Distributed Directed SSSP in Sublinear Time

Jason Li
Carnegie Mellon University

Joint work with Mohsen Ghaffari
(ETH Zurich)

STOC 2018

Distributed Computing, CONGEST Model

Distributed Model

Distributed Computing, CONGEST Model

Distributed Model

- Network graph

Distributed Computing, CONGEST Model

Distributed Model

- Network graph
- Vertices are called nodes

Distributed Computing, CONGEST Model

Distributed Model

- Network graph
- Vertices are called nodes
- Algorithm runs in rounds. In each round:

Distributed Computing, CONGEST Model

Distributed Model

- Network graph
- Vertices are called nodes
- Algorithm runs in rounds. In each round:
  - Every node performs unbounded local computation

Distributed Computing, CONGEST Model

Distributed Model

- Network graph
- Vertices are called nodes
- Algorithm runs in rounds. In each round:
  - Every node performs unbounded local computation
  - Every node sends an $O(\log n)$-bit message to each neighbor

Distributed Computing, CONGEST Model

Distributed Model

- Network graph
- Vertices are called nodes
- Algorithm runs in rounds. In each round:
  - Every node performs unbounded local computation
  - Every node sends an $O(\log n)$-bit message to each neighbor
- The running time is the number of rounds

Distributed Computing, CONGEST Model

Distributed Model

- Network graph
- Vertices are called nodes
- Algorithm runs in rounds. In each round:
  - Every node performs unbounded local computation
  - Every node sends an $O(\log n)$-bit message to each neighbor
- The running time is the number of rounds

SSSP Problem

Distributed Computing, CONGEST Model

Distributed Model

- Network graph
- Vertices are called nodes
- Algorithm runs in rounds. In each round:
  - Every node performs unbounded local computation
  - Every node sends an $O(\log n)$-bit message to each neighbor
- The running time is the number of rounds

SSSP Problem

- Input graph and network graph have same edges

Distributed Computing, CONGEST Model

Distributed Model

- Network graph
- Vertices are called nodes
- Algorithm runs in rounds. In each round:
  - Every node performs unbounded local computation
  - Every node sends an $O(\log n)$-bit message to each neighbor
- The running time is the number of rounds

SSSP Problem

- Input graph and network graph have same edges
- Beginning: every node knows weights of its edges

Distributed Computing, CONGEST Model

Distributed Model

- Network graph
- Vertices are called nodes
- Algorithm runs in rounds. In each round:
  - Every node performs unbounded local computation
  - Every node sends an $O(\log n)$-bit message to each neighbor
- The running time is the number of rounds

SSSP Problem

- Input graph and network graph have same edges
- Beginning: every node knows weights of its edges
- End: every node knows its distance from source

Distributed SSSP

- Trivial $n - 1$ upper bound (Bellman-Ford)

Distributed SSSP

- Trivial $n-1$ upper bound (Bellman-Ford)
- Trivial $\Omega(D)$ lower bound

Distributed SSSP

- Trivial $n - 1$ upper bound (Bellman-Ford)
- Trivial $\Omega(D)$ lower bound
- Unweighted, undirected: BFS in D rounds

Distributed SSSP

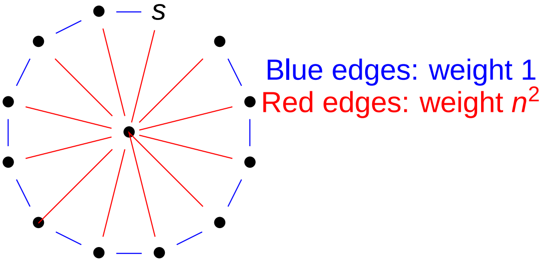
- Trivial $n-1$ upper bound (Bellman-Ford)
- Trivial $\Omega(D)$ lower bound
- Unweighted, undirected: BFS in D rounds
- Weighted, undirected?

Distributed SSSP

- Trivial $n-1$ upper bound (Bellman-Ford)
- Trivial $\Omega(D)$ lower bound
- Unweighted, undirected: BFS in D rounds
- Weighted, undirected?
- BFS, Bellman-Ford?

Distributed SSSP

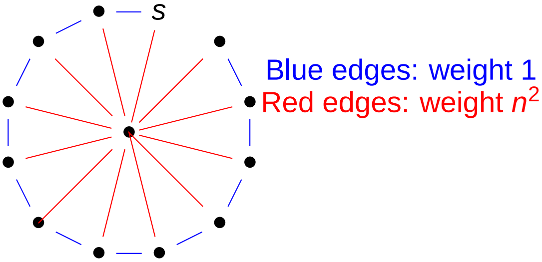
- Trivial $n-1$ upper bound (Bellman-Ford)
- Trivial $\Omega(D)$ lower bound
- Unweighted, undirected: BFS in D rounds
- Weighted, undirected?
- BFS, Bellman-Ford?



-

Distributed SSSP

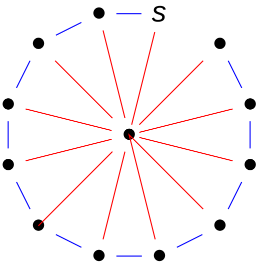
- Trivial $n-1$ upper bound (Bellman-Ford)
- Trivial $\Omega(D)$ lower bound
- Unweighted, undirected: BFS in D rounds
- Weighted, undirected?
- BFS, Bellman-Ford?



-

Distributed SSSP

- Trivial $n-1$ upper bound (Bellman-Ford)
- Trivial $\Omega(D)$ lower bound
- Unweighted, undirected: BFS in D rounds
- Weighted, undirected?
- BFS, Bellman-Ford all $\Theta(n)$ time on a $D=2$ graph!



Blue edges: weight 1
Red edges: weight n^2

-

Prior Work

- Elkin, 2004: Approximate SSSP in sublinear time?

Prior Work

- Elkin, 2004: Approximate SSSP in sublinear time?
- [Nan14, STOC'14] $\tilde{O}(n^{1/2}D^{1/4} + D)$ undirected, slightly worse for directed

Prior Work

- Elkin, 2004: Approximate SSSP in sublinear time?
- [Nan14, STOC'14] $\tilde{O}(n^{1/2}D^{1/4} + D)$ undirected, slightly worse for directed
- [HKN16, STOC'16] $O(n^{1/2+o(1)} + n^{o(1)}D)$ undirected

Prior Work

- Elkin, 2004: Approximate SSSP in sublinear time?
- [Nan14, STOC'14] $\tilde{O}(n^{1/2}D^{1/4} + D)$ undirected, slightly worse for directed
- [HKN16, STOC'16] $O(n^{1/2+o(1)} + n^{o(1)}D)$ undirected
- [BKKL17, DISC'17] $\tilde{O}(n^{1/2} + D)$ undirected

Prior Work

- Elkin, 2004: Approximate SSSP in sublinear time?
- [Nan14, STOC'14] $\tilde{O}(n^{1/2}D^{1/4} + D)$ undirected, slightly worse for directed
- [HKN16, STOC'16] $O(n^{1/2+o(1)} + n^{o(1)}D)$ undirected
- [BKKL17, DISC'17] $\tilde{O}(n^{1/2} + D)$ undirected
- [SHK+11, STOC'11] $\tilde{O}(n^{1/2} + D)$ lower bound

Prior Work

- Elkin, 2004: Approximate SSSP in sublinear time?
- [Nan14, STOC'14] $\tilde{O}(n^{1/2}D^{1/4} + D)$ undirected, slightly worse for directed
- [HKN16, STOC'16] $O(n^{1/2+o(1)} + n^{o(1)}D)$ undirected
- [BKKL17, DISC'17] $\tilde{O}(n^{1/2} + D)$ undirected
- [SHK+11, STOC'11] $\tilde{O}(n^{1/2} + D)$ lower bound

Prior Work

- Elkin, 2004: Approximate SSSP in sublinear time?
- [Nan14, STOC'14] $\tilde{O}(n^{1/2}D^{1/4} + D)$ undirected, slightly worse for directed
- [HKN16, STOC'16] $O(n^{1/2+o(1)} + n^{o(1)}D)$ undirected
- [BKKL17, DISC'17] $\tilde{O}(n^{1/2} + D)$ undirected
- [SHK+11, STOC'11] $\tilde{O}(n^{1/2} + D)$ lower bound

Prior Work

- Elkin, 2004: Approximate SSSP in sublinear time?
- [Nan14, STOC'14] $\tilde{O}(n^{1/2}D^{1/4} + D)$ undirected, slightly worse for directed
- [HKN16, STOC'16] $O(n^{1/2+o(1)} + n^{o(1)}D)$ undirected
- [BKKL17, DISC'17] $\tilde{O}(n^{1/2} + D)$ undirected
- [SHK+11, STOC'11] $\tilde{O}(n^{1/2} + D)$ lower bound
- Elkin, 2004: *Exact* SSSP in sublinear time?

Prior Work

- Elkin, 2004: Approximate SSSP in sublinear time?
- [Nan14, STOC'14] $\tilde{O}(n^{1/2}D^{1/4} + D)$ undirected, slightly worse for directed
- [HKN16, STOC'16] $O(n^{1/2+o(1)} + n^{o(1)}D)$ undirected
- [BKKL17, DISC'17] $\tilde{O}(n^{1/2} + D)$ undirected
- [SHK+11, STOC'11] $\tilde{O}(n^{1/2} + D)$ lower bound
- Elkin, 2004: *Exact* SSSP in sublinear time?
- [Elk17, STOC'17] $\tilde{O}(\max\{n^{5/6}, n^{2/3}D^{1/3}\})$ undirected

Prior Work

- Elkin, 2004: Approximate SSSP in sublinear time?
- [Nan14, STOC'14] $\tilde{O}(n^{1/2}D^{1/4} + D)$ undirected, slightly worse for directed
- [HKN16, STOC'16] $O(n^{1/2+o(1)} + n^{o(1)}D)$ undirected
- [BKKL17, DISC'17] $\tilde{O}(n^{1/2} + D)$ undirected
- [SHK+11, STOC'11] $\tilde{O}(n^{1/2} + D)$ lower bound
- Elkin, 2004: *Exact* SSSP in sublinear time?
- [Elk17, STOC'17] $\tilde{O}(\max\{n^{5/6}, n^{2/3}D^{1/3}\})$ undirected
- This talk: $\tilde{O}(n^{3/4}D^{1/4})$ directed

Prior Work

- Elkin, 2004: Approximate SSSP in sublinear time?
- [Nan14, STOC'14] $\tilde{O}(n^{1/2}D^{1/4} + D)$ undirected, slightly worse for directed
- [HKN16, STOC'16] $O(n^{1/2+o(1)} + n^{o(1)}D)$ undirected
- [BKKL17, DISC'17] $\tilde{O}(n^{1/2} + D)$ undirected
- [SHK+11, STOC'11] $\tilde{O}(n^{1/2} + D)$ lower bound
- Elkin, 2004: *Exact* SSSP in sublinear time?
- [Elk17, STOC'17] $\tilde{O}(\max\{n^{5/6}, n^{2/3}D^{1/3}\})$ undirected
- This talk: $\tilde{O}(n^{3/4}D^{1/4})$ directed
- [KN18, independent] $\tilde{O}(n^{1/2}D^{1/2})$ directed

Prior Work

- Elkin, 2004: Approximate SSSP in sublinear time?
- [Nan14, STOC'14] $\tilde{O}(n^{1/2}D^{1/4} + D)$ undirected, slightly worse for directed
- [HKN16, STOC'16] $O(n^{1/2+o(1)} + n^{o(1)}D)$ undirected
- [BKKL17, DISC'17] $\tilde{O}(n^{1/2} + D)$ undirected
- [SHK+11, STOC'11] $\tilde{O}(n^{1/2} + D)$ lower bound
- Elkin, 2004: *Exact* SSSP in sublinear time?
- [Elk17, STOC'17] $\tilde{O}(\max\{n^{5/6}, n^{2/3}D^{1/3}\})$ undirected
- This talk: $\tilde{O}(n^{3/4}D^{1/4})$ directed
- [KN18, independent] $\tilde{O}(n^{1/2}D^{1/2})$ directed
- [HNS17, FOCS'17] $\tilde{O}(n^{5/4})$ time APSP on directed graphs

Sampling Centers

- [UY91]: Sample a set of *centers*

Sampling Centers

- [UY91]: Sample a set of *centers*
- Each node declares itself a center w.p. k/n (think $k = \sqrt{n}$)

Sampling Centers

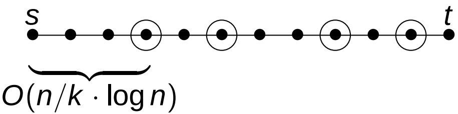
- [UY91]: Sample a set of *centers*
- Each node declares itself a center w.p. k/n (think $k = \sqrt{n}$)
- Chernoff: w.h.p. $\Theta(k)$ centers

Sampling Centers

- [UY91]: Sample a set of *centers*
- Each node declares itself a center w.p. k/n (think $k = \sqrt{n}$)
- Chernoff: w.h.p. $\Theta(k)$ centers
- Key property: for all nodes t, exists shortest $s \to t$ path with centers spaced at most $O(n/k \cdot \log n)$ nodes apart

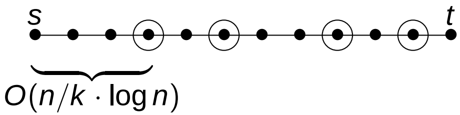
Sampling Centers

- [UY91]: Sample a set of *centers*
- Each node declares itself a center w.p. k/n (think $k = \sqrt{n}$)
- Chernoff: w.h.p. $\Theta(k)$ centers
- Key property: for all nodes t, exists shortest $s \rightarrow t$ path with centers spaced at most $O(n/k \cdot \log n)$ nodes apart



Sampling Centers

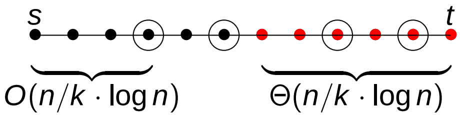
- [UY91]: Sample a set of *centers*
- Each node declares itself a center w.p. k/n (think $k = \sqrt{n}$)
- Chernoff: w.h.p. $\Theta(k)$ centers
- Key property: for all nodes t, exists shortest $s \to t$ path with centers spaced at most $O(n/k \cdot \log n)$ nodes apart



- Proof?

Sampling Centers

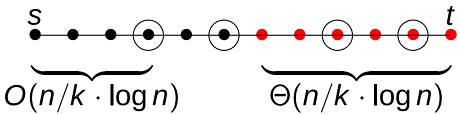
- [UY91]: Sample a set of *centers*
- Each node declares itself a center w.p. k/n (think $k = \sqrt{n}$)
- Chernoff: w.h.p. $\Theta(k)$ centers
- Key property: for all nodes t, exists shortest $s \to t$ path with centers spaced at most $O(n/k \cdot \log n)$ nodes apart



- Proof?

Sampling Centers

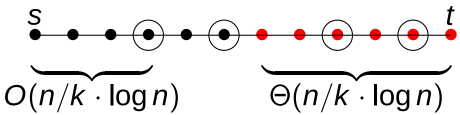
- [UY91]: Sample a set of *centers*
- Each node declares itself a center w.p. k/n (think $k = \sqrt{n}$)
- Chernoff: w.h.p. $\Theta(k)$ centers
- Key property: for all nodes t, exists shortest $s \to t$ path with centers spaced at most $O(n/k \cdot \log n)$ nodes apart



- Proof?
- Pr[no centers within block of $C \cdot n/k \cdot \log n$] $= (1 - k/n)^{C \cdot n/k \cdot \log n} \approx n^{-C}$

Sampling Centers

- [UY91]: Sample a set of *centers*
- Each node declares itself a center w.p. k/n (think $k = \sqrt{n}$)
- Chernoff: w.h.p. $\Theta(k)$ centers
- Key property: for all nodes t, exists shortest $s \to t$ path with centers spaced at most $O(n/k \cdot \log n)$ nodes apart



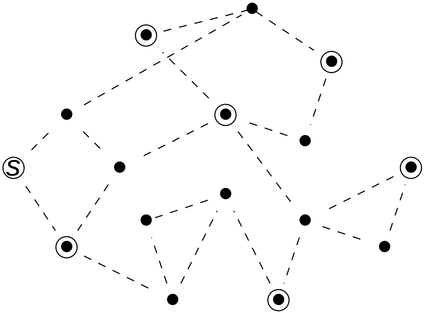
- Proof?
- Pr[no centers within block of $C \cdot n/k \cdot \log n$] $= (1 - k/n)^{C \cdot n/k \cdot \log n} \approx n^{-C}$
- Union bound: all $O(n)$ values of t, all $O(n)$ positions

Sampling Centers

- Key property: for all nodes t, exists shortest $s \to t$ path with centers spaced at most $O(n/k \cdot \log n)$ nodes apart

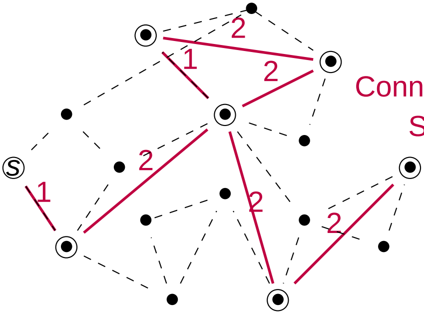
Sampling Centers

- Key property: for all nodes t, exists shortest $s \to t$ path with centers spaced at most $O(n/k \cdot \log n)$ nodes apart



Sampling Centers

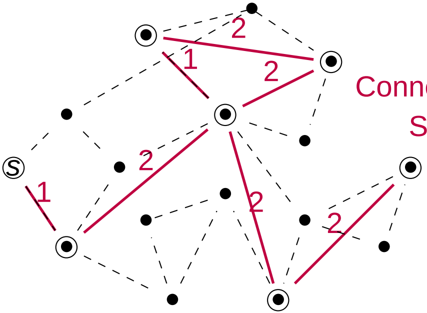
- Key property: for all nodes t, exists shortest $s \to t$ path with centers spaced at most $O(n/k \cdot \log n)$ nodes apart



Connect centers c_1, c_2 if their SP has $\tilde{O}(n/k)$ hops

Sampling Centers

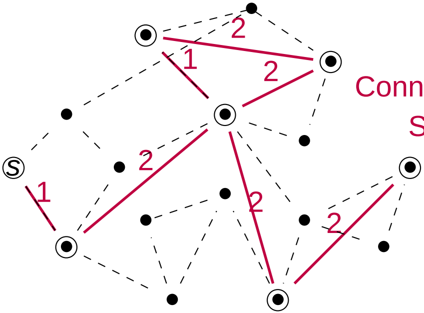
- Key property: for all nodes t, exists shortest $s \to t$ path with centers spaced at most $O(n/k \cdot \log n)$ nodes apart



Connect centers c_1, c_2 if their SP has $\tilde{O}(n/k)$ hops

Sampling Centers

- Key property: for all nodes t, exists shortest $s \to t$ path with centers spaced at most $O(n/k \cdot \log n)$ nodes apart

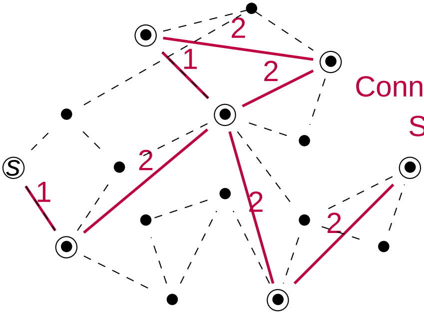


Connect centers c_1, c_2 if their SP has $\tilde{O}(n/k)$ hops

Skeleton graph G'

Sampling Centers

- Key property: for all nodes t, exists shortest $s \to t$ path with centers spaced at most $O(n/k \cdot \log n)$ nodes apart

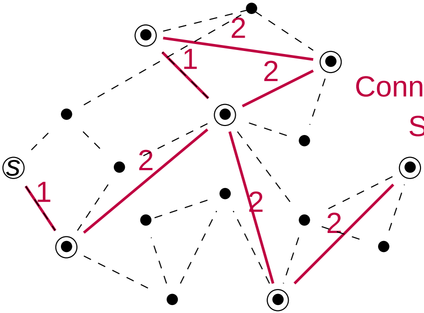


Connect centers c_1, c_2 if their SP has $\tilde{O}(n/k)$ hops

Skeleton graph G'

Sampling Centers

- Key property: for all nodes t, exists shortest $s \to t$ path with centers spaced at most $O(n/k \cdot \log n)$ nodes apart



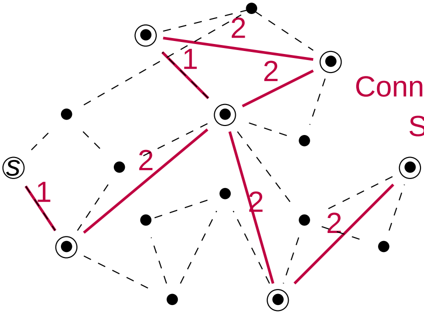
Connect centers c_1, c_2 if their SP has $\tilde{O}(n/k)$ hops

Skeleton graph G'

- Claim: $\forall t$, exists shortest $s \to t$ path in $G \cup G'$ using $\tilde{O}(n/k + k)$ hops

Sampling Centers

- Key property: for all nodes t, exists shortest $s \rightarrow t$ path with centers spaced at most $O(n/k \cdot \log n)$ nodes apart

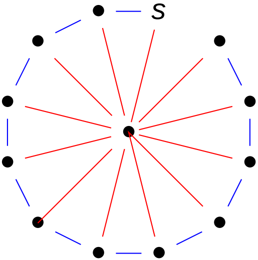


Connect centers c_1, c_2 if their SP has $\tilde{O}(n/k)$ hops

Skeleton graph G'

- Claim: $\forall t$, exists shortest $s \rightarrow t$ path in $G \cup G'$ using $\tilde{O}(n/k + k)$ hops
- G' shortcuts the graph G

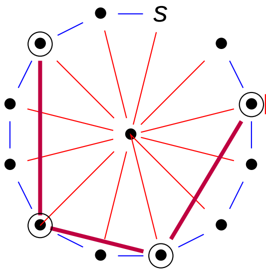
Sampling Centers



Blue edges: weight 1

Red edges: weight n^2

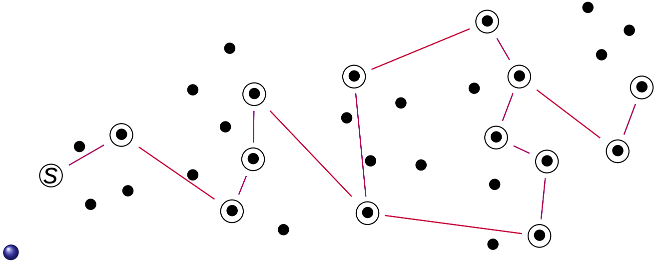
Sampling Centers



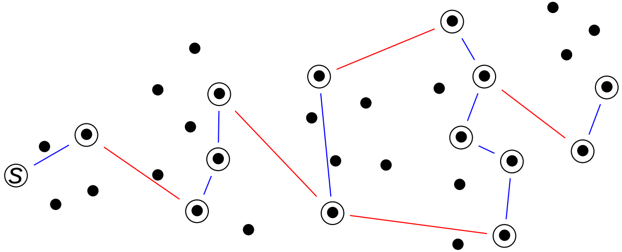
Blue edges: weight 1

Red edges: weight n^2

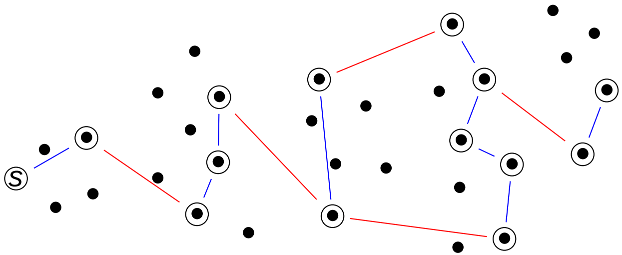
ShortRange [HNS17]



ShortRange [HNS17]

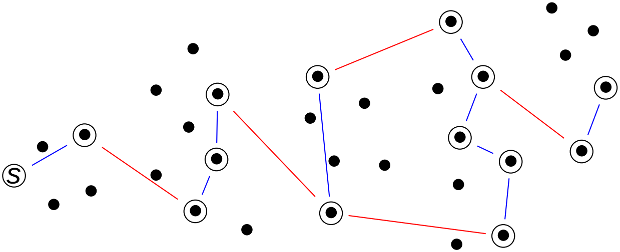


ShortRange [HNS17]



-
- Purple edges divided into blue edges ($d(c, c') \leq \ell$) and red edges ($d(c, c') > \ell$).

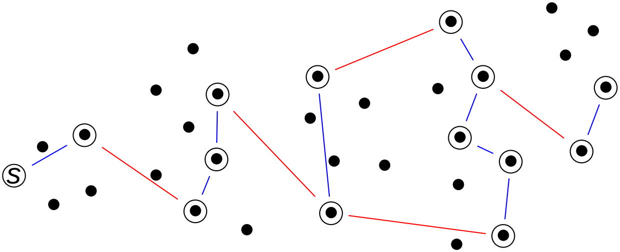
ShortRange [HNS17]



-

- Purple edges divided into blue edges ($d(c, c') \leq \ell$) and red edges ($d(c, c') > \ell$).
- ShortRange [HNS17]: fast distributed algorithm, computes blue edges but not red.

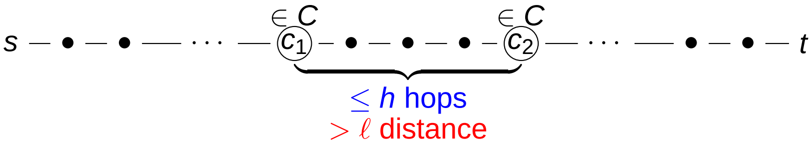
ShortRange [HNS17]



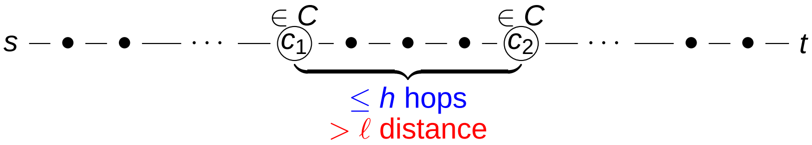
-
- Purple edges divided into blue edges ($d(c, c') \leq \ell$) and red edges ($d(c, c') > \ell$).
- ShortRange [HNS17]: fast distributed algorithm, computes blue edges but not red.
- Our contribution: compute red edges.

Our Contribution: Bucketing

- Red edges: $\leq h$ hops but distance $> \ell$



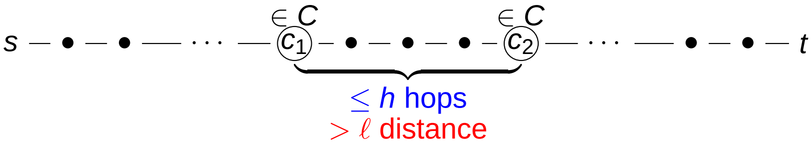
- Red edges: $\leq h$ hops but distance $> \ell$



- B-F works for any distance, so try B-F?

Our Contribution: Bucketing

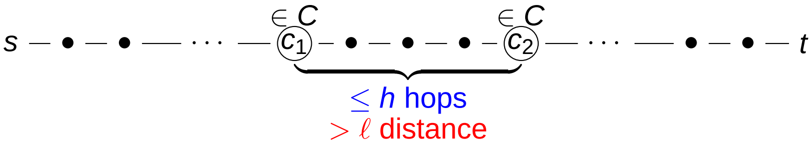
- Red edges: $\leq h$ hops but distance $> \ell$



- B-F works for any distance, so try B-F?
- Idea: bucket the nodes, single B-F for each bucket

Our Contribution: Bucketing

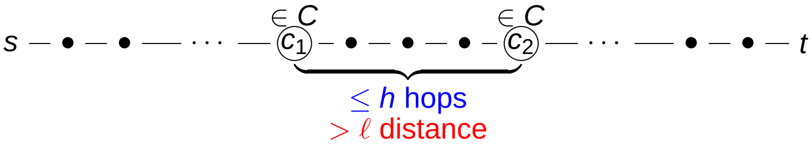
- Red edges: $\leq h$ hops but distance $> \ell$



- B-F works for any distance, so try B-F?
- Idea: bucket the nodes, single B-F for each bucket
  - Ensure: Red edges relevant to SSSP must connect different buckets

Our Contribution: Bucketing

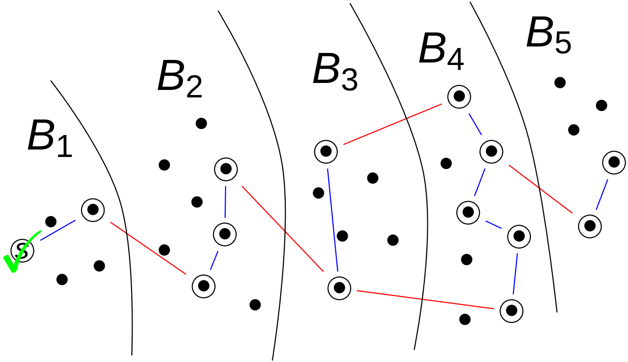
- Red edges: $\leq h$ hops but distance $> \ell$



- B-F works for any distance, so try B-F?
- Idea: bucket the nodes, single B-F for each bucket
  - Ensure: Red edges relevant to SSSP must connect different buckets

Processing Buckets

- Process buckets in increasing order, using blue edges
- Run B-F to depth h after finishing distances in one bucket

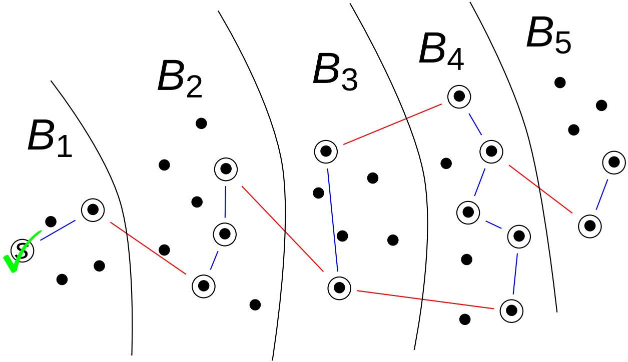


Blue edges: $\leq h$ hops and distance $\leq \ell$, known

Red edges: $\leq h$ hops and distance $> \ell$, unknown

Processing Buckets

- Process buckets in increasing order, using blue edges
- Run B-F to depth h after finishing distances in one bucket

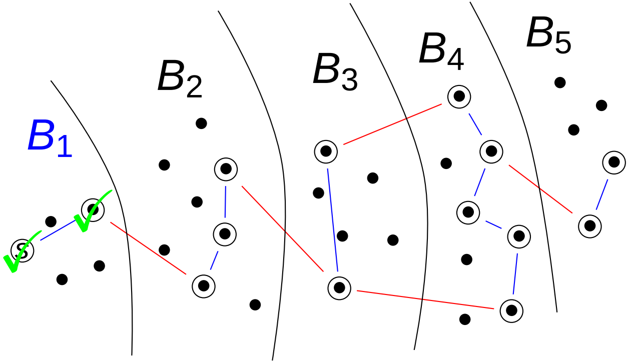


Blue edges: $\leq h$ hops and distance $\leq \ell$, known

Red edges: $\leq h$ hops and distance $> \ell$, unknown

Processing Buckets

- Process buckets in increasing order, using blue edges
- Run B-F to depth h after finishing distances in one bucket

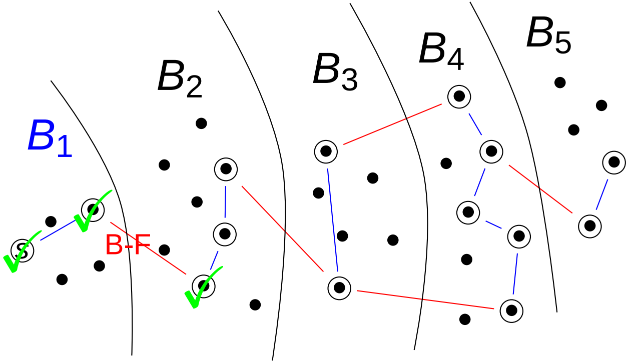


Blue edges: $\leq h$ hops and distance $\leq \ell$, known

Red edges: $\leq h$ hops and distance $> \ell$, unknown

Processing Buckets

- Process buckets in increasing order, using blue edges
- Run B-F to depth h after finishing distances in one bucket

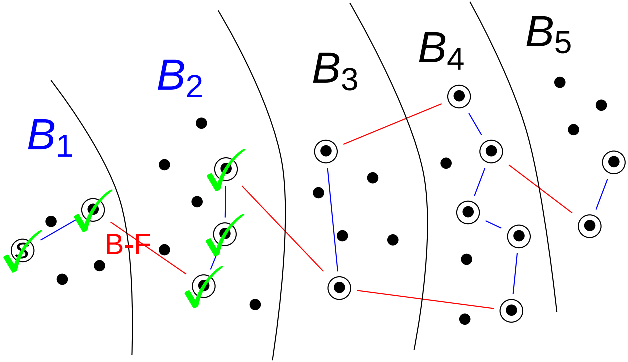


Blue edges: $\leq h$ hops and distance $\leq \ell$, known

Red edges: $\leq h$ hops and distance $> \ell$, unknown

Processing Buckets

- Process buckets in increasing order, using blue edges
- Run B-F to depth h after finishing distances in one bucket

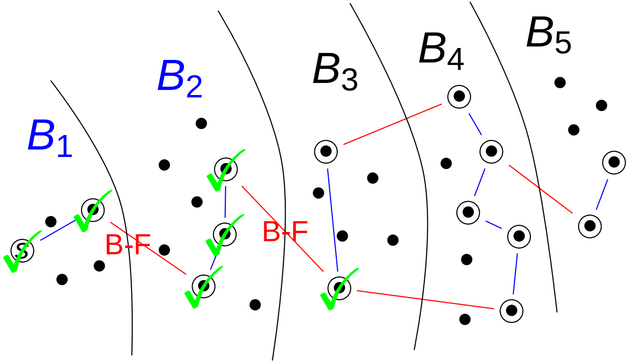


Blue edges: $\leq h$ hops and distance $\leq \ell$, known

Red edges: $\leq h$ hops and distance $> \ell$, unknown

Processing Buckets

- Process buckets in increasing order, using blue edges
- Run B-F to depth h after finishing distances in one bucket

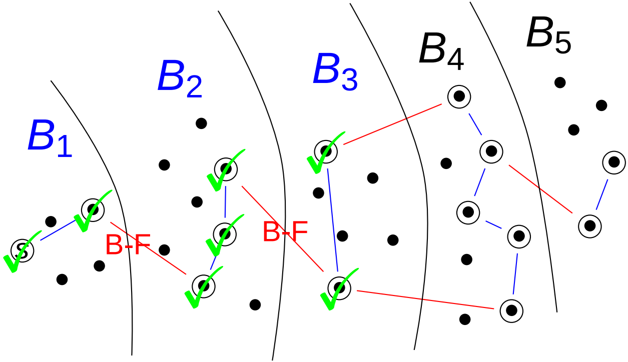


Blue edges: $\leq h$ hops and distance $\leq \ell$, known

Red edges: $\leq h$ hops and distance $> \ell$, unknown

Processing Buckets

- Process buckets in increasing order, using blue edges
- Run B-F to depth *h* after finishing distances in one bucket

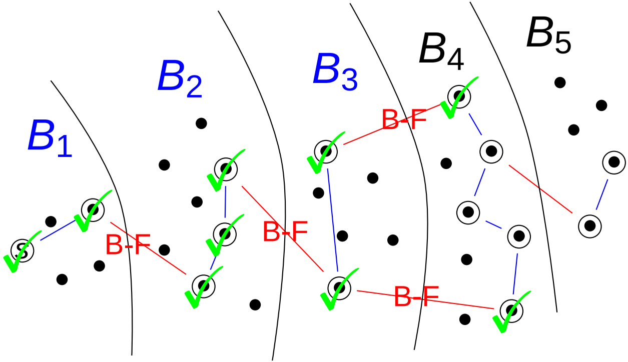


Blue edges: $\leq h$ hops and distance $\leq \ell$, known

Red edges: $\leq h$ hops and distance $> \ell$, unknown

Processing Buckets

- Process buckets in increasing order, using blue edges
- Run B-F to depth h after finishing distances in one bucket

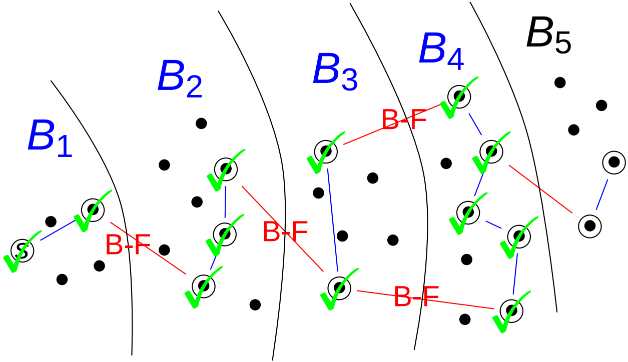


Blue edges: $\leq h$ hops and distance $\leq \ell$, known

Red edges: $\leq h$ hops and distance $> \ell$, unknown

Processing Buckets

- Process buckets in increasing order, using blue edges
- Run B-F to depth h after finishing distances in one bucket

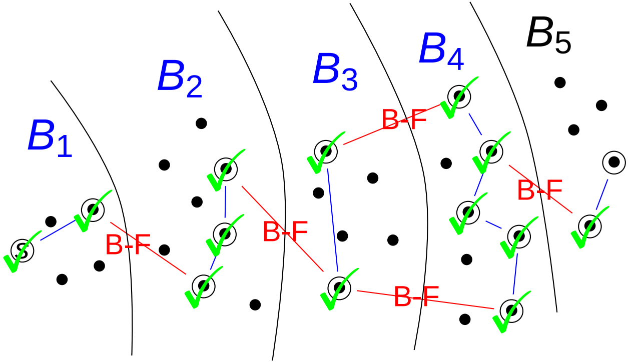


Blue edges: $\leq h$ hops and distance $\leq \ell$, known

Red edges: $\leq h$ hops and distance $> \ell$, unknown

Processing Buckets

- Process buckets in increasing order, using blue edges
- Run B-F to depth *h* after finishing distances in one bucket

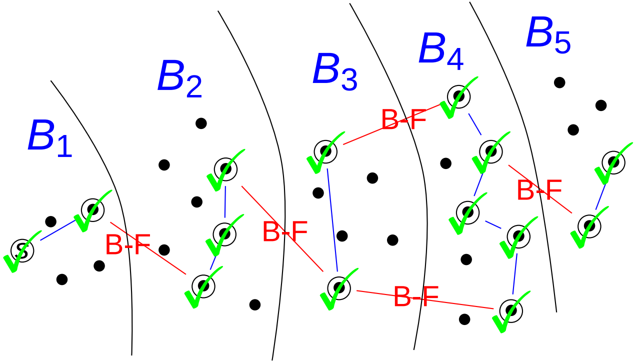


Blue edges: $\leq h$ hops and distance $\leq \ell$, known

Red edges: $\leq h$ hops and distance $> \ell$, unknown

Processing Buckets

- Process buckets in increasing order, using blue edges
- Run B-F to depth h after finishing distances in one bucket



Blue edges: $\leq h$ hops and distance $\leq \ell$, known

Red edges: $\leq h$ hops and distance $> \ell$, unknown

Entire Algorithm

Parameters k, ℓ

1. Every node joins center w.p. k/n

Entire Algorithm

Parameters k, ℓ

1. Every node joins center w.p. k/n
2. $h \leftarrow \Theta(n \log n/k)$

Entire Algorithm

Parameters k, ℓ

1. Every node joins center w.p. k/n
2. $h \leftarrow \Theta(n \log n/k)$
3. $\texttt{ShortRange}(\text{centers } c, \leq h \text{ hops, distance} \leq \ell)$

Entire Algorithm

Parameters k, ℓ

1. Every node joins center w.p. k/n
2. $h \leftarrow \Theta(n \log n/k)$
3. `ShortRange`(centers c, $\leq h$ hops, distance $\leq \ell$)
4. **for** each bucket in order **do**

Entire Algorithm

Parameters k, ℓ

1. Every node joins center w.p. k/n
2. $h \leftarrow \Theta(n \log n/k)$
3. `ShortRange`(centers c, $\leq h$ hops, distance $\leq \ell$)
4. **for** each bucket in order **do**
   1. Process bucket using blue edges

Entire Algorithm

Parameters k, ℓ

1. Every node joins center w.p. k/n
2. $h \leftarrow \Theta(n \log n/k)$
3. `ShortRange`(centers c, $\leq h$ hops, distance $\leq \ell$)
4. **for** each bucket in order **do**
    1. Process bucket using blue edges
    2. Run B-F for h rounds

Entire Algorithm

Parameters k, ℓ

1. Every node joins center w.p. k/n
2. $h \leftarrow \Theta(n \log n/k)$
3. `ShortRange`(centers c, $\leq h$ hops, distance $\leq \ell$)
4. **for** each bucket in order **do**
   1. Process bucket using blue edges
   2. Run B-F for h rounds

Entire Algorithm

Parameters k, ℓ

1. Every node joins center w.p. k/n
2. $h \leftarrow \Theta(n \log n/k)$
3. `ShortRange`(centers c, $\leq h$ hops, distance $\leq \ell$)
4. **for** each bucket in order **do**
   1. Process bucket using blue edges
   2. Run B-F for h rounds

Running time: optimize $k, \ell \rightarrow \tilde{O}(n^{3/4}D^{1/4})$ time

Open Questions

- Even approximate SSSP on directed graphs still unresolved

Open Questions

- Even approximate SSSP on directed graphs still unresolved
  - Beat $\tilde{O}(\min\{n^{2/3}, n^{1/2}D^{1/2}\})$ for $(1+\epsilon)$-approximate

Open Questions

- Even approximate SSSP on directed graphs still unresolved
  - Beat $\tilde{O}(\min\{n^{2/3}, n^{1/2}D^{1/2}\})$ for $(1+\epsilon)$-approximate
- Exact SSSP

Open Questions

- Even approximate SSSP on directed graphs still unresolved
  - Beat $\tilde{O}(\min\{n^{2/3}, n^{1/2}D^{1/2}\})$ for $(1+\epsilon)$-approximate
- Exact SSSP
  - $\tilde{O}(n^{1/2}D^{1/2})$ [KN18] is optimal for $D = \text{polylog}(n)$

Open Questions

- Even approximate SSSP on directed graphs still unresolved
  - Beat $\tilde{O}(\min\{n^{2/3}, n^{1/2}D^{1/2}\})$ for $(1+\epsilon)$-approximate
- Exact SSSP
  - $\tilde{O}(n^{1/2}D^{1/2})$ [KN18] is optimal for $D = \text{polylog}(n)$
  - Lower bound: $\tilde{O}(\sqrt{n} + D)$

Open Questions

- Even approximate SSSP on directed graphs still unresolved
  - Beat $\tilde{O}(\min\{n^{2/3}, n^{1/2}D^{1/2}\})$ for $(1+\epsilon)$-approximate
- Exact SSSP
  - $\tilde{O}(n^{1/2}D^{1/2})$ [KN18] is optimal for $D = \text{polylog}(n)$
  - Lower bound: $\tilde{O}(\sqrt{n} + D)$
  - For higher D, [KN18] get $\tilde{O}(\sqrt{n}D^{1/4} + n^{3/5})$ time

Open Questions

- Even approximate SSSP on directed graphs still unresolved
  - Beat $\tilde{O}(\min\{n^{2/3}, n^{1/2}D^{1/2}\})$ for $(1+\epsilon)$-approximate
- Exact SSSP
  - $\tilde{O}(n^{1/2}D^{1/2})$ [KN18] is optimal for $D = \text{polylog}(n)$
  - Lower bound: $\tilde{O}(\sqrt{n} + D)$
  - For higher D, [KN18] get $\tilde{O}(\sqrt{n}D^{1/4} + n^{3/5})$ time