Low-congestion Shortcuts without Embedding

Bernhard Haeupler [1], Taisuke Izumi [2], **Goran Žužić** [1]

[1]Carnegie Mellon University, USA

[2]Nagoya Institute of Technology, Japan

February 20, 2017

Problem

- Solve MST in CONGEST model

Minimum Spanning Tree (MST)

Given graph G with weights on edges, compute a spanning tree with minimum sum of weights of edges.

CONGEST model

Graph G with n nodes and diameter D. Computation in synchronized rounds. In each round all nodes send $O(\log n)$-bits to all their neighbors. In the end, every vertex outputs the MST weight.

- Lower bound $\tilde{\Omega}(D + \sqrt{n})$
- for MST, Min-Cut, Shortest Path, ... ☹

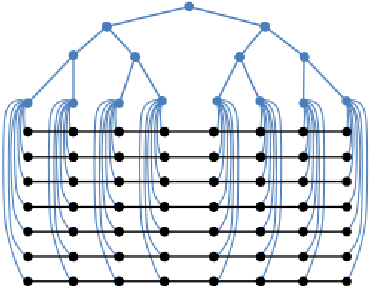


Figure : Lower bound graph, [Ghaffari and Haeupler; SODA'16]

- $\tilde{\Omega}(\cdot)$, $\tilde{O}(\cdot)$ supressed $\log^{O(1)} n$ factors

- In practice
  - Internet-like graphs
  - n is huge (as is $\sqrt{n}$)
  - D is logarithmic
  - **lots** of structure
  - Can we do better than $\tilde{O}(D + \sqrt{n})$?
- People care: Spanning Tree Protocol [Perlman 1985]

Our Contribution

- Can we do better than $\tilde{O}(D + \sqrt{n})$: YES (for some graphs)

Our Contribution

- Can we do better than $\tilde{O}(D + \sqrt{n})$: YES (for some graphs)
- Central topic: Tree-Restricted Shortcuts

Our Contribution

- Can we do better than $\tilde{O}(D + \sqrt{n})$: YES (for some graphs)
- Central topic: Tree-Restricted Shortcuts

	simpler	$\tilde{O}(D)$-round	planar graphs
	new	$\tilde{O}(gD)$-round	genus-g graphs
[DISC'16]	**new**	$\tilde{O}(\sqrt{g}D)$-round	genus-g graphs
[DISC'16]	**new**	$\tilde{O}(kD)$-round	treewidth-k graphs

- [SODA'16] has $\tilde{O}(D)$ planar algorithm - but it requires a planar embedding (hard!)

Solving Strategy

Graph G has good TR-shortcuts

$$\Downarrow$$

Construct universally optimal TR-shortcuts in G

$$\Downarrow$$

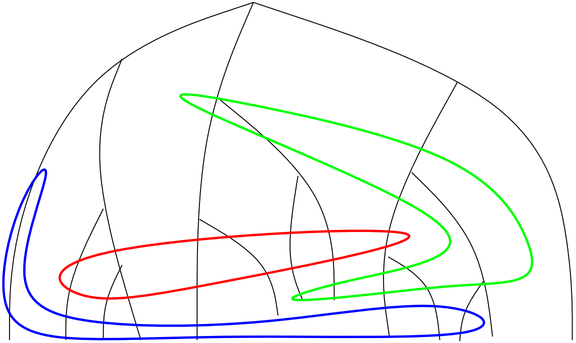
Construct fast distrib. algs for G

Overview

1. What are tree-restricted shortcuts?
2. How to use them? [in Boruvka]
3. Graphs with good TR-shortcuts
4. How to construct universally nearly optimal TR-shortcuts?

What are Tree-Restricted shortcuts?

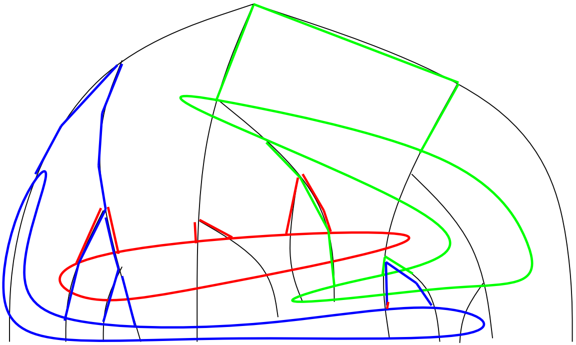
- Fix any **connected** vertex partition
- Fix any (spanning) BFS tree T
- add edges of T to parts in order to reduce its parameter



What are Tree-Restricted shortcuts?

congestion

all edges used in $\leq c$ shortcuts



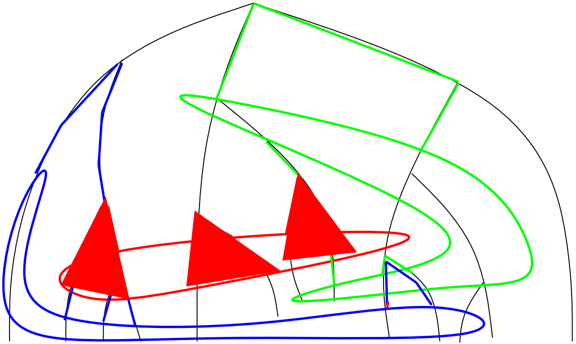
What are Tree-Restricted shortcuts?

congestion

all edges used in $\leq c$ shortcuts

block number

all parts have $\leq b$ blocks



How to use TR-shortcuts?

- MST using Boruvka

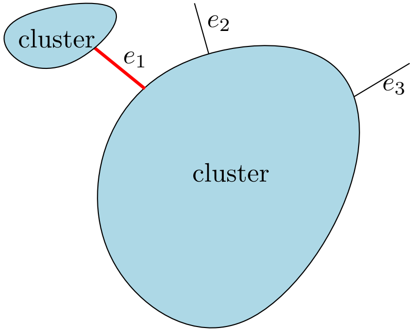


Figure : Main step - find minimum outgoing edge in each part of partition

How to use TR-shortcuts?

- MST using Boruvka

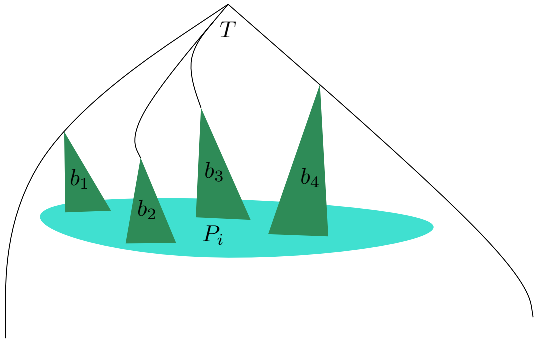


Figure : Spreading information within part in $O(bD)$

- for all parts together in $O(b(D + c))$

Graphs with good TR-shortcuts

Family	Congestion c	Block parameter b	$O(b(D+c))$
Planar graphs	$\tilde{O}(D)$	$\tilde{O}(1)$	$\tilde{O}(D)$
Genus-g graphs	$\tilde{O}(gD)$	$\tilde{O}(1)$	$\tilde{O}(gD)$
Treewidth-k graphs	$\tilde{O}(k)$	$\tilde{O}(k)$	$\tilde{O}(kD)$

How to Construct Universally Optimal TR-Shortcuts?

Theorem

Given a tree T spanning a graph G such that there exists a ***block**-b* ***congestion**-c *TR-shortcut*

$\Longrightarrow$

we can construct a ***block**-$3b$* ***congestion**-$O(c \log n)$ *TR shortcut.*

Running time: $\tilde{O}(b(D + c))$-rounds (with high probability).

- tl;dr If a graph has good TR-shortcuts, we can find them efficiently.

How to Construct Universally Optimal TR-Shortcuts?

Algorithm

1. Each part tries to take all the T-edges above it
2. If edge is used by $> 2c$ times, delete it
3. In the end, constant fraction of parts with have good shortcuts, so repeat $O(\log n)$ times

- A bit more details:
  - First, D-level edges are taken, then $D - 1$-level, ...
  - Use part-wise random sampling for efficiency

Final Note

- Also works for Min-Cut [SODA'16]
- In "practice" (not knowing the exact topology)
  - exponential search for $\max(bD, c)$
  - try to construct TR-shortcut
  - if successful, use it
  - conjectured to be good in practice