Shortest Paths

Hengfeng Wei

hengxin0912@gmail.com

June 13, 2016

Shortest Paths

Dijkstra's algorithm

Invariant: maintain $R \subseteq V$: $\forall u \in R : s \leadsto u$ is known

1. How to choose the next v and (u, v)?
2. Update $R = R + \{v\}$ and set $\text{dist}[v] = \text{dist}[u] + w(u, v)$.
3. How to update $\text{dist}[w]$ for $(v, w) \in E \wedge w \notin R$?

Dijkstra's algorithm

Invariant: maintain $R \subseteq V$: $\forall u \in R : s \leadsto u$ is known

1. How to choose the next v and (u, v)?

$$\min_{u \in R} \mathsf{dist}[u] + w(u, v)$$

To prove $w(s \leadsto u \to v)$ is the shortest distance from s to v.

Proof.

- $\forall s \leadsto v : w(s \leadsto v) \geq w(s \leadsto u \to v)$
- $s \in R, v \notin R \Rightarrow \exists u' \in R, v' \notin R$: $s \leadsto v = s \leadsto u' \to v' \leadsto v$
- given $w(s \leadsto u' \to v) \geq w(s \leadsto u \to v)$
- required $w(v' \leadsto v) \geq 0 (v' \notin R)$

$\square$

Dijkstra's algorithm

Negative edges [Problem: 3.7.9]

Dijkstra's algorithm on graphs with negative edges

Dijkstra's algorithm

Negative edges leaving s [Problem: 3.7.17]

- digraph $G = (V, E, w)$
- all negative edges are from s

Solution.

required $w(v' \rightsquigarrow v) \geq 0$

Proof.

$$v' \notin R, s \in R \Rightarrow v' \neq s \Rightarrow w(v' \rightsquigarrow v) \geq 0$$

$\square$

Dijkstra's algorithm

$w'(e) = w(e) + 1$ [Problem: 3.7.8]

- digraph $G = (V, E, w), w(e) > 0, s \in V$
- T: MST of G; T_s: shortest path tree from s
- $w'(e) = w(e) + 1$
- Does T or T_s change?

Solution.

T does not change; T_s may change.

Dijkstra's algorithm

Shortest paths from $S \subset V$ to $T \subset V$

- digraph $G = (V, E, w), w(e) \geq 0$
- $S \subset V$ to $T \subset V, S \cap T = \emptyset$
- to compute $\forall s \in S, \forall t \in T, s \rightsquigarrow t$ shortest paths
- $O(m \log n)$

Solution.

- adding s_0
- $s_0 \to s \in S$
- $w(s_0 \to s) = 0$

Shortest Paths

Dijkstra's algorithm

Invariant: maintain $R \subseteq V$: $\forall u \in R : s \leadsto u$ is known

3. How to update dist$[w]$ for $(v, w) \in E \wedge w \notin R$?

new estimator for dist$[w]$

```
forall (e = (v,w) in E) and (w not in R)
if dist[w] > dist[v] + w(v,w)
  dist[w] = dist[v] + w(v,w)
```

Dijkstra's algorithm as skeleton

```
Initialization:
  dist[s] = 0;
  dist[v] = infty for others

Q = MakePriorityQueue(V) with dist[v] as keys
                   (using min-heap)

while (Q is not empty)
  v = deleteMin(Q)
  foreach edge (v,w) in E // w in Q
    if dist[w] > dist[v] + w(v,w)
      dist[w] = dist[v] + w(v,w)
      decreaseKey(Q, w)
    else if dist[w] = dist[v] + w(v,w)
    //  do nothing
```

Dijkstra's algorithm as skeleton

Uniqueness of shortest path [Problem: 3.7.7]

- $G = (V, E, w), s \in V, w(e) > 0$
- Is shortest path $s \rightsquigarrow t$ unique?
- $\forall t$: compute the number of shortest paths from s to t.

Dijkstra's algorithm as skeleton

Solution.

```
Initialization:
  num[s] = 1; num[v] = 0 for others
  usp[s] = true; usp[v] = false for others

// update num[w] and usp[w]:
if dist[w] > dist[v] + w(v,w)
  dist[w] = dist[v] + w(v,w)
  num[w] = num[v]
  usp[w] = usp[v]
else if dist[w] = dist[v] + w(v,w)
  num[w] = num[w] + num[v]
  usp[w] = false
```

Dijkstra's algorithm as skeleton

Shortest path with fewest edges [Problem: 3.7.19]

- $G = (V, E, w), w(e) > 0, s \in V$
- best$[u]$: minimum number of edges in a shortest path from s to u
- (an example here)

Solution.

```
Initialization:
  best[s] = 0; best[v] = infty for others
// update best[w]
if dist[w] > dist[v] + w(v,w)
  dist[w] = dist[v] + w(v,w)
  best[w] = best[v] + 1
else if dist[w] = dist[v] + w(v,w)
  if best[w] > best[v] + 1
    best[w] = best[v] + 1
```

Dijkstra's algorithm as skeleton

Bottleneck shortest path [Problem: 3.7.20]

- min-max path: bottleneck length and bottleneck distance
- single source, all-pairs

Solution.

```
Q = MakePQ(V) with b-dist[v] as keys (using min-heap)
v = deleteMin(Q)

if b-dist[w] > max(b-dist[v], w(v,w))
   b-dist[w] = max(b-dist[v], w(v,w))
```

For max-min path [Problem: 3.7.21 (3.7.23, 3.7.24)]

Shortest Paths

Cycles

4-Cycle in undirected graph [Problem: 3.7.1]

- undirected graph $G = (V, E)$
- simple cycle of length 4
- $O(n^3)$

Solution.

Cycles

Shortest cycle in digraph [Problem: 3.7.4]

Solution.

Floyd-Warshall: $\min_i D[i][i]$
Initialization: $D^{(0)}[i][i] = \infty$

Remark.

Does not apply to undirected graph.

Cycles

Shortest cycle in undirected graph [Problem: 3.7.14]

- $G = (V, E), w(e) = 1$
- DFS: back edge $\iff$ cycle
- $u \to v$: $\text{level}[u] - \text{level}[v] + 1$

Solution.

A counterexample here.

Cycles

Shortest cycle containing a specific edge [Problem: 3.7.5]

- undirected graph $G = (V, E, w), w(e) > 0, e \in E$
- shortest cycle containing e

Solution.

$P_{u \rightsquigarrow v} + (u, v)$

Cycles

Hamiltonian path in tournament graph [Problem: 3.7.18]

- digraph $G = (V, E)$
- $\forall u, v : (u \to v \vee v \to u) \wedge \neg(u \to v \wedge v \to u)$
- hamiltonial path

Solution.

- existence: induction on n
- algorithm $1 + 2 + \cdots + (n-1) = O(n^2)$


THANK YOU