강의 18: 최단 경로 찾기 IV - 다익스트라 알고리즘 가속화

강좌 목차

- 단일 시작 단일 도착 다익스트라
- 양방향 탐색
- 목표 지향 탐색 - 잠재력과 랜드마크

읽을거리

Wagner, Dorothea, and Thomas Willhalm. ”Speed-up Techniques for Shortest-Path Computations.” Proceedings of the 24th International Symposium on Theoretical Aspects of Computer Science (STACS 2007): 23-36. Read up to Section 3.2.

단일 시작 단일 도착 다익스트라

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

관찰: s에서 t까지의 최단 경로만 필요하다면 Q에서 제거되었을 때 알고리즘을 멈춘다. i.e., $u = t$ 일 때 종료한다.

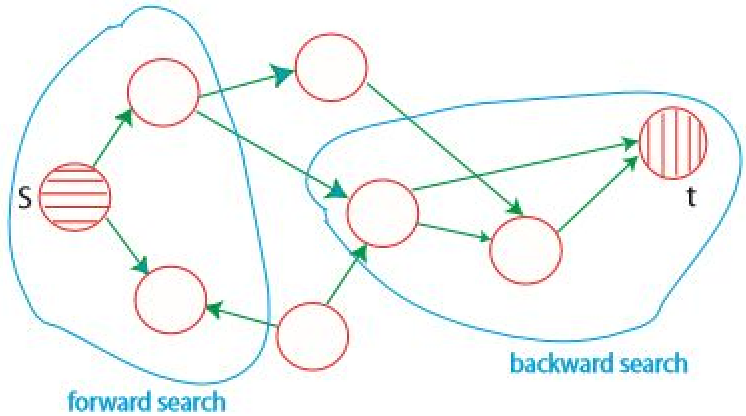

그림 1: 양방향 탐색 아이디어.

양방향 탐색

주의: 여기에서 다루는 속도를 가속화 시키는 방법은 최악의 경우 복잡도를 바꾸지 않지만 실제에 적용되었을 때 방문하는 정점의 개수를 줄인다

양방향 탐색

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

알고리즘은 어떤 정점 w가 계산 되었을 때 끝난다. 즉 전진 탐색과 후진 탐색 모두의 우선순위 큐 Qf 와 Qb 에서 제거되었을 때 종료한다.
탐색이 종료하고 $df(x) + db(x)$의 값이 최소인 정점 x를 찾는다. 왼쪽의 예시에서처럼 x가 정점 w가 아닐 수도 있다.
Πf를 이용해서 s에서 x까지의 최단 경로를 찾고 Πb를 이용해서 t에서 x까지의 최단 경로를 거꾸로 찾는다. 이때 x는 Qf와 Qb모두에서 제거되었다.

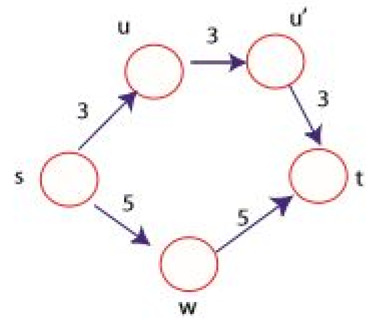

그림 2: 양방향 탐색 예시

한번의 탐색에서 처리된 모든 정점에 대해 $df(x)+ db(x)$ 최소값을 찾는다.
(그림 3 참조):

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

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

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

목적 지향 탐색 또는 A*

정점에 대해 잠재력 함수에 따라 간선의 가중치를 수정한다.

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

그림 4에서 보여진 대로 도착점을 향해 탐색한다.

정확성

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

가중치 $\overline{w}$ 로 수정된 그래프에서도 최단 경로가 유지된다. (그림 5 참조).
다익스트라를 적용하기 위해서 모든 간선 (u, v)에 대해서 $\overline{w}(u, v) \geq 0$ 를 만족해야한다. 적절하고 조건을 만족하는 잠재력 함수를 찾아야한다.

랜드마크

그래프의 정점의 부분집합 랜드마크 집합 LCV가 있다. 모든 정점 $u \in V$, $l \in L$에 대해 $\delta(u, l)$를 미리 계산한다.

각 랜드마크 l에 대한 잠재력 함수 $\lambda_t^{(l)}(u) = \delta(u, l) - \delta(t, l)$ 이다.

주장: $\lambda_t^{(l)}$ 은 사용가능하다.

사용 가능성

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

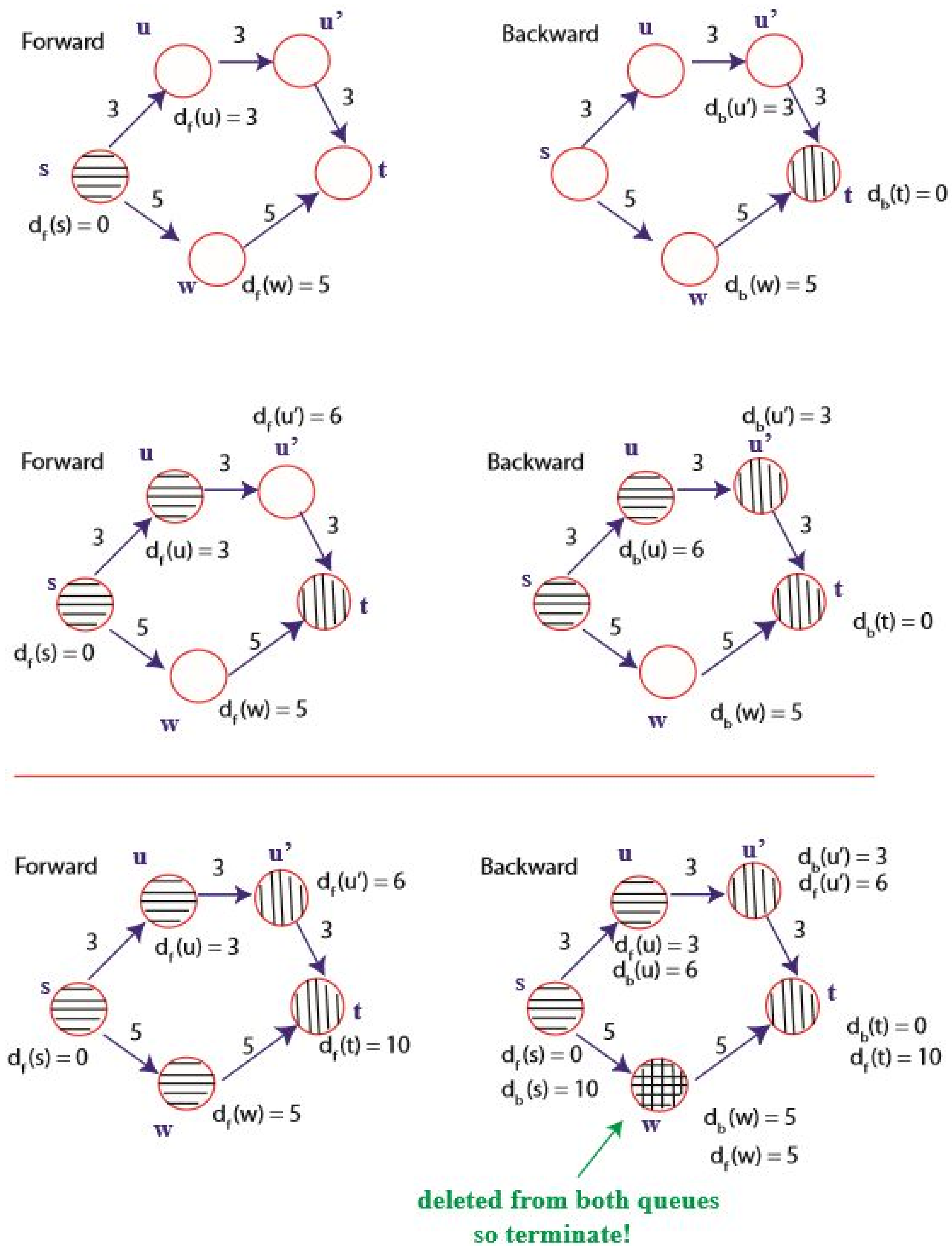

그림 3: 전진 탐색 후진 탐색 그리고 종료.

그림 4: 목표 지향적 탐색

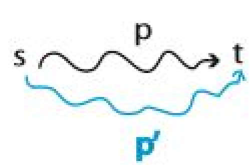

그림 5: 간선의 가중치 수정하기.

MIT OpenCourseWare
http://ocw.mit.edu

6.006 Introduction to Algorithms
Fall 2011

For information about citing these materials or our Terms of Use, visit: http://ocw.mit.edu/terms.