第六章 图论

6.3 最短路问题

修贤超

机电工程与自动化学院
上海大学

xcxiu@shu.edu.cn

最短路问题

问题描述

- 最短路问题是网络理论中应用最广泛的问题之一，例如设备更新、管道铺设、线路安排、厂区布局等。

最短路问题

问题描述

- 最短路问题是网络理论中应用最广泛的问题之一，例如设备更新、管道铺设、线路安排、厂区布局等。
- 设 $G=(V,E)$ 为连通图，图中各边 (v_i,v_j) 有权 l_{ij}（$l_{ij}=\infty$ 表示 v_i,v_j 之间没有边），v_s,v_t 为图中任意两点，求一条路 μ，使它为从 v_s 到 v_t 的所有路中总权最短。即: $L(\mu)=\sum\limits_{(v_i,v_j)\in\mu} l_{ij}$ 最小。

最短路问题

问题描述

- 最短路问题是网络理论中应用最广泛的问题之一，例如设备更新、管道铺设、线路安排、厂区布局等。
- 设 $G=(V,E)$ 为连通图，图中各边 (v_i,v_j) 有权 l_{ij} （$l_{ij}=\infty$ 表示 v_i,v_j 之间没有边），v_s,v_t 为图中任意两点，求一条路 μ，使它为从 v_s 到 v_t 的所有路中总权最短。即: $L(\mu)=\sum\limits_{(v_i,v_j)\in\mu} l_{ij}$ 最小。
- Dijkstra 算法是在 1959 年提出来的。目前公认，在所有的权 $w_{ij}\geq 0$ 时，这个算法是寻求最短路问题最好的算法。并且，这个算法实际上也给出了寻求从一个始定点 v_s 到任意一个点 v_j 的最短路。

最短路问题

Dijkstra 算法

- 给始点 v_s 以 P 标号 $P(v_s)=0$，这表示从 v_s 到 v_s 的最短距离为 0，其余节点均给 T 标号，$T(v_i)=+\infty$ $(i=2,3,\ldots,n)$。

最短路问题

Dijkstra 算法

- 给始点 v_s 以 P 标号 $P(v_s)=0$，这表示从 v_s 到 v_s 的最短距离为 0，其余节点均给 T 标号，$T(v_i)=+\infty\ (i=2,3,\ldots,n)$。
- 设节点 v_i 为刚得到 P 标号的点，考虑点 v_j，其中 $(v_i,v_j)\in E$，且 v_j 为 T 标号。对 v_j 的 T 标号进行如下修改:

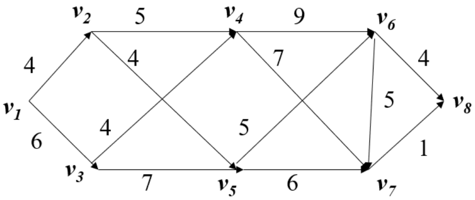
$$T(v_j)=\min[T(v_j),P(v_j)+l_{ij}]$$

- 比较所有具有 T 标号的节点，把最小者改为 P 标号，即:

$$P(v_k)=\min[T(v_i)]$$

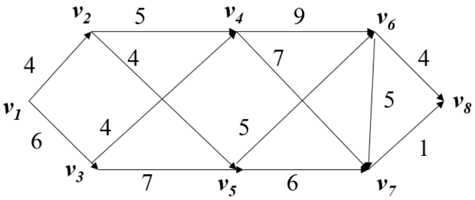
  当存在两个以上最小者时，可同时改为 P 标号。若全部节点均为 P 标号，则停止，否则用 v_k 代替 v_i，返回上一步。

- 求下图从 v_1 到 v_8 的最短路



最短路问题

- 求下图从 v_1 到 v_8 的最短路



- 首先给 v_1 以 P 标号，$P(v_1)=0$; 给其余所有点 T 标号，$T(v_i)=+\infty$ $(i=2,3,\ldots,8)$。
- $T(v_2)=\min[T(v_2),P(v_1)+l_{12}]=\min[+\infty,0+4]=4$
  $T(v_3)=\min[T(v_3),P(v_1)+l_{13}]=\min[+\infty,0+6]=6$
  比较所有 T 标号，$T(v_2)$ 最小，令 $P(v_2)=4$，并记录路径 (v_1,v_2)。
- $T(v_4)=\min[T(v_4),P(v_2)+l_{24}]=\min[+\infty,4+5]=9$
  $T(v_5)=\min[T(v_5),P(v_2)+l_{25}]=\min[+\infty,4+4]=8$
  比较所有 T 标号，$T(v_3)$ 最小，令 $P(v_3)=6$，并记录路径 (v_1,v_3)。

最短路问题

- **求下图从 v_1 到 v_8 的最短路**
  - $T(v_4)=\min[T(v_4),P(v_3)+l_{34}]=\min[9,4+9]=9$
    $T(v_5)=\min[T(v_5),P(v_3)+l_{35}]=\min[8,6+7]=8$
    比较所有 T 标号，$T(v_5)$ 最小，令 $P(v_5)=8$，并记录路径 (v_2,v_3)。
  - $T(v_6)=\min[T(v_6),P(v_5)+l_{56}]=\min[+\infty,8+5]=13$
    $T(v_7)=\min[T(v_7),P(v_5)+l_{57}]=\min[+\infty,8+6]=14$
    比较所有 T 标号，$T(v_4)$ 最小，令 $P(v_4)=9$，并记录路径 (v_2,v_4)。
  - $T(v_6)=\min[T(v_6),P(v_4)+l_{46}]=\min[13,9+9]=13$
    $T(v_7)=\min[T(v_7),P(v_4)+l_{47}]=\min[14,9+7]=14$
    比较所有 T 标号，$T(v_6)$ 最小，令 $P(v_6)=13$，并记录路径 (v_5,v_6)。
  - $T(v_7)=\min[T(v_6),P(v_6)+l_{67}]=\min[14,13+5]=14$
    $T(v_8)=\min[T(v_8),P(v_6)+l_{68}]=\min[+\infty,13+4]=17$
    比较所有 T 标号，$T(v_7)$ 最小，令 $P(v_7)=14$，并记录路径 (v_7,v_8)。
  - $T(v_8)=\min[T(v_8),P(v_7)+l_{78}]=\min[17,14+1]=15$
    因为只有一个 T 标号 $T(v_8)$ 最小，令 $P(v_8)=15$, 并记录路径 (v_7,v_8)，v_1 到 v_8 之最短路为: $v_1\to v_2\to v_5\to v_7\to v_8$

最短路问题

- Floyd 算法
  - 可直接求出网络中任意两点间的最短路。

最短路问题

Floyd 算法

- 可直接求出网络中任意两点间的最短路。
- 令网络的权矩阵为 $D=(d_{ij})_{n\times n}$, l_{ij} 为 v_i 到 v_j 的距离

$$d_{ij}=\begin{cases} l_{ij} & \text{当}(v_i,v_j)\in E \\ \infty & \text{其他} \end{cases}$$

最短路问题

Floyd 算法

- 可直接求出网络中任意两点间的最短路。
- 令网络的权矩阵为 $D=(d_{ij})_{n\times n}$，l_{ij} 为 v_i 到 v_j 的距离

$$d_{ij}=\begin{cases} l_{ij} & \text{当}(v_i,v_j)\in E \\ \infty & \text{其他} \end{cases}$$

- 算法基本步骤
  - 输入权矩阵 $D^{(0)}=D$
  - 计算 $D^{(k)}=(d_{ij}^{(k)})_{n\times n}$ $(k=1,2,\ldots,n)$，其中
    $d_{ij}=\min[d_{ij}^{(k-1)},d_{ik}^{(k-1)}+d_{kj}^{k-1}]$
  - $D^{(n)}=(d_{ij}^{(n)})_{n\times n}$ 中元素 $d_{ij}^{(n)}$ 就是 v_i 到 v_j 的最短路长。

最短路问题

小结

- Dijkstra 算法
  - 求无负权网络最短路问题的最好方法
  - 指定两点间的最短路
  - 标号法
- Floyd 算法
  - 任意两点间的最短路

Q&A

Thank you!

感谢您的聆听和反馈