计算机问题求解 – 论题3-8
- 单源最短通路算法

2016年10月26日
什么是最短通路问题？

In a *shortest-paths problem*, we are given a weighted, directed graph \( G = (V, E) \), with weight function \( w : E \to \mathbb{R} \) mapping edges to real-valued weights. The *weight* \( w(p) \) of path \( p = (v_0, v_1, \ldots, v_k) \) is the sum of the weights of its constituent edges:

\[
    w(p) = \sum_{i=1}^{k} w(v_{i-1}, v_i).
\]

We define the *shortest-path weight* \( \delta(u, v) \) from \( u \) to \( v \) by

\[
    \delta(u, v) = \begin{cases} 
    \min\{w(p) : u \xrightarrow{p} v\} & \text{if there is a path from } u \text{ to } v, \\
    \infty & \text{otherwise}.
    \end{cases}
\]

A *shortest path* from vertex \( u \) to vertex \( v \) is then defined as any path \( p \) with weight \( w(p) = \delta(u, v) \).
问题2：
为什么说可以将单源最短路问题的解看成一个树？你认为这个树与两种图遍历搜索树相比，更可能象哪一个？
As in breadth-first search, we shall be interested in the \textbf{predecessor subgraph} $G_\pi = (V_\pi, E_\pi)$ induced by the $\pi$ values. Here again, we define the vertex set $V_\pi$ to be the set of vertices of $G$ with non-NIL predecessors, plus the source $s$:

$$V_\pi = \{v \in V : v.\pi \neq \text{NIL}\} \cup \{s\}.$$ 

The directed edge set $E_\pi$ is the set of edges induced by the $\pi$ values for vertices in $V_\pi$:

$$E_\pi = \{(v.\pi, v) \in E : v \in V_\pi - \{s\}\}.$$ 

A \textbf{shortest-paths tree} rooted at $s$ is a directed subgraph $G' = (V', E')$, where $V' \subseteq V$ and $E' \subseteq E$, such that

1. $V'$ is the set of vertices reachable from $s$ in $G$,
2. $G'$ forms a rooted tree with root $s$, and
3. for all $v \in V'$, the unique simple path from $s$ to $v$ in $G'$ is a shortest path from $s$ to $v$ in $G$.

\textbf{Predecessor-subgraph property} (Lemma 24.17)

Once $v.d = \delta(s, v)$ for all $v \in V$, the predecessor subgraph is a shortest-paths tree rooted at $s$. 
问题3：
能否借助上图说明最简单的greedy策略不一定能正确解决最短通路问题？这是单源最短通路问题具有“最优子结构”矛盾吗？
问题4：
具有负值权的回路对于单源最短通路问题的解有什么影响？非负值权的回路呢？
问题5：
在本章中介绍的算法基本思路是一样的，那是什么？
“预估” 与 “修正”

```plaintext
INITIALIZE-SINGLE-SOURCE(G, s)
1  for each vertex v ∈ G.V
2    v.d = ∞
3    v.π = NIL
4  s.d = 0

RELAX(u, v, w)
1  if v.d > u.d + w(u, v)
2    v.d = u.d + w(u, v)
3    v.π = u
```
BELLMAN-FORD \((G, w, s)\)

1. **INITIALIZE-SINGLE-SOURCE** \((G, s)\)
2. \(\text{for} \ i = 1 \ \text{to} \ |G.V| - 1\)
3. \(\text{for each edge} \ (u, v) \ \in G.E\)
4. \(\text{RELAX} \ (u, v, w)\)
5. \(\text{for each edge} \ (u, v) \ \in G.E\)
6. \(\text{if} \ v.d > u.d + w(u, v)\)
7. \(\text{return} \ \text{FALSE}\)
8. \(\text{return} \ \text{TRUE}\)
问题6：Relax中的“修正”到底在干什么？
当我们有\(u.d\)这么一个预估值后，\(v.d\)这个预估值必须小于\(u.d+w(u,v)\)（三角不等式），如果\(relax\)时不小于，修正\(v.d\)为\(u.d+w(u,v)\)
修正后的\(v.d\)满足三角不等式的可能性大大提高
Lemma 24.10 (Triangle inequality)

Let $G = (V, E)$ be a weighted, directed graph with weight function $w : E \rightarrow \mathbb{R}$ and source vertex $s$. Then, for all edges $(u, v) \in E$, we have

$$\delta(s, v) \leq \delta(s, u) + w(u, v).$$

Proof Suppose that $p$ is a shortest path from source $s$ to vertex $v$. Then $p$ has no more weight than any other path from $s$ to $v$. Specifically, path $p$ has no more weight than the particular path that takes a shortest path from source $s$ to vertex $u$ and then takes edge $(u, v)$. 

Lemma 24.11 (Upper-bound property)
Let $G = (V, E)$ be a weighted, directed graph with weight function $w : E \rightarrow \mathbb{R}$. Let $s \in V$ be the source vertex, and let the graph be initialized by INITIALIZE-SINGLE-SOURCE$(G, s)$. Then, $v.d \geq \delta(s, v)$ for all $v \in V$, and this invariant is maintained over any sequence of relaxation steps on the edges of $G$. Moreover, once $v.d$ achieves its lower bound $\delta(s, v)$, it never changes.

**Proof** We prove the invariant $v.d \geq \delta(s, v)$ for all vertices $v \in V$ by induction over the number of relaxation steps.

For the basis, $v.d \geq \delta(s, v)$ is certainly true after initialization, since $v.d = \infty$ implies $v.d \geq \delta(s, v)$ for all $v \in V - \{s\}$, and since $s.d = 0 \geq \delta(s, s)$ (note that $\delta(s, s) = -\infty$ if $s$ is on a negative-weight cycle and $0$ otherwise).

For the inductive step, consider the relaxation of an edge $(u, v)$. By the inductive hypothesis, $x.d \geq \delta(s, x)$ for all $x \in V$ prior to the relaxation. The only $d$ value that may change is $v.d$. If it changes, we have

$$v.d = u.d + w(u, v)$$

$$\geq \delta(s, u) + w(u, v) \quad \text{(by the inductive hypothesis)}$$

$$\geq \delta(s, v) \quad \text{(by the triangle inequality)},$$

and so the invariant is maintained.

To see that the value of $v.d$ never changes once $v.d = \delta(s, v)$, note that having achieved its lower bound, $v.d$ cannot decrease because we have just shown that $v.d \geq \delta(s, v)$, and it cannot increase because relaxation steps do not increase $d$ values.

既不会再减小，也不会增大。
问题6：
“修正”最终“可能”导致\(\nu.d = \delta(s,\nu)\)。但“可能”怎么能变成“一定”？
如果 $u_i.d = \delta(s, u_i)$, 我们在某一轮 relax 中对所有 $(u_i, v)$ 边进行 relax, 会有什么结果？

必定在某边 $(U_i, V)$ 的 relax 中，$v.d = \delta(s, v)$ 并且在之后轮次 relax 中，$v.d$ 不会改变。
Lemma 24.14 (Convergence property)

Let \( G = (V, E) \) be a weighted, directed graph with weight function \( w : E \to \mathbb{R} \), let \( s \in V \) be a source vertex, and let \( s \leadsto u \to v \) be a shortest path in \( G \) for some vertices \( u, v \in V \). Suppose that \( G \) is initialized by INITIALIZE-SINGLE-SOURCE\((G, s)\) and then a sequence of relaxation steps that includes the call \( \text{RELAX}(u, v, w) \) is executed on the edges of \( G \). If \( u.d = \delta(s, u) \) at any time prior to the call, then \( v.d = \delta(s, v) \) at all times after the call.

**Proof** By the upper-bound property, if \( u.d = \delta(s, u) \) at some point prior to relaxing edge \((u, v)\), then this equality holds thereafter. In particular, after relaxing edge \((u, v)\), we have

\[
\begin{align*}
  v.d & \leq u.d + w(u, v) & & \text{(by Lemma 24.13)} \\
  & = \delta(s, u) + w(u, v) \\
  & = \delta(s, v) & & \text{(by Lemma 24.1)}.
\end{align*}
\]

By the upper-bound property, \( v.d \leq \delta(s, v) \), from which we conclude that \( v.d = \delta(s, v) \), and this equality is maintained thereafter. \( \blacksquare \)
在刚刚这一轮中，我们relax\((ui,v)\)，得到\(v.d=\delta(s,v)\), \(ui.d=\delta(s,ui)\)何时得到？

\[ui.d=\delta(s,ui)\]必定在前面的某轮relax中完成

\[ui.\pi.d=\delta(s, ui.\pi)\]必定在更前面的某轮relax中完成

其实，一旦完成收敛，最短路已经形成
Lemma 24.15 (Path-relaxation property)

Let \( G = (V, E) \) be a weighted, directed graph with weight function \( w : E \to \mathbb{R} \), and let \( s \in V \) be a source vertex. Consider any shortest path \( p = (v_0, v_1, \ldots, v_k) \) from \( s = v_0 \) to \( v_k \). If \( G \) is initialized by \textsc{Initialize-Single-Source}(\( G, s \)) and then a sequence of relaxation steps occurs that includes, in order, relaxing the edges \((v_0, v_1), (v_1, v_2), \ldots, (v_{k-1}, v_k)\), then \( v_k \cdot d = \delta(s, v_k) \) after these relaxations and at all times afterward. This property holds no matter what other edge relaxations occur, including relaxations that are intermixed with relaxations of the edges of \( p \).

\textbf{Proof} \quad \text{We show by induction that after the} \ i \text{th edge of path} \ p \text{ is relaxed, we have} \ v_i \cdot d = \delta(s, v_i). \text{For the basis,} i = 0, \text{and before any edges of} \ p \text{ have been relaxed, we have from the initialization that} v_0 \cdot d = s \cdot d = 0 = \delta(s, s). \text{By the upper-bound property, the value of} s \cdot d \text{ never changes after initialization.}

\text{For the inductive step, we assume that} v_{i-1} \cdot d = \delta(s, v_{i-1}), \text{and we examine what happens when we relax edge} (v_{i-1}, v_i). \text{By the convergence property, after relaxing this edge, we have} v_i \cdot d = \delta(s, v_i), \text{and this equality is maintained at all times thereafter.} \quad \blacksquare
这会影响什么？
**Lemma 24.2**
Let $G = (V, E)$ be a weighted, directed graph with source $s$ and weight function $w : E \to \mathbb{R}$, and assume that $G$ contains no negative-weight cycles that are reachable from $s$. Then, after the $|V| - 1$ iterations of the for loop of lines 2–4 of BELLMAN-FORD, we have $v.d = \delta(s, v)$ for all vertices $v$ that are reachable from $s$.

**Proof** We prove the lemma by appealing to the path-relaxation property. Consider any vertex $v$ that is reachable from $s$, and let $p = \langle v_0, v_1, \ldots, v_k \rangle$, where $v_0 = s$ and $v_k = v$, be any shortest path from $s$ to $v$. Because shortest paths are simple, $p$ has at most $|V| - 1$ edges, and so $k \leq |V| - 1$. Each of the $|V| - 1$ iterations of the for loop of lines 2–4 relaxes all $|E|$ edges. Among the edges relaxed in the $i$th iteration, for $i = 1, 2, \ldots, k$, is $(v_{i-1}, v_i)$. By the path-relaxation property, therefore, $v.d = v_k.d = \delta(s, v_k) = \delta(s, v)$.  

为什么？
Bellman-Ford ($G, w, s$)
1. Initialize-Single-Source($G, s$)
2. for $i = 1$ to $|G.V| - 1$
3. for each edge $(u, v) \in G.E$
4. Relax($u, v, w$)
5. for each edge $(u, v) \in G.E$
6. if $v.d > u.d + w(u, v)$
7. return FALSE
8. return TRUE

Suppose that graph $G$ contains no negative-weight cycles that are reachable from the source $s$. We first prove the claim that at termination, $v.d = \delta(s, v)$ for all vertices $v \in V$. If vertex $v$ is reachable from $s$, then Lemma 24.2 proves this claim. If $v$ is not reachable from $s$, then the claim follows from the no-path property. Thus, the claim is proven. The predecessor-subgraph property, along with the claim, implies that $G_{\pi}$ is a shortest-paths tree. Now we use the claim to show that Bellman-Ford returns TRUE. At termination, we have for all edges $(u, v) \in E$,

$$v.d = \delta(s, v)$$
$$\leq \delta(s, u) + w(u, v) \quad \text{(by the triangle inequality)}$$
$$= u.d + w(u, v),$$

and so none of the tests in line 6 causes Bellman-Ford to return FALSE. Therefore, it returns TRUE.
问题7：Bellman-Ford算法的复杂度是$O(VE)$，你是否觉得relax操作太多了一些？有什么办法吗？
换一种边的顺序，可能减少边的relax次数！

**Bellman-Ford**

1. `INITIALIZE-SINGLE-SOURCE(G, s)`
2. for \( i = 1 \) to \( |V| - 1 \)
   3. for each edge \((u, v) \in G.E\)
   4. `RELAX(u, v, w)`
5. for each edge \((u, v) \in G.E\)
   6. if \( v.d > u.d + w(u, v) \)
   7. return `FALSE`
8. return `TRUE`
如果没有回路...

```
DAG-SHORTEST-PATHS(G, w, s)
1  topologically sort the vertices of G
2  INITIALIZE-SINGLE-SOURCE(G, s)
3  for each vertex u, taken in topologically sorted order
4      for each vertex v ∈ G.Adj[u]
5          RELAX(u, v, w)
```
问题8：
为什么不需要做那么多次relax操作了？

关键是被relax的边的顺序。
If the dag contains a path from vertex \( u \) to vertex \( v \), then \( u \) precedes \( v \) in the topological sort.

**Theorem 24.5**

If a weighted, directed graph \( G = (V, E) \) has source vertex \( s \) and no cycles, then at the termination of the DAG-SHORTEST-PATHS procedure, \( v.d = \delta(s, v) \) for all vertices \( v \in V \), and the predecessor subgraph \( G_\pi \) is a shortest-paths tree.

**Proof** We first show that \( v.d = \delta(s, v) \) for all vertices \( v \in V \) at termination. If \( v \) is not reachable from \( s \), then \( v.d = \delta(s, v) = \infty \) by the no-path property. Now, suppose that \( v \) is reachable from \( s \), so that there is a shortest path \( p = (v_0, v_1, \ldots, v_k) \), where \( v_0 = s \) and \( v_k = v \). Because we process the vertices in topologically sorted order, we relax the edges on \( p \) in the order \((v_0, v_1), (v_1, v_2), \ldots, (v_{k-1}, v_k)\). The path-relaxation property implies that \( v_i.d = \delta(s, v_i) \) at termination for \( i = 0, 1, \ldots, k \). Finally, by the predecessor-subgraph property, \( G_\pi \) is a shortest-paths tree.
问题9:
没有回路的要求过高了，有什么办法达到类似的效果呢？
Dijkstra\((G, w, s)\)
1. Initialize-Single-Source\((G, s)\)
2. \(S = \emptyset\)
3. \(Q = G.V\)
4. while \(Q \neq \emptyset\)
   5. \(u = \text{Extract-Min}(Q)\)
   6. \(S = S \cup \{u\}\)
   7. for each vertex \(v \in G.\text{Adj}[u]\)
      8. \(\text{Relax}(u, v, w)\)

问题10：
为什么这被认为是Greedy算法？
Dijkstra算法的正确性

**循环不变式:**

At the start of each iteration of the **while** loop of lines 4–8, \(v.d = \delta(s, v)\) for each vertex \(v \in S\).

用反证法证明关键的一步: 任给一次特定循环, 即将加入S的顶点\(u\)必须满足 \(u.d = \delta(s, u)\)。

在左图的形势下(\(s\)到\(u\)的最短路), \(u.d\)既不能大于\(y.d\)（否则不可能选\(u\)加入\(S\), 也不能小于\(y.d\)（\(y.d=\delta(s, y)\leq \delta(s, u)\))。

因此，只能是 \(u.d = y.d = \delta(s, d) = \delta(s, u)\)。
问题11：
Dijkstra算法对每条边最多relax一次，而且不要求输入是DAG，它付出的代价是什么？为什么必须如此？
Dijkstra(G, w, s)
1  Initialize-Single-Source(G, s)
2  S = ∅
3  Q = G.V
4  while Q ≠ ∅
5      u = Extract-Min(Q)
6      S = S ∪ {u}
7      for each vertex v ∈ G.Adj[u]
8          Relax(u, v, w)

问题12：
为什么说Dijkstra算法的复杂度与其实现方法有关？
问题13：
你能比较一下Dijkstra算法与计算最小生成树的Prim算法吗？Dijkstra算法的结果是否一定是一个最小生成树？
课外作业

- TC Ex.24.1: 2, 3, 4
- TC Ex.24.2: 2
- TC Ex.24.3: 2, 4, 7
- TC Ex.24.5: 2, 5
- TC Prob.24: 2, 3