Efficiently Testing T-Interval Connectivity in Dynamic Graphs

A. Casteigts [1], R. Klasing [1], Y. M. Neggaz [1], J. G. Peters [2]

[1] LaBRI, CNRS, University of Bordeaux, France

[2] School of Computing Science, Simon Fraser University, Burnaby, BC, Canada

AlgoTel 2015

June 02, 2015

Overview

Dynamic Networks

Highly dynamic networks.

Ex :      

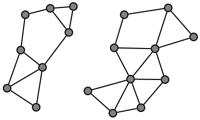
Dynamic Networks

Highly dynamic networks.

Ex : 

How changes are perceived ?

- ~~Faults and Failures ?~~
- Nature of the system. Change is normal.

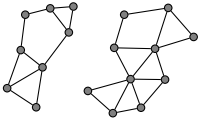


Highly dynamic networks.

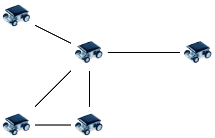
Ex : 

How changes are perceived ?

- ~~Faults and Failures ?~~
- Nature of the system. Change is normal.



Example scenario

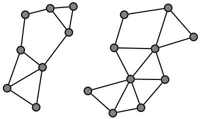


Highly dynamic networks.

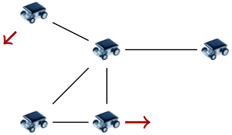
Ex : 

How changes are perceived ?

- ~~Faults and Failures ?~~
- Nature of the system. Change is normal.



Example scenario



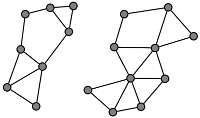
Dynamic Networks

Highly dynamic networks.

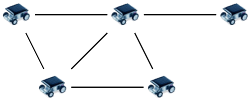
Ex : 

How changes are perceived ?

- ~~Faults and Failures ?~~
- Nature of the system. Change is normal.



Example scenario



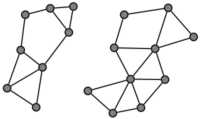
Dynamic Networks

Highly dynamic networks.

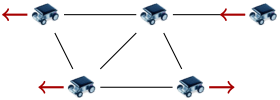
Ex : 

How changes are perceived ?

- ~~Faults and Failures ?~~
- Nature of the system. Change is normal.



Example scenario



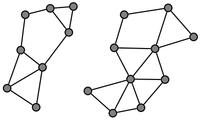
Dynamic Networks

Highly dynamic networks.

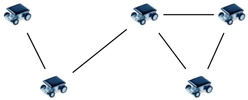
Ex : 

How changes are perceived ?

- ~~Faults and Failures ?~~
- Nature of the system. Change is normal.



Example scenario

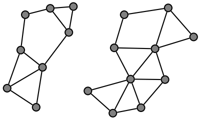


Dynamic Networks

Highly dynamic networks.

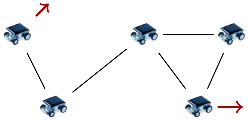
Ex : 

How changes are perceived ?



- ~~Faults and Failures ?~~
- Nature of the system. Change is normal.

Example scenario

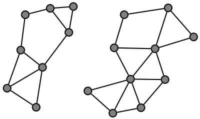


Highly dynamic networks.

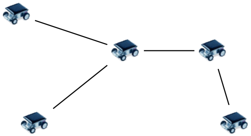
Ex : 

How changes are perceived ?

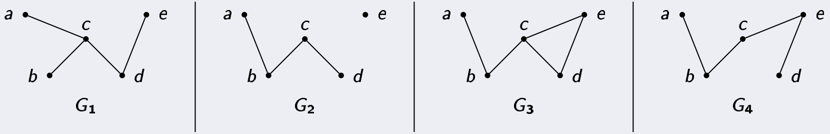
- ~~Faults and Failures ?~~
- Nature of the system. Change is normal.



Example scenario

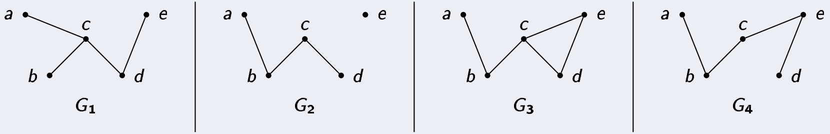


Dynamic graphs : Various forms : TVG, Evolving graphs ...

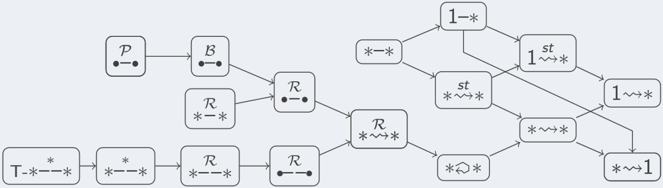


Dynamic graphs :

Various forms : TVG, Evolving graphs ...



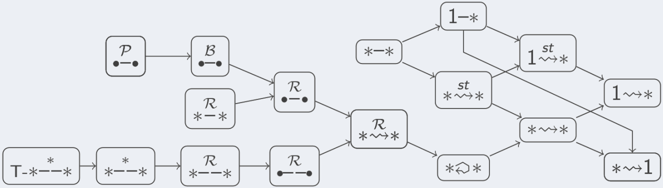
Dynamic graphs classes : *[Casteigts, Flocchini, Quattrociocchi et Santoro, 2011]*



Dynamic Graphs Classification

- Testing Temporal Connectivity in Sparse Dynamic Graphs
  [Barjon, Casteigts, Chaumette, Johnen, Neggaz, Algotel 2014]
  - Transitive closure of journeys

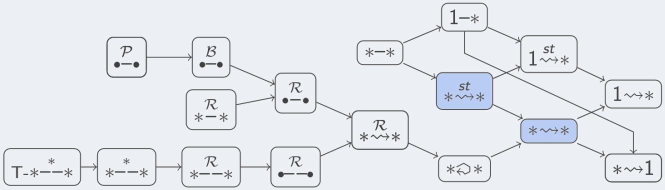
Dynamic graphs classes : *[Casteigts, Flocchini, Quattrociocchi et Santoro, 2011]*



Dynamic Graphs Classification

- Testing Temporal Connectivity in Sparse Dynamic Graphs
  [Barjon, Casteigts, Chaumette, Johnen, Neggaz, Algotel 2014]
  - Transitive closure of journeys

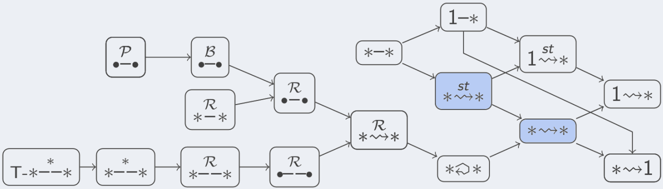
Dynamic graphs classes : *[Casteigts, Flocchini, Quattrociocchi et Santoro, 2011]*



Dynamic Graphs Classification

- Testing Temporal Connectivity in Sparse Dynamic Graphs
  [Barjon, Casteigts, Chaumette, Johnen, Neggaz, Algotel 2014]
  - Transitive closure of journeys
- Shortest, Fastest, and Foremost Broadcast in Dynamic Networks
  [Casteigts, Flocchini, Mans, Santoro, IJFCS 2015]
  - Feasibility requires distinct features on the evolution
  - Re-appearance of edges : recurrent, bounded-recurrent, periodic

Dynamic graphs classes : *[Casteigts, Flocchini, Quattrociocchi et Santoro, 2011]*

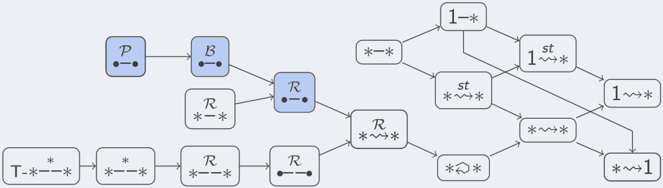


Dynamic Graphs Classification

- Testing Temporal Connectivity in Sparse Dynamic Graphs
  [Barjon, Casteigts, Chaumette, Johnen, Neggaz, Algotel 2014]
  - Transitive closure of journeys

- Shortest, Fastest, and Foremost Broadcast in Dynamic Networks
  [Casteigts, Flocchini, Mans, Santoro, IJFCS 2015]
  - Feasibility requires distinct features on the evolution
  - Re-appearance of edges : recurrent, bounded-recurrent, periodic

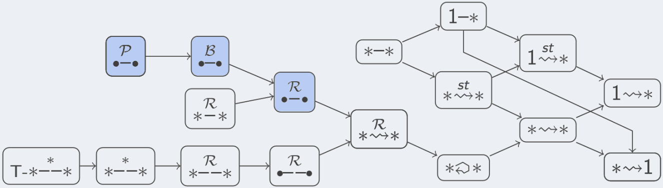
Dynamic graphs classes : *[Casteigts, Flocchini, Quattrociocchi et Santoro, 2011]*



T-interval Connectivity

- In this work : Testing T-Interval Connectivity in Dynamic Graphs

Dynamic graphs classes : *[Casteigts, Flocchini, Quattrociocchi et Santoro, 2011]*



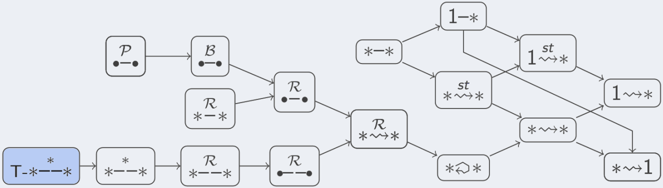
T-interval Connectivity

- In this work : Testing T-Interval Connectivity in Dynamic Graphs

Definition : T-interval connectivity

A dynamic graph $\mathcal{G}$ of length δ is T-interval connected if and only if every T length sequence of graphs has a common connected spanning sub-graph.

Dynamic graphs classes : *[Casteigts, Flocchini, Quattrociocchi et Santoro, 2011]*



[Kuhn, Lynch, Oshman. STOC 2010]

T-interval Connectivity

- In this work : Testing T-Interval Connectivity in Dynamic Graphs

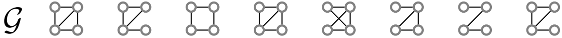
Definition : T-interval connectivity

A dynamic graph $\mathcal{G}$ of length δ is T-interval connected if and only if every T length sequence of graphs has a common connected spanning sub-graph.

- In this work : Testing T-Interval Connectivity in Dynamic Graphs

Definition : T-interval connectivity

A dynamic graph $\mathcal{G}$ of length δ is T-interval connected if and only if every T length sequence of graphs has a common connected spanning sub-graph.

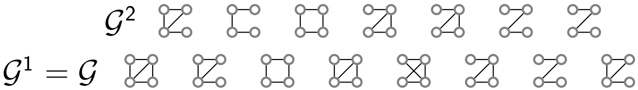


T-interval Connectivity

- In this work : Testing T-Interval Connectivity in Dynamic Graphs

Definition : T-interval connectivity

A dynamic graph $\mathcal{G}$ of length δ is T-interval connected if and only if every T length sequence of graphs has a common connected spanning sub-graph.

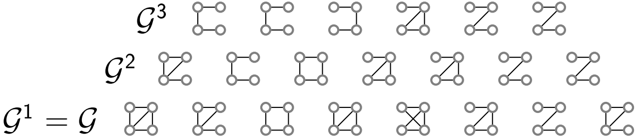


T-interval Connectivity

- In this work : Testing T-Interval Connectivity in Dynamic Graphs

Definition : T-interval connectivity

A dynamic graph $\mathcal{G}$ of length δ is T-interval connected if and only if every T length sequence of graphs has a common connected spanning sub-graph.

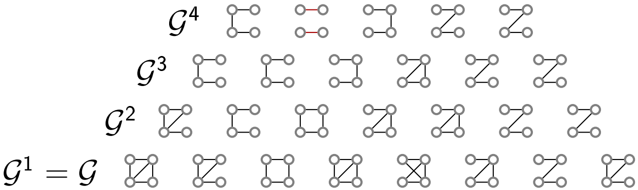


T-interval Connectivity

- In this work : Testing T-Interval Connectivity in Dynamic Graphs

Definition : T-interval connectivity

A dynamic graph $\mathcal{G}$ of length δ is T-interval connected if and only if every T length sequence of graphs has a common connected spanning sub-graph.



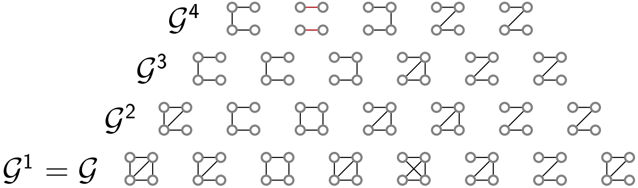
T-interval Connectivity

- <u>In this work</u> : Testing T-Interval Connectivity in Dynamic Graphs

Definition : T-interval connectivity

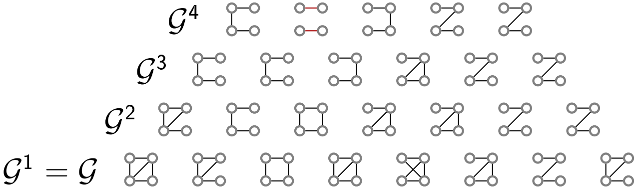
A dynamic graph $\mathcal{G}$ of length δ is T-interval connected if and only if every T length sequence of graphs has a common connected spanning sub-graph.

- <u>T-INTERVAL-CONNECTIVITY</u> : Test whether a dynamic graph is T-interval connected for a given T
- <u>INTERVAL-CONNECTIVITY</u> : Find the largest T for which a given dynamic graph is T-interval connected



T-interval Connectivity

- Our approach :
  - High-level strategies that work directly at the graph level
  - Two elementary graph-level operations : *Binary intersection* and *Connectivity testing* (comparable costs)



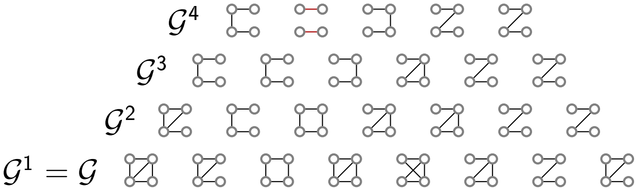
Lower Bound

Lower Bound (by contradiction)

- Let A be an algorithm that decides if $\mathcal{G}$ is T-interval connected in $\delta - 1$

  $\Rightarrow$ At least one graph $G \in \mathcal{G}$ is never accessed by A

- G could be connected or disconnected



Lower Bound (by contradiction)

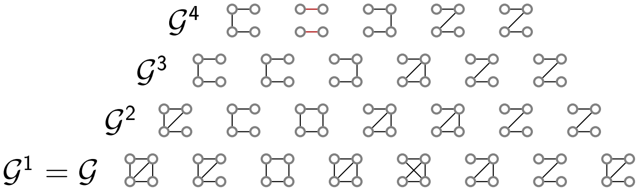
- Let A be an algorithm that decides if $\mathcal{G}$ is T-interval connected in $\delta - 1$

  $\Rightarrow$ At least one graph $G \in \mathcal{G}$ is never accessed by A

- G could be connected or disconnected

  $\Rightarrow \Omega(\delta)$ elementary operations are necessary to solve T-INTERVAL-CONNECTIVITY

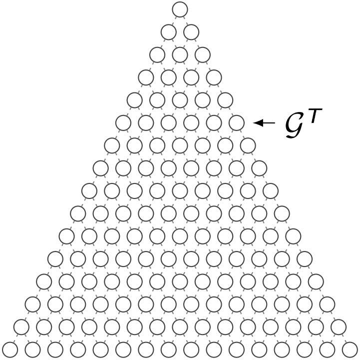
- Same argument for INTERVAL-CONNECTIVITY



Row-Based Strategy

Row-Based Strategy

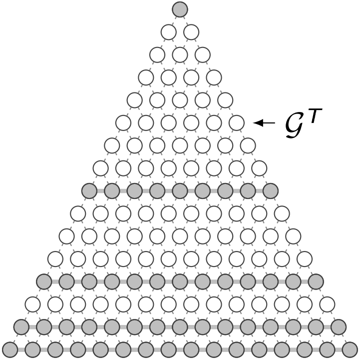
- T-INTERVAL-CONNECTIVITY (Given T)



Row-Based Strategy

- T-Interval-Connectivity (Given T)

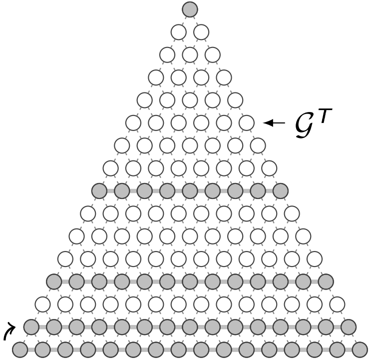
- Compute only "power rows" : $\mathcal{G}^{2^i}$



Row-Based Strategy

- T-Interval-Connectivity (Given T)

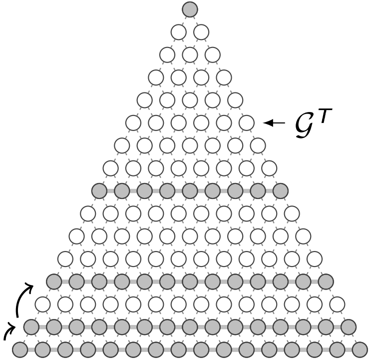
- Compute only "power rows" : $\mathcal{G}^{2^i}$



Row-Based Strategy

- T-Interval-Connectivity (Given T)

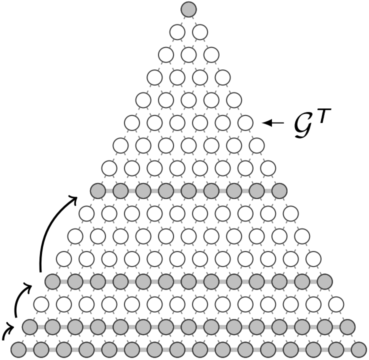
- Compute only "power rows" : $\mathcal{G}^{2^i}$



Row-Based Strategy

- T-Interval-Connectivity (Given T)

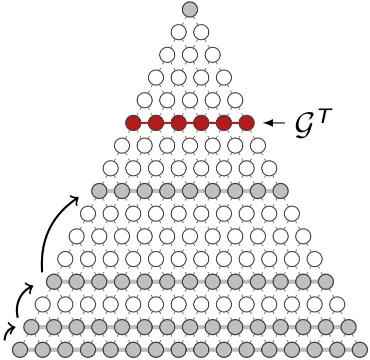
- Compute only "power rows" : $\mathcal{G}^{2^i}$
- $O(\delta)$ intersections per row



Row-Based Strategy

- T-INTERVAL-CONNECTIVITY (Given T)

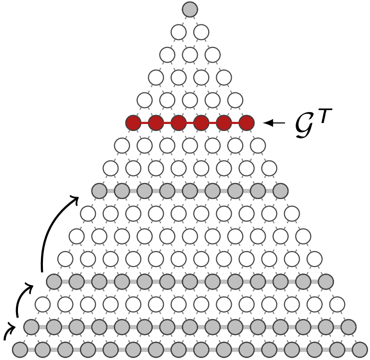
- Compute only "power rows" : $\mathcal{G}^{2^i}$
- $O(\delta)$ intersections per row



Row-Based Strategy

- T-Interval-Connectivity (Given T)

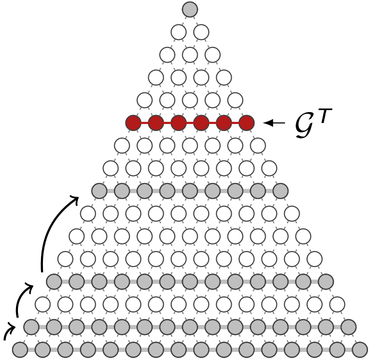
- Compute only "power rows" : $\mathcal{G}^{2^i}$
- $O(\delta)$ intersections per row
- $O(\log \delta)$ rows



Row-Based Strategy

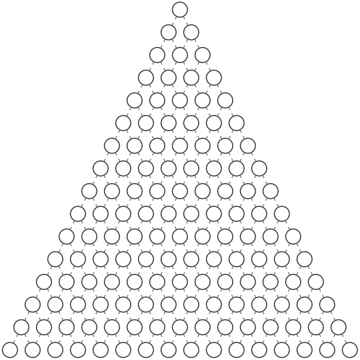
- T-INTERVAL-CONNECTIVITY (Given T) **is solvable with** $O(\delta \log \delta)$ **elementary operations**

- Compute only "power rows" : $\mathcal{G}^{2^i}$
- $O(\delta)$ intersections per row
- $O(\log \delta)$ rows



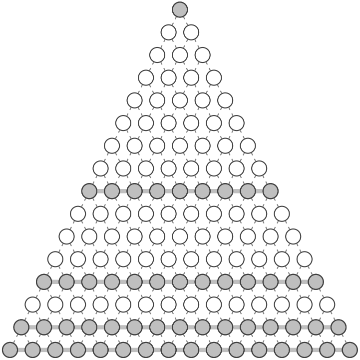
Row-Based Strategy

- INTERVAL-CONNECTIVITY (Find T)



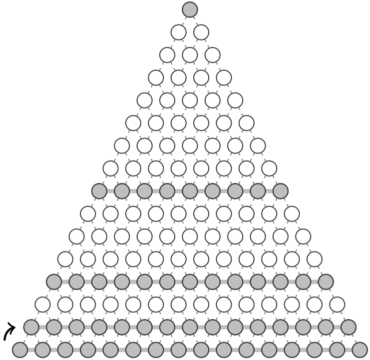
Row-Based Strategy

- Interval-Connectivity (Find T)

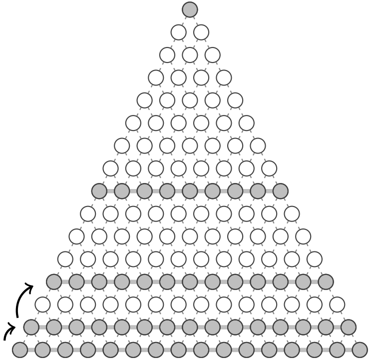


Row-Based Strategy

- Interval-Connectivity (Find T)



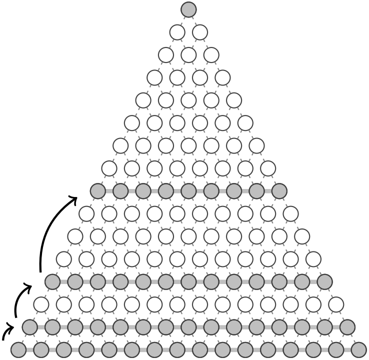
- Interval-Connectivity (Find T)



Row-Based Strategy

- Interval-Connectivity (Find T)

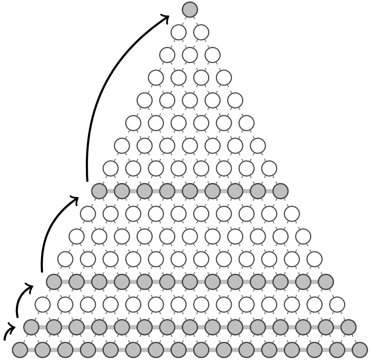
- $O(\delta)$ intersections per row
- $O(\delta)$ connectivity tests per row



Row-Based Strategy

- Interval-Connectivity (Find T)

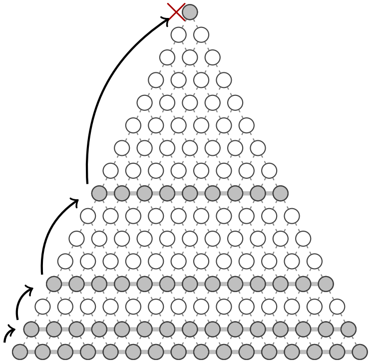
- $O(\delta)$ intersections per row
- $O(\delta)$ connectivity tests per row



Row-Based Strategy

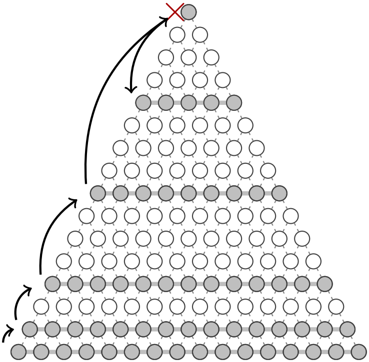
- INTERVAL-CONNECTIVITY (Find T)

- $O(\delta)$ intersections per row
- $O(\delta)$ connectivity tests per row



Row-Based Strategy

- Interval-Connectivity (Find T)

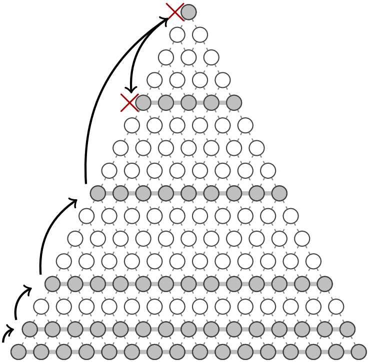


- $O(\delta)$ intersections per row
- $O(\delta)$ connectivity tests per row

Row-Based Strategy

- Interval-Connectivity (Find T)

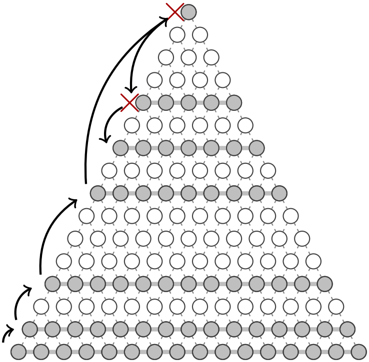
- $O(\delta)$ intersections per row
- $O(\delta)$ connectivity tests per row



Row-Based Strategy

- Interval-Connectivity (Find T)

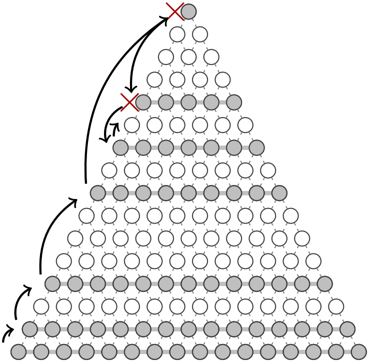
- $O(\delta)$ intersections per row
- $O(\delta)$ connectivity tests per row



Row-Based Strategy

- Interval-Connectivity (Find T)

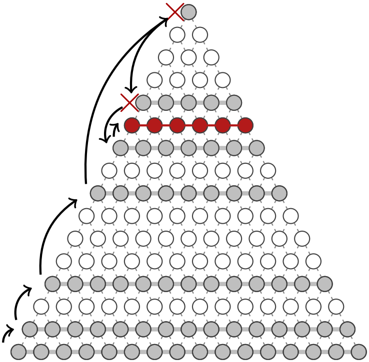
- $O(\delta)$ intersections per row
- $O(\delta)$ connectivity tests per row



Row-Based Strategy

- INTERVAL-CONNECTIVITY (Find T)

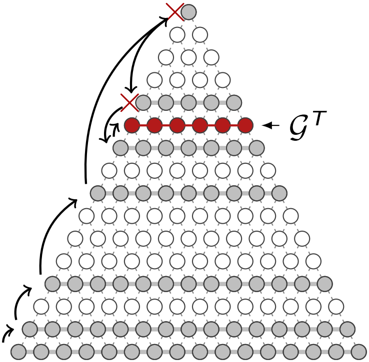
- $O(\delta)$ intersections per row
- $O(\delta)$ connectivity tests per row



Row-Based Strategy

- Interval-Connectivity (Find T)

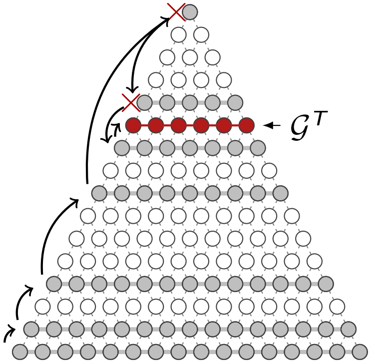
- $O(\delta)$ intersections per row
- $O(\delta)$ connectivity tests per row



Row-Based Strategy

- INTERVAL-CONNECTIVITY (Find T)

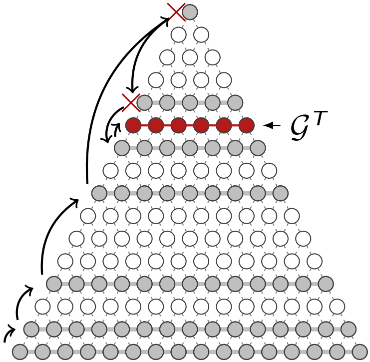
- $O(\delta)$ intersections per row
- $O(\delta)$ connectivity tests per row
- $O(\log \delta)$ rows



Row-Based Strategy

- INTERVAL-CONNECTIVITY (Find T) **is solvable with** $O(\delta \log \delta)$ **elementary operations**

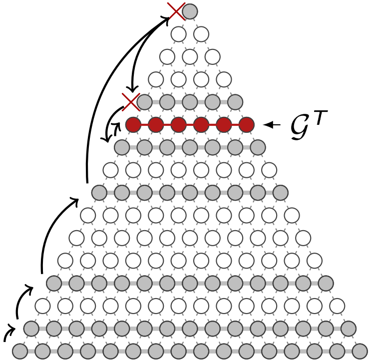
- $O(\delta)$ intersections per row
- $O(\delta)$ connectivity tests per row
- $O(\log \delta)$ rows



T-interval Connectivity on EREW PRAM

- T-Interval-Connectivity and Interval-Connectivity are in Nick's class

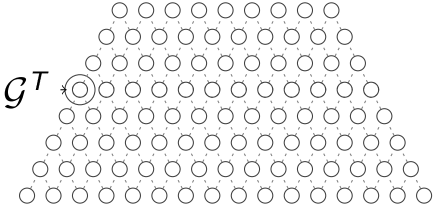
- T-Interval-Connectivity is solvable in $O(\log \delta)$ on an **EREW PRAM** with $O(\delta)$ processors
- Interval-Connectivity is solvable in $O(\log^2 \delta)$ on an **EREW PRAM** with $O(\delta)$ processors



Optimal Solution

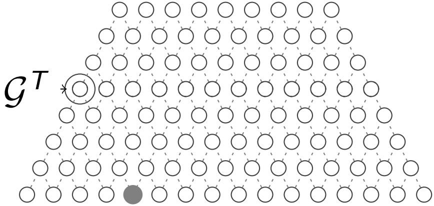
Optimal Solution

- T-Interval-Connectivity



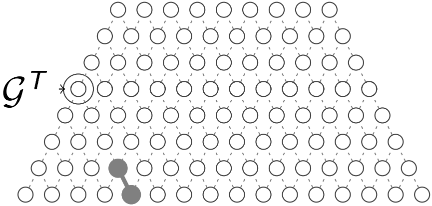
Optimal Solution

- T-Interval-Connectivity



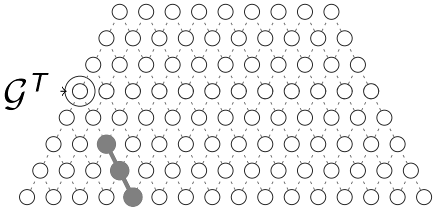
Optimal Solution

- T-Interval-Connectivity



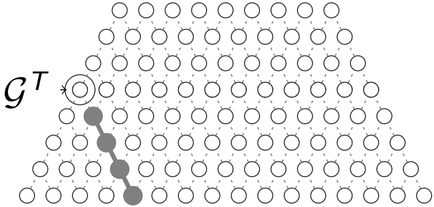
Optimal Solution

- T-Interval-Connectivity



Optimal Solution

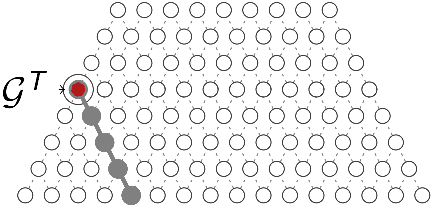
- T-INTERVAL-CONNECTIVITY



Optimal Solution

- T-INTERVAL-CONNECTIVITY

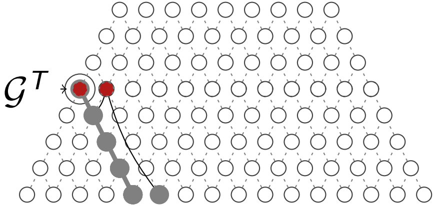
- A ladder of length l costs $l - 1$ binary intersections



Optimal Solution

- T-Interval-Connectivity

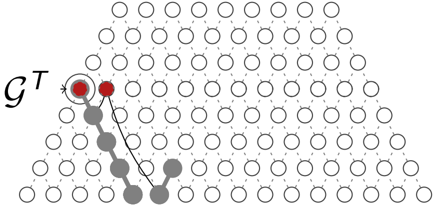
- A ladder of length l costs $l - 1$ binary intersections



Optimal Solution

- T-Interval-Connectivity

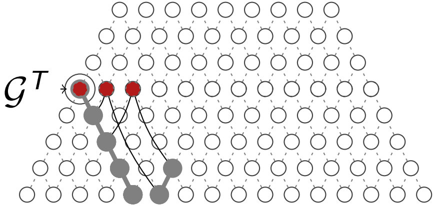
- A ladder of length l costs $l-1$ binary intersections



Optimal Solution

- T-INTERVAL-CONNECTIVITY

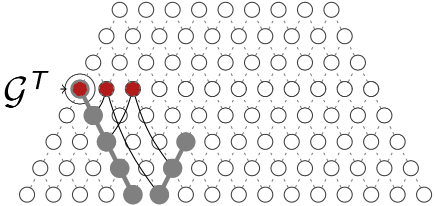
- A ladder of length l costs $l - 1$ binary intersections



Optimal Solution

- T-INTERVAL-CONNECTIVITY

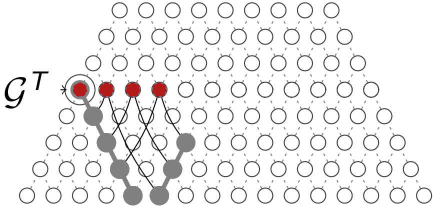
- A ladder of length l costs $l - 1$ binary intersections



Optimal Solution

- T-INTERVAL-CONNECTIVITY

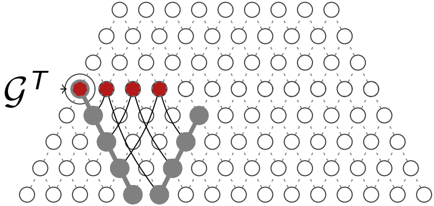
- A ladder of length l costs $l - 1$ binary intersections



Optimal Solution

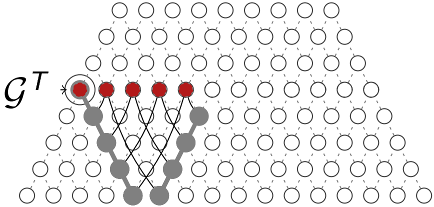
- T-Interval-Connectivity

- A ladder of length l costs $l-1$ binary intersections



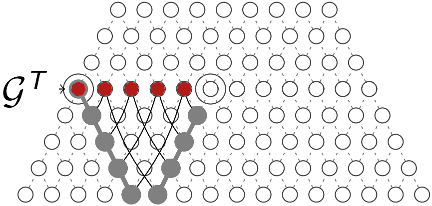
- T-INTERVAL-CONNECTIVITY

- A ladder of length l costs $l - 1$ binary intersections
- Any graph "between" (red graphs) two ladders can be computed by a single binary intersection



- T-INTERVAL-CONNECTIVITY

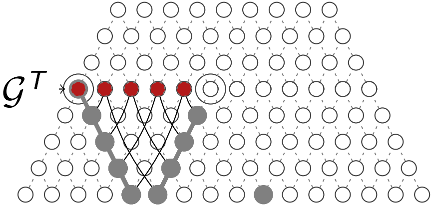
- A ladder of length l costs $l-1$ binary intersections
- Any graph "between" (red graphs) two ladders can be computed by a single binary intersection



Optimal Solution

- T-INTERVAL-CONNECTIVITY

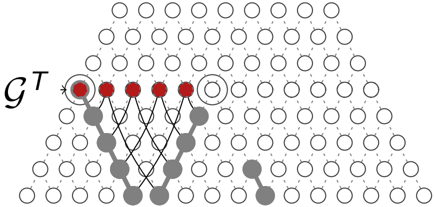
- A ladder of length l costs $l-1$ binary intersections
- Any graph "between" (red graphs) two ladders can be computed by a single binary intersection



Optimal Solution

- T-Interval-Connectivity

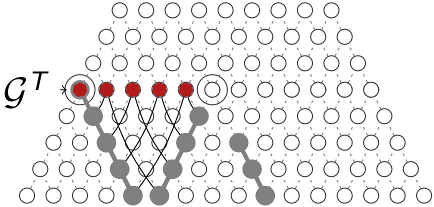
- A ladder of length l costs $l-1$ binary intersections
- Any graph "between" (red graphs) two ladders can be computed by a single binary intersection



Optimal Solution

- T-Interval-Connectivity

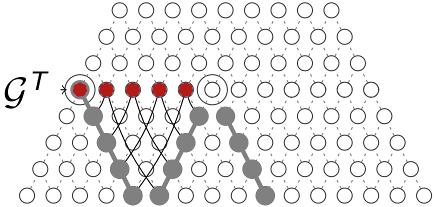
- A ladder of length l costs $l - 1$ binary intersections
- Any graph "between" (red graphs) two ladders can be computed by a single binary intersection



Optimal Solution

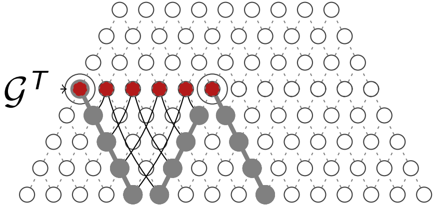
- T-INTERVAL-CONNECTIVITY

- A ladder of length l costs $l-1$ binary intersections
- Any graph "between" (red graphs) two ladders can be computed by a single binary intersection



- T-Interval-Connectivity

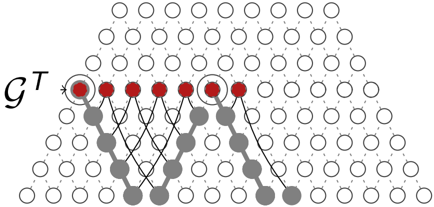
- A ladder of length l costs $l - 1$ binary intersections
- Any graph "between" (red graphs) two ladders can be computed by a single binary intersection



Optimal Solution

- T-INTERVAL-CONNECTIVITY

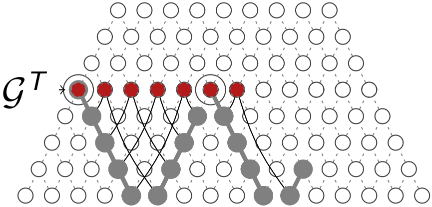
- A ladder of length l costs $l-1$ binary intersections
- Any graph "between" (red graphs) two ladders can be computed by a single binary intersection



Optimal Solution

- T-Interval-Connectivity

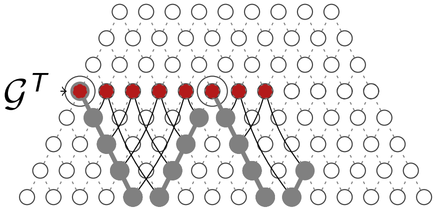
- A ladder of length l costs $l - 1$ binary intersections
- Any graph "between" (red graphs) two ladders can be computed by a single binary intersection



Optimal Solution

- T-INTERVAL-CONNECTIVITY

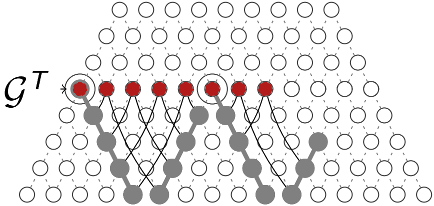
- A ladder of length l costs $l - 1$ binary intersections
- Any graph "between" (red graphs) two ladders can be computed by a single binary intersection



Optimal Solution

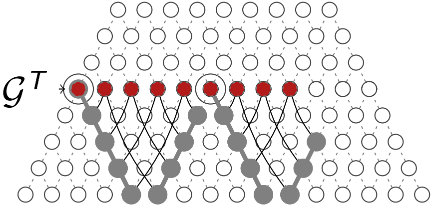
- T-INTERVAL-CONNECTIVITY

- A ladder of length l costs $l - 1$ binary intersections
- Any graph "between" (red graphs) two ladders can be computed by a single binary intersection



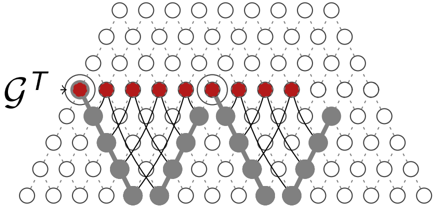
- T-INTERVAL-CONNECTIVITY

- A ladder of length l costs $l - 1$ binary intersections
- Any graph "between" (red graphs) two ladders can be computed by a single binary intersection



- T-Interval-Connectivity

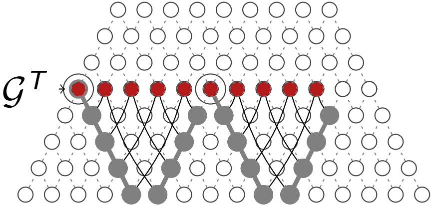
- A ladder of length l costs $l - 1$ binary intersections
- Any graph "between" (red graphs) two ladders can be computed by a single binary intersection



Optimal Solution

- T-Interval-Connectivity

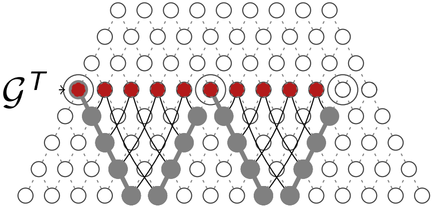
- A ladder of length l costs $l-1$ binary intersections
- Any graph "between" (red graphs) two ladders can be computed by a single binary intersection



Optimal Solution

- T-Interval-Connectivity

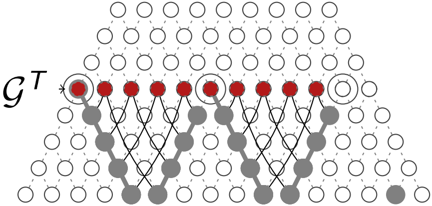
- A ladder of length l costs $l - 1$ binary intersections
- Any graph "between" (red graphs) two ladders can be computed by a single binary intersection



Optimal Solution

- T-INTERVAL-CONNECTIVITY

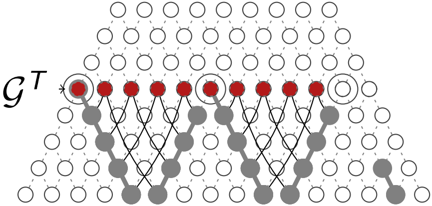
- A ladder of length l costs $l - 1$ binary intersections
- Any graph "between" (red graphs) two ladders can be computed by a single binary intersection



Optimal Solution

- T-Interval-Connectivity

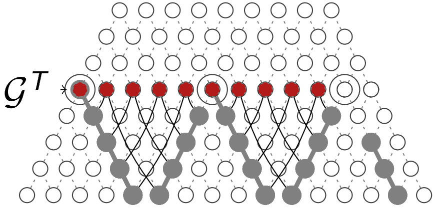
- A ladder of length l costs $l-1$ binary intersections
- Any graph "between" (red graphs) two ladders can be computed by a single binary intersection



Optimal Solution

- T-Interval-Connectivity

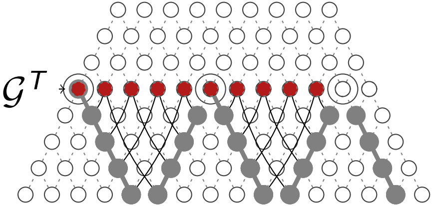
- A ladder of length l costs $l - 1$ binary intersections
- Any graph "between" (red graphs) two ladders can be computed by a single binary intersection



Optimal Solution

- T-Interval-Connectivity

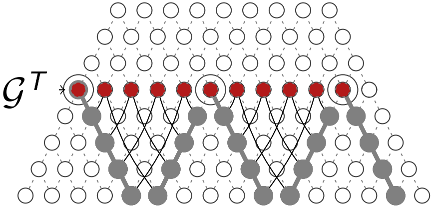
- A ladder of length l costs $l - 1$ binary intersections
- Any graph "between" (red graphs) two ladders can be computed by a single binary intersection



Optimal Solution

- T-INTERVAL-CONNECTIVITY

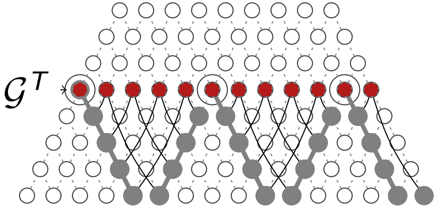
- A ladder of length l costs $l - 1$ binary intersections
- Any graph "between" (red graphs) two ladders can be computed by a single binary intersection



Optimal Solution

- T-Interval-Connectivity

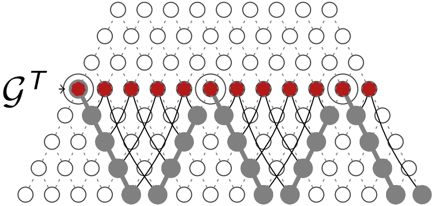
- A ladder of length l costs $l - 1$ binary intersections
- Any graph "between" (red graphs) two ladders can be computed by a single binary intersection



Optimal Solution

- T-INTERVAL-CONNECTIVITY

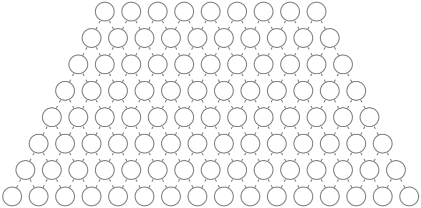
- A ladder of length l costs $l - 1$ binary intersections
- Any graph "between" (red graphs) two ladders can be computed by a single binary intersection



Theorem 1 : T-INTERVAL-CONNECTIVITY is solvable with $O(\delta)$ elementary operations

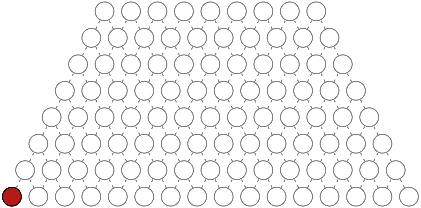
Optimal Solution

- INTERVAL-CONNECTIVITY (Find T)



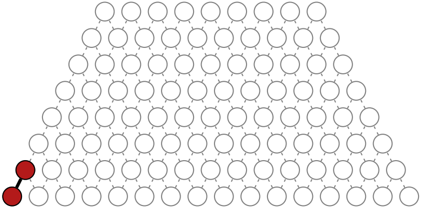
Optimal Solution

- Interval-Connectivity (Find T)



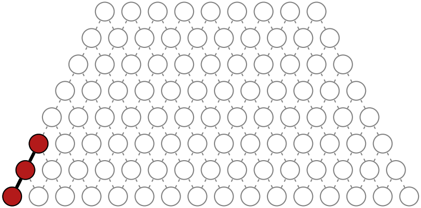
Optimal Solution

- INTERVAL-CONNECTIVITY (Find T)



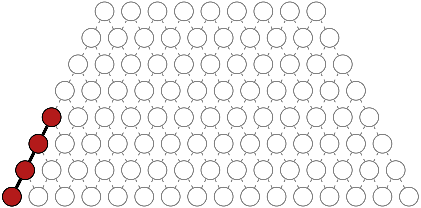
Optimal Solution

- INTERVAL-CONNECTIVITY (Find T)



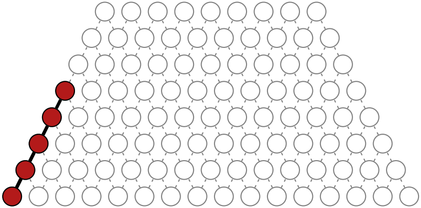
Optimal Solution

- INTERVAL-CONNECTIVITY (Find T)



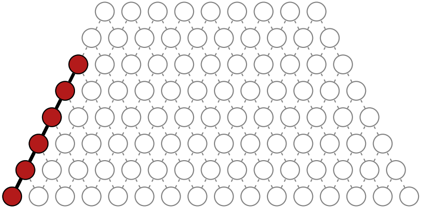
Optimal Solution

- INTERVAL-CONNECTIVITY (Find T)



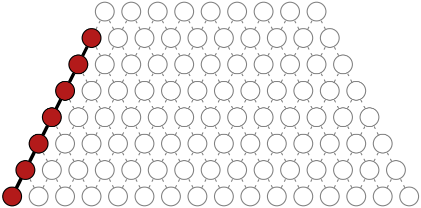
Optimal Solution

- INTERVAL-CONNECTIVITY (Find T)



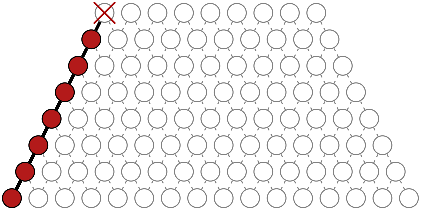
Optimal Solution

- INTERVAL-CONNECTIVITY (Find T)



Optimal Solution

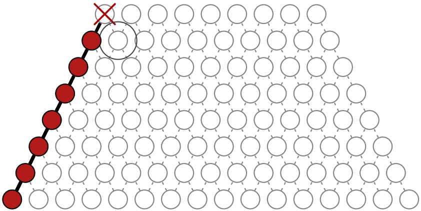
- Interval-Connectivity (Find T)



Optimal Solution

- Interval-Connectivity (Find T)

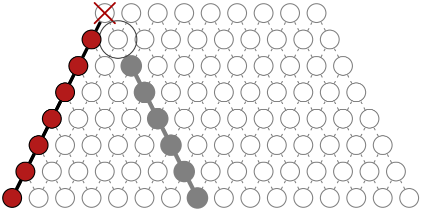
- Strategy : descending walk



Optimal Solution

- Interval-Connectivity (Find T)

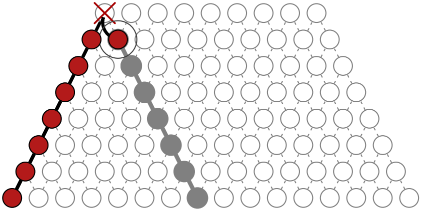
- Strategy : descending walk



Optimal Solution

- INTERVAL-CONNECTIVITY (Find T)

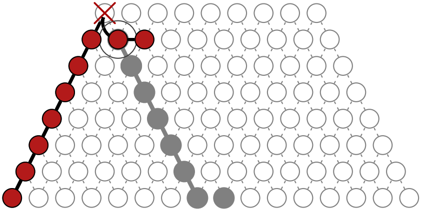
- Strategy : descending walk



Optimal Solution

- INTERVAL-CONNECTIVITY (Find T)

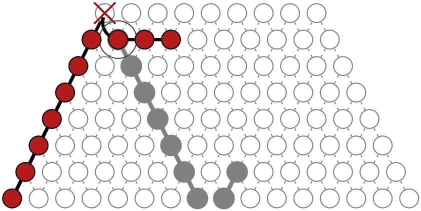
- Strategy : descending walk



Optimal Solution

- INTERVAL-CONNECTIVITY (Find T)

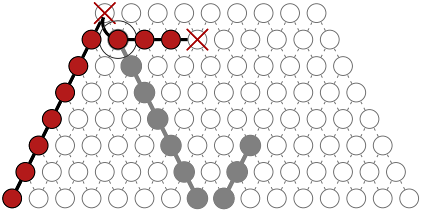
- Strategy : descending walk



Optimal Solution

- Interval-Connectivity (Find T)

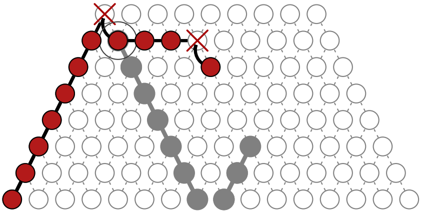
- Strategy : descending walk



Optimal Solution

- Interval-Connectivity (Find T)

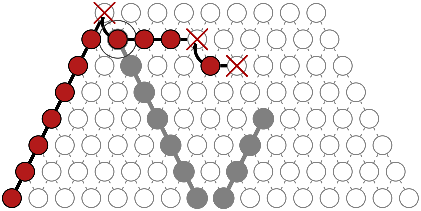
- Strategy : descending walk



Optimal Solution

- INTERVAL-CONNECTIVITY (Find T)

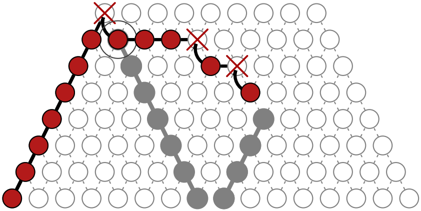
- Strategy : descending walk



Optimal Solution

- Interval-Connectivity (Find T)

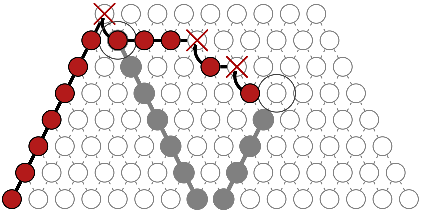
- Strategy : descending walk



Optimal Solution

- INTERVAL-CONNECTIVITY (Find T)

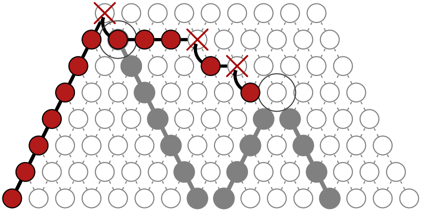
- Strategy : descending walk



Optimal Solution

- INTERVAL-CONNECTIVITY (Find T)

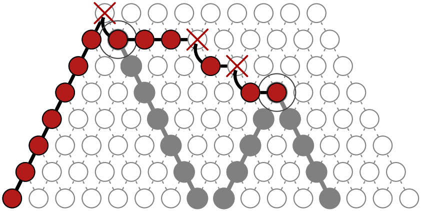
- Strategy : descending walk



Optimal Solution

- Interval-Connectivity (Find T)

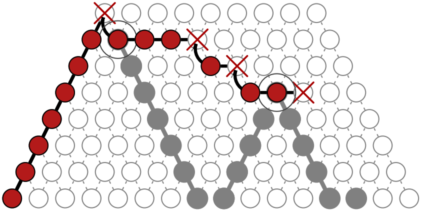
- Strategy : descending walk



Optimal Solution

- Interval-Connectivity (Find T)

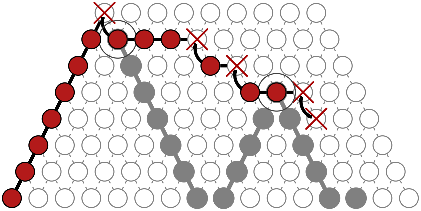
- Strategy : descending walk



Optimal Solution

- Interval-Connectivity (Find T)

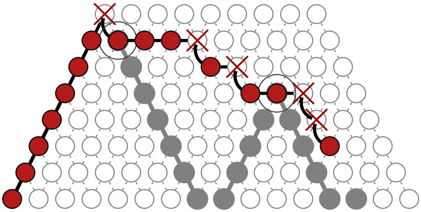
- Strategy : descending walk



Optimal Solution

- INTERVAL-CONNECTIVITY (Find T)

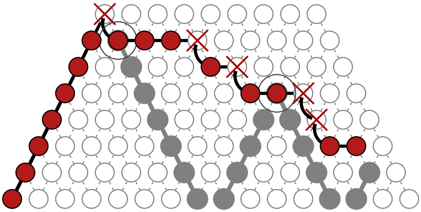
- Strategy : descending walk



Optimal Solution

- Interval-Connectivity (Find T)

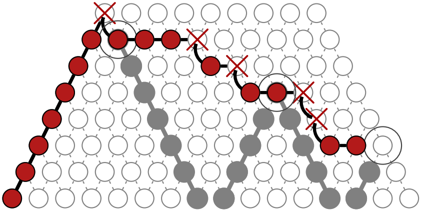
- Strategy : descending walk



Optimal Solution

- INTERVAL-CONNECTIVITY (Find T)

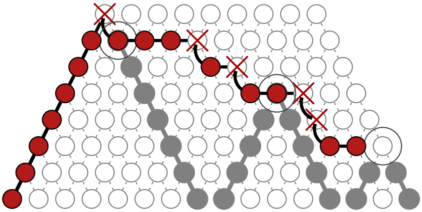
- Strategy : descending walk



Optimal Solution

- Interval-Connectivity (Find T)

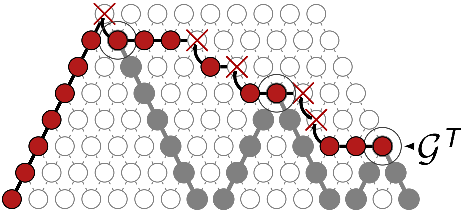
- Strategy : descending walk



Optimal Solution

- INTERVAL-CONNECTIVITY (Find T)

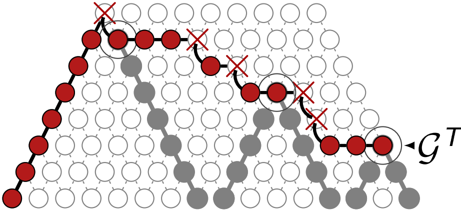
- Strategy : descending walk



Optimal Solution

- INTERVAL-CONNECTIVITY (Find T)

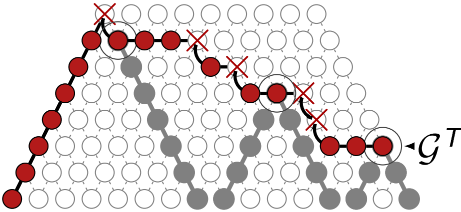
- Strategy : descending walk
- The total length of the ladders is $O(\delta)$
- At most $O(\delta)$ binary intersections and connectivity tests



Optimal Solution

- INTERVAL-CONNECTIVITY (Find T)

- Strategy : descending walk
- The total length of the ladders is $O(\delta)$
- At most $O(\delta)$ binary intersections and connectivity tests



Theorem 2 : INTERVAL-CONNECTIVITY is solvable with $O(\delta)$ elementary operations

Online Algorithms

Online Algorithms

- The optimal algorithms for T-Interval-Connectivity and Interval-Connectivity can be adapted to an **online setting**
- The sequence of graphs $G_1, G_2, G_3, \dots$ of $\mathcal{G}$ is processed in the order of reception
- T-Interval-Connectivity and Interval-Connectivity can be solved online with an **amortized cost of** $O(1)$ elementary operations per graph received

Conclusion and Future works

- Conclusions :
  - Efficient algorithms that use only $O(\delta)$ elementary operations, asymptotically matching the lower bound of $\Omega(\delta)$
  - Both problems are efficiently parallelizable on PRAM (in Nick's class)
  - Online algorithms with amortized cost of $O(1)$ elementary operations per graph received
- Future work :
  - Use specific data structure and low-level operations
  - Sliding window online algorithms
  - How about other classes ?
  - How about distributed testing ?

Thank you !