Distributed Shortest Path

Recap: Scheduling Distributed Algorithms

$\mathcal{A}_i, \ldots, \mathcal{A}_x \rightarrow$ algs to run in parallel.

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

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

$c(e) = \max_{i=1}^{x} c_i(e)$

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

Gaffari 2015:

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

$\text{UB: } \tilde{\Omega}(c + d)$ → 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 $d(s, u)$

$\text{LB: } \Omega\left(D + \sqrt{n}\right)$

$\rightarrow D + n^{\frac{2}{3}} \ (\text{Today's Algo})$

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

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

$j^h(u,v) = \text{collection of all } u$-$v \text{ paths with } \leq h \text{ edges.}$

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

$\text{SPD} = \min h \ s.t.$

$d^h(u,v) = d(u,v) \ \forall u,v$
Lemma: There is an algorithm for computing SSSP in $O(SPD)$ time.

**Distributed Belman-Ford**

\[ \delta(v) = \infty, \text{ for } v \neq s \]
\[ \delta(s) = 0 \]
- $s$ sends $\delta(s)$ to its neighbors
- for SPD rounds:
  - Every vertex $u$, upon receiving $\delta(v), v \in N(u)$:
    - 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 $h = SPD$, $\delta_h(v) = d^{SPD}(s,v) = d(s,v)$

**Proof:** By induction on $h$.

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

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

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

\[ \delta_{h+1}(v) = \delta_h(v) + w(u,v) = d^{h+4}(s,v). \]

**Theorem [Nanogkai 2014]:**

Given $n$-vertex weighted graph, $s \in V$ and $\epsilon \in (0,1)$ there is an alg for computing $(1+\epsilon)$ approx SSSP in $\tilde{O}(\frac{d+n^{2+\epsilon}}{\epsilon})$ 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]

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 $O(\frac{n}{\varepsilon})$ rounds.

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

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

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

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

Step 1: Sample $O(n^{1/2} \log n)$ sources $S'$.

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

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

Output: $\delta^h_n(s',u) \forall s' \in S$.

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

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

Run Dijkstra in $G_v$ and outputs $d(s,v,G_v)$
Round Complexity:

Step 1: \( h\)-hop \( S \times V \):
\[
\widetilde{O}(|S| + \frac{h}{\varepsilon}) = \widetilde{O}\left(n^{\frac{1}{3}} + \frac{n^{\frac{1}{3}}}{\varepsilon}\right)
\]
\[
O(D + |S|) = \widetilde{O}\left(n^{\frac{1}{3}} + D\right)
\]

Step 2: \( O(|S|^2 + D) = \widetilde{O}\left(n^{2/3} + D\right)\)

Correctness:

Case 1: \( d^h(s,u) = d(s,u) \quad \checkmark \)

Case 2: \( d^h(s,u) > d(s,u) \)

\( s-u \) shortest path \( P \)

- 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

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

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

\( w \) 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:

\((h, \varepsilon)\)-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 \( |H|=\widetilde{O}(n) \) and \( h=\widetilde{O}(\sqrt{n}) \).

- \( S \) sample of \( \widetilde{O}(\sqrt{n}) \) 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) $W \leq d(s,t) \leq 2W \quad \rightarrow \quad W$ is a const. we will need to guess.

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

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

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

\[ G \quad \overset{\varepsilon W}{\sim} \quad \overset{\varepsilon W}{\sim} \quad G' \]

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

\# rounds: \[ \frac{d(s,t)}{\varepsilon W/h} \leq \frac{2W}{\varepsilon W/h} \leq \frac{2h}{E} \]

**Algorithm**

Subalgorithms $A_1, \ldots, A_t$. $A_i$ guesses $W = 2^i$, i.e. $O(\log W_{\text{max}})$

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

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

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: $δ^h_i(u) ≥ d(s,u)$

Proof: $w_i(u,v) ≥ w(u,v)$

Fix $u$, let $i$ be s.t. $2^i ≤ d(s,u) ≤ 2^{i+1}$

Show: $δ^h_i(u) ≤ (1+ε) d''(s,u)$

\[
\frac{w(u,v)}{εW_h} \cdot \frac{εW}{h}
\]

Total Error: \( \frac{εW}{h} \cdot h \)

\[
\frac{d^h(s,t)}{εW/h} = O\left(\frac{n}{ε}\right)
\]