16—Shortest Paths and Dijkstra's Algorithm

May 17, 2002

---

Weighted Graphs

Quest for Food
(all distances in yards)

---

Serious About My Lunch

Single-Source Shortest Path

Compute shortest distance to all nodes from my desk
Dijkstra's Algorithm

Two kinds of vertices
- **Finished** vertices
  Shortest distance computed
- **Unknown** vertices
  Have *tentative* distance

At each step:
1. Pick **closest** unknown vertex
2. Add it to **finished** vertices
3. Update distances

**Dijkstra's Algorithm vs. BFS**

At each step:
1. Pick closest unknown vertex
2. Add it to **visited** vertices
3. Update *queue* with neighbors

<table>
<thead>
<tr>
<th>Dijkstra's Algorithm</th>
<th>BFS</th>
</tr>
</thead>
</table>
1. *Finished* vertices in the middle
2. Vectors from fringe added at each step

Dijkstra’s and BFS

Yay! Example!

- All vertices unknown
- Start vertex distance 0
- All other vertices at $\infty$

Initialize

At each step:
1. Pick closest unknown vertex
2. Add it to finished vertices
3. Update distances
At each step:
1. Pick closest unknown vertex
2. Add it to finished vertices
3. Update distances

Why does this work?
Why are the distances of finished vertices correct?
When Dijkstra's Wouldn't Work

Positive-only edge weights is essential

Key Lemma

We finish with the closest vertices first

If \( w \) finished immediately after \( v \), then
\[
\text{FinalDistance}(w) \geq \text{FinalDistance}(v)
\]

Inductive Proof

If \( w \) finished immediately after \( v \), then
\[
\text{FinalDistance}(w) \geq \text{FinalDistance}(v)
\]

Tentative distance of \( w \) no bigger than \( v \)'s

Updates from \( v \) may shrink distance of \( w \),
but can't get smaller than \( v \)'s distance
The Big Kahuna

Suppose \( v \)'s final distance is *not* correct

Then exists a shorter path to \( v \)
- \( \text{dist}(w) + x < \text{dist}(v) \)
- Final distances on path are correct
- \( v \) finalized *after* all other nodes on the path

Gotcha!

When \( w_k \) finalized, \( \text{dist}(v) \leftarrow \text{dist}(w) + x \)
- Final \( \text{dist}(v) \) *greater* than this
- \( \text{dist}(v) \) only *decreases* during the algorithm
- Contradiction!

A Number is Not a Path

Sure, we know how far away \( v \) is...

...but how do we *get* there?
Dijkstra(Graph G, Vertex s)
{
    PQ pq;
    for each (v in G) pq.Insert(v, INFINITY);
    pq.Change(s, 0);
    while (!pq.Empty()) {
        v = pq.DeleteMin();
        for each (w in v.Neighbors())
            if (dist(w) > dist(v) + w(v,w))
                pq.Change(w, dist(v)+w(v,w));
    }
}

Implementing the PQ
PQ as a Heap

```c
PQ::SwapUp(i)
{
    int p = (i-1)/2;
    while (p >= 0 && heap[p].key > heap[i].key) {
        swap(heap[p],heap[i]);
        i = p;
        p = (i-1)/2;
    }
}
```