';
c = sparse(w); %构造稀疏矩阵
[x,y,z] = graphshortestpath(c,1,10,'Directed',0); %无向图，0/false
h = view(biograph(c,[],'ShowArrows','off','ShowWeights','off')); %画出无向图
Edges = getedgesbynodeid(h); %提取句柄 h 中的边集
set(Edges,'LineColor',[0,0,0]); %为了打印清楚，边画成黑色
set(Edges,'LineWidth',1.5) %线型宽度设置为 1.5
```

References

[1] 徐俊明. 图论及其应用.

[2] https://zhuanlan.zhihu.com/p/129373740.