6.006 Introduction to Algorithms
Spring 2008

Lecture 18: Shortest Paths IV - Speeding up Dijkstra

Lecture Overview

- Single-source single-target Dijkstra
- Bidirectional search
- Goal directed search - potentials and landmarks

Readings

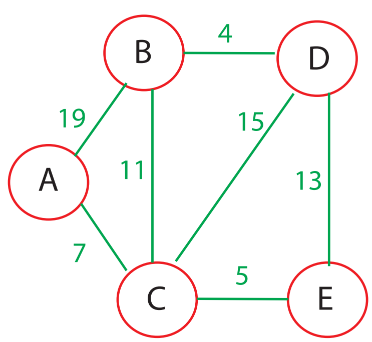
Wagner, Dorothea, and Thomas Willhalm. "Speed-Up Techniques for Shortest-Path Computations." In *Lecture Notes in Computer Science: Proceedings of the 24th Annual Symposium on Theoretical Aspects of Computer Science*. Berlin/Heidelberg, MA: Springer, 2007. ISBN: 9783540709176. Read up to section 3.2.

DIJKSTRA single-source, single-target

Initialize()
$Q \leftarrow V[G]$
while $Q \neq \phi$
    do $u \leftarrow$ EXTRACT_MIN(Q) (stop if $u = t$!)
    for each vertex $v \;\epsilon$ Adj$[u]$
        do RELAX(u, v, w)

Observation: If only shortest path from s to t is required, stop when t is removed from Q, i.e., when $u = t$

DIJKSTRA Demo



A C E B D
7 12 18 22

D B E C A
4 13 15 22

E C A D B
5 12 13 16

Figure 1: Dijkstra Demonstration with Balls and String

Bi-Directional Search

Note: Speedup techniques covered here do not change worst-case behavior, but reduce the number of visited vertices in practice.

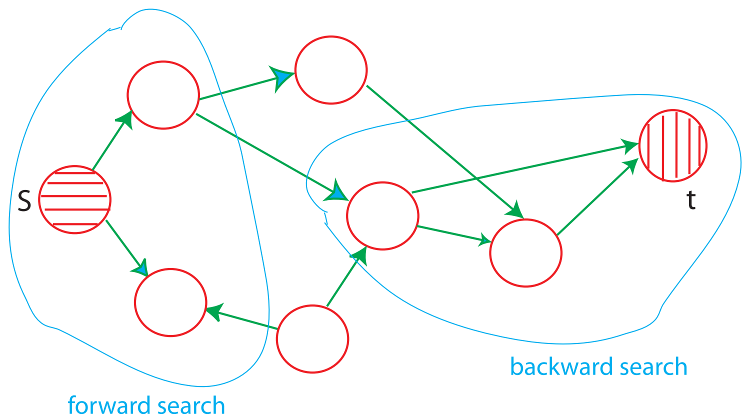


Figure 2: Bi-directional Search

Bi-D Search

Alternate forward search from s

backward search from t

(follow edges backward)

$d_f(u)$ distances for forward search

$d_b(u)$ distances for backward search

Algorithm terminates when some vertex w has been processed, i.e., deleted from the queue of both searches, Q_f and Q_b

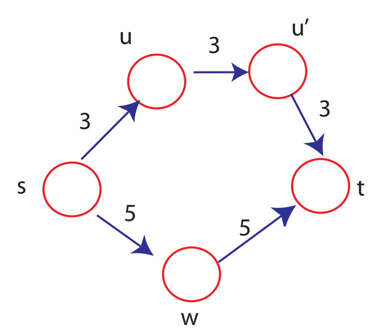


Figure 3: Bi-D Search

Subtlety: After search terminates, find node x with minimum value of $d_f(x)+d_b(x)$. x may not be the vertex w that caused termination as in example to the left!
Find shortest path from s to x using Π_f and shortest path backwards from t to x using Π_b.
Note: x will have been deleted from either Q_f or Q_b or both.

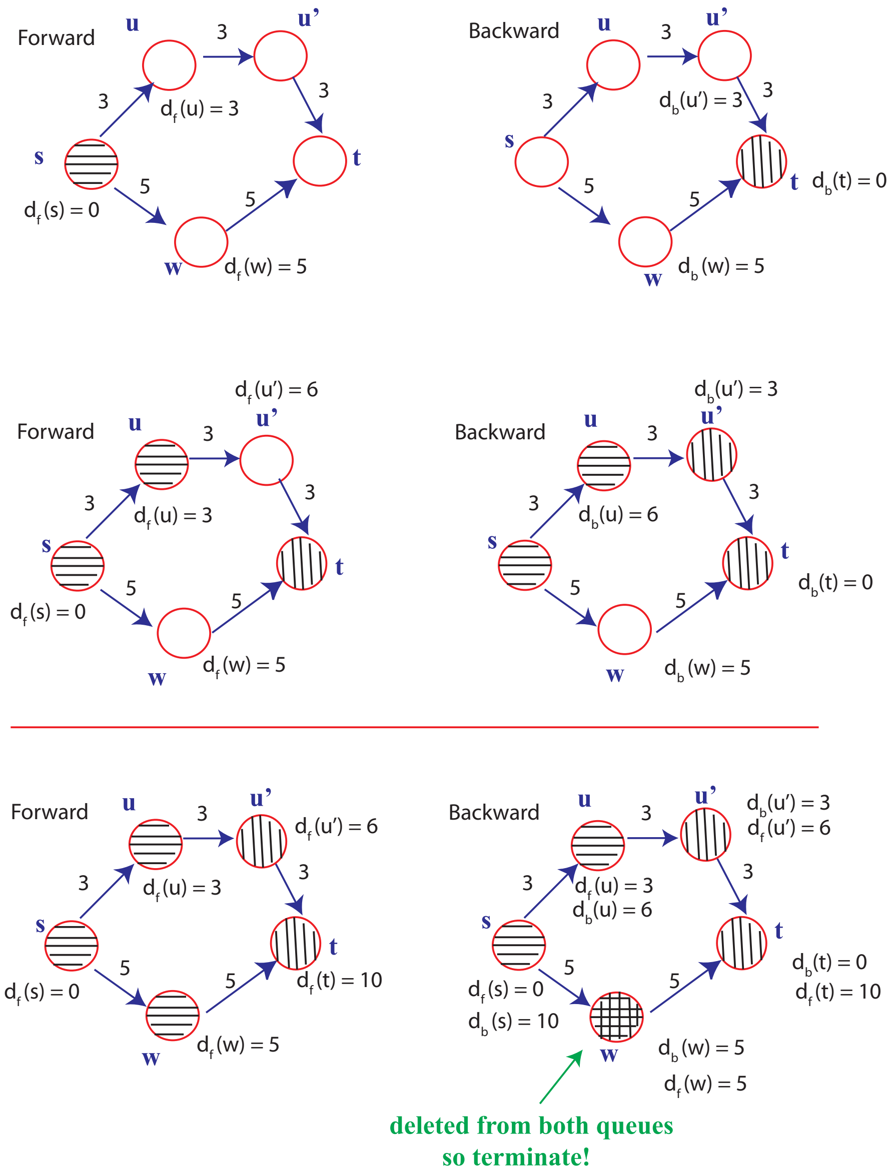


Figure 4: Forward and Backward Search

Minimum value for $d_f(x)+d_b(x)$ over all vertices that have been processed in at least one search

$$d_f(u)+d_b(u)=3+6=9$$

$$d_f(u') + d_b(u') = 6 + 3 = 9$$

$$d_f(w) + d_b(w) = 5 + 5 = 10$$

Goal-Directed Search or A^*

Modify edge weights with potential function over vertices.

$$\overline{w}\ (u, v) = w\ (u, v) - \lambda(u) + \lambda(v)$$

Search toward target:

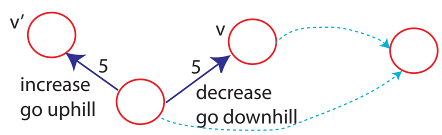


Figure 5: Targeted Search

Correctness

$$\overline{w}(p) = w(p) - \lambda_t(s) + \lambda_t(t)$$

So shortest paths are maintained in modified graph with $\overline{w}$ weights.

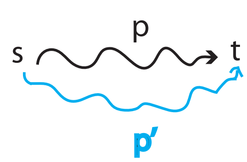


Figure 6: Modifying Edge Weights

To apply Dijkstra, we need $\overline{w}(u, v) \geq 0$ for all (u, v).
Choose potential function appropriately, to be feasible.

Landmarks

Small set of landmarks LCV. For all $u \epsilon V, l \epsilon L$, pre-compute $\delta(u, l)$. Potential $\lambda_t^{(l)}(u) = \delta(u, l) = \delta(t, l)$ for each l.
CLAIM: $\lambda_t^{(l)}$ is feasible.

Feasibility

$$
\begin{aligned}
\overline{w}(u,v) &= w(u,v) - \lambda_t^{(l)}(u) + \lambda_t^{(l)}(v) \\
&= w(u,v) - \delta(u,l) + \delta(t,l) + \delta(v,l) - \delta(t,l) \\
&= w(u,v) - \delta(u,l) + \delta(v,l) \geq 0 \quad \text{by the } \Delta \text{ -inequality} \\
\lambda_t(u) &= \max_{l \, \epsilon \, L} \lambda_t^{(l)}(u) \;\; \text{is also feasible}
\end{aligned}
$$