Dijkstra's Algorithm

Prasangsha Ganguly

January 2, 2021

Dijkstra's algorithm is used to identify single source shortest path for a graph with non-negative edge costs.

Input and parameters

$G(V, E)$ is a graph. A source vertex s is given along with the cost function associated with E. For each vertex $u \in V$, $d[u]$ denotes the upper bound of the distance from s. For each vertex $u \in V$, let $\pi[u]$ denote the shortest parent of u in the shortest path $s \to u$.

Let, S be the set that holds the finished vertices or the vertices for which the shortest distance has been already computed.

Q be the set of vertices that are not in S. This is maintained as a priority queue. The minimum distance node is kept at the head of the priority queue Q.

Algorithm

Algorithm 1: Dijkstra's

Input: $(G, s, w(*))$
Output: Shortest distance from s to all the points
begin
    Initialize(G,s)
    $S \leftarrow \phi$
    $Q \leftarrow V$
    while *($Q \neq \phi$)* **do**
        $u \leftarrow$ Extract-Min(Q)
        $S \leftarrow S \cup \{u\}$
        for *each* $v \in V$ **do**
            Relax($u, v, w(*)$)
    return d

Algorithm 2: Initialize(G, s)

Input: (G, s)

begin

  for *each* $u \in V$ **do**

    $d[u] = \infty$

    $\pi[u] = NIL$

  $d[s] = 0$

Algorithm 3: Relax$(u, v, w(*))$

Input: $(u, v, w(*))$

begin

  if $d[v] > d[u] + w(u \to v)$ **then**

    $d[v] = d[u] + w(u \to v)$

    $\pi[v] = u$