Distributed Shortest Path

30/6/2021
Lecture 12
Scribe: Avi Cohen

Recap: Scheduling Distributed Algorithms

$\mathcal{A}_1, \dots, \mathcal{A}_\ell$ → algs to run in parallel.

d ↙ dilation: max run time of $\mathcal{A}_i$'s

c ↙ congestion: $\forall e \quad c_i(e) = \#$msgs $\mathcal{A}_i$ sends through e.

$$c(e) = \sum_{i=1}^{\ell} c_i(e)$$

$$\text{cong} = \max_e c(e)$$

Gaffari 2015: LB: $\Omega\left(c + d \frac{\log n}{\log\log n}\right)$

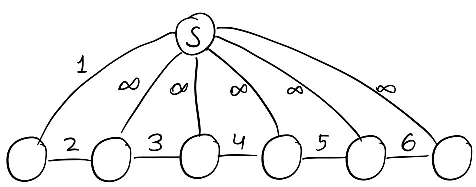
UB: $\tilde{O}(c+d)$ → Note that this does not depend on the diameter!

Input: n-vertex weighted graph $G=(V,E,w)$

all weights are unique positive integers in $[1, \text{Poly}(n)]$

Source Vertex s.

Output: each u should hold $d(s,u)$



LB: $\Omega(D + \sqrt{n})$
↪ $D + n^{2/3}$ (Today's Alg)

Def: h-hop path: path with h-edges.

h-hop distance: shortest h-hop u-v path

$\rho^h(u,v) =$ collection of all u-v paths with $\le h$ edges.

$d^h(u,v) = \min\{w(P),\ P \in \rho^h(u,v)\}$

SPD = min h s.t.

$$d^h(u,v) = d(u,v) \quad \forall u,v$$

Lemma: There is an algorithm for computing SSSP in $O(SPD)$ time.

Distributed Belman-Ford

$\delta(v) = \infty$, for $v \neq s$

$\delta(s) = 0$

- s sends $\delta(s)$ to its neighbors
- for SPD rounds:

  Every vertex u, upon recieving $\delta(v)$, $v \in N(u)$:

  If $\delta(u) > \delta(v) + w(u,v)$

  - $\delta(u) = \delta(v) + w(u,v)$
  - sends $\delta(u)$ to neighbors.

Denote $\delta_h(v)$: v's estimate after h rounds.

Claim: After h rounds, $\delta_h(v) = d^h(s,v)$

$\Rightarrow$ for $h = SPD$, $\delta_h(v) = d^{SPD}(s,v) = d(s,v)$

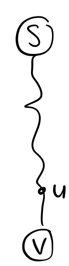
$\longrightarrow$ $\boxed{\text{Cor: h-hop distances after h-rounds}}$

Proof: By induction on h.

Assume the claim holds upto h and show for $h+1$.

Fix v.

Fix $h+1$ path P s.t. $w(P) = d^{h+1}(s,v)$



By ind: $\delta_h(u) = d^h(s,u) = w(P[s,u])$

$$\delta_{h+1}(v) = \delta_h(v) + w(u,v) = d^{h+1}(s,v).$$

Theorem [Nanogkai 2014]:

Given n-vertex weighted graph, $s \in V$ and $\varepsilon \in (0,1)$ there is an alg for computing $(1+\varepsilon)$ approx SSSP in $\tilde{O}(D + \frac{n^{2/3}}{\varepsilon})$ rounds.

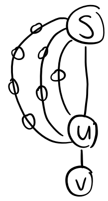
Two Limitations of Dist. BF

1) runs in SPD time which might be linear.

[adding edges to the graph to reduce SPD]

2) h-hop BF alg has large congestion.

[light alg for comp. $(1+\varepsilon)$-approx. of h-hop distance]



Key Lemma: [light comp. of h-hop distances]

light: sending $\tilde{O}(1)$ msgs through each edge.

Given n-vertex graph G, $s \in V$, h hop bound, $\varepsilon \in (0,1)$

there is a light alg that computes $(1+\varepsilon)$-approx. h-hop $s \times V$ distances

within $O(\frac{h}{\varepsilon})$ rounds.

Output: $d^h(s,u) \leq \delta_h(u) \leq (1+\varepsilon) d^h(s,u)$

Cor: Given $S \subseteq V$, compute $(1+\varepsilon)$ approx h-hop $S \times V$ distances in $\tilde{O}(|S| + \frac{h}{\varepsilon})$ rounds.

↳ By Scheduling Lemma + Key Lemma.

SSSP alg in $\tilde{O}(D + n^{2/3})$ rounds

Set $h = n^{2/3}$

Step 1: Sample $O(n^{1/3} \log n)$ sources S'.

$S = S' \cup \{s\}$

Apply key lemma to compute $(1+\varepsilon)$-approx h-hop $S \times V$ distances.

↳ Output: $\delta_h(s', u)$ $\forall s' \in S$.

Step 2: Every $s' \in S$, broadcast $\{\delta_h(s', s''), \forall s'' \in S\}$

Step 3: Every vertex v locally computes a graph $G_v = (S \cup \{v\}, E_v)$, $w(x,y) = \delta_h(x,y)$ $\forall x,y \in S \cup \{v\}$

Run Dijskstra in G_v and outputs $d(s, v, G_v)$

Round Complexity:

Step 1: h-hop $S \times V$:

$$\tilde{O}\left(|S| + \frac{h}{\varepsilon}\right) = \tilde{O}\left(n^{1/3} + \frac{n^{2/3}}{\varepsilon}\right)$$

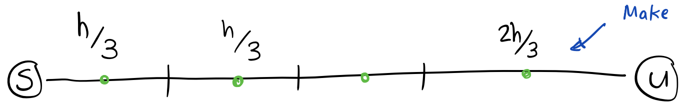
$$O(D + |S|) = \tilde{O}(n^{1/3} + D)$$

Step 2: $O(|S|^2 + D) = \tilde{O}(n^{2/3} + D)$

Correctness:

Case 1: $d^h(s,u) = d(s,u)$ ✓

Case 2: $d^h(s,u) > d(s,u)$ s-u shortest path P



Make the last segment longer to include the residual vertices

- Partition P into segments $\sim \frac{h}{3}$ hops.
- w.h.p. each segment P_i intersects with S, let's call it b_i.
- $d^h(b_i, b_{i+1}) = d(b_i, b_{i+1})$

After Step 1+2

$b_1 = s, b_2, \ldots, b_k = u$ s.t. $d^h(b_i, b_{i+1}) = d(b_i, b_{i+1})$

$w(b_i, b_{i+1}) \leq (1+\varepsilon)\, d^h(b_i, b_{i+1})$

weight of the edge (b_i, b_{i+1}) in G_u.

$$d(s, u, G_u) \leq \sum_{i=1}^{k} w(b_i, b_{i+1}) = (1+\varepsilon)\, d(s,u)$$

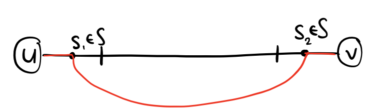
Graph-theoretic Objective:

(h,ε)-hopset of a graph G is a small collection of edges $H \subseteq V \times V$, s.t.:

$$d(u,v,G) \leq d^h(u,v,G \cup H) \leq (1+\varepsilon)\, d(u,v,G)$$

Observation: Every n-vertex graph admits an exact hopset H with $|H| = \tilde{O}(n)$ and $h = O(\sqrt{n})$.

→ S sample of $\tilde{O}(\sqrt{n})$ sources. Add weighted edges $S \times S$.



Proof of the Key Lemma (Light computation of approx h-hop distances)

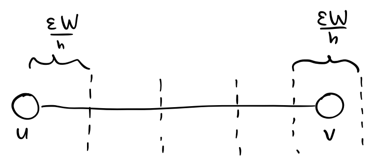
Intuition: want to compute $d(s,t)$, and we know that W, h:

1) $d^h(s,t) = d(s,t)$

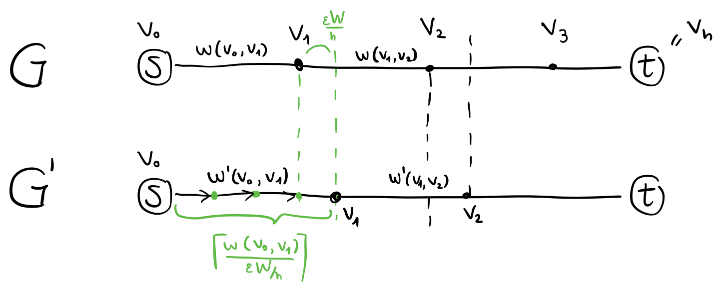
2) $W \leq d(s,t) \leq 2W \rightarrow$ W is a const. we will need to guess.

"Round" each $w(u,v)$ up to some multiple of $\frac{\varepsilon W}{h}$

$$w'(u,v) = \left\lceil \frac{w(u,v)}{\varepsilon W/h} \right\rceil \cdot \frac{\varepsilon W}{h}$$



$\hookrightarrow \quad w(u,v) \leq w'(u,v) \leq w(u,v) + \frac{\varepsilon W}{h}$



Total Error: $\frac{\varepsilon W}{h} \cdot h = \varepsilon\, d(s,t)$

rounds: $\frac{d(s,t)}{\varepsilon W/h} \leq \frac{2W}{\varepsilon W/h} \leq \frac{2h}{\varepsilon}$

Algorithm

Subalgorithms $A_1, \dots, A_t$. A_i guesses $W = 2^i$, $i \in (0, \log n\, W_{max})$

Alg $A_i(s, 2^i, h)$:

1) $w_i(u,v) = \left\lceil \frac{w(u,v)}{\varepsilon W/h} \right\rceil \cdot \frac{\varepsilon W}{h}$, $\forall (u,v) \in E$

2) Run "BFS-like" alg. for $O(\frac{h}{\varepsilon})$ rounds.

3) Set $\delta_i^h(u)$ to be the output distance.

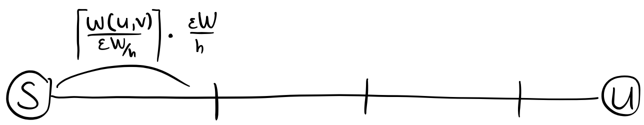
Final Output: $\delta_h(u) = \min_i \delta_i^h(u)$

Claim: $\delta_i^h(u) \geq d(s,u)$

Proof: $w_i(u,v) \geq w(u,v)$

Fix u, let i be s.t. $\overset{= W}{2^i} \leq d(s,u) \leq 2^{i+1}$

Show: $\delta_i^h(u) \leq (1+\varepsilon)\, d^n(s,u)$



Total Error: $\frac{\varepsilon W}{h} \cdot h$

$$\frac{d^h(s,t)}{\varepsilon W/h} = O\left(\frac{n}{\varepsilon}\right)$$