

5-3 最短路径问题

第五章：图论 · 带权图与 Dijkstra 算法

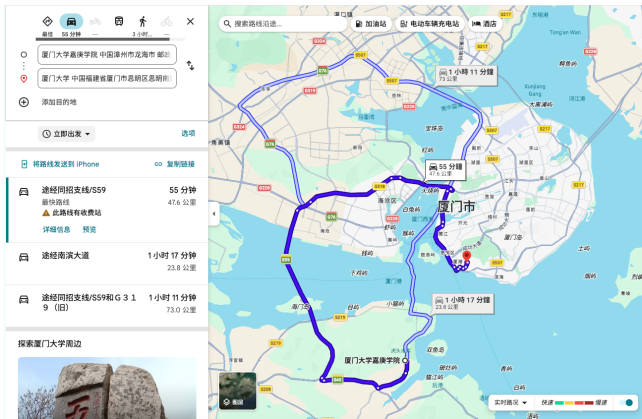
张 振



厦门大学嘉庚学院

XIAMEN UNIVERSITY TAN KAH KEE COLLEGE

- 1 带权图
- 2 Dijkstra 算法的基本想法
- 3 Dijkstra 算法的执行步骤



- 多条候选路线。
- 同一网络中的不同路径。
- 时间、距离、费用可作为权重。

算法入口

如何在带权图中找最短路径？

给图中的边加上距离、时间或费用

PART 1

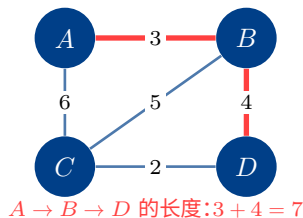
图不仅要描述“是否相连”，还要描述“通行代价”。

定义

给每条边 e 指定实数 $W(e)$ ，所得图称为 **带权图**。

路径长度为各边权重之和：

$$W(P) = \sum_{i=1}^k W(e_i).$$



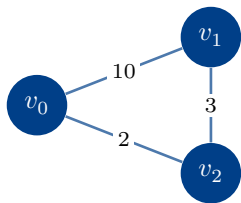
- 顶点:地点或路口。
- 边:可直接通行的道路。
- 权重:通行代价。

最短路径问题

从起点到终点,找权重之和最小的路径。

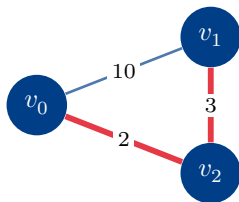
从“目前最近的点”开始逐步扩展

PART 2



题: 思考

从 v_0 到 v_1 , 直接走还是绕行更短? 先确定哪个顶点?



- 从 v_0 出发, 得到两个临时距离。

$v_1 : 10$, 来自 v_0 ,

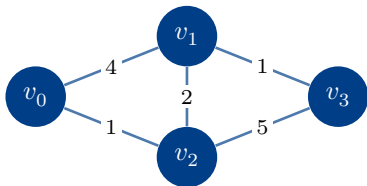
$v_2 : 2$, 来自 v_0 .

- 比较 $2 < 10$, 先确定 v_2 。

- 经 v_2 到 v_1 更短, 改写 v_1 的标号。

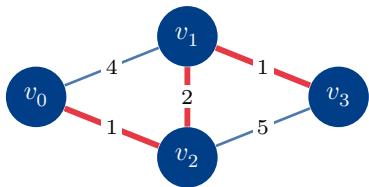
$v_1 : 10 \rightarrow 5$, 来自 v_2

例 2:先确定当前最近的点



■ 题:思考

从 v_0 出发,先确定 v_1 还是 v_2 ?



- 从 v_0 探测,先确定 v_2 。

$v_1 : 4$, 来自 v_0 ,

$v_2 : 1$, 来自 v_0 .

- 经 v_2 改进 v_1 。

$v_1 : 4 \rightarrow 3$, 来自 v_2

同时得到 v_3 的临时距离:

$v_3 : 1 + 5 = 6$, 来自 v_2

- 再由 v_1 改进 v_3 。

$v_3 : 6 \rightarrow 4$, 来自 v_1

前面的动作可以统一写成顶点标号：

Dijkstra 算法为每个顶点维护一个标号

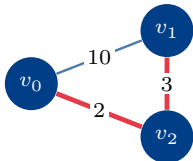
$$(l_i, p_i).$$

- l_i : 当前已知最短距离。
- p_i : 前驱顶点。
- λ : 没有前驱。
- $*$: 永久标号。

Dijkstra 算法的三个动作

1. 初始: 设定起点。
2. 探测: 更新邻点。
3. 固定: 选择最小临时标号。

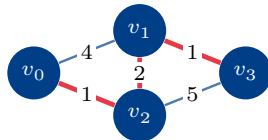
例 1



t	v_0	v_1	v_2
1	$(0, \lambda)^*$	$(+\infty, \lambda)$	$(+\infty, \lambda)$
2		$(10, v_0)$	$(2, v_0)^*$
3		$(5, v_2)^*$	

$$v_0 \rightarrow v_2 \rightarrow v_1, \quad W = 5.$$

例 2



t	v_0	v_1	v_2	v_3
1	$(0, \lambda)^*$	$(+\infty, \lambda)$	$(+\infty, \lambda)$	$(+\infty, \lambda)$
2		$(4, v_0)$	$(1, v_0)^*$	$(+\infty, \lambda)$
3		$(3, v_2)^*$		$(6, v_2)$
4				$(4, v_1)^*$

$$v_0 \rightarrow v_2 \rightarrow v_1 \rightarrow v_3, \quad W = 4.$$

洪峰到达预测

带权图 $\rightarrow$ 河流管网 边权 $\rightarrow$ 流经时间

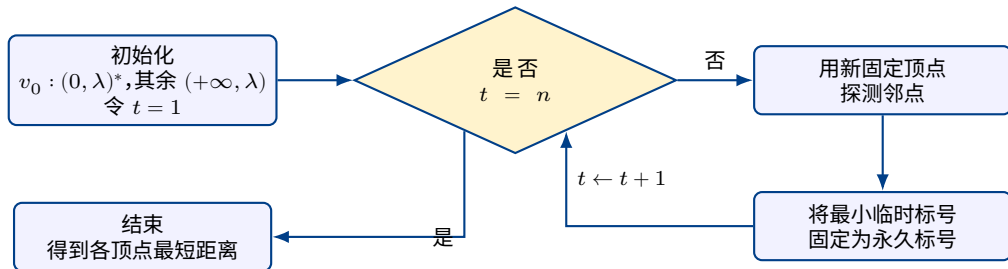
- **并发蔓延**:洪峰从源点 S 向外同步推演。
- **永久固定**:首次抵达节点(等价于确定最短路径)。
- **目标锚定**:最早到达终点 T 的水流轨迹。

► 动画演示

[点击播放视频](#)

探测更新，选择并固定，直到所有顶点完成

PART 3



探测测试

若经过新固定顶点更短,就改写标号。

固定永久标号

从临时标号中选最小者,将其固定为永久标号。

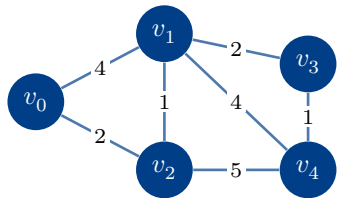
给定非负带权图,求从 v_0 出发的最短路径。

1. 初始化: $v_0 : (0, \lambda)^*$,其余为 $(+\infty, \lambda)$ 。
2. 探测:若经过 v_i 到 v_j 更短,则更新

$$(l_j, p_j) \leftarrow \begin{cases} (l_i + W(v_i, v_j), v_i), & l_i + W(v_i, v_j) < l_j, \\ (l_j, p_j), & l_i + W(v_i, v_j) \geq l_j. \end{cases}$$

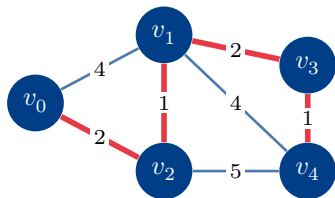
3. 固定:选最小临时标号,改为永久标号。

每步固定一个顶点,直到所有顶点完成。



题:练习

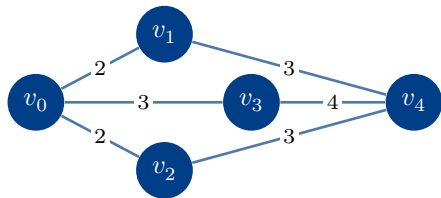
以 v_0 为起点,填写 Dijkstra 标号表。



t	v_0	v_1	v_2	v_3	v_4
1	$(0, \lambda)^*$	$(+\infty, \lambda)$	$(+\infty, \lambda)$	$(+\infty, \lambda)$	$(+\infty, \lambda)$
2		$(4, v_0)$	$(2, v_0)^*$	$(+\infty, \lambda)$	$(+\infty, \lambda)$
3		$(3, v_2)^*$		$(+\infty, \lambda)$	$(7, v_2)$
4				$(5, v_1)^*$	$(7, v_2)$
5					$(6, v_3)^*$

回溯前驱, 得到 v_0 到 v_4 的最短路径:

$$v_0 \rightarrow v_2 \rightarrow v_1 \rightarrow v_3 \rightarrow v_4, \quad W = 6.$$

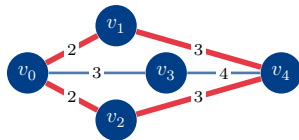


题:练习

以 v_0 为起点,填写 Dijkstra 标号表,并写出从 v_0 到 v_4 的所有最短路径。

提示

当两个临时标号相等时,先固定哪个顶点?



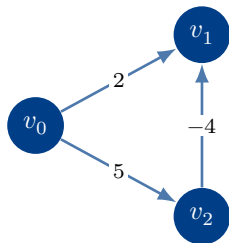
红色边给出两条同长最短路径。

t	v_0	v_1	v_2	v_3	v_4
1	$(0, \lambda)^*$	$(+\infty, \lambda)$	$(+\infty, \lambda)$	$(+\infty, \lambda)$	$(+\infty, \lambda)$
2		$(2, v_0)^*$	$(2, v_0)$	$(3, v_0)$	$(+\infty, \lambda)$
3			$(2, v_0)^*$	$(3, v_0)$	$(5, \{v_1, v_2\})$
4				$(3, v_0)^*$	$(5, \{v_1, v_2\})$
5					$(5, \{v_1, v_2\})^*$

若只求一条路径,相等时可任选;若求所有路径,相等时保留多个前驱。

到 v_4 的两个前驱都可达到距离 5:

$$v_0 \rightarrow v_1 \rightarrow v_4, \quad v_0 \rightarrow v_2 \rightarrow v_4, \quad W = 5.$$



失效原因

负权边使“绕行后只会更长”失效。

■ 初始: $v_1 : 2, v_2 : 5$ 。

■ 先固定 v_1 : 认定 $d(v_1) = 2$ 。

■ 再固定 v_2 : 发现

$$v_0 \rightarrow v_2 \rightarrow v_1 : 5 + (-4) = 1.$$

■ 矛盾: v_1 已固定, 算法不会回头更新。

核心思想

每一步都固定当前最近的临时顶点,再用它改进邻点。

补充观点

完整执行后,可得到起点到所有可达顶点的最短距离。

- 贪心策略:当前最近。
- 临时标号:还可能变。
- 永久标号:不再改变。
- 前驱顶点:用于回溯。
- 非负权值:保证固定有效。

局限与拓展

Dijkstra 算法不能处理**负权边**。含负权边的图需使用 Bellman-Ford 等算法。

本节结束