模型-运筹学-图论-最短路模型【czy】

1. 模型名称

2. 适用范围

最短路问题是网络理论中应用最广泛的问题之一，许多优化问题可以使用这个模型，如管道铺设、设备更新，线路安排等。

最短路问题可以细分成单源最短路问题，两点最短路问题和k短路问题。

3. 定义和性质

1. 定义一：设u,v是图G的顶点，图G的一条u-v链是由有限的顶点，和相邻顶点之间的边组成的交替序列，内部点互不同的链称为路，链中出现的边数称为链的长度。
2. 性质一：最短路最优性原理（最优子结构性质）}两顶点之间的最短路径包括路径上其它顶点的最短路径。

   即如果$p =< v_1, v_2, \ldots, v_k >$是从$v_1$到$v_k$的最短路径，那么任意子路径，
   $p_{ij} =< v_i, v_{i+1}, \ldots, v_j >$，$1 \leq i \leq j \leq k$，也是顶点$v_i$到$v_j$的最短路径。所以这里的分析和动态规划非常类似。事实上，每两点间最短路问题的floyd-warshall算法就是基于动态规划的

4. 模型类型

4.1 单源最短路问题

4.1.1 模型名称

单源最短路问题（single-source shortest paths，SSSP）

4.1.2 定义

- 定义二：从源s到某一顶点v的某一条路称为临时最短路，定义$dist(v)$是临时最短路树中源s到v的长度，称为点v的标号，则$dist(v)$是s到v最短路长度的上界。
- 定义三：定义$prev(v)$为临时最短路树中s-v中v的前驱，w(u,v)是边(u,v)的距离，若$dist(u)+w(u,v)<dist(v)$，称边(u,v)称是紧的，可以对它进行松弛：$dist(v)=dist(u)+w(u,v)$

给定加权图和一个起点s，求s到所有点的最短路，即边权之后最小。求解单源最短路径的算法主要是Dijkstra算法和Bellman-Ford算法，其中Dijkstra算法主要解决所有边的权为非负的单源最短路径问题，而Bellman-Ford算法可以适用于更一般的问题，图中边的权值可以为负。

4.1.3 性质

- 性质二：在单源最短路问题中，最短路一定不存在环。

  如果环上的权和是正的，那么没有必要走这个环；如果权和是负的，那么没有最短路，因为可以沿这个负环兜圈子，则路径长可以为无限小。
- 性质三：设共有n个顶点，如果最短路存在，则每个顶点最多经过一次，因此不超过n-1条边。
- 性质四：临时最短路树的标号均为真实最短路长**当且仅当**图中没有紧边。

由性质三我们可以自然想到通用SSSP算法：不断找出紧边并进行松弛，更新dist和prev，直到不存在紧边。[1]

4.1.4 单源最短路求解算法

4.1.4.1 dijkstra算法

1. 适用范围

   只能解决带有**非负**权值的图中的单源最短路径问题。
2. 思路

   令$d'(v)=min\{d[u]+w(u,v)|u$的最短距离已求出且边(u,v)存在$\}$，设d'中最小的元素为i，则每次令$d_i=d'i$（即求出了i的最短路长度）。每次求出一个d值，n次以后就可以得到所有点的距离。
3. 算法复杂度

   算法的复杂度为$O((m+n)\log m)$

4.1.4.2 Bellman-Ford算法

1. 思路

   它是最优性原理的直接应用。

   这个算法是基于性质一和性质三，路径边数上限为k时的最短路可以由边数上限为k-1时的最短路加一条边来求出。如果一共有n个顶点，最多只需要迭代n-1次就可以求出最短路。
2. 算法伪代码

```
for k:=1 to n-1 do
    for 每条边(u,v) do
        if ($dist(u)<INFINITY$) and ($dist(v)>dist(u)+w(u,v)$) then
            dist(v):=dist(u)+w(u,v)
```

4.2 多源最短路问题

4.2.1 定义

多源最短路问题要求任意两对结点的最短路。[2]

4.2.2 多源最短路求解算法

4.2.2.1 Floyd-warshall算法（适合稠密图）

1. 基本思想

   令d[i,j,k]为从i到j只经过结点1...k的最短路，则这样的路有两种，一种是包含点k的，一种是不包含的。显然不包含的情况等于d[i,j,k-1]，而包含的情况如下图，它等于d[i,k,k-1]+d[k,j,k-1]，它反应了最优性原理。

   其实可以看出这实际上在做动态规划，由于在地推过程中k是递增的，所以只需要一个二维数组就可以了

```
for k:=1 to n do
for i:=1 to n do
for j:=1 to n do
    if ($d[i,k]<无穷$) and ($d(k,j)<无穷$) and ($d[i,k]+d[k,j]<d[i,j]$)
        $d[i,j]:=d[i,k]+d[k,j]$
```

4.2.2.2 Johnson算法 （适合稀疏图）

1. 基本思路

   dijkstra算法速度快，但只适用于非负权图；bellman-ford算法使用任意权图，但时间慢。

   Johnson算法的巧妙之处在于综合了两个算法的优点：用1次bellman-ford对图进行**重加权**，使得每个源点出发的最短路树都保持不变，但权变成非负，然后用n次dijkstra求出每两点间的最短路。

2. 重加权方法

   增加人工结点s，直接到所有点连一条弧，权均为0，然后以s为起点运行bellman-ford，求出dist(v)。如果有负权圈则退出，否则对于原图中的每个条边(u,v)，设新权
   $w'(u,v) = dist(u) + w(u,v) - dist(v)$，则它是非负的。[3]

重加权后每个源点的最短路树都不变。

5. 模型应用

5.1 城市网络

运用最短路原理，解决交通运输管理系统的问题

根据实时交通状况，赋予城市路网中每段线路以时间权值，利用最短路原理，计算出车辆运行时间最短的路线并汇总，通过新媒体及时向广大群众发布信息，指导广大群众选择行驶路线，进一步提高现有道路通行能力，提高道路服务水平，满足现代化高速发展的需求。

5.2 舰船通道

利用图论的经典理论和人群流量理论，研究舰船人员通道路线的优化设计及最优线路选择。

对船舶通道进行路网抽象，建立网络图，然后根据人群流动的相关理论，选取不同拥挤情况下的人员移动速度，从而确定各条路段(包括楼梯)的行程时间。以行程时间作为通道网络的路权，得出路阻矩阵以选择一对起点/终点的最短时间路线为目标,建立最短路径问题的数学模型,利用经典的Floyd算法确定最短路径。

将此方法应用于某舰艇多层甲板的通道网络中,计算结果并进行讨论,最后在此研究的基础上对通道设计相关问题的深化和拓展进行了探讨和总结,并提出设想。

6. 实例（国赛2007B）

6.1 问题表述

分别在（1）只考虑公共汽车线路，（2）同时考虑公共汽车与地铁线路，（3）同时考虑公共汽车、地铁线路及所有站点之间的步行时间，这三种情况下，

根据附录已有公交线路及基本参数设定，针对乘客的不同需求，给出任意两站点之间线路选择的数学模型。

6.2 基本思路

一个通常的思路是从实际交通网络中抽象出相应的图论模型，将站点抽象成图中的节点，用图中的有向边表示两节点之间的直达关系。

每条边设计**两个权值**，一个代表乘坐某线路所花费的**时间**，一个代表乘坐该线路所需要的**费用**，分别得到关于交通网络的时间图和费用图。

针对不同类型的乘客，分别建立**不同优先次序**的**多目标优化模型**。

比如说模型一对于外来人口，对北京线路不熟悉，偏重选择换乘次数少的线路；

模型二对于赶时间的乘客，首先考虑的是时间因素；

模型三是优先考虑更加经济的路线。

6.3 算法设计

由实际情况可知，所建立的图是一个**边权全部为正的图**，

Dijkstra算法就很适合本题。

这里简单陈述一篇比赛论文的具体计算方法。他是运用BFS算法求解模型一，用Dijkstra算法求解模型二、三，在Dijkstra算法中，采取堆的结构把线性数据组织为树形数据，使算法时间复杂度从$O(n^2)$降到$O(n*logn)$

7. 参考资料

1. 数学建模培训营----最短路模型

---

1. 而我们之后要介绍的dijkstra算法和bellman-ford算法其实都是通用SSSP算法的特例。 ↩

2. 虽然它可以通过对每个结点出发做一次SSSP来求解，但不太高效，对于非负权图，可以运行n次；对于任意权图，必须运行n次bellman-ford。 ↩

3. 为什么新权是非负的？因为如果$w'(u,v)<0$，即$dist(u)+w(u,v)<dist(v)$，意味着(u,v)是紧的，与bellman-ford没找到负圈矛盾。 ↩