Recent Progress on Distributed CONGEST Algorithms for Specific Graph Classes

Taisuke Izumi(NITECH, Japan)

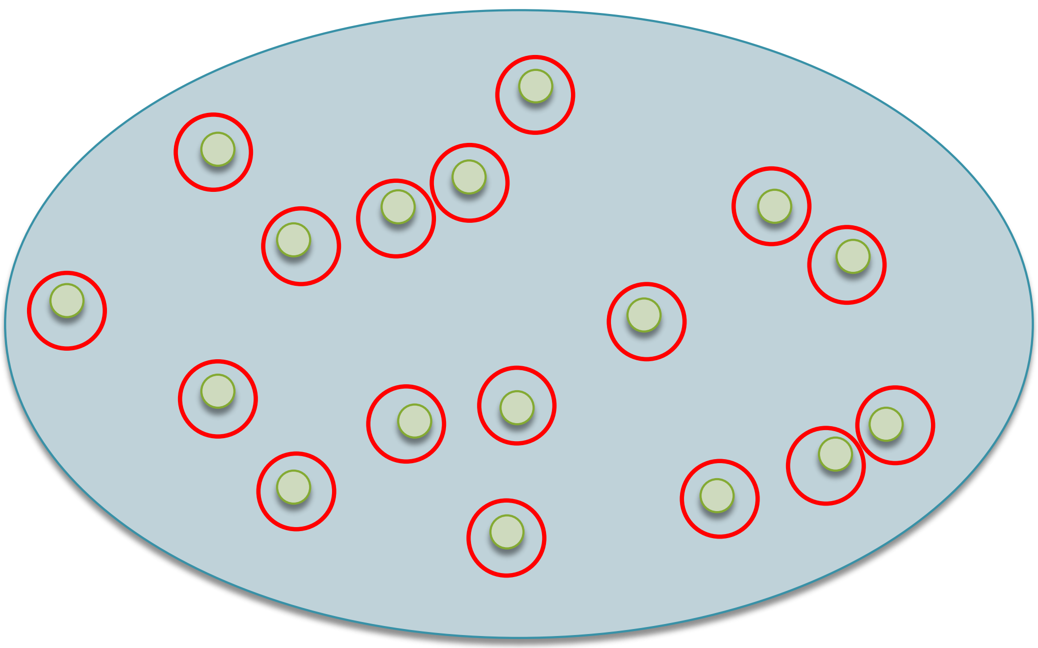
Model

- CONGEST model
  - Round-based synchrony
  - Network is a graph $G = \left(V(G), V(E)\right)$ of n nodes
  - Each link transmits $O(\log n)$ bits / round
    - Reliable

- Coping with low bandwidth is a primary difficulty
  - Many hardness results: MST, Diameter, Min-cut, etc.

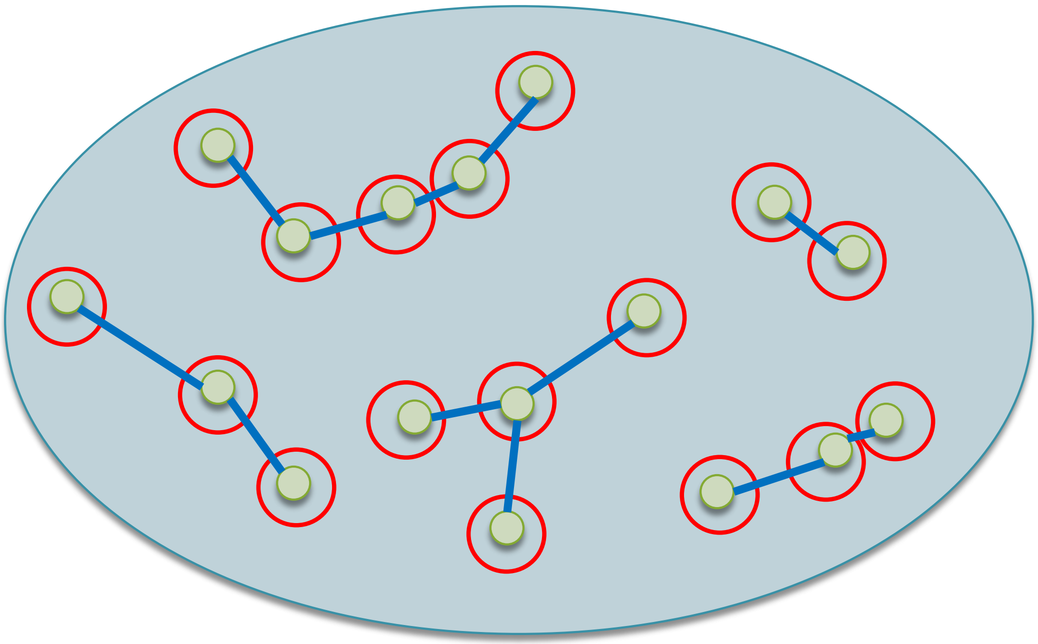
Warm-up : MST

- Classical GHS algorithm (= Distributed Boruvka)
  - Growing the fragments of MST
    - Each fragment finds its minimum outgoing edge (MOE)



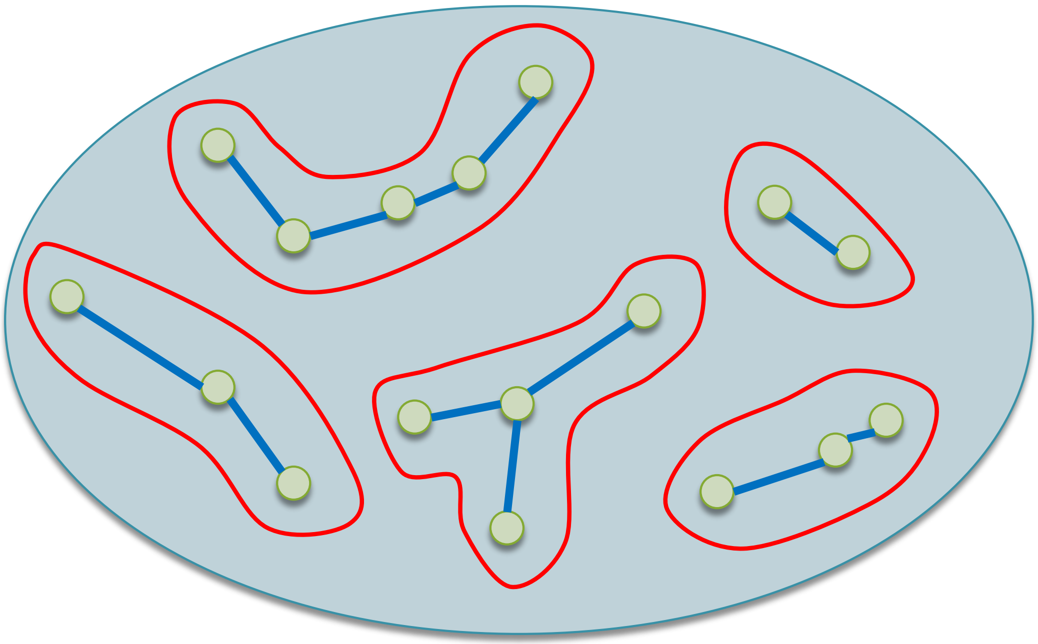
Warm-up : MST

- Classical GHS algorithm (= Distributed Boruvka)
  - Growing the fragments of MST
    - Each fragment finds its minimum outgoing edge (MOE)



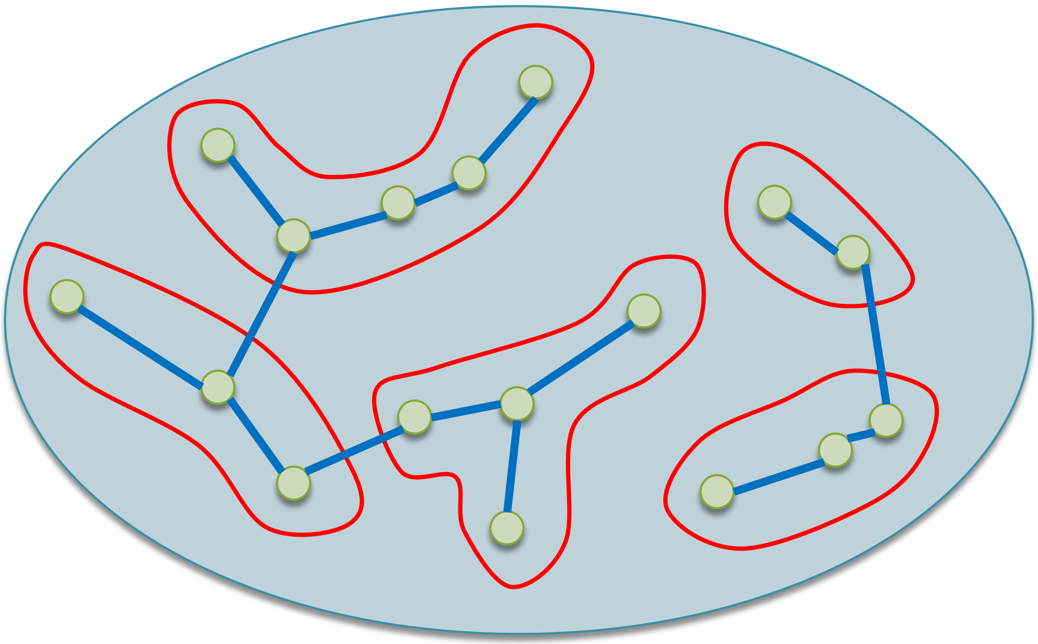
Warm-up : MST

- Classical GHS algorithm (= Distributed Boruvka)
  - Growing the fragments of MST
    - Each fragment finds its minimum outgoing edge (MOE)



Warm-up : MST

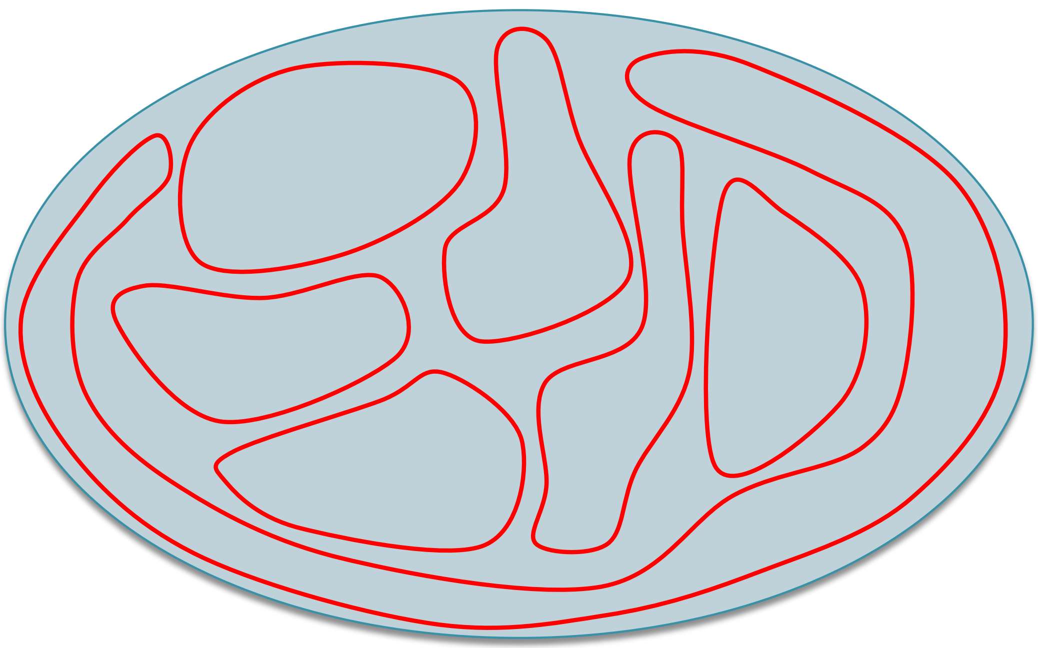
- Classical GHS algorithm (= Distributed Boruvka)
  - Growing the fragments of MST
    - Each fragment finds its minimum outgoing edge (MOE)



Warm-up : MST

- Finding MOEs is not necessarily fast
  - Even if the diameter of graph G is $D \ll n$, a fragment can have an $\Omega(\mathrm{n})$ diameter

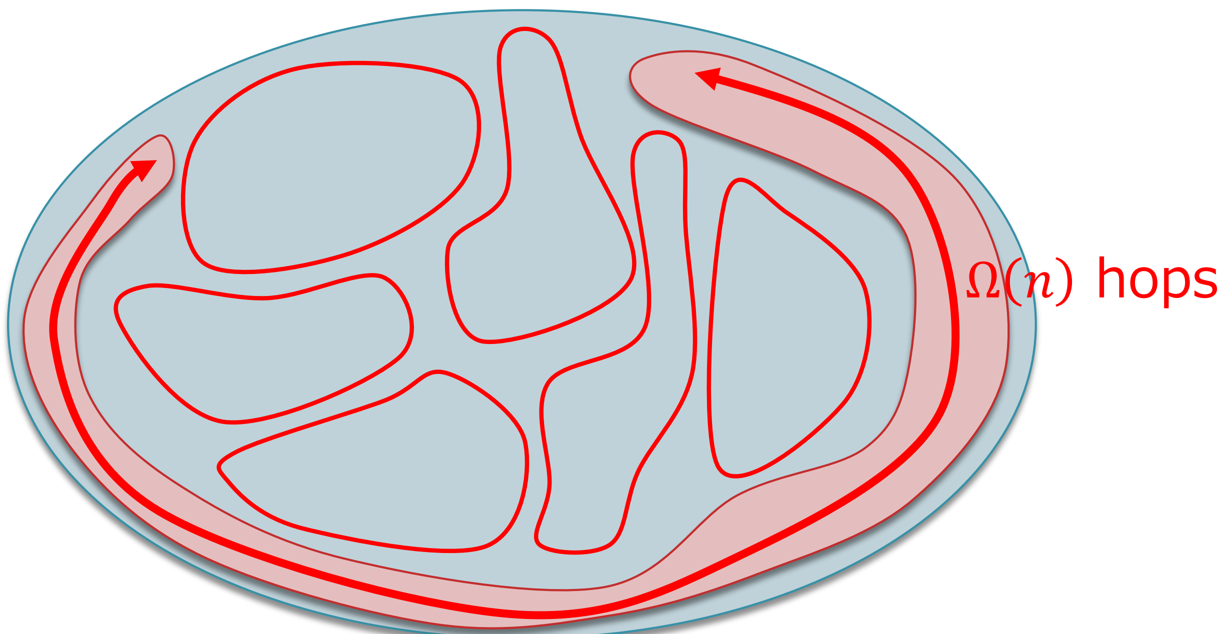
Naive in-fragment aggregation is slow !



Warm-up : MST

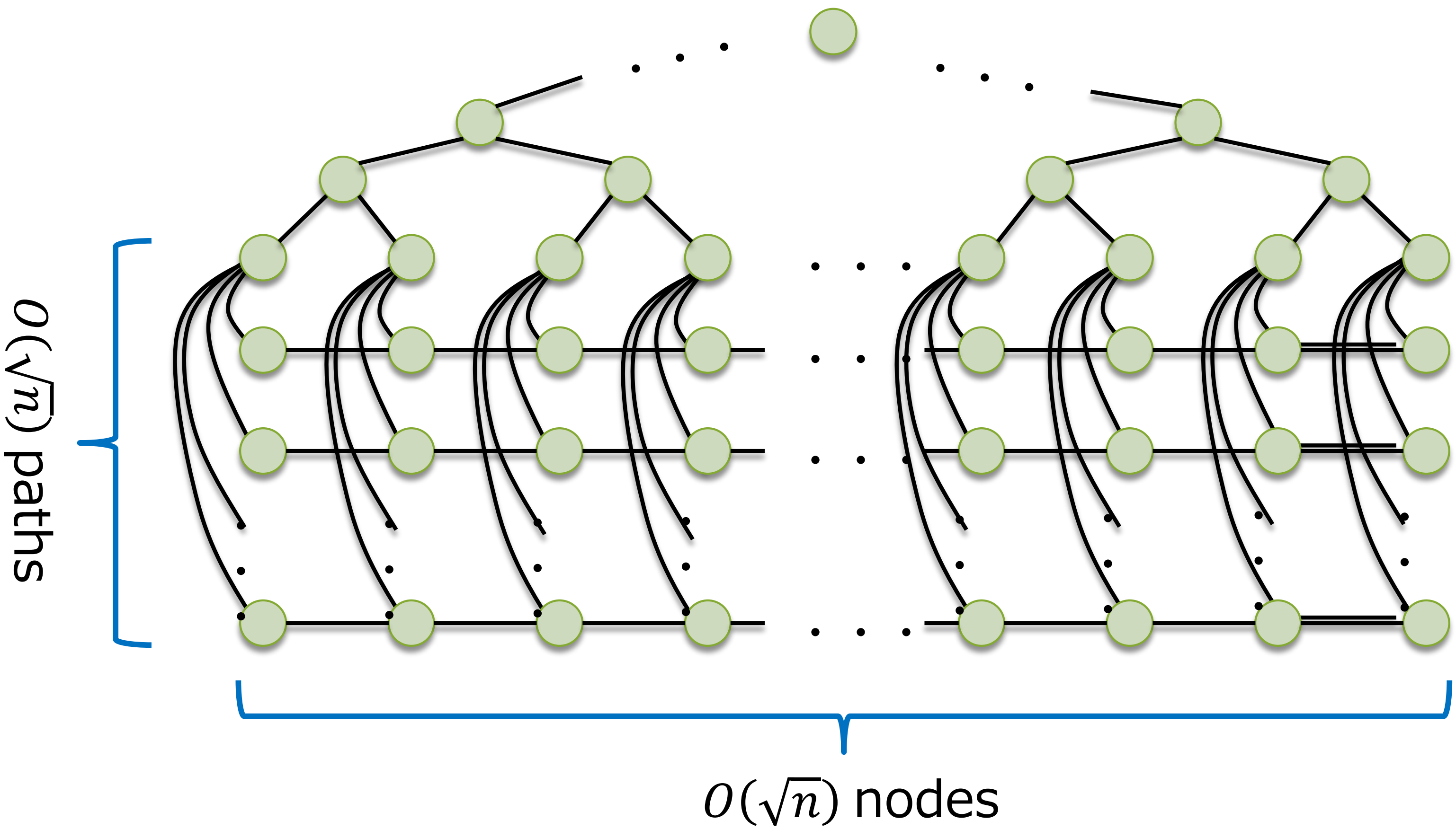
- Finding MOEs is not necessarily fast
  - Even if the diameter of graph G is $D \ll n$, a fragment can have an $\Omega(\mathrm{n})$ diameter

Naive in-fragment aggregation is slow !



A Hard-Core Instance for MST

- … and many other problems

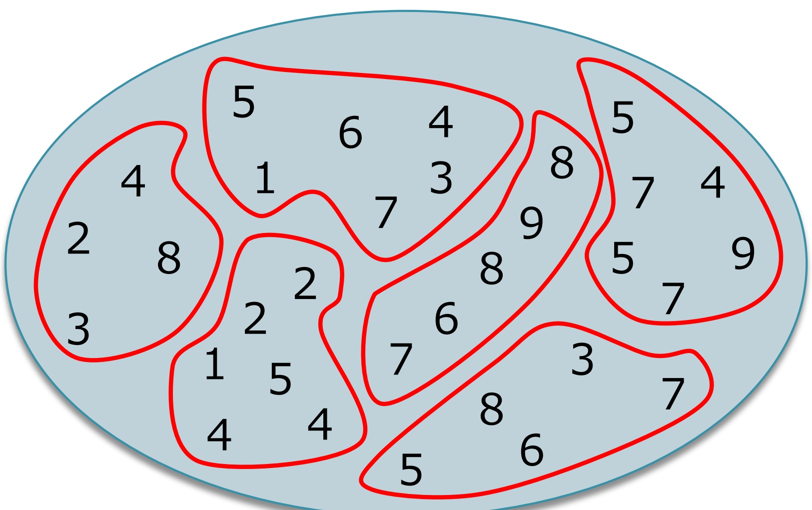


$\Omega(\sqrt{n} + D)$-round lower bound !

Partwise Aggregation(Minimum)

Definition :

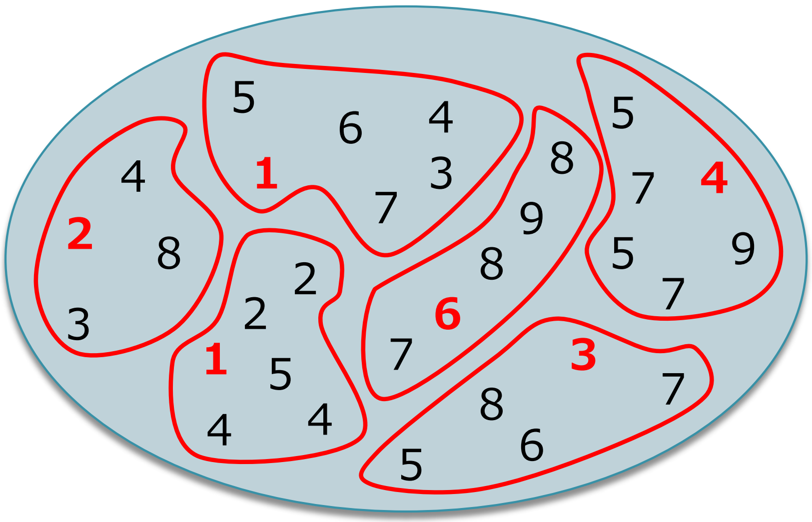
- Each node has one value ($O(\log n)$ bits)
- Each link can transmit $O(\log n)$ bits / round
- $V(G)$ is partitioned into a number of connected subgraphs $P_1, P_2, \cdots, P_N$
- For all P_i $(1 \leq i \leq N)$, find the minimum value in P_i independently



Partwise Aggregation(Minimum)

Definition :

- Each node has one value ($O(\log n)$ bits)
- Each link can transmit $O(\log n)$ bits / round
- $V(G)$ is partitioned into a number of connected subgraphs $P_1, P_2, \cdots, P_N$
- For all P_i ($1 \leq i \leq N$), find the minimum value in P_i independently



Motivation

- Partwise aggregation plays an important role for designing distributed algorithms in CONGEST model

  (CONGEST model : Round-based synchrony + $O(\log n)$-bit bandwidth)

- Meta-Theorem [Folklore + Ghaffari and Haeupler' 16]

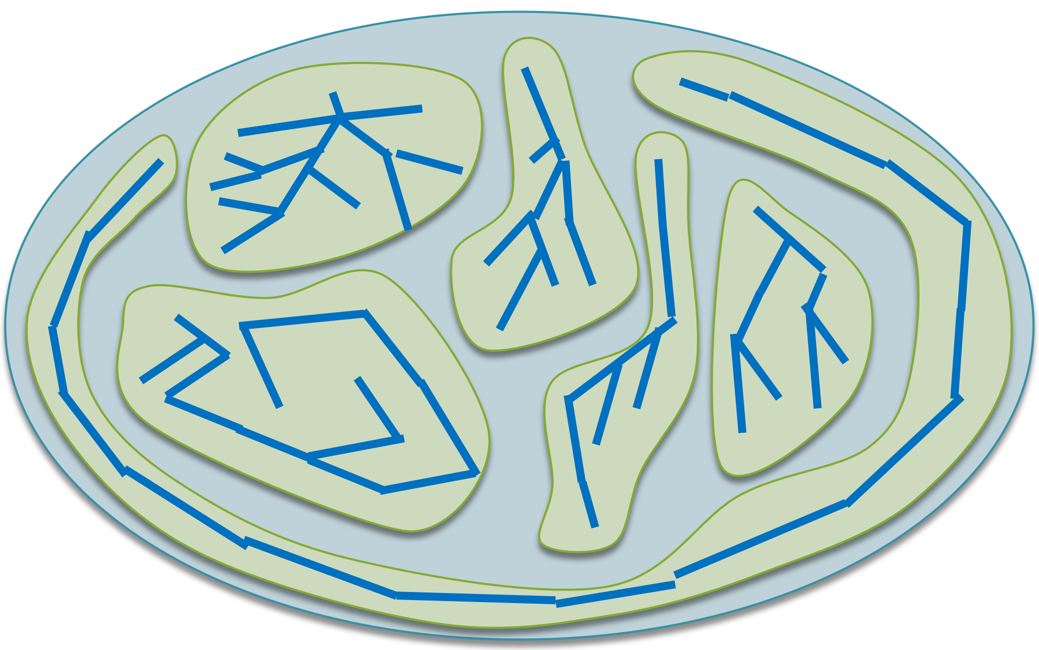
Efficient partwise aggregation

⇓

Efficient distributed algorithm for MST,
min-cut, weighted shortest path, and so on...

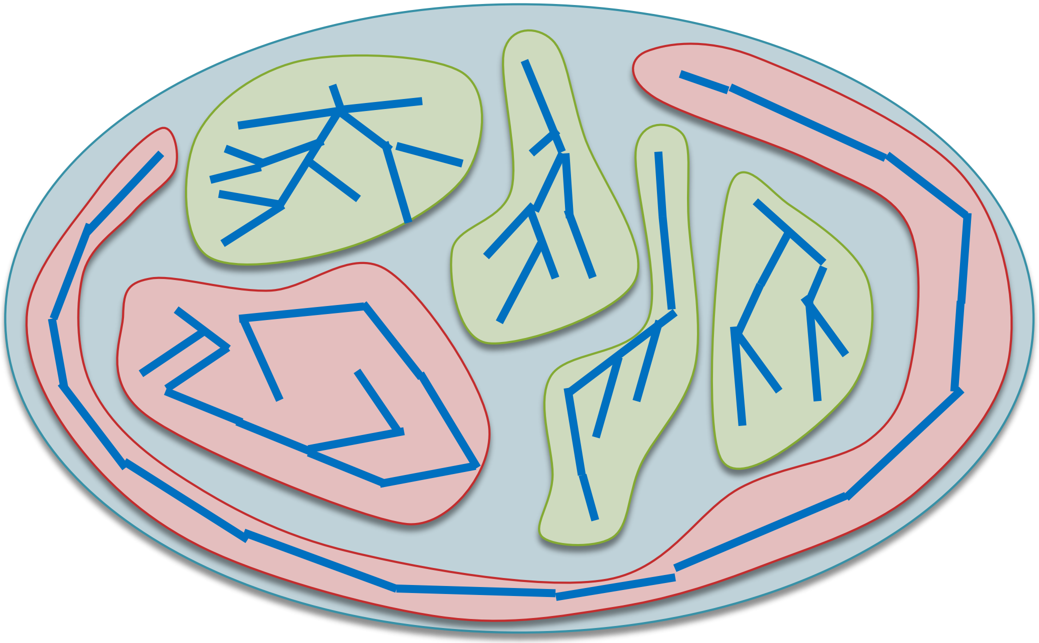
Naive Solution(1)

- In-part aggregation
  - BFS trees in parts might have a large diameter
    - The diameter even becomes $O(n)$, so $O(n)$ rounds



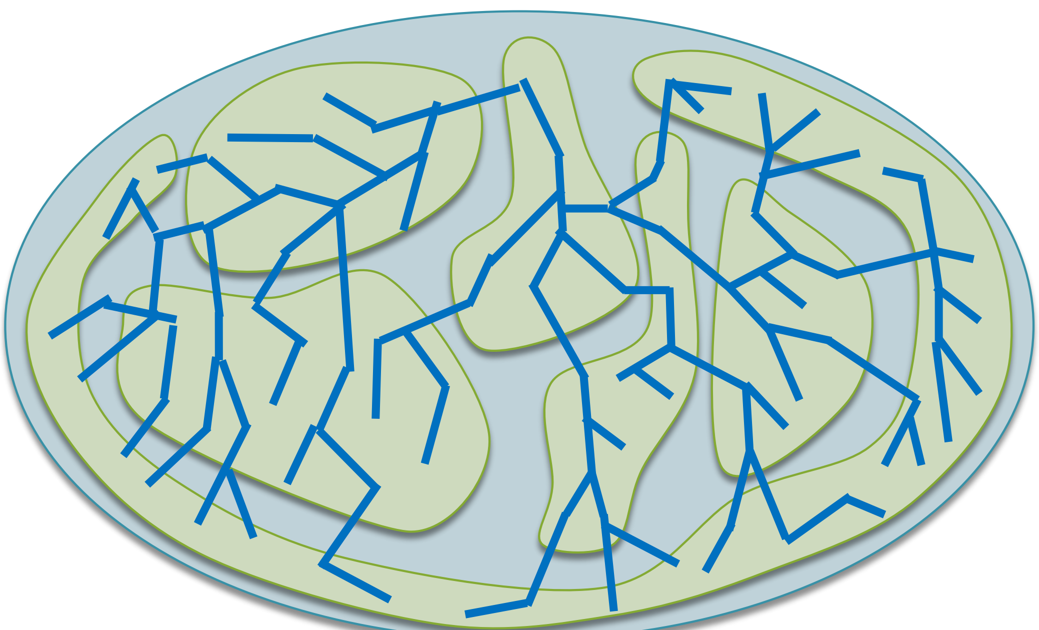
Naive Solution(1)

- In-part aggregation
  - BFS trees in parts might have a large diameter
    - The diameter even becomes $O(n)$, so $O(n)$ rounds



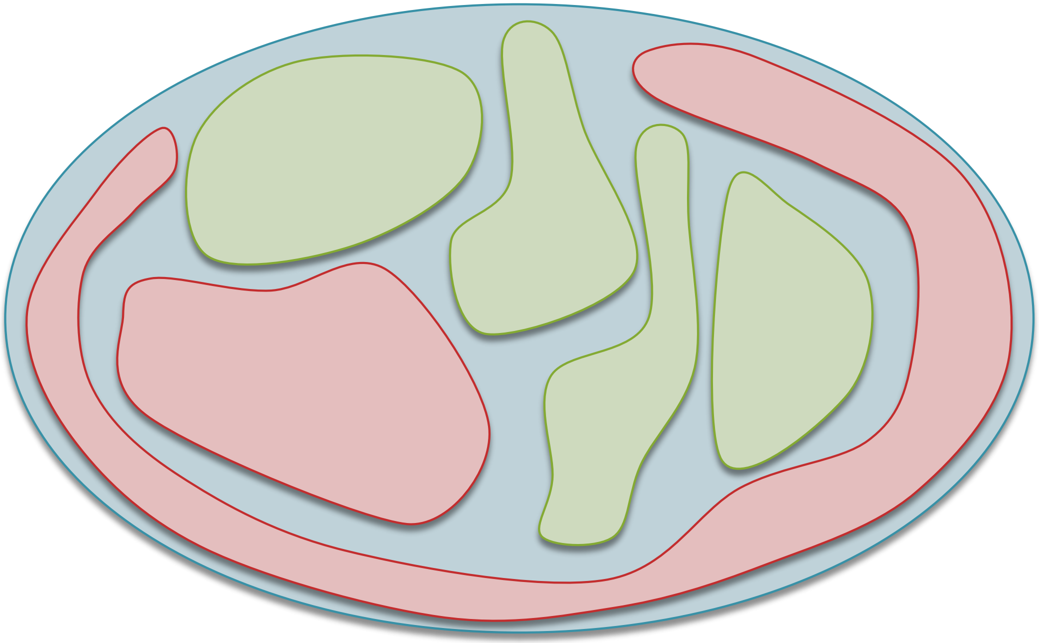
Naive Solution(2)

- Aggregation via a global BFS tree
  - Pipelined scheduling achieves $O(D+N)$ rounds
    - N can become $O(n)$, so $O(n)$ rounds



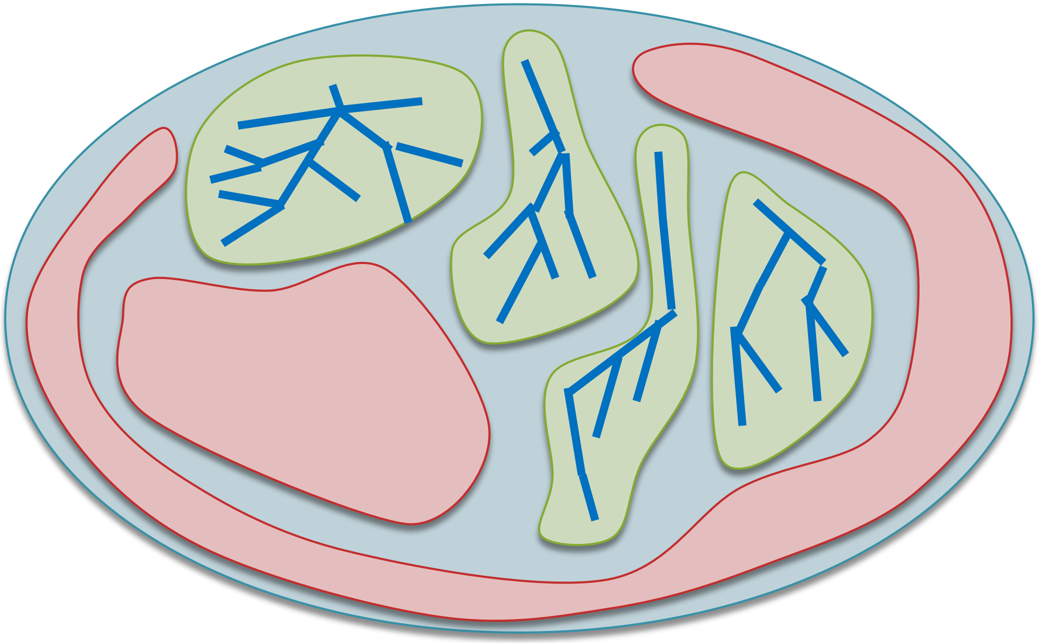
The Optimal Solution

- $|V(P_i)| \leq \sqrt{n}$: Naive in-part aggregation
- $|V(P_i)| > \sqrt{n}$: Use a BFS tree of the whole network + pipelined scheduling



The Optimal Solution

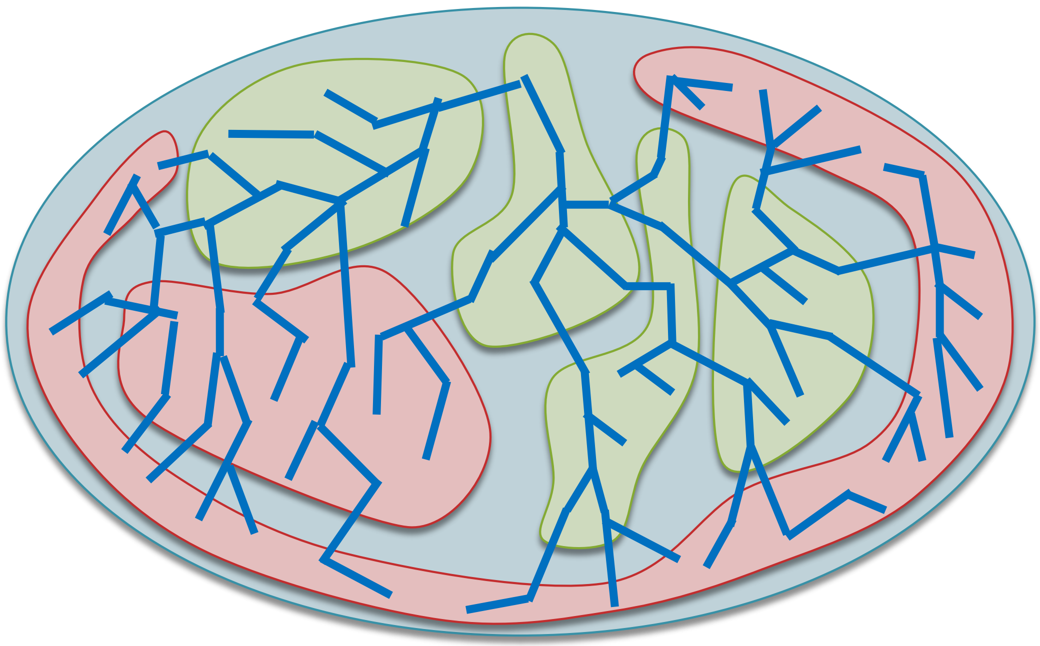
- $|V(P_i)| \leq \sqrt{n}$: Naive in-part aggregation
- $|V(P_i)| > \sqrt{n}$: Use a BFS tree of the whole network + pipelined scheduling



The Optimal Solution

- $|V(P_i)| \leq \sqrt{n}$: Naive in-part aggregation
- $|V(P_i)| > \sqrt{n}$: Use a BFS tree of the whole network + pipelined scheduling

$\tilde{O}\left(\sqrt{n} + D\right)$-round solution



Good Algorithms for Good Graphs

- This is an existential lower bound
  - There exists “an instance” exhibiting expensive cost
- We can expect much faster aggregation for many “not-so-bad” instances
  - Universal Lower bound : $\Omega(D)$ rounds

Problem

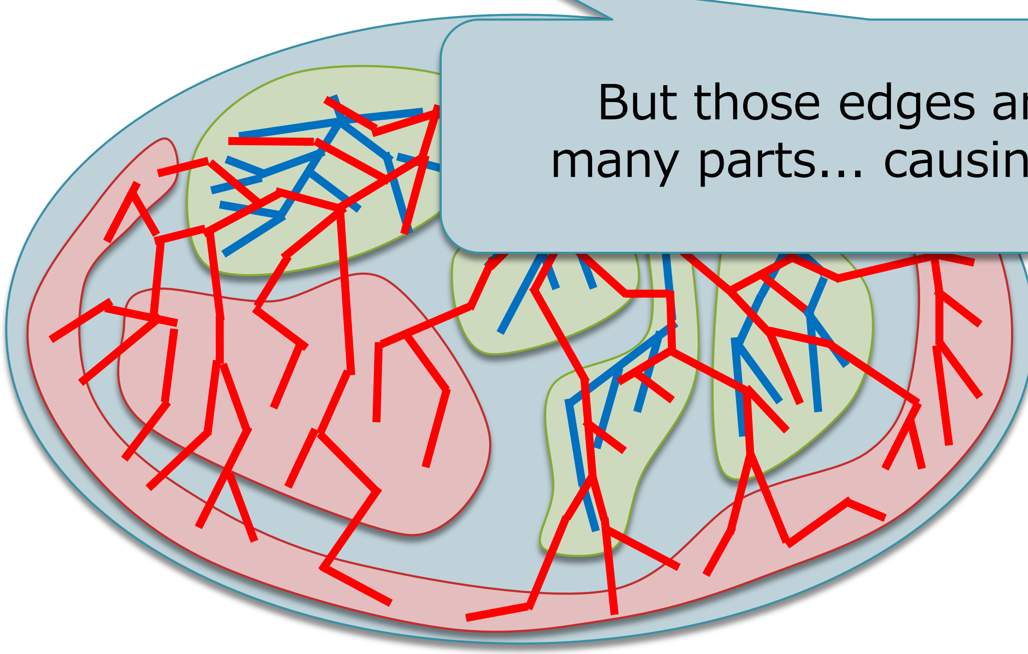
What graphs (classes) allow faster aggregation?

Shortcuts - An alternative view of P.A.

- $|V(P_i)| \leq \sqrt{n}$: Naive in-part aggregation
- $|V(P_i)| > \sqrt{n}$: Use a BFS tree of the whole network + pipelined scheduling

Augmenting the edges outside of the part for faster aggregation

But those edges are shared by many parts... causing congestion !



(d,c)-shortcut

- Given a connected partition $P_1, P_2, \cdots, P_N$ of G

- (d, c)-shortcut is a subgraph $H_1, H_2, \cdots, H_N$ s.t.
  - For any i, $P_i + H_i$ has diameter at most d (dilation)
  - Each edge $e \in E(G)$ is used as a shortcut edge at most c times

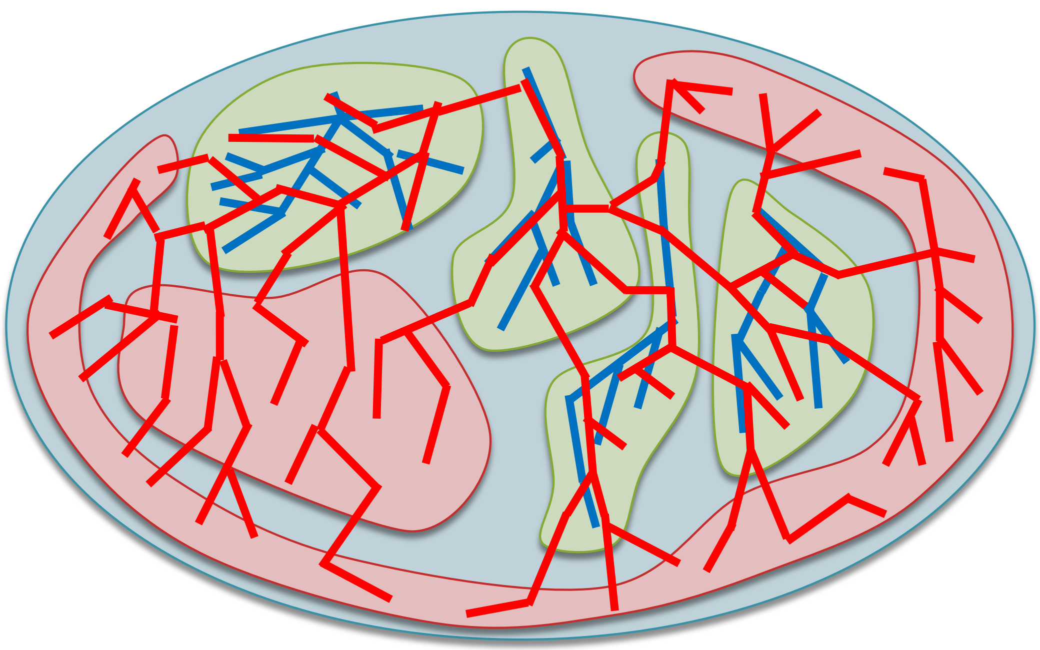
- An algorithm constructing (d,c)-shortcut for any partition with $O(f)$ rounds induces $\tilde{O}(d + c + f)$-round algorithms for partwise aggregation !

➡ For measuring quality, $\max\{d, c\}$ is usually enough.
We state simply by k-shortcuts if $k = \max\{d, c\}$

Shortcuts - An alternative view of P.A.

- $|V(P_i)| \leq \sqrt{n}$: Naive in-part aggregation — dilation : $\sqrt{n}$
- $|V(P_i)| > \sqrt{n}$: Use a BFS tree of the whole network + pipelined scheduling — congestion : $\sqrt{n}+1$

$O(\sqrt{n})$-shortcut



Shortcut and Graph Classes : Known Results

Graph Family	Quality	Construction	Lower Bound
Genus-g [GH16, HIZ16]	$O(\sqrt{g}D\log D)$	$O(\sqrt{g}D\log D)$	$\Omega\left(\frac{\sqrt{g}D}{\log g}\right)$
Treewidth-k [HIZ16]	$O(kD\log n)$	$O(kD\log n)$	$\Omega(kD)$
Minor-Free [HLZ18]	$\tilde{O}(D^2)$	$\tilde{O}(D^2)$	$\Omega(D)$(trivial)
Mixing Time τ [GKS17]	$O(\tau 2^{\sqrt{\log n \log\log n}}D)$	$O(\tau 2^{\sqrt{\log n \log\log n}}D)$	$\Omega(D)$(trivial)
k-chordal [KKIO19, in prep.]	$O(kD)$	$O(1)$	$\Omega(kD)$
Douling Dimesion-α [KKIO19, in prep.]	$O(D^\alpha)$	$O(1)$	$\Omega(D^\alpha)$
Cliquewidth-c [KKIO19, in prep.]	$O(\sqrt{n})$	$O(\sqrt{n})$	$\Omega(\sqrt{n})$ for $c = O(1)$
Small Diameter [KKI19, in prep.]	$\tilde{O}(n^{\frac{1}{2}-\frac{1}{2D-2}})$ for $D = 3,4$	$\tilde{O}(n^{\frac{1}{2}-\frac{1}{2D-2}})$ for $D = 3,4$	$\tilde{\Omega}(n^{\frac{1}{2}-\frac{1}{2D-2}})$ for any D [DHKKPPW13]

Shortcut and Graph Classes : Known Results

Spanning Tree-based approach

Graph Family	Quality	Construction	Lower Bound
Genus-g [GH16, HIZ16]	$O(\sqrt{g}D\log D)$	$O(\sqrt{g}D\log D)$	$\Omega\left(\frac{\sqrt{g}D}{\log g}\right)$
Treewidth-k [HIZ16]	$O(kD\log n)$	$O(kD\log n)$	$\Omega(kD)$
Minor-Free [HLZ18]	$\tilde{O}(D^2)$	$\tilde{O}(D^2)$	$\Omega(D)$(trivial)
Mixing Time τ [GKS17]	$O(\tau 2^{\sqrt{\log n \log\log n}}D)$	$O(\tau 2^{\sqrt{\log n \log\log n}}D)$	$\Omega(D)$(trivial)
k-chordal [KKIO19, in prep.]	$O(kD)$	$O(1)$	$\Omega(kD)$
Douling Dimesion-α [KKIO19, in prep.]	$O(D^\alpha)$	$O(1)$	$\Omega(D^\alpha)$
Cliquewidth-c [KKIO19, in prep.]	$O(\sqrt{n})$	$O(\sqrt{n})$	$\Omega(\sqrt{n})$ for $c = O(1)$
Small Diameter [KKI19, in prep.]	$\tilde{O}(n^{\frac{1}{2}-\frac{1}{2D-2}})$ for $D = 3,4$	$\tilde{O}(n^{\frac{1}{2}-\frac{1}{2D-2}})$ for $D = 3,4$	$\tilde{\Omega}(n^{\frac{1}{2}-\frac{1}{2D-2}})$ for any D [DHKKPPW13]

Shortcut and Graph Classes : Known Results

Random-Walk based approach →

Graph Family	Quality	Construction	Lower Bound
Genus-g [GH16, HIZ16]	$O(\sqrt{g}D\log D)$	$O(\sqrt{g}D\log D)$	$\Omega\left(\frac{\sqrt{g}D}{\log g}\right)$
Treewidth-k [HIZ16]	$O(kD\log n)$	$O(kD\log n)$	$\Omega(kD)$
Minor-Free [HLZ18]	$\tilde{O}(D^2)$	$\tilde{O}(D^2)$	$\Omega(D)$(trivial)
Mixing Time τ [GKS17]	$O(\tau 2^{\sqrt{\log n \log\log n}}D)$	$O(\tau 2^{\sqrt{\log n \log\log n}}D)$	$\Omega(D)$(trivial)
k-chordal [KKIO19, in prep.]	$O(kD)$	$O(1)$	$\Omega(kD)$
Douling Dimesion-α [KKIO19, in prep.]	$O(D^\alpha)$	$O(1)$	$\Omega(D^\alpha)$
Cliquewidth-c [KKIO19, in prep.]	$O(\sqrt{n})$	$O(\sqrt{n})$	$\Omega(\sqrt{n})$ for $c = O(1)$
Small Diameter [KKI19, in prep.]	$\tilde{O}(n^{\frac{1}{2}-\frac{1}{2D-2}})$ for $D = 3,4$	$\tilde{O}(n^{\frac{1}{2}-\frac{1}{2D-2}})$ for $D = 3,4$	$\tilde{\Omega}(n^{\frac{1}{2}-\frac{1}{2D-2}})$ for any D [DHKKPPW13]

Shortcut and Graph Classes : Known Results

Graph Family	Quality	Construction	Lower Bound
Genus-g [GH16, HIZ16]	$O(\sqrt{g}D\log D)$	$O(\sqrt{g}D\log D)$	$\Omega\left(\frac{\sqrt{g}D}{\log g}\right)$
Treewidth-k [HIZ16]	$O(kD\log n)$	$O(kD\log n)$	$\Omega(kD)$
Minor-Free [HLZ18]	$\tilde{O}(D^2)$	$\tilde{O}(D^2)$	$\Omega(D)$(trivial)
Mixing Time τ [GKS17]	$O(\tau 2^{\sqrt{\log n \log\log n}}D)$	$O(\tau 2^{\sqrt{\log n \log\log n}}D)$	$\Omega(D)$(trivial)
k-chordal [KKIO19, in prep.]	$O(kD)$	$O(1)$	$\Omega(kD)$
Douling Dimesion-α [KKIO19, in prep.]	$O(D^{\alpha})$	$O(1)$	$\Omega(D^{\alpha})$
Cliquewidth-c [KKIO19, in prep.]	$O(\sqrt{n})$	$O(\sqrt{n})$	$\Omega(\sqrt{n})$ for $c = O(1)$
Small Diameter [KKI19, in prep.]	$\tilde{O}(n^{\frac{1}{2}-\frac{1}{2D-2}})$ for $D = 3,4$	$\tilde{O}(n^{\frac{1}{2}-\frac{1}{2D-2}})$ for $D = 3,4$	$\tilde{\Omega}(n^{\frac{1}{2}-\frac{1}{2D-2}})$ for any D [DHKKPPW13]

1-hop extension approach

Shortcut and Graph Classes : Known Results

Graph Family	Quality	Construction	Lower Bound
Genus-g [GH16, HIZ16]	$O(\sqrt{g}D\log D)$	$O(\sqrt{g}D\log D)$	$\Omega\left(\frac{\sqrt{g}D}{\log g}\right)$
Treewidth-k [HIZ16]	$O(kD\log n)$	$O(kD\log n)$	$\Omega(kD)$
Minor-Free [HLZ18]	$\tilde{O}(D^2)$	$\tilde{O}(D^2)$	$\Omega(D)$(trivial)
Mixing Time τ [GKS17]	$O(\tau 2^{\sqrt{\log n \log\log n}}D)$	$O(\tau 2^{\sqrt{\log n \log\log n}}D)$	$\Omega(D)$(trivial)
k-chordal [KKIO19, in prep.]	$O(kD)$	$O(1)$	$\Omega(kD)$
Douling Dimesion-α [KKIO19, in prep.]	$O(D^\alpha)$	$O(1)$	$\Omega(D^\alpha)$
Cliquewidth-c [KKIO19, in prep.]	$O(\sqrt{n})$	$O(\sqrt{n})$	$\Omega(\sqrt{n})$ for $c = O(1)$
Small Diameter [KKI19, in prep.]	$\tilde{O}(n^{\frac{1}{2}-\frac{1}{2D-2}})$ for $D = 3,4$	$\tilde{O}(n^{\frac{1}{2}-\frac{1}{2D-2}})$ for $D = 3,4$	$\tilde{\Omega}(n^{\frac{1}{2}-\frac{1}{2D-2}})$ for any D [DHKKPPW13]

Lower Bound → (Cliquewidth-c)

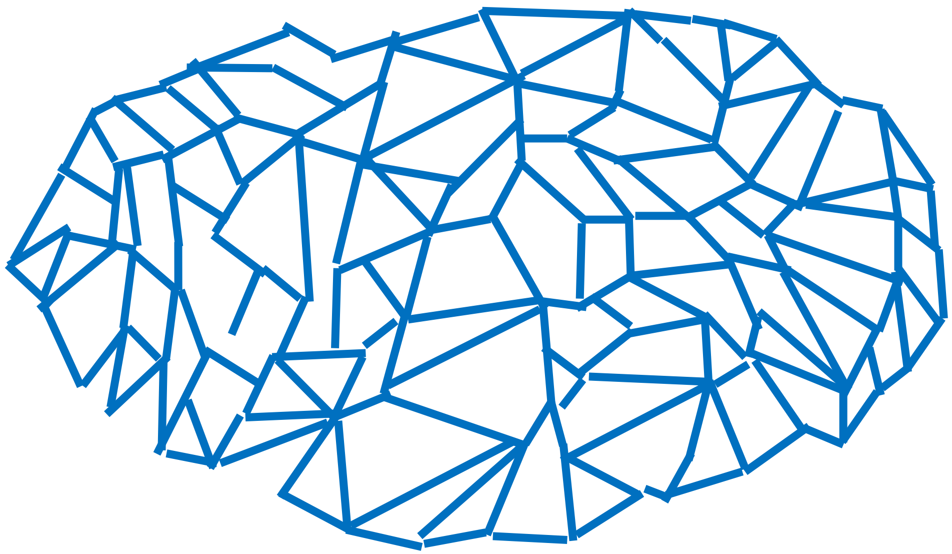
Shortcut and Graph Classes : Known Results

Graph Family	Quality	Construction	Lower Bound
Genus-g [GH16, HIZ16]	$O(\sqrt{g}D\log D)$	$O(\sqrt{g}D\log D)$	$\Omega\left(\frac{\sqrt{g}D}{\log g}\right)$
Treewidth-k [HIZ16]	$O(kD\log n)$	$O(kD\log n)$	$\Omega(kD)$
Minor-Free [HLZ18]	$\tilde{O}(D^2)$	$\tilde{O}(D^2)$	$\Omega(D)$(trivial)
Mixing Time τ [GKS17]	$O(\tau 2^{\sqrt{\log n \log\log n}} D)$	$O(\tau 2^{\sqrt{\log n \log\log n}} D)$	$\Omega(D)$(trivial)
k-chordal [KKIO19, in prep.]	$O(kD)$	$O(1)$	$\Omega(kD)$
Douling Dimesion-α [KKIO19, in prep.]	$O(D^\alpha)$	$O(1)$	$\Omega(D^\alpha)$
Cliquewidth-c [KKIO19, in prep.]	$O(\sqrt{n})$	$O(\sqrt{n})$	$\Omega(\sqrt{n})$ for $c = O(1)$
Small Diameter [KKI19, in prep.]	$\tilde{O}(n^{\frac{1}{2}-\frac{1}{2D-2}})$ for $D = 3,4$	$\tilde{O}(n^{\frac{1}{2}-\frac{1}{2D-2}})$ for $D = 3,4$	$\tilde{\Omega}(n^{\frac{1}{2}-\frac{1}{2D-2}})$ for any D [DHKKPPW13]

1-hop extension + α

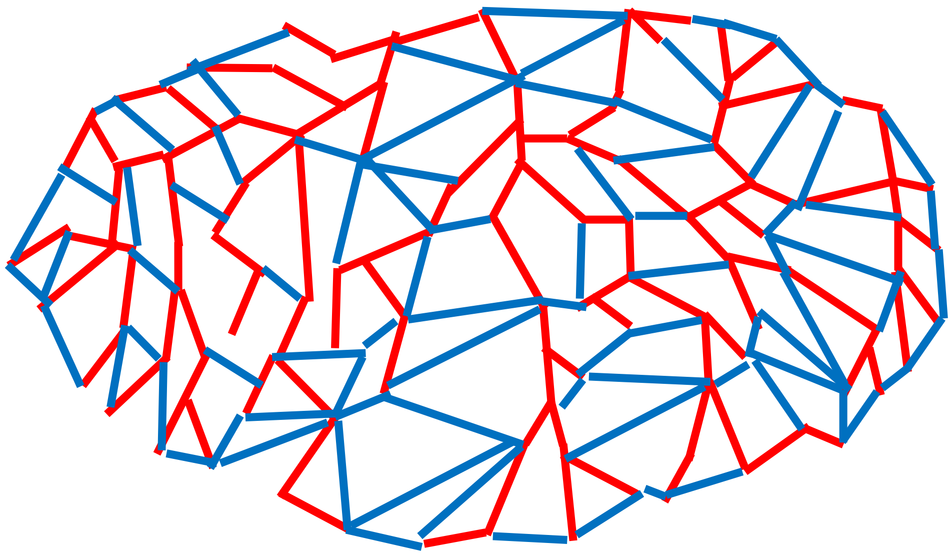
ST-approach : $O(D^2)$ quality for Planar Graphs

1. Construct a spanning tree



Planar Graph : $O(D^2)$ quality construction

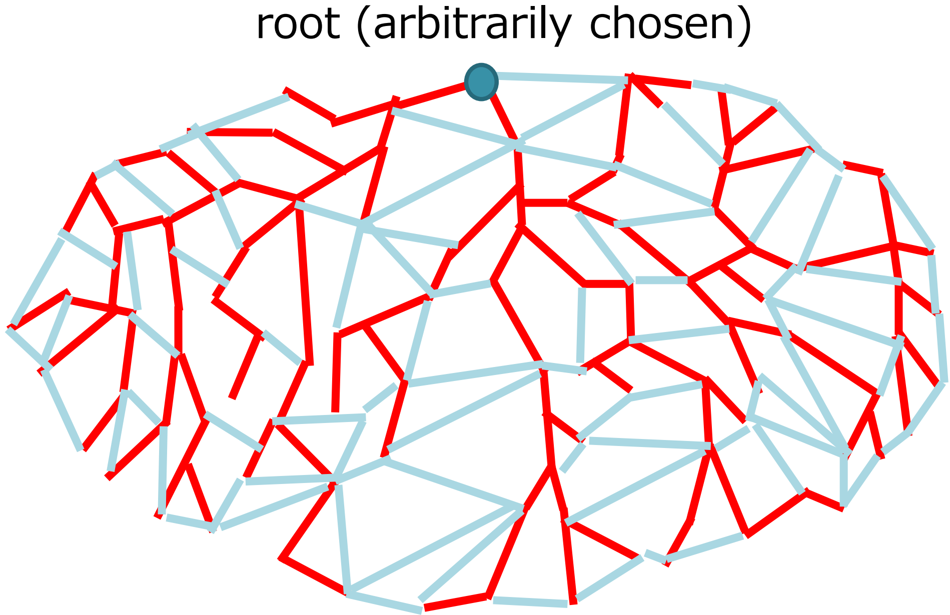
1. Construct a spanning tree



ST-approach : $O(D^2)$ quality for Planar Graphs

1. Construct a spanning tree

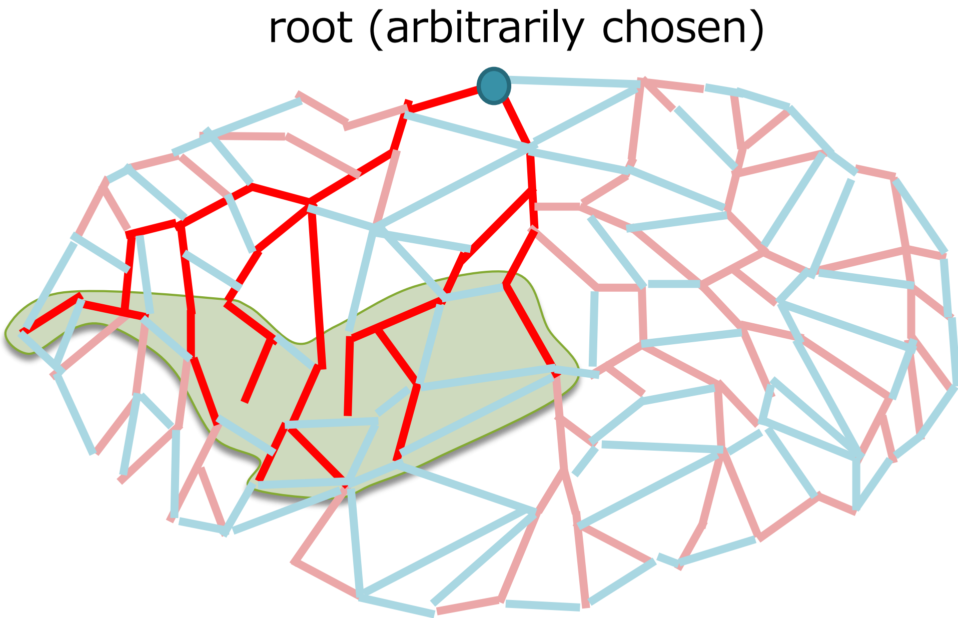
2. Each part takes upward path of the tree, except for the leftmost and rightmost paths



ST-approach : $O(D^2)$ quality for Planar Graphs

1. Construct a spanning tree

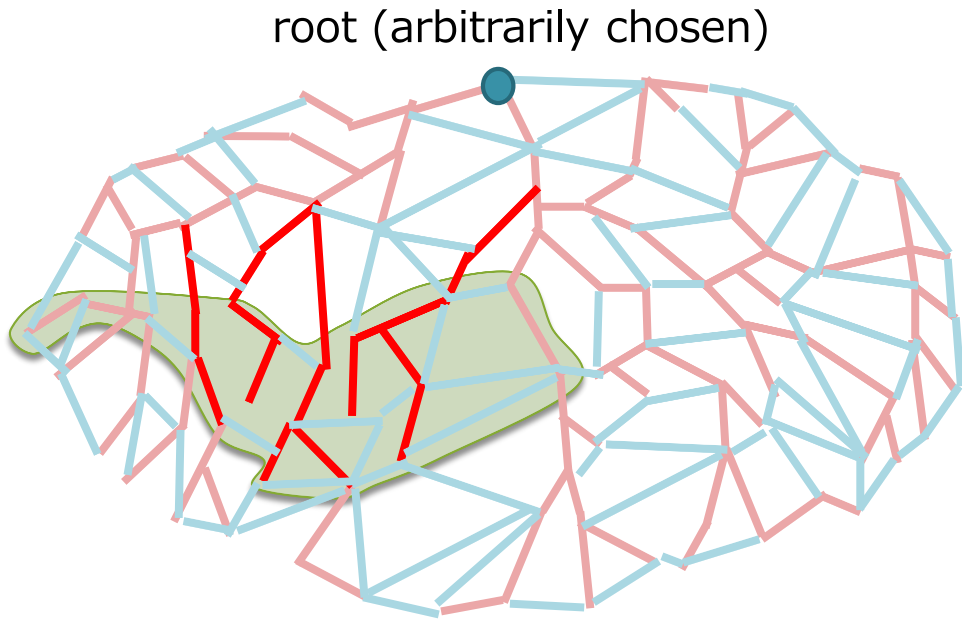
2. Each part takes upward path of the tree, except for the leftmost and rightmost paths



Planar Graph : $O(D^2)$ quality construction

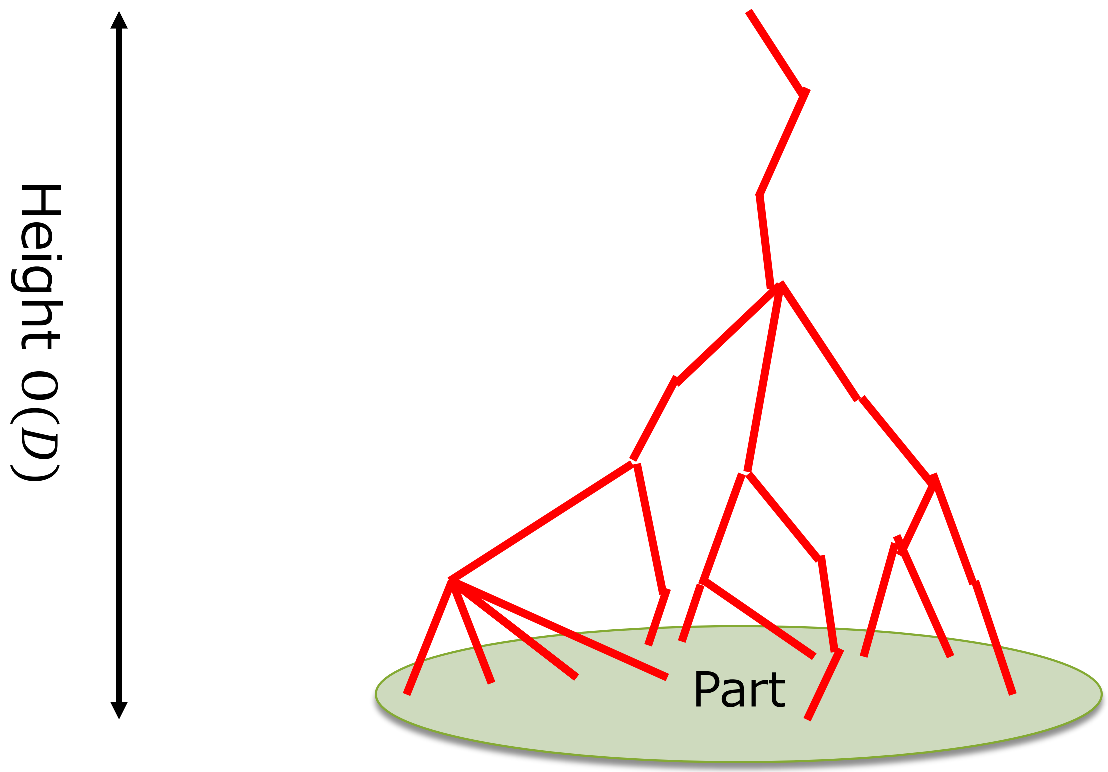
1. Construct a spanning tree

2. Each part takes upward path of the tree, except for the leftmost and rightmost paths



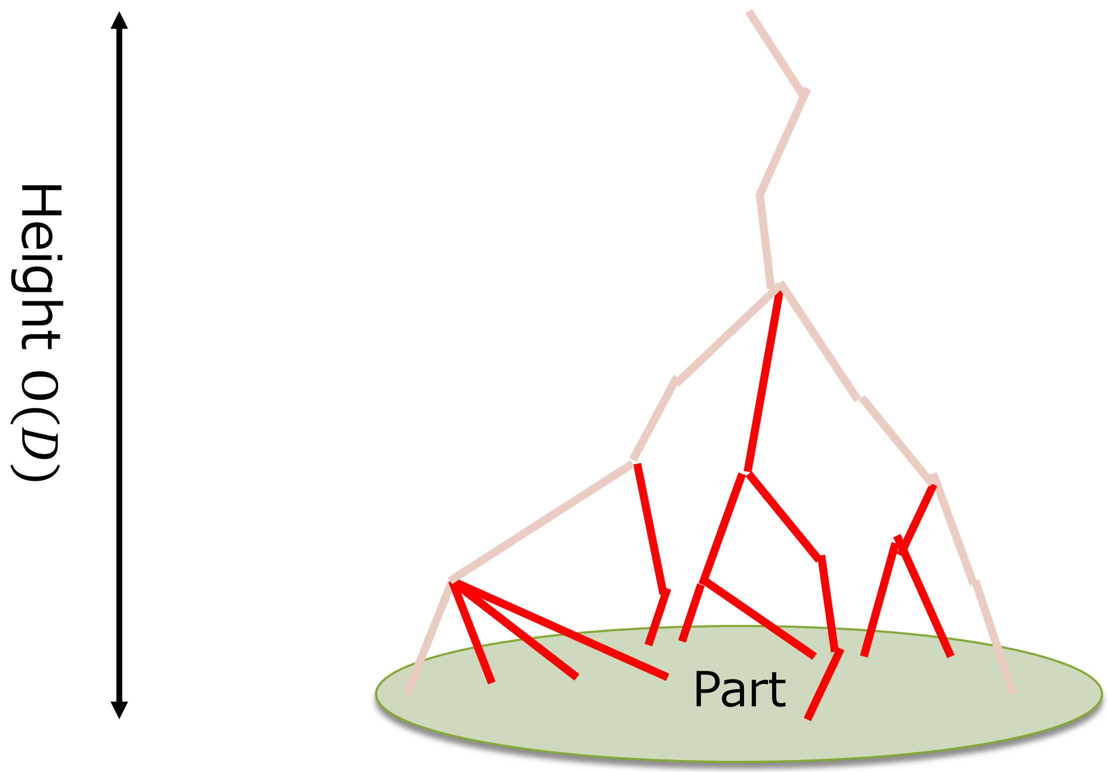
Proving the Quality

- Taking a BFS tree, this construction achieves
  - $O(D^2)$ dilation
  - $O(D)$ congestion



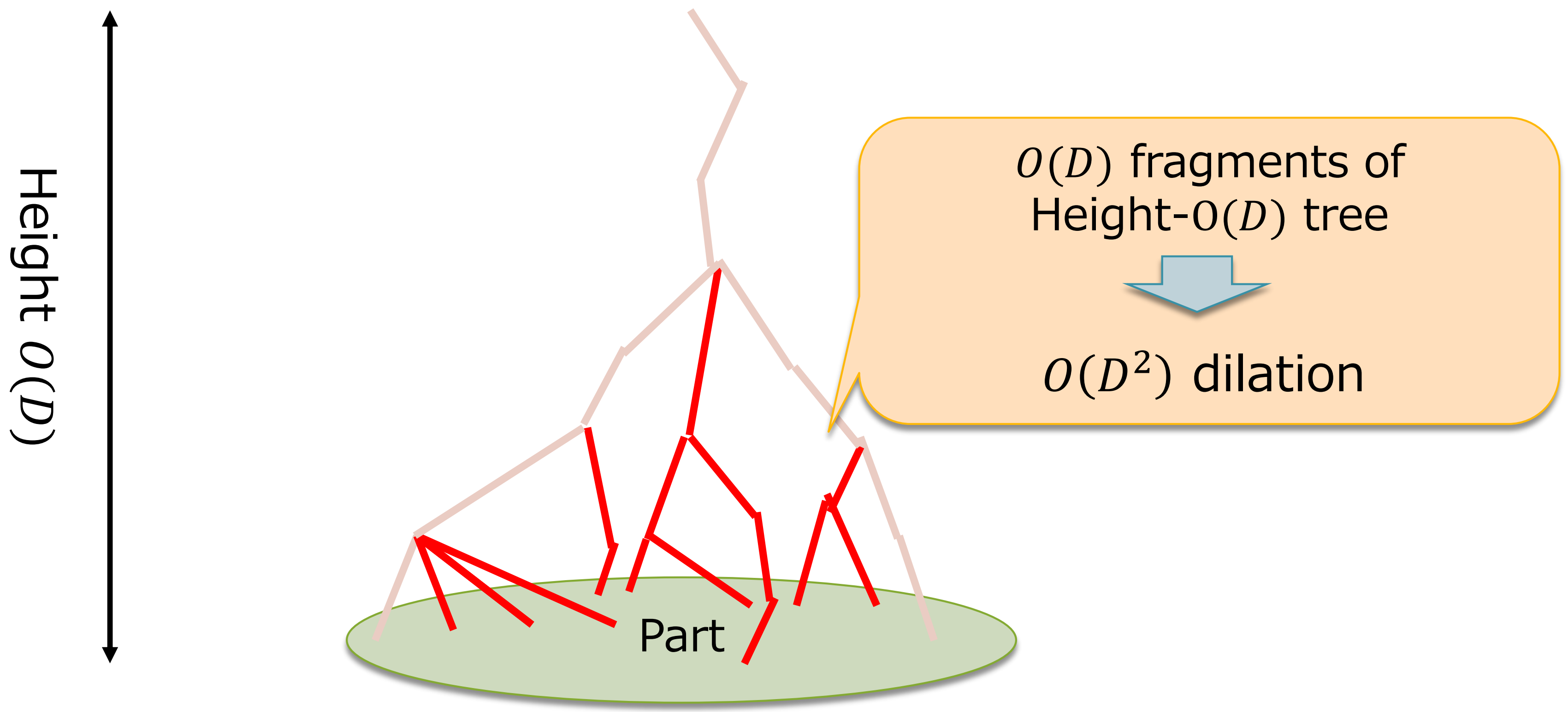
Proving the Quality

- Taking a BFS tree, this construction achieves
  - $O(D^2)$ dilation
  - $O(D)$ congestion



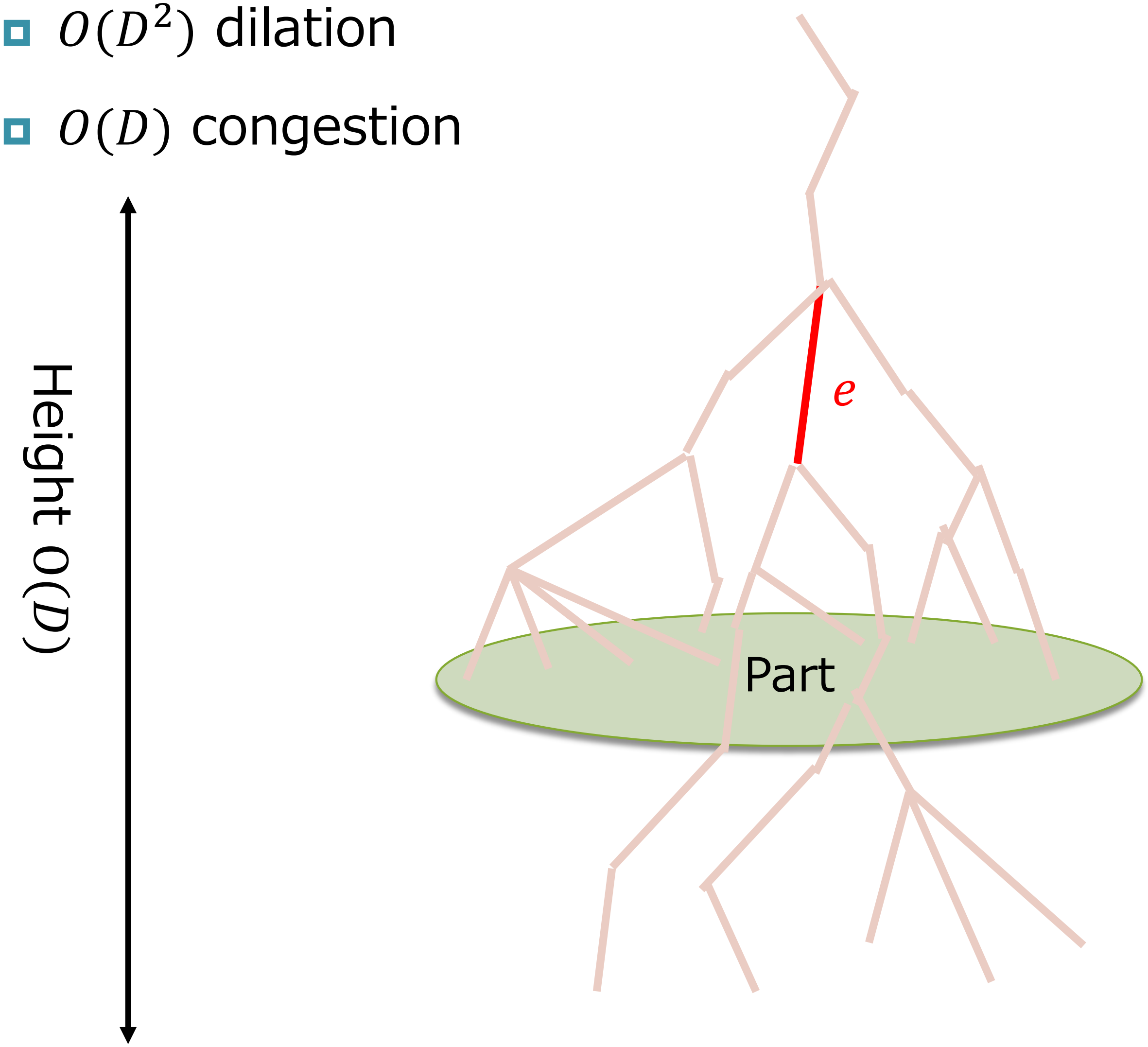
Proving the Quality

- Taking a BFS tree, this construction achieves
  - $O(D^2)$ dilation
  - $O(D)$ congestion



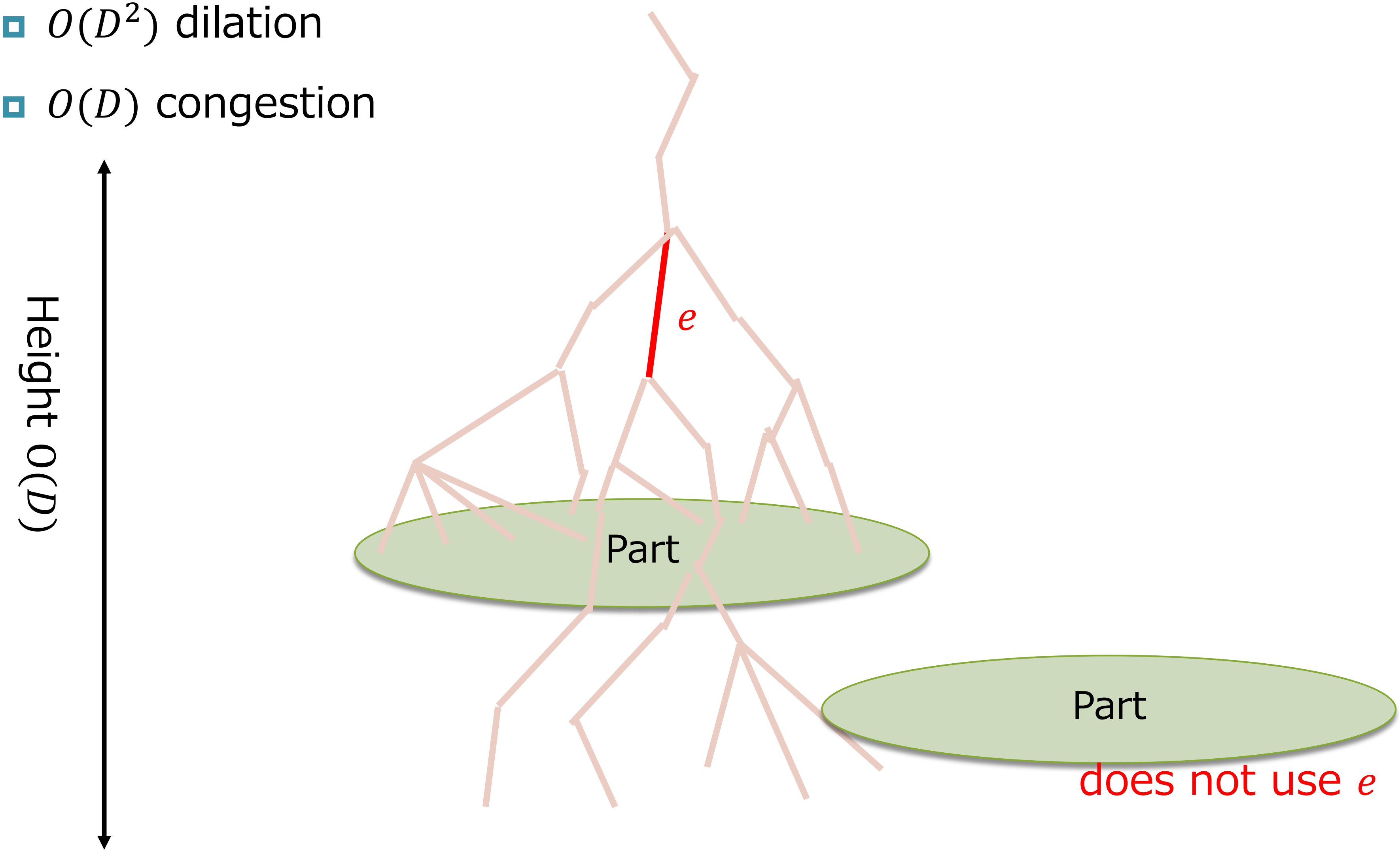
Proving the Quality

- Taking a BFS tree, this construction achieves
  - $O(D^2)$ dilation
  - $O(D)$ congestion



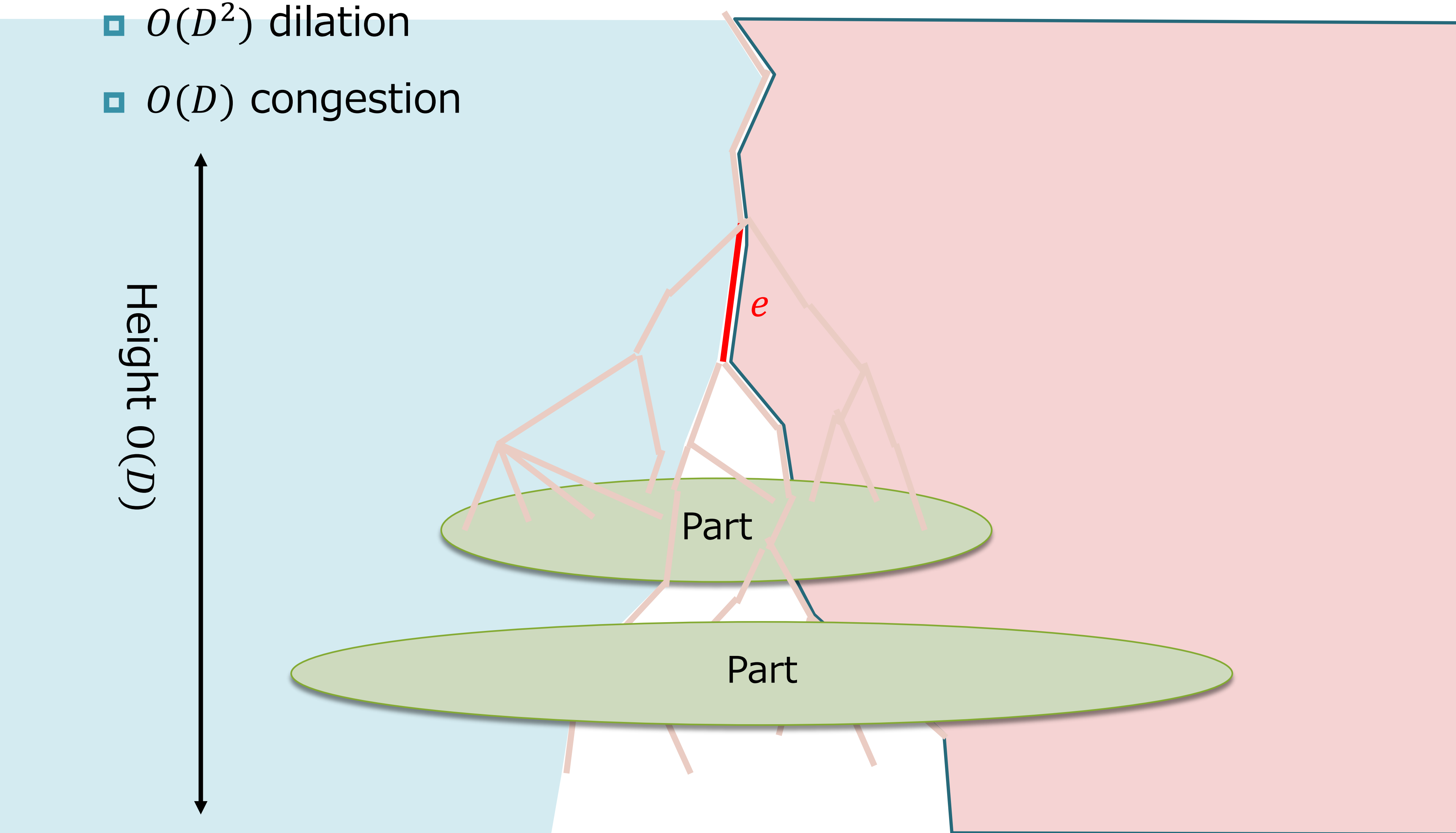
Proving the Quality

- Taking a BFS tree, this construction achieves
  - $O(D^2)$ dilation
  - $O(D)$ congestion



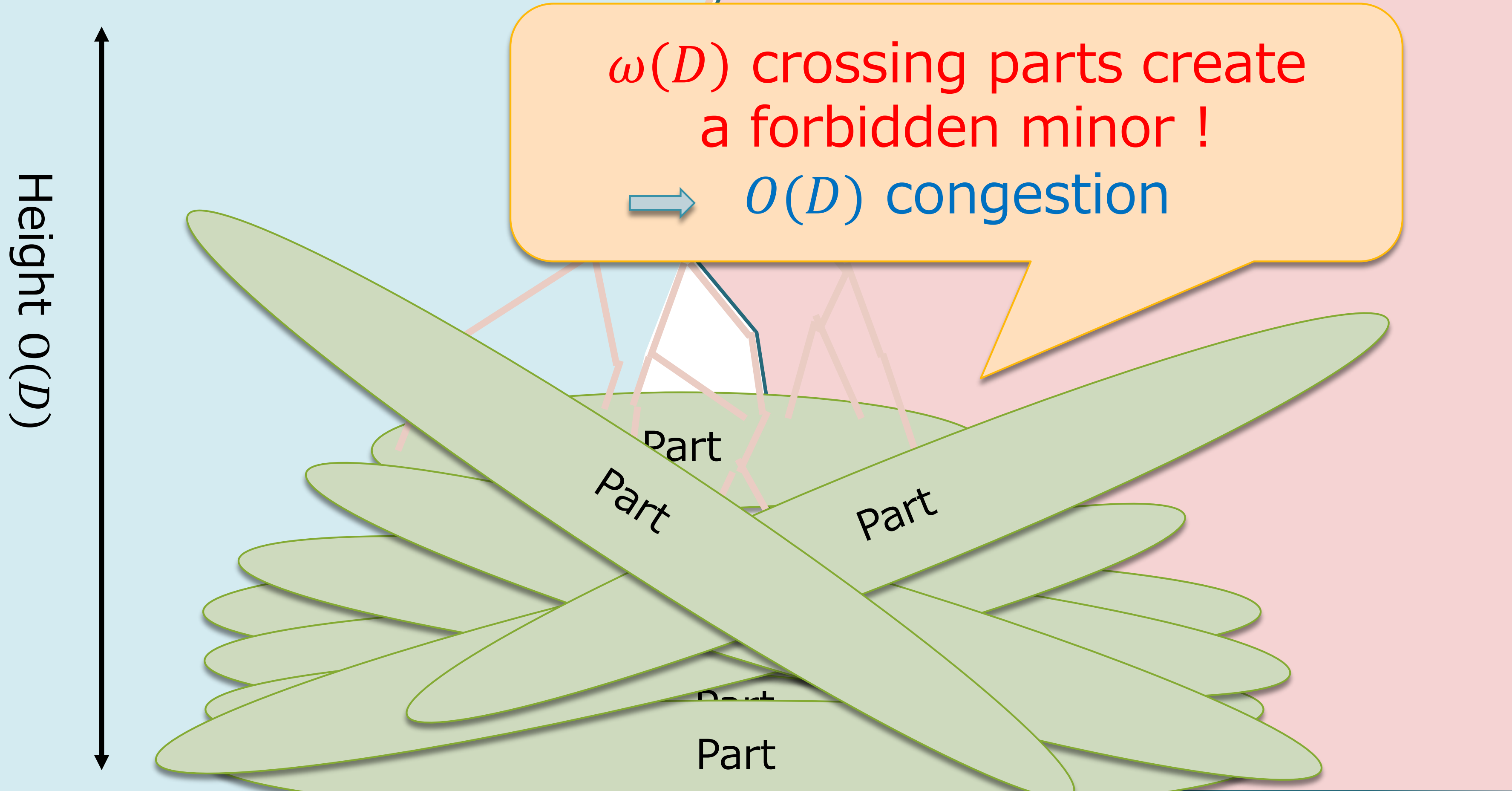
Proving the Quality

- Taking a BFS tree, this construction achieves
  - $O(D^2)$ dilation
  - $O(D)$ congestion



Proving the Quality

- Taking a BFS tree, this construction achieves
  - $O(D^2)$ dilation
  - $O(D)$ congestion



Distributed Construction

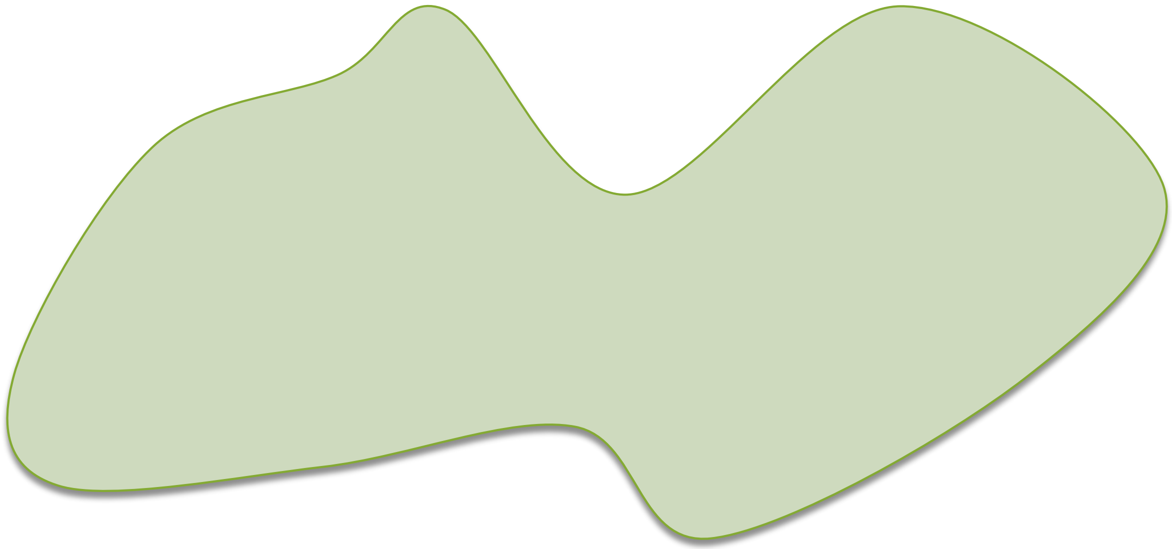
- The construction requires a planar embedding
  - It is possible (in distributed mannar)
    [Ghaffari and Haeupler, PODC'16]

- There also exists an algorithm without embedding
  [Haeupler, I, Zuzic, '16]
  - A versatile algorithm (not only for planar graphs)
  - Find any spanning-tree based shortcuts (efficiently)

    → Only existential proofs suffice!

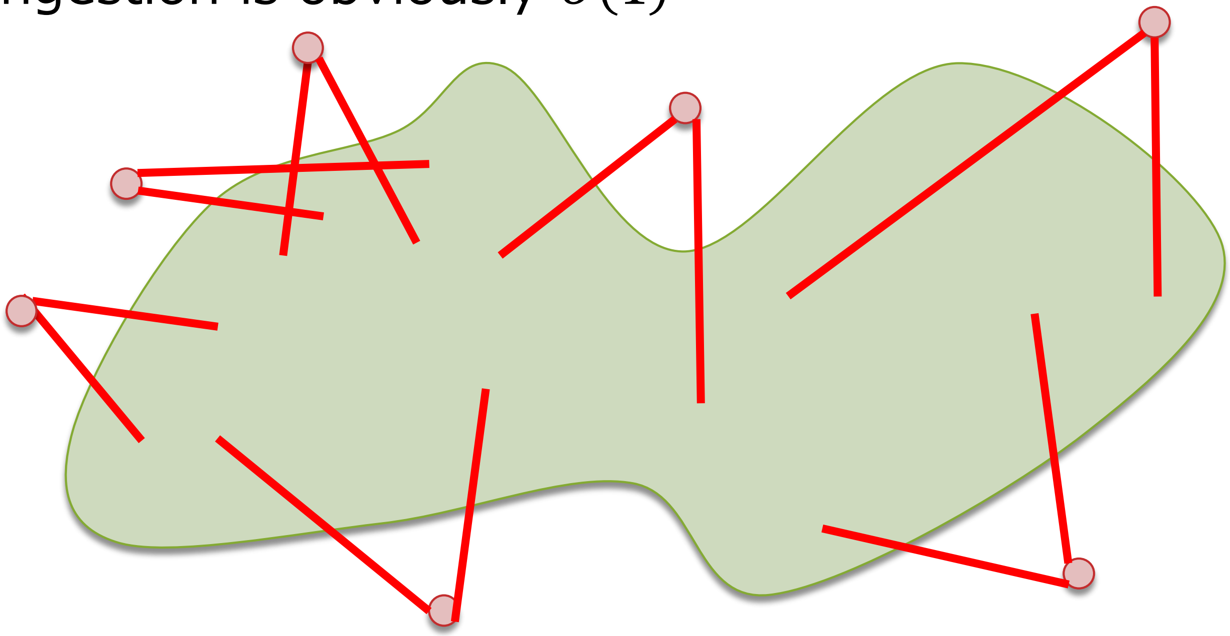
1-hop Extension Approach

- Take all the edges touching each part



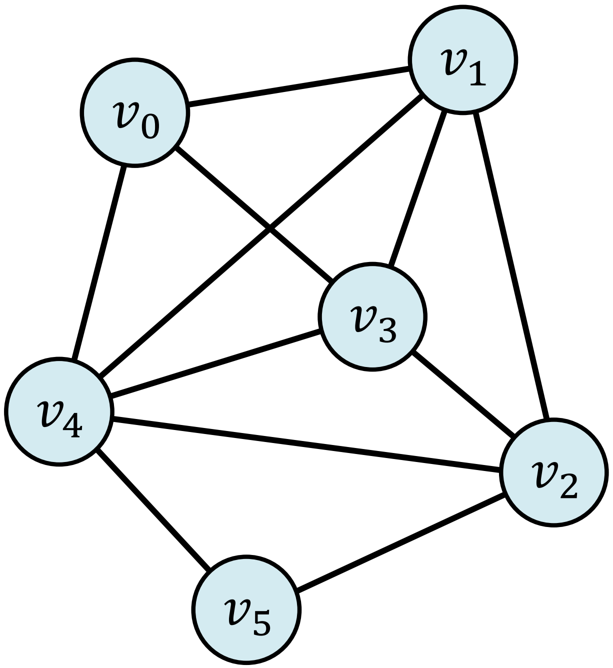
1-hop Extension Approach

- Take all the edges touching each part
  - Congestion is obviously $O(1)$



Application : k-chordal graphs

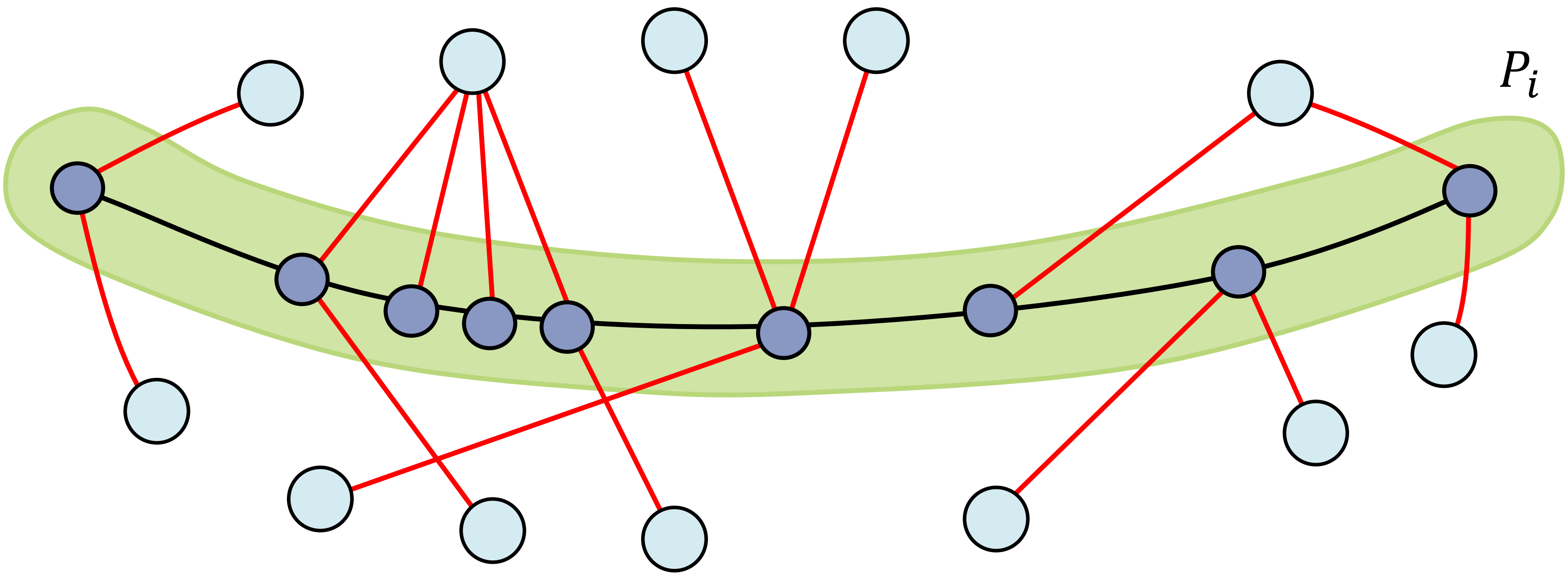
- k-chordal graphs = any induced cycle has length at most k



3-chordal (chordal) 4-chordal 5-chordal

1-hop extension for k-chordal graphs

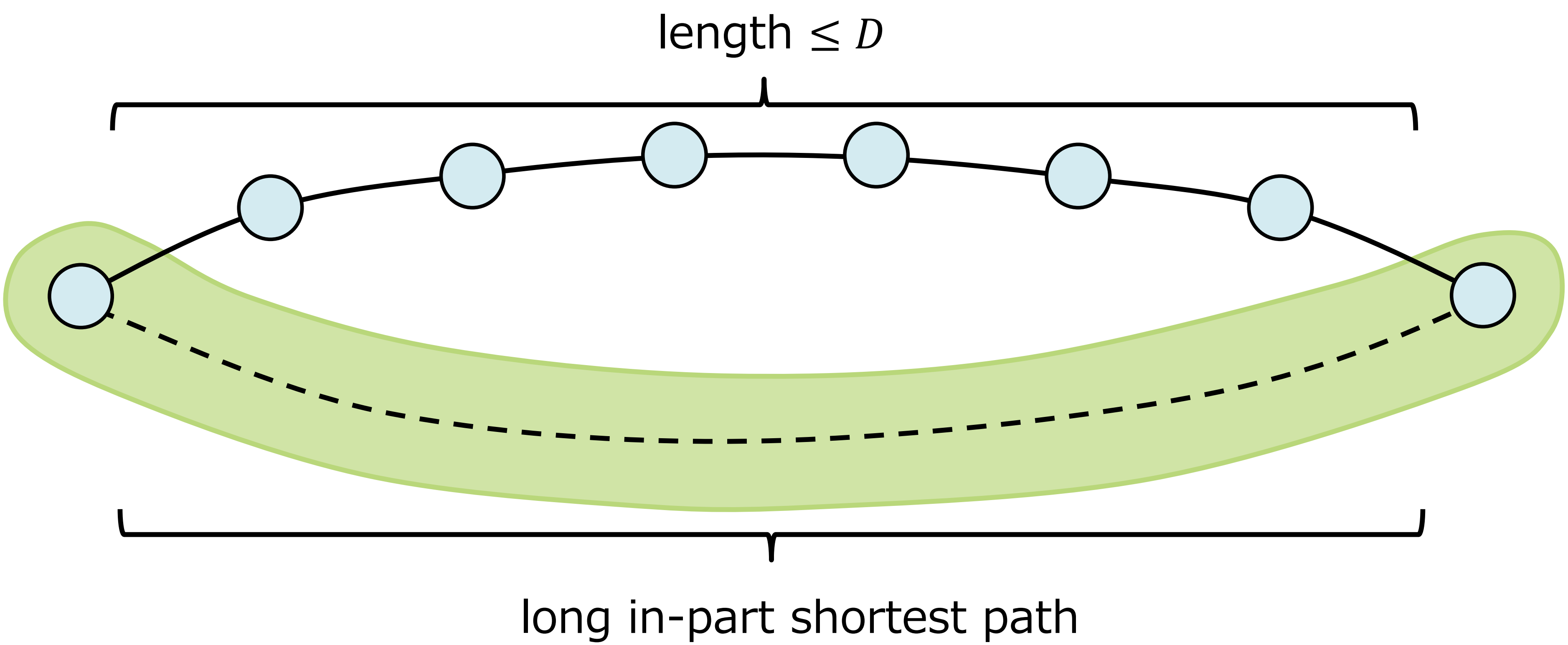
- 1-hop extension shrinks the diameter of any subgraph of k-chordal graphs!



1-hop extension for k-chordal graphs

- Take two nodes far apart in the part
  - Shortest path in the part is long
- They have a (shortest) path ≤ diameter D

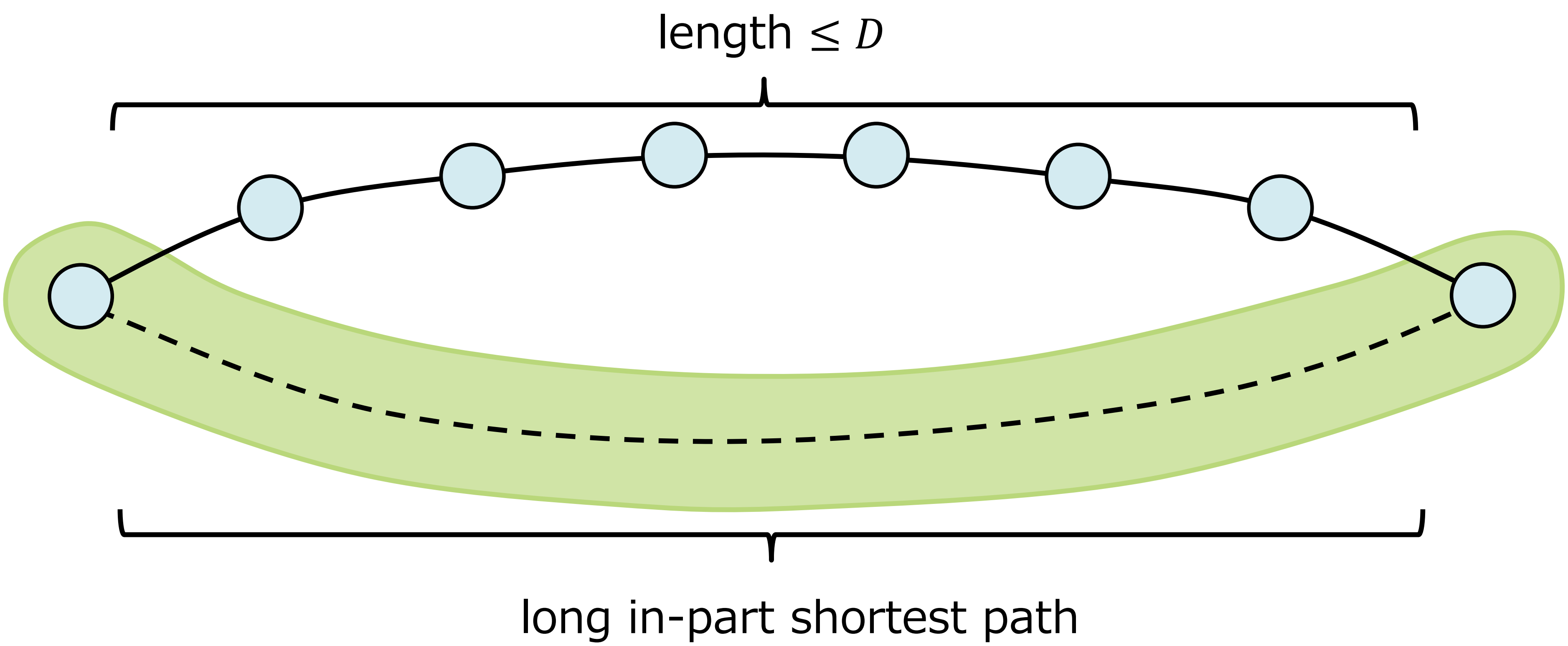
Assume their disjointness for simplicity



1-hop extension for k-chordal graphs

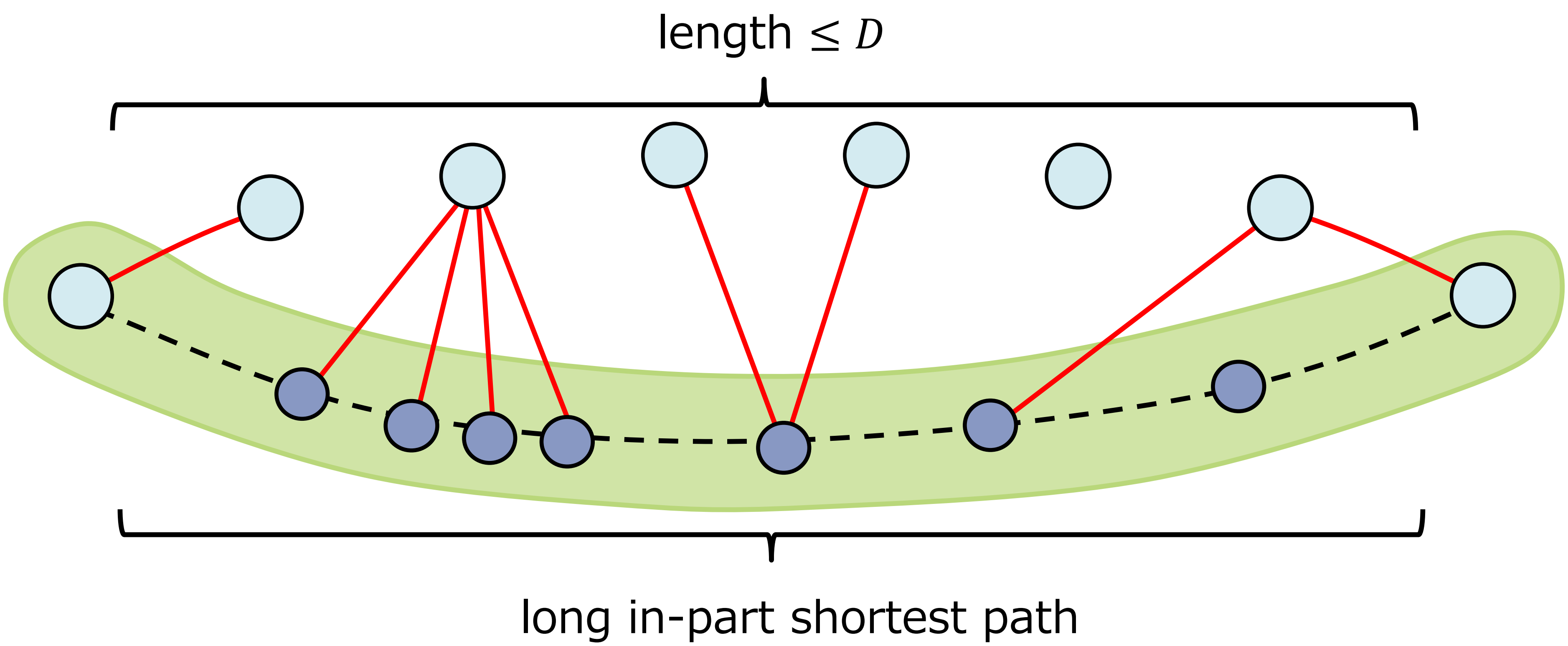
- Take two nodes far apart in the part
  - Shortest path in the part is long
- They have a (shortest) path ≤ diameter D

Assume their disjointness for simplicity



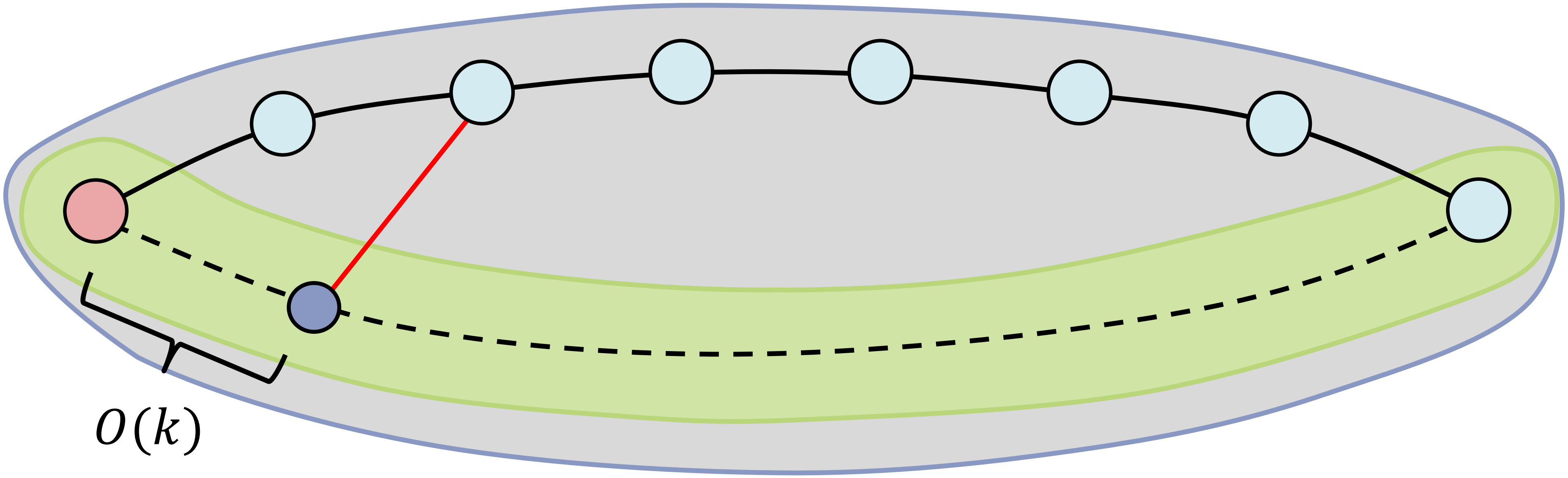
1-hop extension for k-chordal graphs

- What happens taking 1-hop extension edges



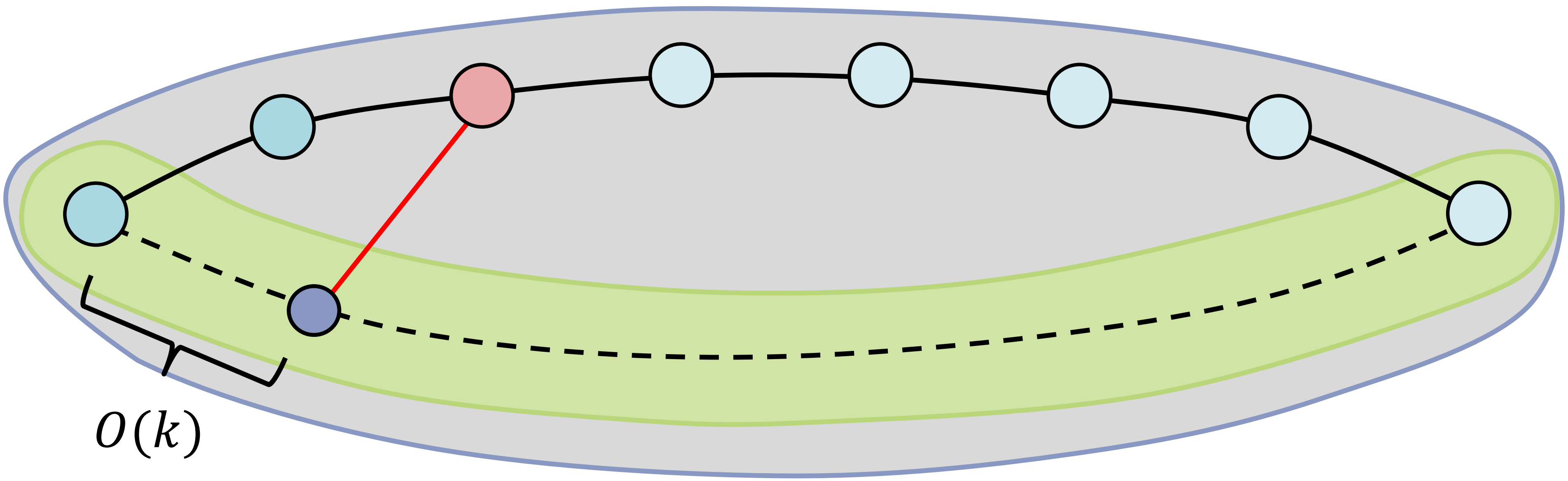
1-hop extension for k-chordal graphs

- Exploration from the left
  - Can find one shortcut edge within distance $O(k)$ because of k-chordality



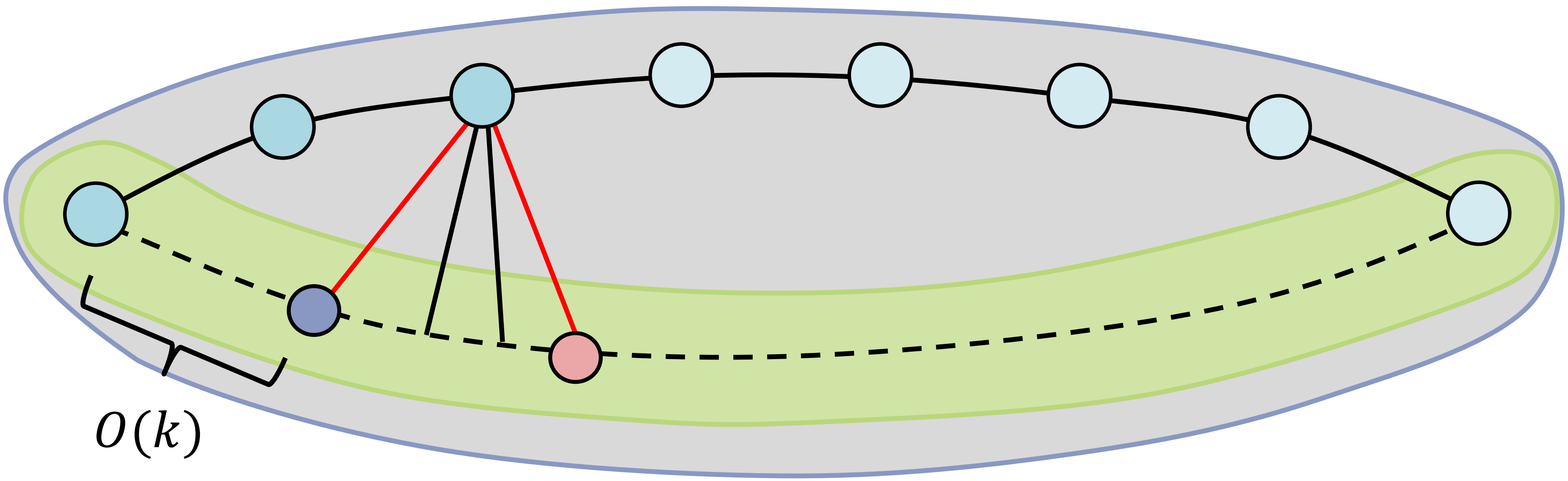
1-hop extension for k-chordal graphs

- Exploration from the left
  - Can find one shortcut edge within distance $O(k)$ because of k-chordality



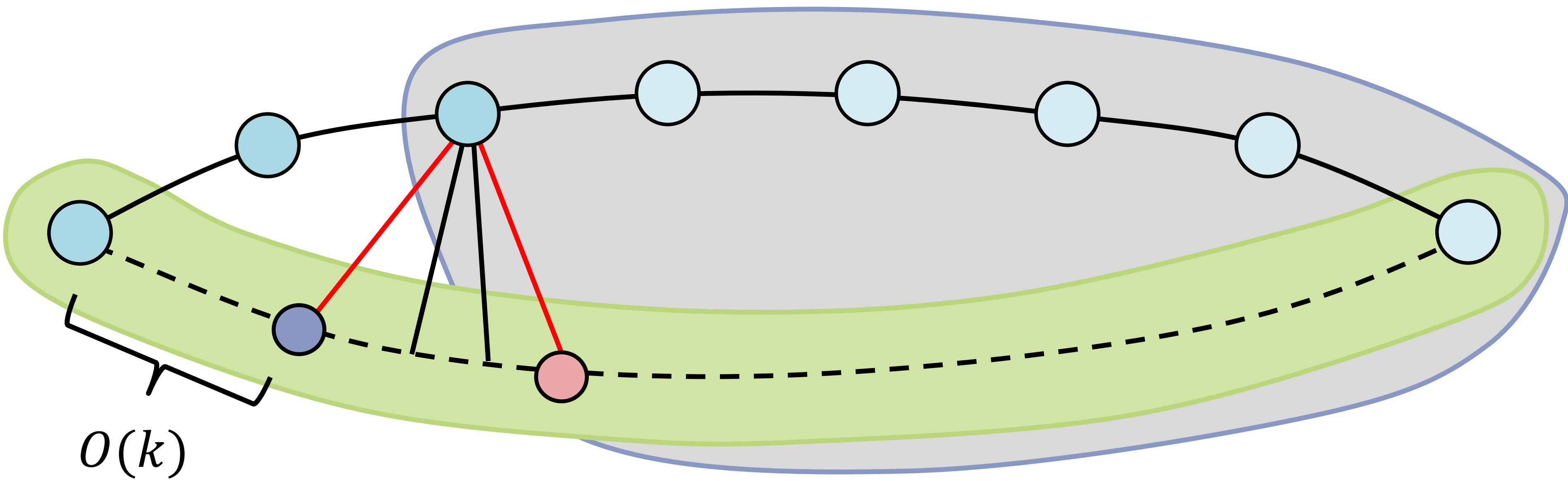
1-hop extension for k-chordal graphs

- Go back to the part (by taking the best edge)



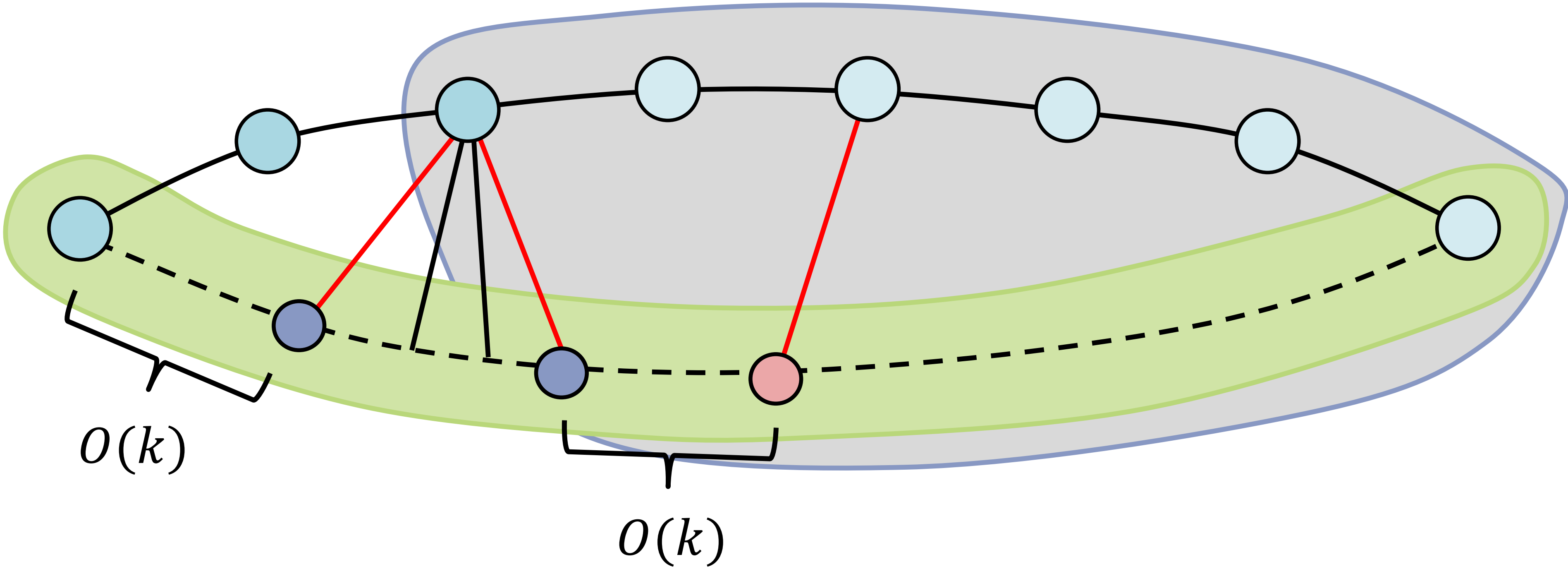
1-hop extension for k-chordal graphs

- Do the same thing for the remaining cycle



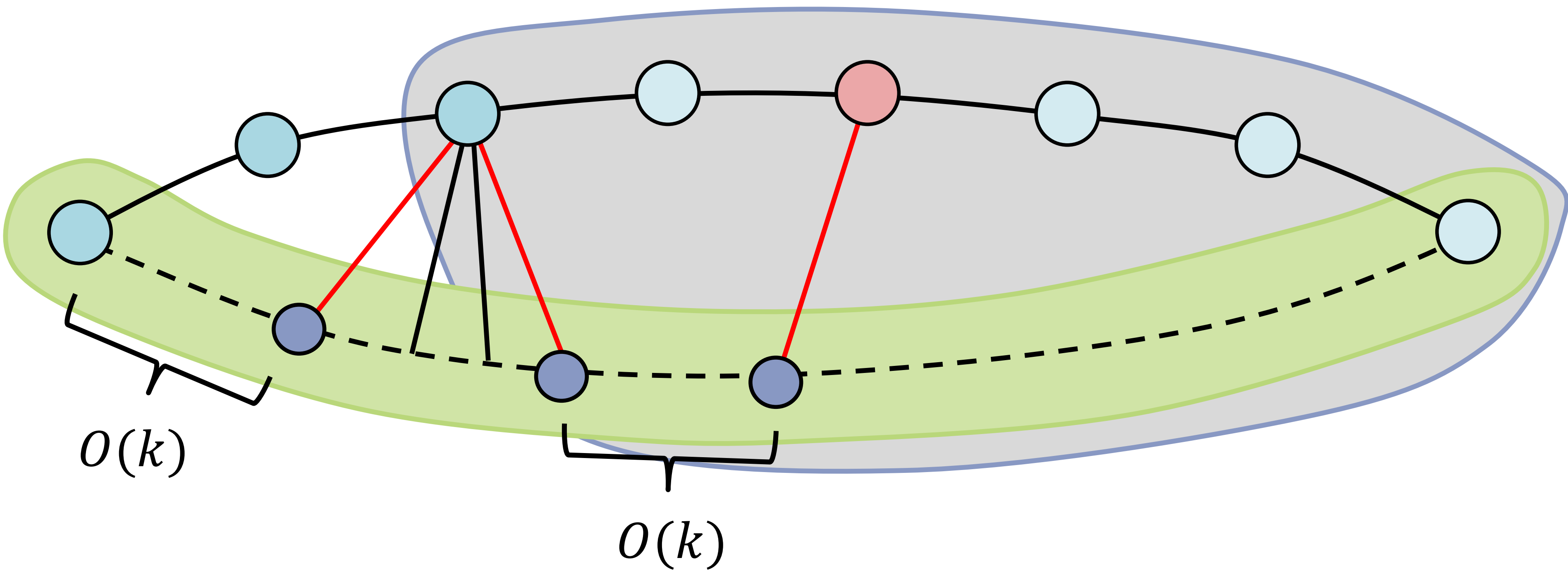
1-hop extension for k-chordal graphs

- Do the same thing for the remaining cycle



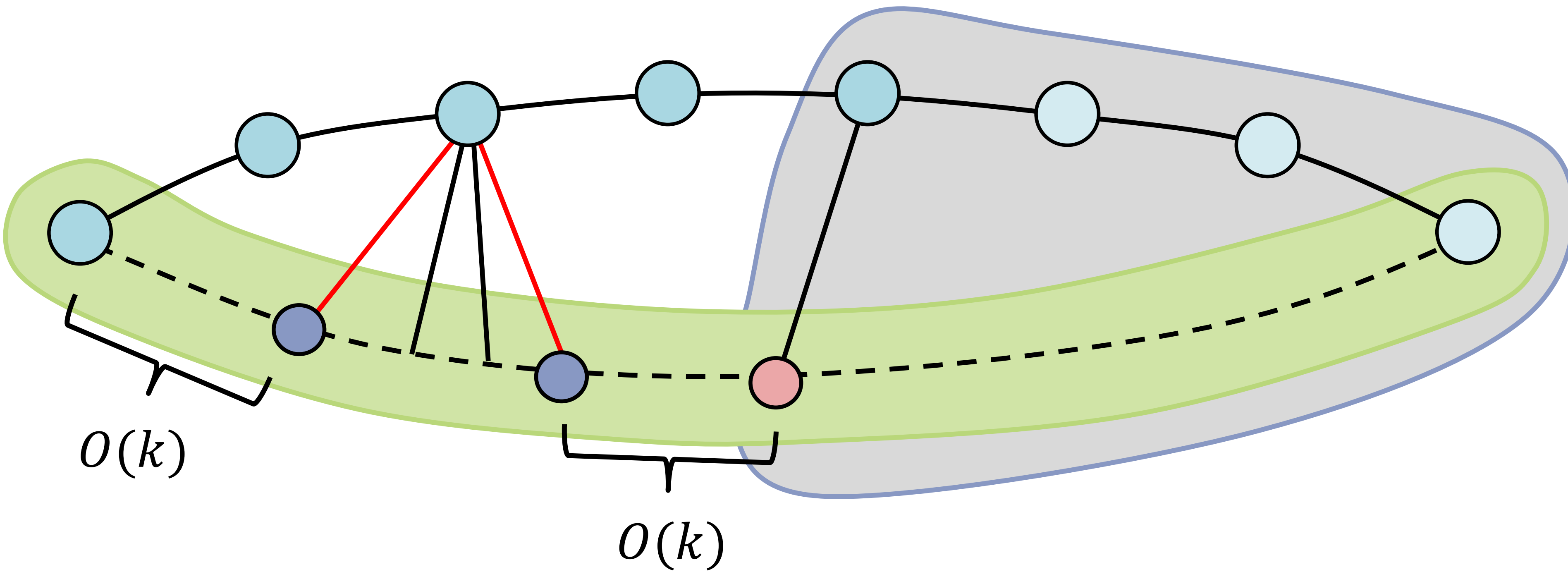
1-hop extension for k-chordal graphs

- Do the same thing for the remaining cycle



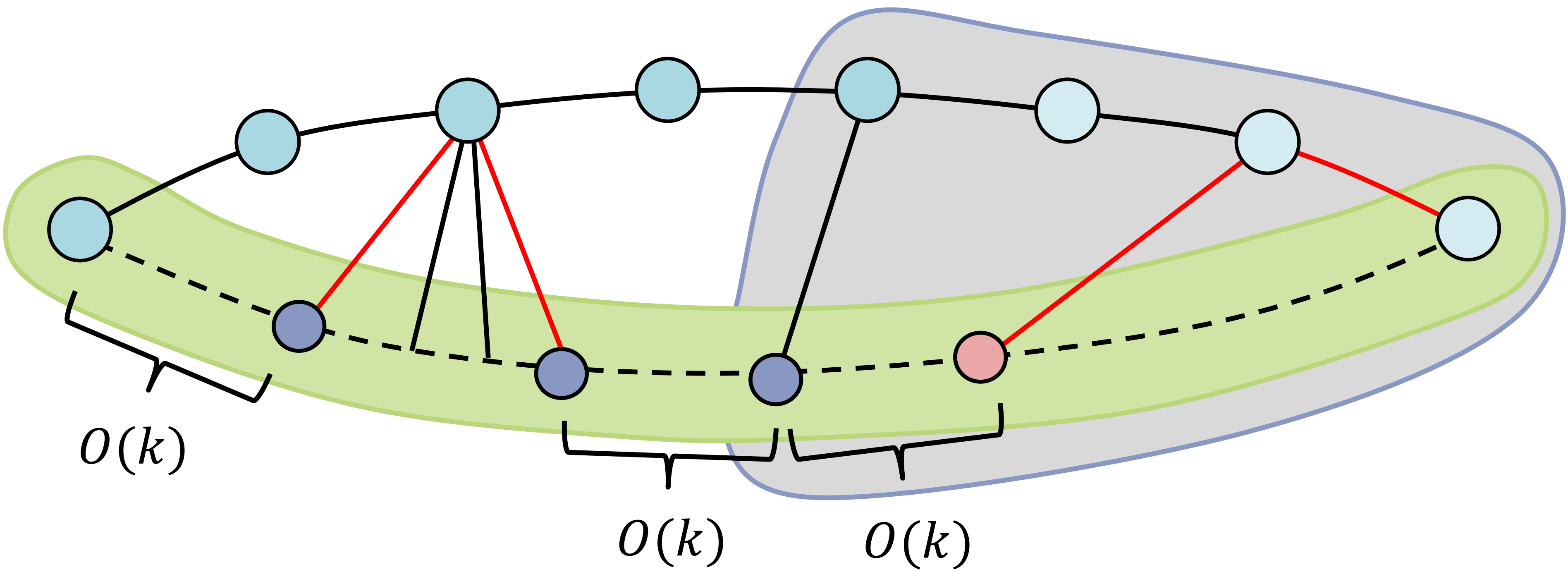
1-hop extension for k-chordal graphs

- Do the same thing for the remaining cycle



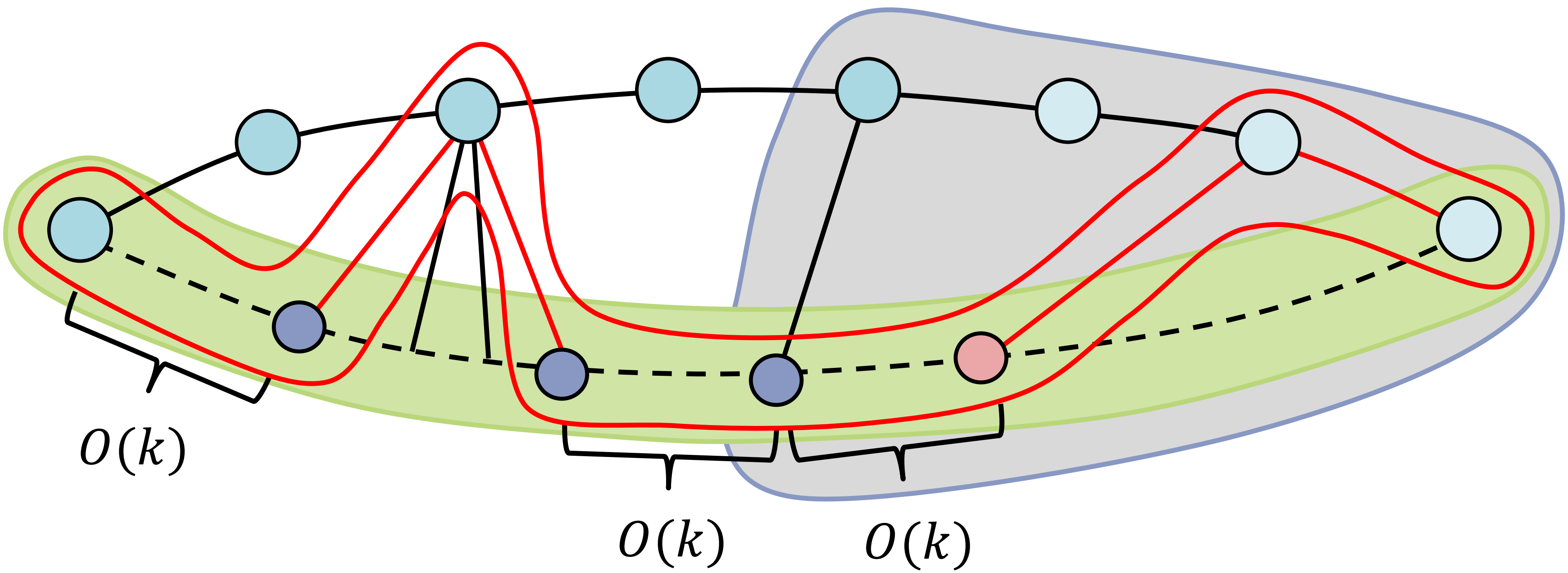
1-hop extension for k-chordal graphs

- Do the same thing for the remaining cycle



1-hop extension for k-chordal graphs

- The shortest path length using 1-hop extension edges is $O(kD)$



Open Problems

- On graph classes
  - Optimal shortcuts for minor-closed family (generalization of bounded genus/treewidth graphs)
  - Everywhere sparse graphs (further generalization ?)
  - Highly-connected graphs
- Versatile algorithms
  - Automatic transformer from existential results to constructability results

Open Problems

- How about other problems?

Theorem [GH16]]
$\tilde{O}(f)$-round PA $\rightarrow$ $\tilde{O}(f)$-round MST

Theorem[GH16]
$\tilde{O}(f)$-round PA $\rightarrow$ $\tilde{O}(f)$-round $(1+\epsilon)$-approx. min-cut

Theorem[HL18]
$\tilde{O}(f)$-round PA$\rightarrow$ For $\beta = (\log n)^{\Omega(1)}$,
$\tilde{O}(\beta f)$-round $O(n^{\frac{\log\log n}{\log \beta}})$ -approx. SSSP

- Known that it does not help the diameter or APSP