Single Source Shortest Path (Dijkstra's Algorithm)

Idea: Dijkstra's algorithm maintains a set of vertices S such that for all $v \in S$ the cost of the shortest path from the starting vertex v_0 to v is known. At each step, we add a new vertex to S.

Dijkstra's algorithm also maintains an array $D[v]$ of cost estimates known so far. A key to understanding the correctness proof for Dijkstra's algorithm is a precise interpretation of the values contained in $D[v]$:

- If $v \in S$ then $D[v] =$ the cost of the shortest path from v_0 to v.
- If $v \notin S$ then $D[v] =$ the cost of the shortest path from v_0 to v that lies[1] entirely within S, except for the endpoint v. It is possible that no such path to v exists, and in this case $D[v] = +\infty$.

Input: Graph $G = (V, E)$ and a weight function $f : V \times V \to R^+ \cup +\infty$
Output: Array D such that $D[v]$ is the cost of the shortest path from v_0 to v.

```
S = {v0}
D[v0] = 0
for all v ∈ (V − {v0})
    D[v] = f(v0, v)
end for all
while ( S ≠ V )
    Select vertex w in V − S such that D[w] is minimal
    S = S ∪ { w }
    for all v in V − S do
        D[v] = min(D[v], D[w] + f(w, v))
    end for all
end while
```

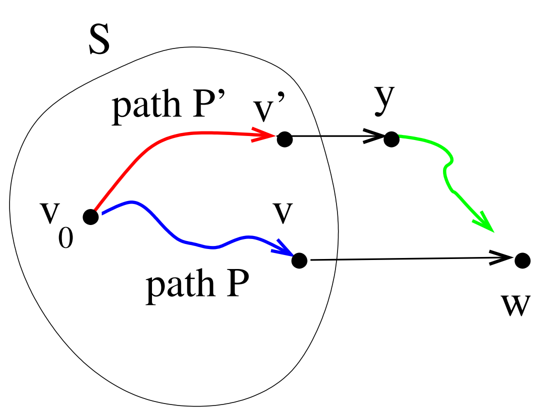

Figure 1: Paths P and P' from v_0 to w

[1] A path P lies entirely within a vertex set S if and only if every vertex w along path P is also in S.

Correctness Proof: (by induction on $|S|$)

Base Case:

Initially, $S = \{v_0\}$. We can make two self-evident claims:

- $D[v_0] = 0$ is the cost of the shortest path from v_0 to v_0.
- For all $v \notin S$, $D[v_0] = f(v_0, v)$ is the cost of the shortest path from v_0 to v that lies entirely within S except for the endpoint v.

These two facts establish the base case.

General Case:

By inductive hypothesis, we have:

- If $v \in S$ then $D[v]$ is the cost of the shortest path from v_0 to v.
- If $v \notin S$ then $D[v]$ is the cost of the shortest path from v_0 to v that lies entirely within S, except for the endpoint v.

To conclude the proof, we refer to the diagram in Figure 1.

Let w denote the vertex chosen on line 7. Let v denote the last vertex within P along path P from v_0 to w. We wish to establish that the path P from v_0 through (some) v to w is the shortest path from v_0 to w (without qualification). We approach this part of our proof by contradiction: suppose there is a (strictly) shorter path P'.

Let v' denote the last vertex in S on path P'. We claim that by the inductive hypothesis, path P' must contain a vertex y which is not in S and further, v' and y are connected by a single edge.

Sub-proof: If the vertex y does not exist along P', then path P' goes from v_0, through v' and then directly to w. In this case, P' lies entirely within S except for the endpoint w. By the inductive hypothesis, path P is a shortest such path, and therefore P' can not be (strictly) shorter.

We now re-consider path P', knowing that $y \notin S$ exists along path P' from v_0 to w. Further, the sub-path from v_0 to y lies entirely within S except for a single edge from v' to y. The sub-path from v_0 to y is (strictly) shorter than P' and therefore also (strictly) shorter than P. This contradicts the manner in which w is chosen on line 7. If path P' exists, the algorithm would have chosen y instead of w, a contradiction.

Path P' does not exist. We conclude that path P is a shortest path from v_0 to w.

$\blacksquare$