Lower Bounds for Local Monotonicity Reconstruction from Transitive Closure Spanners

Madhav Jha

Penn State

Joint work with

Arnab Bhattacharyya MIT
Elena Grigorescu MIT
Kyomin Jung KAIST
Sofya Raskhodnikova Penn State
David Woodruff IBM Almaden

What is this paper about?

1. A new connection between

efficiency of a randomized algorithm : monotonicity filter on a graph

and

a combinatorial property of the graph : |2-transitive-closure-spanner|

What is this paper about?

1. A new connection between
   efficiency of a randomized algorithm : monotonicity filter on a graph

   and

   a combinatorial property of the graph : |2-transitive-closure-spanner|

2. Bounds on the |2-transitive-closure-spanner| of hypergrid graphs.

What is this paper about?

1. A new connection between
   efficiency of a randomized algorithm : monotonicity filter on a graph

   and

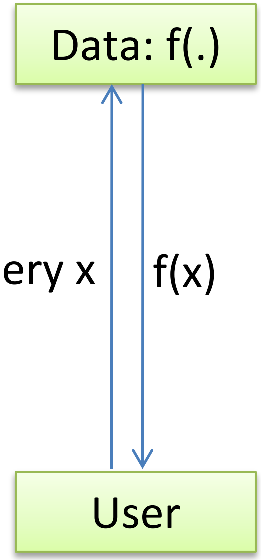
   a combinatorial property of the graph : |2-transitive-closure-spanner|

2. Bounds on the |2-transitive-closure-spanner| of hypergrid graphs.

3. (1) and (2) $\Rightarrow$ Lower bounds on monotonicity filter

Local Property Reconstruction [Saks Seshadhri 10]

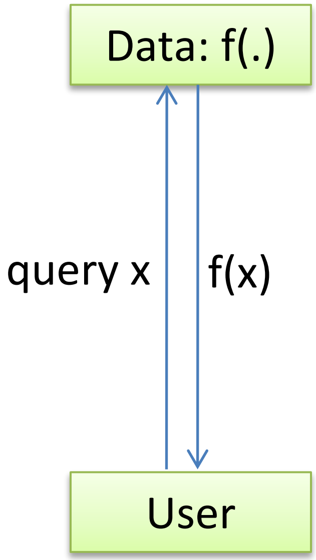
Extends [Ailon Chazelle Comandur Liu 08]



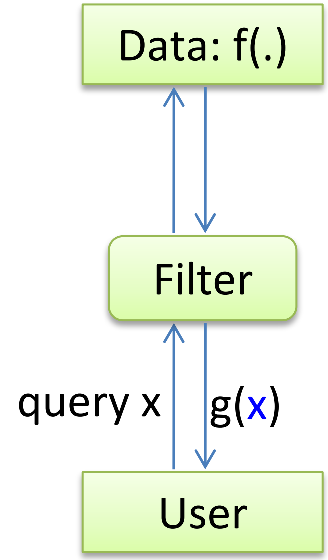
User expects f to satisfy property P

Local Property Reconstruction [Saks Seshadhri 10]

Extends [Ailon Chazelle Comandur Liu 08]



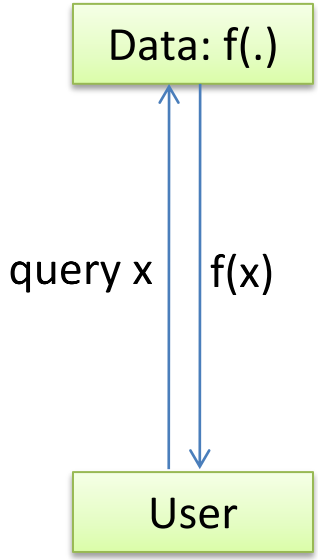
User expects f to satisfy property P



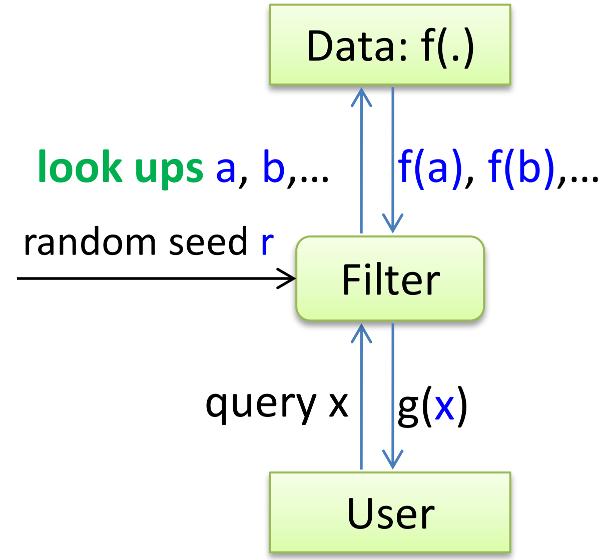
Reconstruction of property P

Local Property Reconstruction [Saks Seshadhri 10]

Extends [Ailon Chazelle Comandur Liu 08]



User expects f to satisfy property P



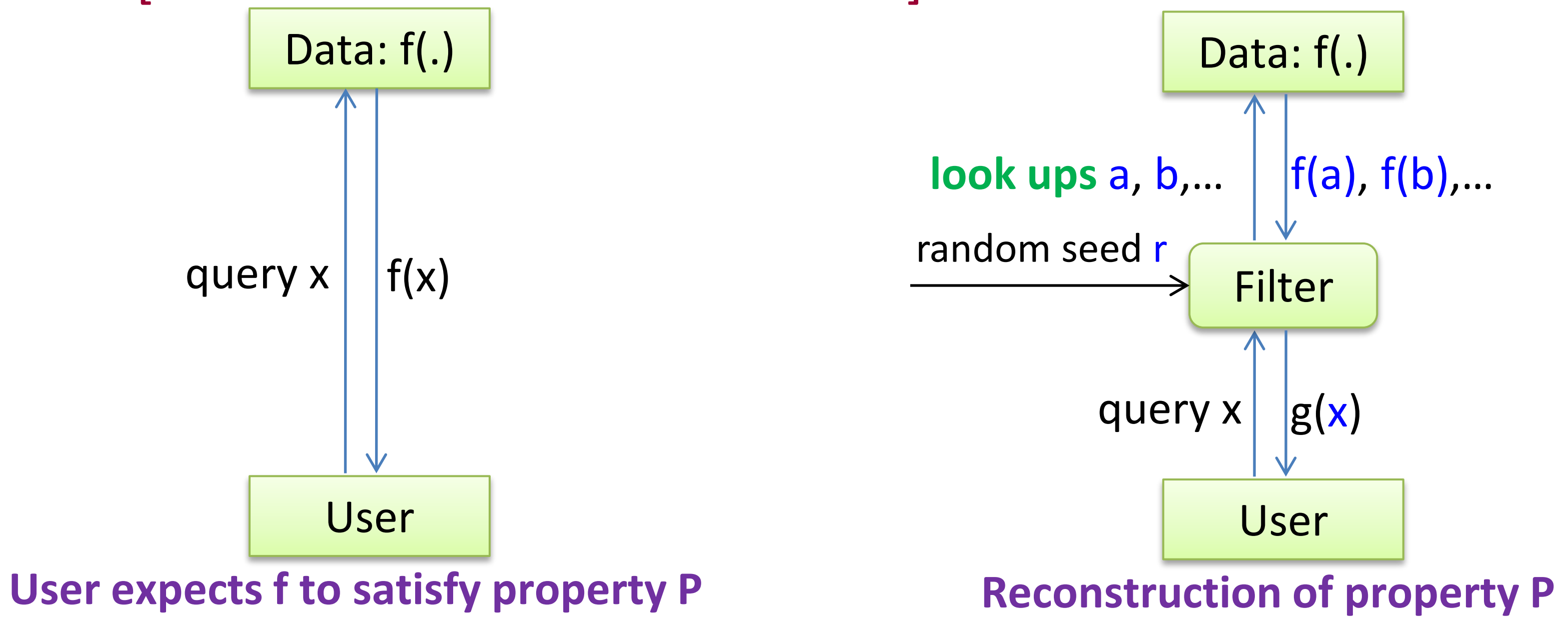
Reconstruction of property P

Filter can make look ups to data set

Filter can flip coins: Randomized

Local Property Reconstruction [Saks Seshadhri 10]

Extends [Ailon Chazelle Comandur Liu 08]



Filter can make look ups to data set

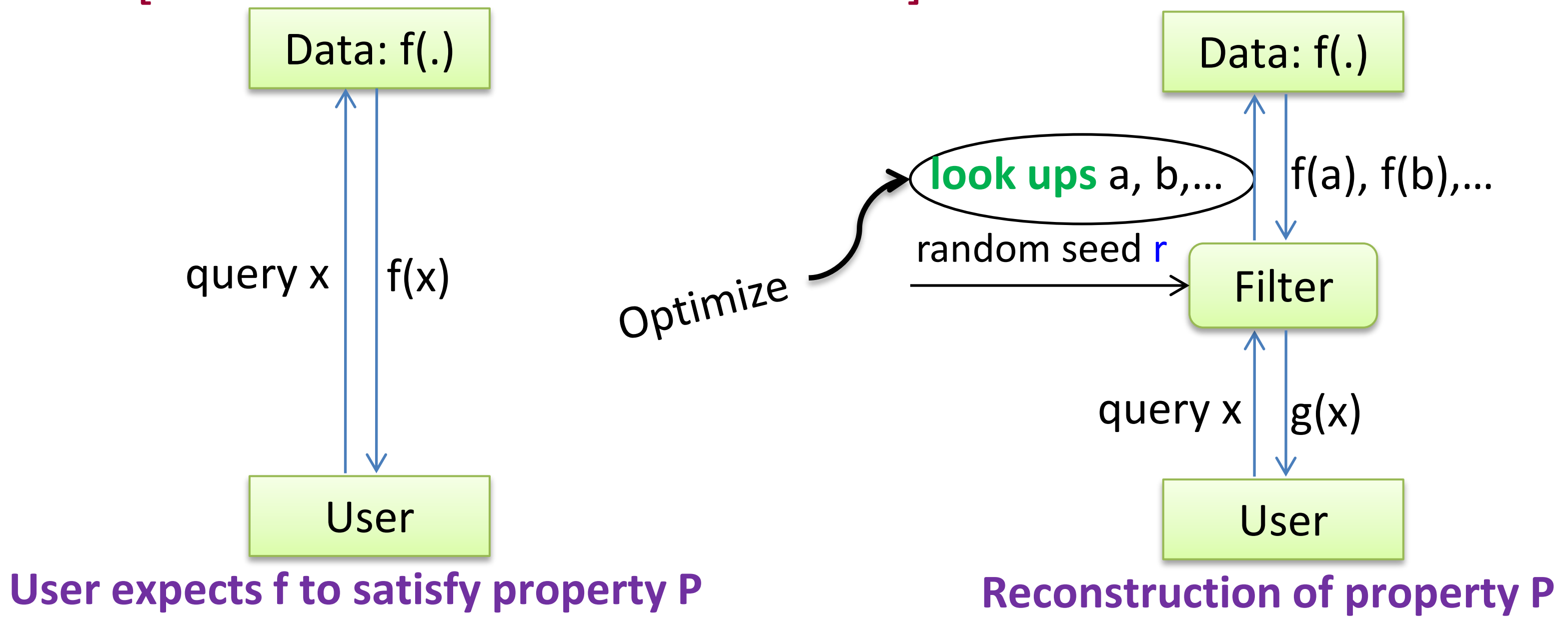
Filter can flip coins: Randomized

- For each f and r, function g satisfies the property P

  Notice: $g(\cdot)=\text{Filter}_{f,r}(\cdot)$ is deterministic
- w.h.p. g is close to f ($f(x)=g(x)$ for many x)

Local Property Reconstruction [Saks Seshadhri 10]

Extends [Ailon Chazelle Comandur Liu 08]



Filter can make look ups to data set

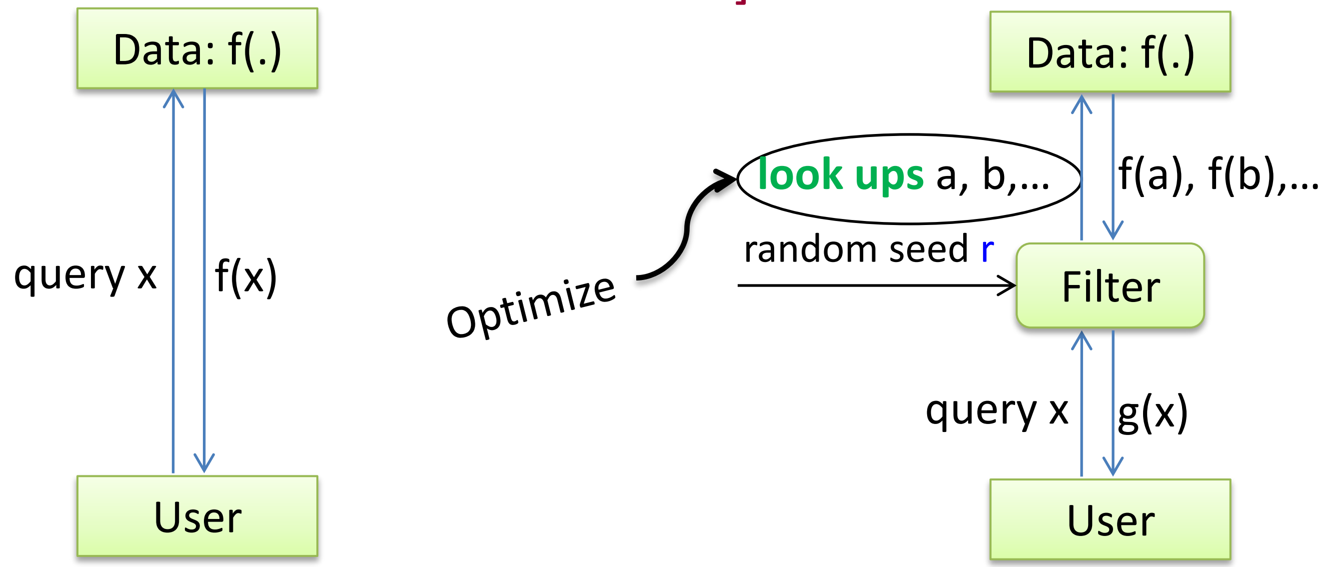
Filter can flip coins: Randomized

- For each f and r, function g satisfies the property P

  Notice: $g(\cdot)=\text{Filter}_{f,r}(\cdot)$ is deterministic
- w.h.p. g is close to f ($f(x)=g(x)$ for many x)

Local Monotonicity Reconstruction [Saks Seshadhri 10]

Extends [Ailon Chazelle Comandur Liu 08]



User expects f to satisfy monotonicity **Reconstruction of monotonicity**

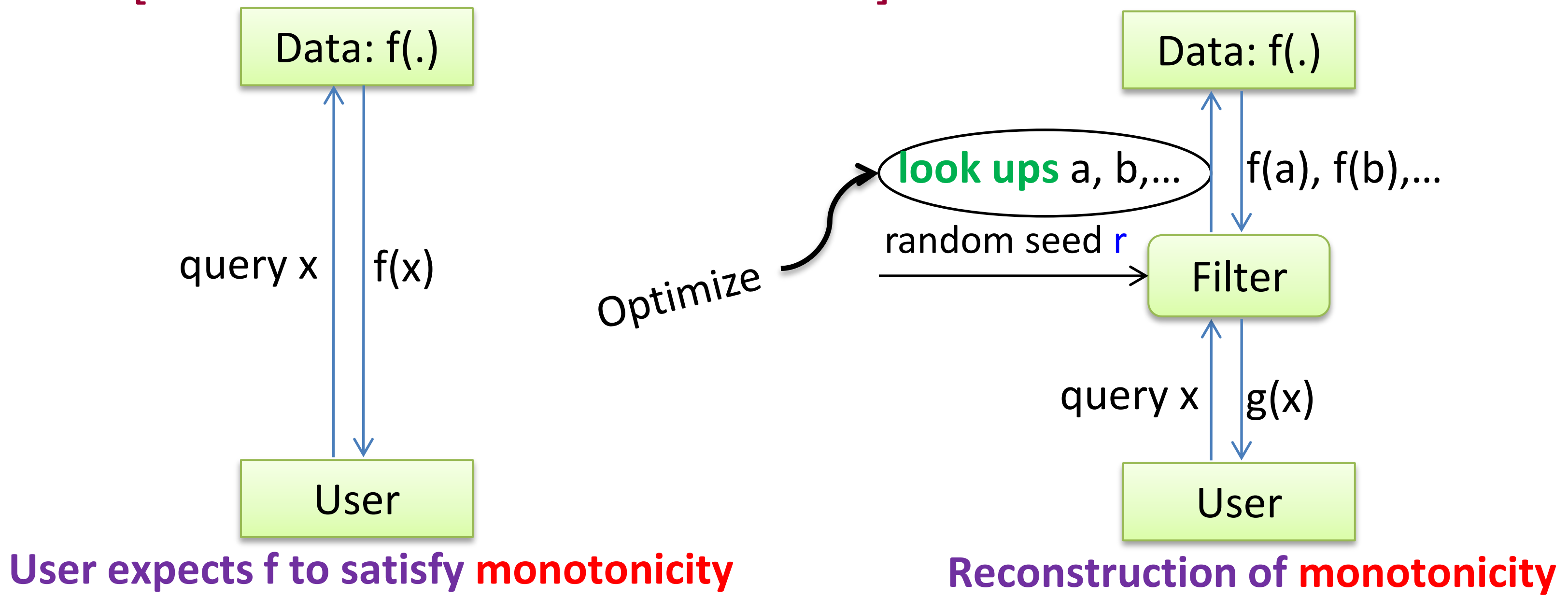
Filter can make look ups to data set

Filter can flip coins: Randomized

- For each f and r, function g is monotone
  Notice: $g(\cdot)=\text{Filter}_{f,r}(\cdot)$ is deterministic
- w.h.p. g is identical to f if f is monotone

Local Monotonicity Reconstruction [Saks Seshadhri 10]

Extends [Ailon Chazelle Comandur Liu 08]



- Local: Output g does not depend on order of queries
- Non-adaptive: All look ups are given in advance
- Distance-respecting: f and g do not differ too much

Monotonicity

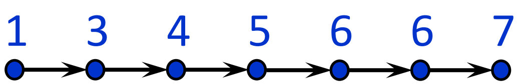
Graph = domain; nodes = points in the domain;

edges = “less than equal to”; node labels = function values

R: real numbers

$f : \{1,\ldots,m\} \to R$

Line



Monotonicity

Graph = domain; nodes = points in the domain;

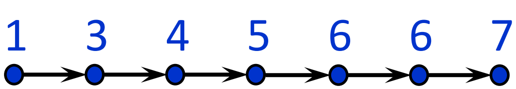
edges = "less than equal to"; node labels = function values

R: real numbers

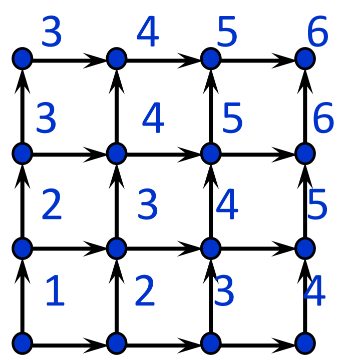
$f : \{1,\ldots,m\} \rightarrow R$

$f : \{1,\ldots,m\}^2 \rightarrow R$

Line



Grid



Monotonicity

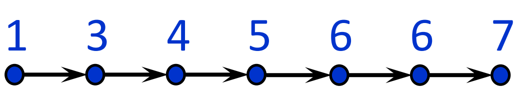
Graph = domain; nodes = points in the domain;

edges = "less than equal to"; node labels = function values

R: real numbers

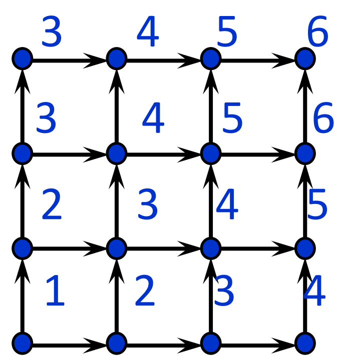
$f : \{1,\dots,m\} \to R$

Line



$f : \{1,\dots,m\}^d \to R$

Hypergrid



Monotonicity

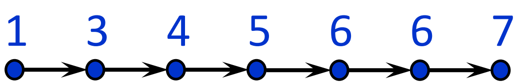
Graph = domain; nodes = points in the domain;

edges = "less than equal to"; node labels = function values

R: real numbers

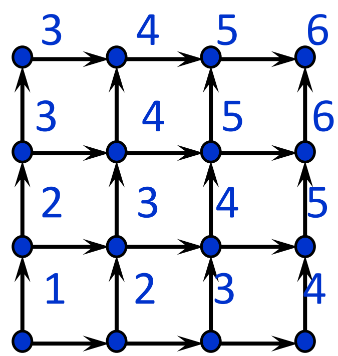
$f : \{1,\ldots,m\} \rightarrow R$

Line



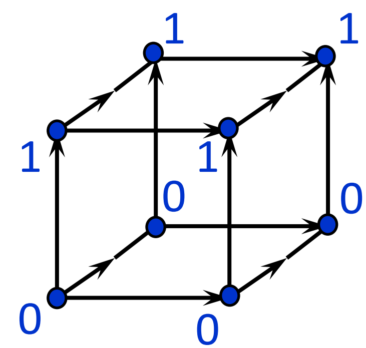
$f : \{1,\ldots,m\}^d \rightarrow R$

Hypergrid



$f : \{0,1\}^d \rightarrow R$

Hypercube



Monotonicity

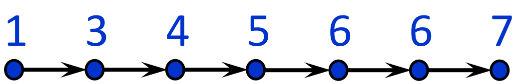
Graph = domain; nodes = points in the domain;

edges = "less than equal to"; node labels = function values

R: real numbers

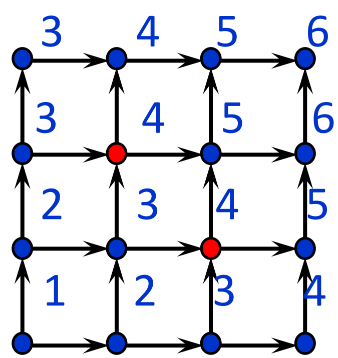
$f : \{1,\ldots,m\} \rightarrow R$

Line



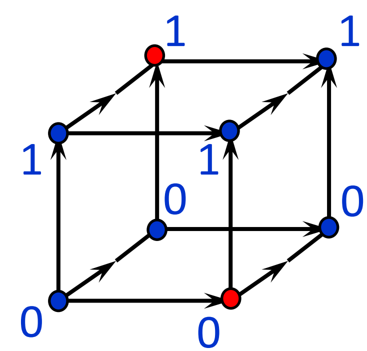
$f : \{1,\ldots,m\}^d \rightarrow R$

Hypergrid



$f : \{0,1\}^d \rightarrow R$

Hypercube



Not all nodes are comparable (have a path connecting them)

Example: red nodes are incomparable

Previous Work (Local Monotonicity Reconstruction)

Upper bounds

- Line (f : {1,…,m} → R)
  - O(log m) [Ailon Chazelle Comandur Liu 08, Saks Seshadhri 10]
- Hypergrid (f : $\{1,\ldots,m\}^d \to R$)
  - $(\log m)^{O(d)}$ [Saks Seshadhri 10]

Both filters are *non-adaptive* and *distance-respecting*

Lower bounds

- Hypercube
  - $\Omega(2^{\alpha d})$ [Saks Seshadhri 10]

    applies only to *distance-respecting* filters

Our Results (Part 1)

Lower Bounds for non-adaptive local monotonicity filter

- Hypergrid : $\Omega(\log^d m \,/\, (2d \log\log m)^{d-1})$
- Hypercube : $\Omega(2^{cd})$

  $c \approx 0.1620$.

1. Holds even for non-*distance-respecting* filters
2. Adaptive filters: later in the talk

Our Results (Part 1)

Lower Bounds for non-adaptive local monotonicity filter

- Hypergrid : $\Omega(\log^d m / (2d \log \log m)^{d-1})$
- Hypercube : $\Omega(2^{cd})$
  $c \approx 0.1620$.

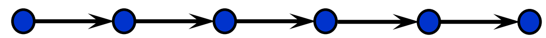
1. Holds even for non-*distance-respecting* filters
2. Adaptive filters: later in the talk

Main Tool:
Transitive-Closure Spanners
[Bhattacharyya Grigorescu Jung Raskhodnikova Woodruff 09]

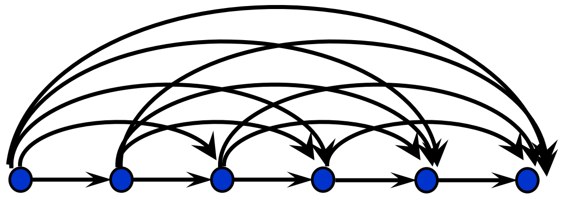
Transitive-Closure Spanners [BGJRW09]

Transitive closure TC(*G*) has an edge from *u* to *v* iff

G has a path from *u* to *v*



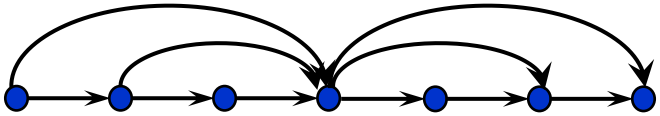
G



TC(*G*)

k-TC-spanner *H* of *G* has $distance_H(u,v) \leq k$ iff

G has a path from *u* to *v*

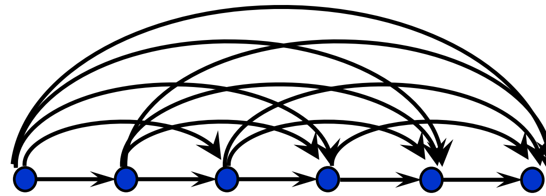


Transitive-Closure Spanners [BGJRW09]

Transitive closure TC(G) has an edge from u to v iff

G has a path from u to v

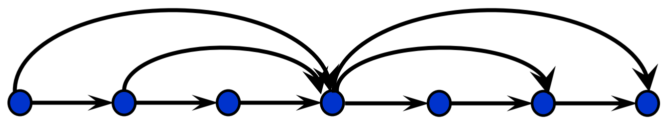
G



TC(G)

k-TC-spanner H of G has $distance_H(u,v) \le k$ iff

G has a path from u to v



- Common abstraction for several applications [BGJRW09]
  - property testing
  - property reconstruction
  - access control
  - data structures

What is a sparsest k-TC-spanner for a given graph family?

Previous Work (TC-Spanners for specific graph families)

- Planar graphs

[Thorup 95]

- Directed trees

[Bodlaender Tel Santoro 94, Thorup 97, Yao 82, Alon Schieber 87, Chazelle 87, Dodis Goldreich Lehman Raskhodnikova Ron Samorodnitsky 99, Attalah Frikken Blanton 05, Bhattacharyya Grigorescu Jung Raskhodnikova Woodruff 09]

- Path-separable graphs

[Bhattacharyya Grigorescu Jung Raskhodnikova Woodruff 09]

- Low-dimensional posets

  [Bhattacharyya Grigorescu Raskhodnikova Woodruff, Santis Ferrara Masucci 07, Attalah Blanton Fazio Frikken 09]

Our Results (Part 2)

- Local monotonicity reconstruction and 2-TC-Spanners

- Structural results on TC-spanners

Our Results (Part 2)

- Local monotonicity reconstruction and 2-TC-Spanners

(Filter-to-TC-Spanner-Reduction)

Non-adaptive local monotonicity filter on G with s(n) look up

$$\Downarrow$$

G has a 2-TC-spanner with $n \cdot s(n)$ edges

- Structural results on TC-spanners

Our Results (Part 2)

- Local monotonicity reconstruction and 2-TC-Spanners

(Filter-to-TC-Spanner-Reduction)

Non-adaptive local monotonicity filter on G with s(n) look up

$$\Downarrow$$

G has a 2-TC-spanner with $n \cdot s(n)$ edges

- Structural results on TC-spanners

(Hypergrid)

$$\frac{m^d \log^d m}{(2d \log \log m)^{d-1}} \leq |\text{sparsest 2-TC-Spanner of grid } [m]^d| \leq m^d \log^d m$$

Our Results (Part 2)

- Local monotonicity reconstruction and 2-TC-Spanners

(Filter-to-TC-Spanner-Reduction)

Non-adaptive local monotonicity filter on G with s(n) look up

$$\Downarrow$$

G has a 2-TC-spanner with $n \cdot s(n)$ edges

- Structural results on TC-spanners

(Hypergrid)

$$\frac{m^d \log^d m}{(2d \log \log m)^{d-1}} \leq |\text{sparsest 2-TC-Spanner of grid } [m]^d| \leq m^d \log^d m$$

(Hypercube)

$$2^{c\,d} \leq |\text{sparsest 2-TC-Spanner of grid } \{0,1\}^d| \leq 2^{c\,d}\,\mathrm{poly}(d)$$

$c \approx 1.1620$

Our Results (Part 2)

- Local monotonicity reconstruction and 2-TC-Spanners

(Filter-to-TC-Spanner-Reduction)

Non-adaptive local monotonicity filter on G with $s(n)$ look up

$$\Downarrow$$

G has a 2-TC-spanner with $n \cdot s(n)$ edges

Construction of 2-TC-Spanner using filter

Intuition: Consider a deterministic non-adaptive filter

Filter cannot modify a monotone function $\Rightarrow$ must find a violation to modify a point

Construction of 2-TC-Spanner using filter

Intuition: Consider a deterministic non-adaptive filter

Filter cannot modify a monotone function $\Rightarrow$ must find a violation to modify a point

Construction: Connect each x to all comparable points looked up on x

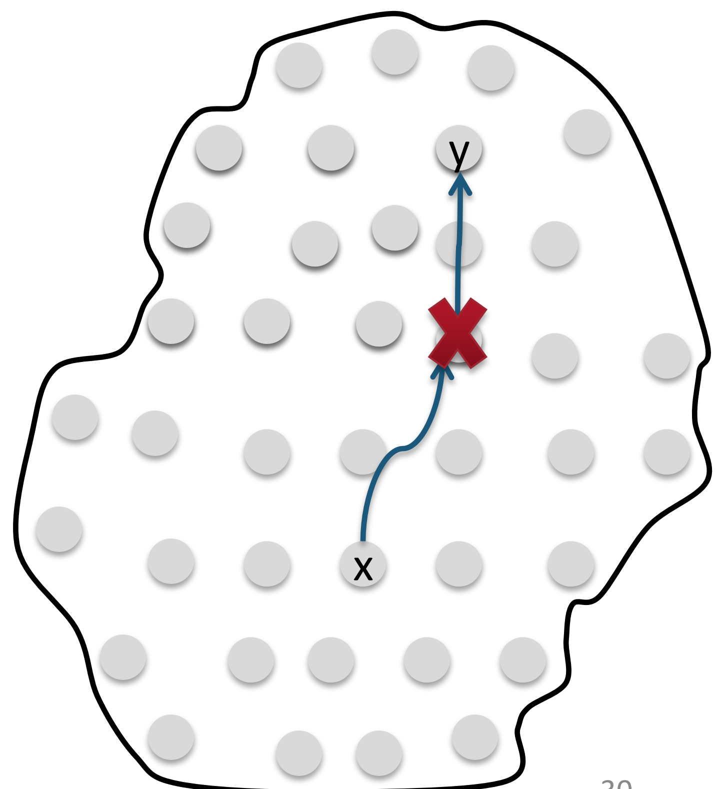
Construction of 2-TC-Spanner using filter

Intuition: Consider a deterministic non-adaptive filter

Filter cannot modify a monotone function $\Rightarrow$ must find a violation to modify a point

Construction: Connect each x to all comparable points looked up on x

Exists (x,y) with path from x to y in G but no path of length at most 2



Construction of 2-TC-Spanner using filter

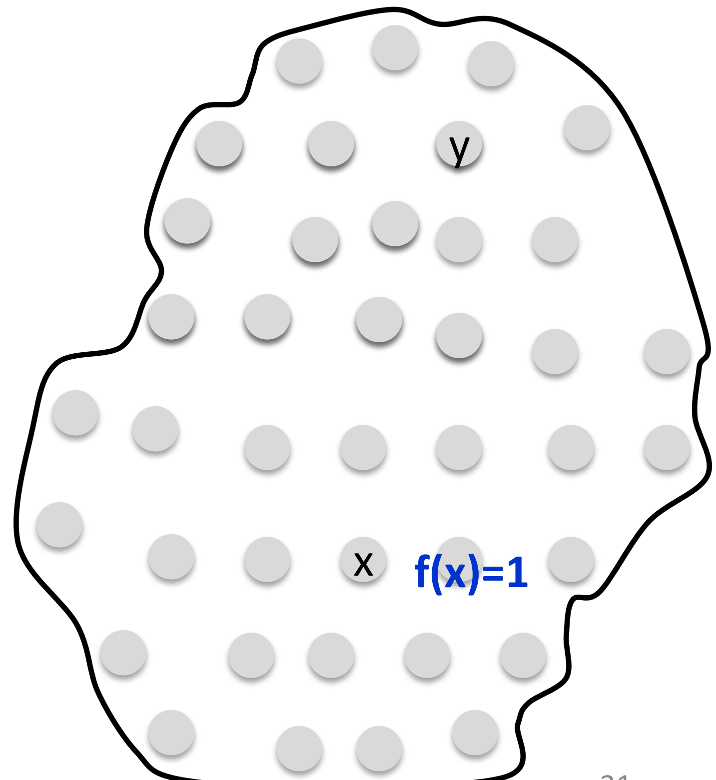
Intuition: Consider a deterministic non-adaptive filter

Filter cannot modify a monotone function $\Rightarrow$ must find a violation to modify a point

Construction: Connect each x to all comparable points looked up on *x*

Exists (x,y) with path from x to y in G but no path of length at most 2

- Set f(x)=1



Construction of 2-TC-Spanner using filter

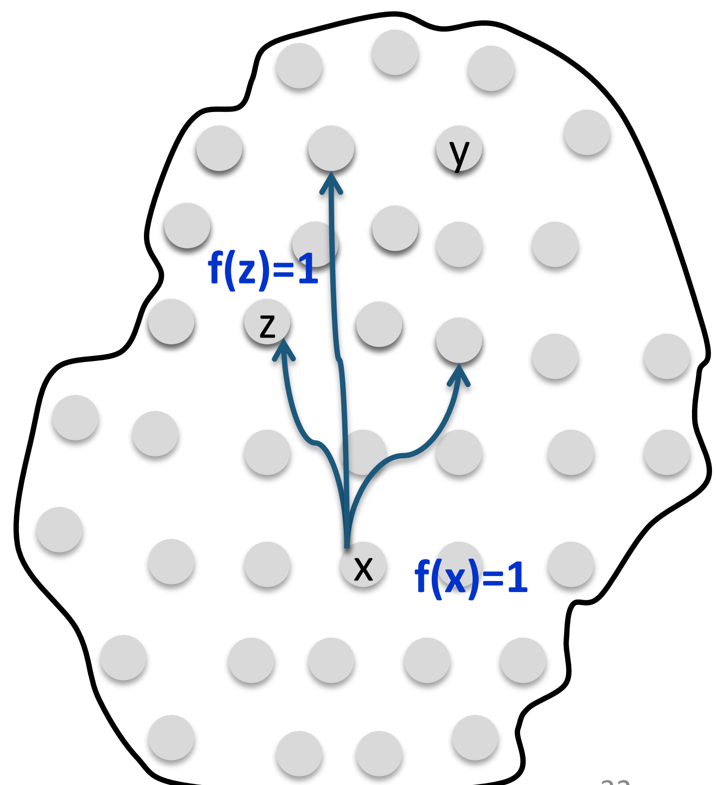
Intuition: Consider a deterministic non-adaptive filter

Filter cannot modify a monotone function $\Rightarrow$ must find a violation to modify a point

Construction: Connect each x to all comparable points looked up on *x*

Exists (x,y) with path from x to y in G but no path of length at most 2

- Set f(x)=1
- Set f(z)=1 for all edges (x,z)



Construction of 2-TC-Spanner using filter

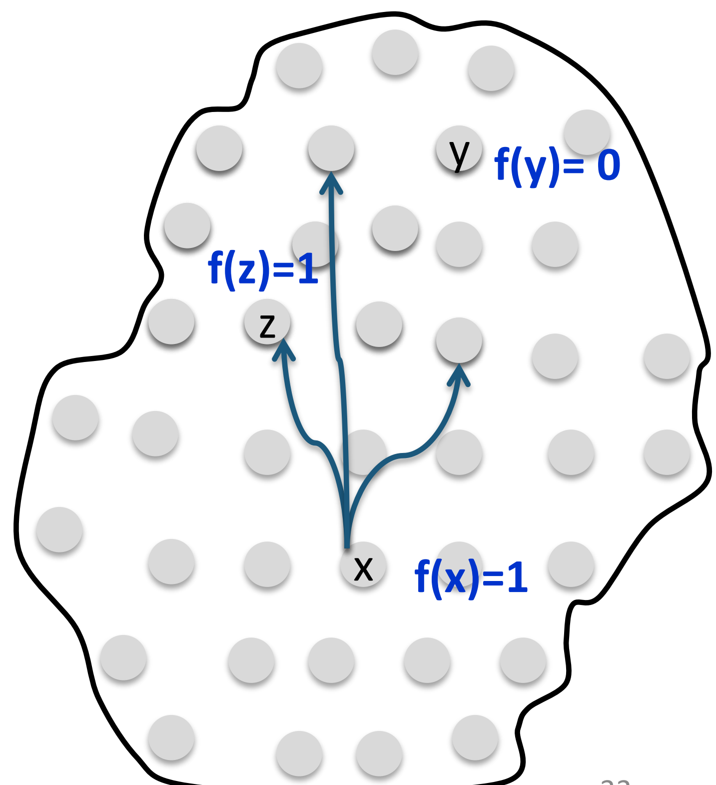
Intuition: Consider a deterministic non-adaptive filter

Filter cannot modify a monotone function $\Rightarrow$ must find a violation to modify a point

Construction: Connect each x to all comparable points looked up on x

Exists (x,y) with path from x to y in G but no path of length at most 2

- Set f(x)=1
- Set f(z)=1 for all edges (x,z)
- Set f to 0 on remaining points



Construction of 2-TC-Spanner using filter

Intuition: Consider a deterministic non-adaptive filter

Filter cannot modify a monotone function $\Rightarrow$ must find a violation to modify a point

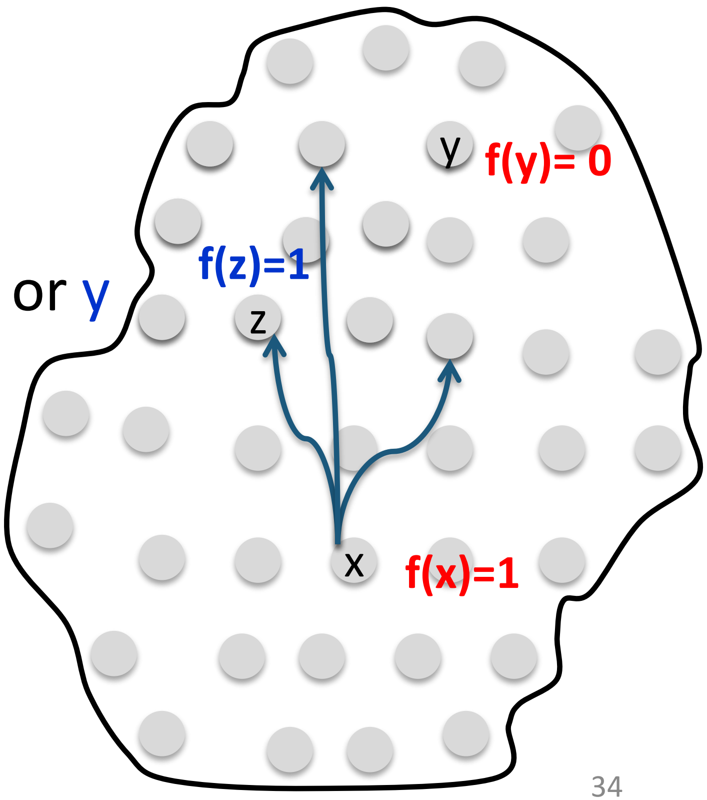
Construction: Connect each x to all comparable points looked up on *x*

Exists (x,y) with path from x to y in G but no path of length at most 2

- Set f(x)=1
- Set f(z)=1 for all edges (x,z)
- Set f to 0 on remaining points

Either x or y has to be changed,

but no violation will be detected on query x or y



Construction of 2-TC-Spanner using filter

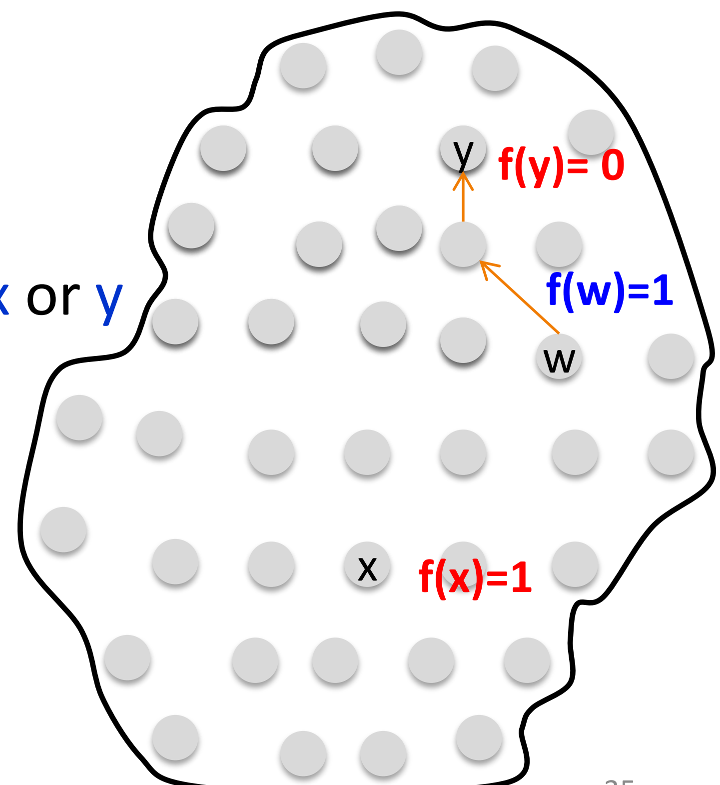
Intuition: Consider a deterministic non-adaptive filter
Filter cannot modify a monotone function $\Rightarrow$ must find a violation to modify a point

Construction: Connect each x to all comparable points looked up on *x*

Exists (x,y) with path from x to y in G but no path of length at most 2

- Set f(x)=1
- Set f(z)=1 for all edges (x,z)
- Set f to 0 on remaining points

Either x or y has to be changed,
but no violation will be detected on query x or y



Consider any violation involving y: (w,y)

- Path from w to y in G and f(w) = 1

Construction of 2-TC-Spanner using filter

Intuition: Consider a deterministic non-adaptive filter

Filter cannot modify a monotone function $\Rightarrow$ must find a violation to modify a point

Construction: Connect each x to all comparable points looked up on *x*

Exists (x,y) with path from x to y in G but no path of length at most 2

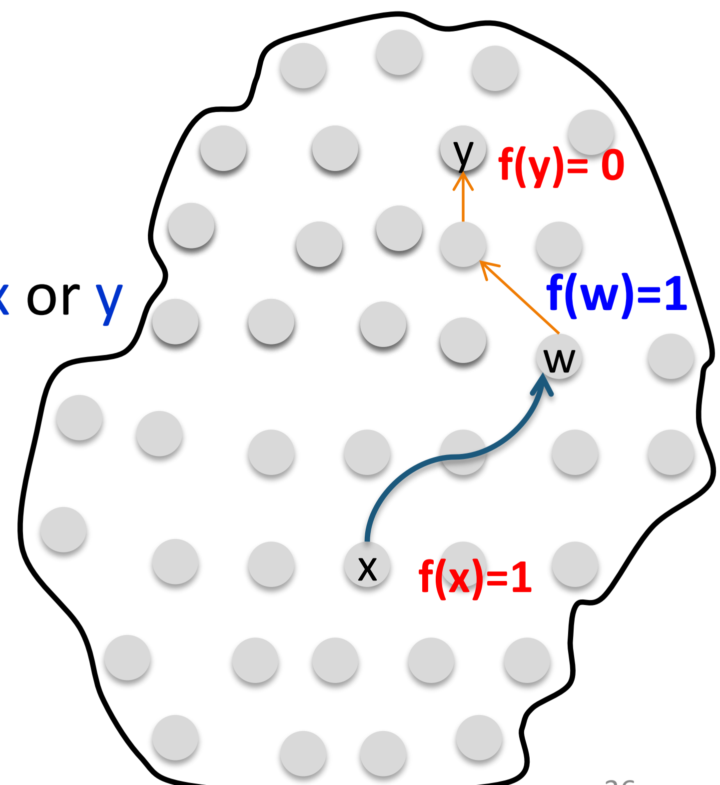
- Set f(x)=1
- Set f(z)=1 for all edges (x,z)
- Set f to 0 on remaining points

Either x or y has to be changed,

but no violation will be detected on query x or y

Consider any violation involving y: (w,y)

- Path from w to y in G and f(w) = 1
- f(w) = 1 $\Rightarrow$ w is either x or
  (x,w) an edge in construction.



Construction of 2-TC-Spanner using filter

Intuition: Consider a deterministic non-adaptive filter

Filter cannot modify a monotone function $\Rightarrow$ must find a violation to modify a point

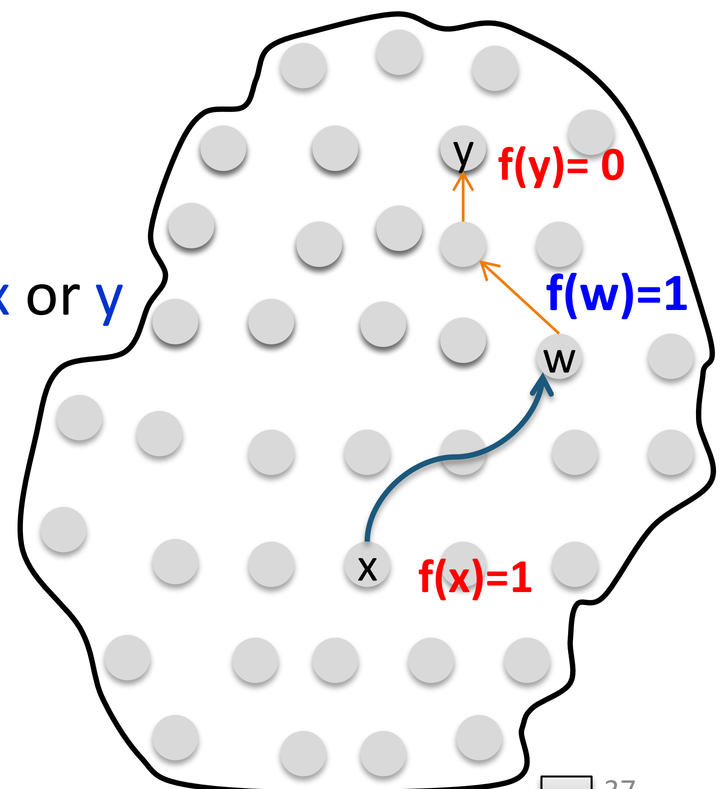
Construction: Connect each x to all comparable points looked up on *x*

Exists (x,y) with path from x to y in G but no path of length at most 2

- Set f(x)=1
- Set f(z)=1 for all edges (x,z)
- Set f to 0 on remaining points

Either x or y has to be changed,

but no violation will be detected on query x or y



Consider any violation involving y: (w,y)

- Path from w to y in G and f(w) = 1
- f(w) = 1 $\Rightarrow$ w is either x or (x,w) an edge in construction.

Either way, Filter(y) does not look up w

2-TC-spanners from Adaptive Local Monotonicity Filters

An additional factor: $2^{k(n)}$

$k(n)$: number of look ups made on
"incomparable" points

- Points that do not have path TO or FROM the query point

Open problem:

1. Can this dependence be improved?
2. Strength of look ups made to incomparable points?

Our Results (Part 2)

- Local monotonicity reconstruction and 2-TC-Spanners

(Filter-to-TC-Spanner-Reduction)

Non-adaptive local monotonicity filter on G with s(n) look up

$$\Downarrow$$

G has a 2-TC-spanner with $n \cdot s(n)$ edges

- Structural results on TC-spanners

(Hypergrid)

$$\frac{m^d \log^d m}{(2d \log \log m)^{d-1}} \leq |\text{sparsest 2-TC-Spanner of grid } [m]^d| \leq m^d \log^d m$$

(Hypercube)

$$2^{c\,d} \leq |\text{sparsest 2-TC-Spanner of grid } \{0,1\}^d| \leq 2^{c\,d}\,\text{poly}(d)$$

- m = 2: $c \approx 1.1620$

2-TC-spanners of an mxm grid

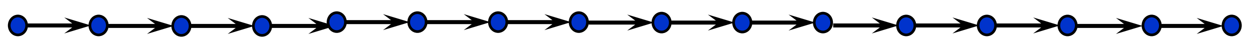
Lemma. A sparsest 2-TC-spanner of an $m \times m$ grid has $m^2 \log^2 m$ and $\Omega(m^2 \log^2 m / \log \log m)$ edges.

Upper bound: graph product of two 2-TC-spanners of the line.

Lower bound: a tradeoff argument, balancing number of edges of different types.

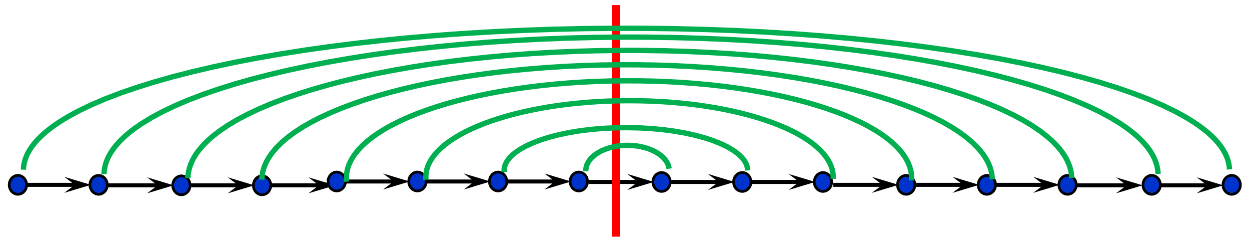
2-TC-Spanner of a line

Lemma. A sparsest 2-TC-spanner of a line with m nodes has $\Omega(m \log m)$ edges.



2-TC-Spanner of a line

Lemma. A sparsest 2-TC-spanner of a line with m nodes has $\Omega(m \log m)$ edges.

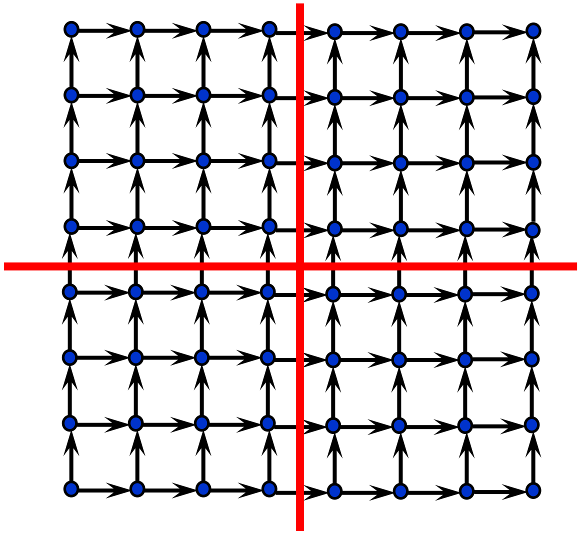


Proof:

- Each pair of nodes connected by a green arc contributes an edge crossing the midline.
- $\geq m/2$ edges cross the midline.
- Continue recursively to obtain $\Omega(m \log m)$ bound.

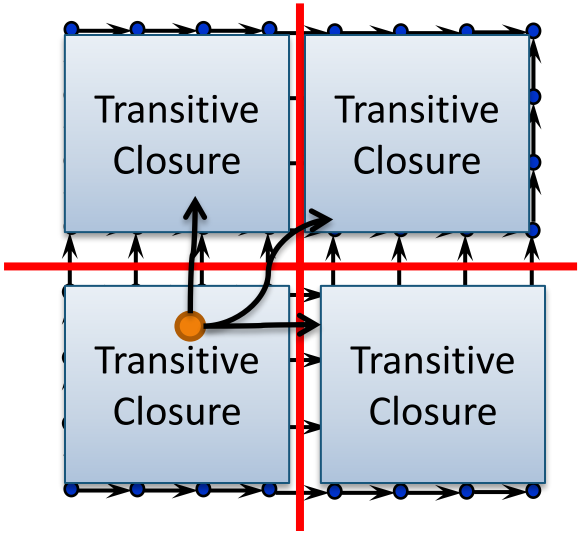
Lower Bound for an m × m Grid: First Attempt

Approach: Recursively halve the grid in both dimensions hoping that each time $\Omega(m^2 \log m)$ *edges are cut.*



Lower Bound for an m × m Grid: First Attempt

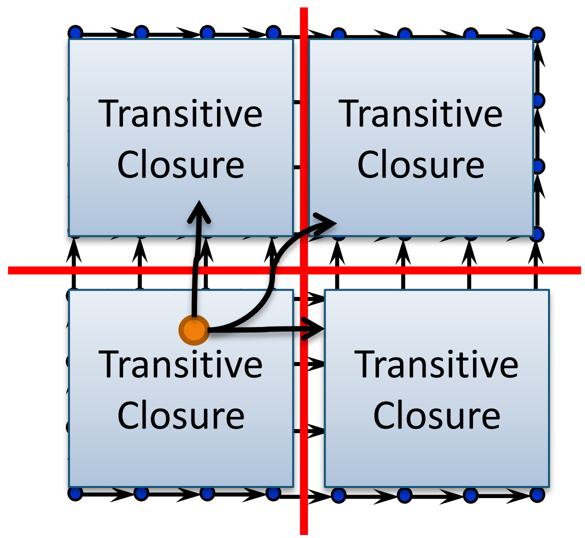
Approach: Recursively halve the grid in both dimensions hoping that each time $\Omega(m^2 \log m)$ *edges are cut.*



Problem: Exists 2-TC-spanner with only $3m^2$ edges crossing the cut.

Lower Bound for an m × m Grid: First Attempt

Approach: Recursively halve the grid in both dimensions hoping that each time $\Omega(m^2 \log m)$ *edges are cut.*

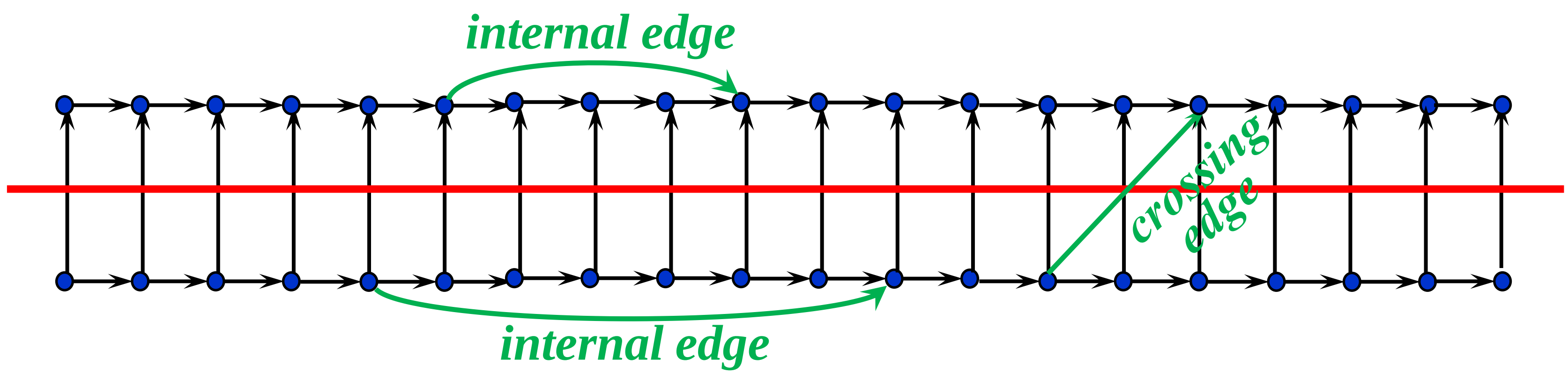


Problem: Exists 2-TC-spanner with only $3m^2$ edges crossing the cut.

There is a tradeoff between the number of internal edges and the number of edges crossing the cut.

Two-Line Tradeoff Lemma

Lemma. Consider a 2-TC-spanner of an $m \times 2$ grid. Cut it horizontally. If it has $O(m \log^2 m)$ internal edges, it has $\Omega(m \log m / \log \log m)$ crossing edges



Internal edges: Edges that are on the same side of the red line

Crossing edges: Edges that cut the red line

Conclusions: What was this paper about?

1. A new connection between

   local monotonicity filter on a graph

   and

   |2-transitive-closure-spanner|

2. Bounds on the |2-transitive-closure-spanner| of hypergrid graphs

3. Lower bounds for local monotonicity filter on hypergrid graphs

Many open problems