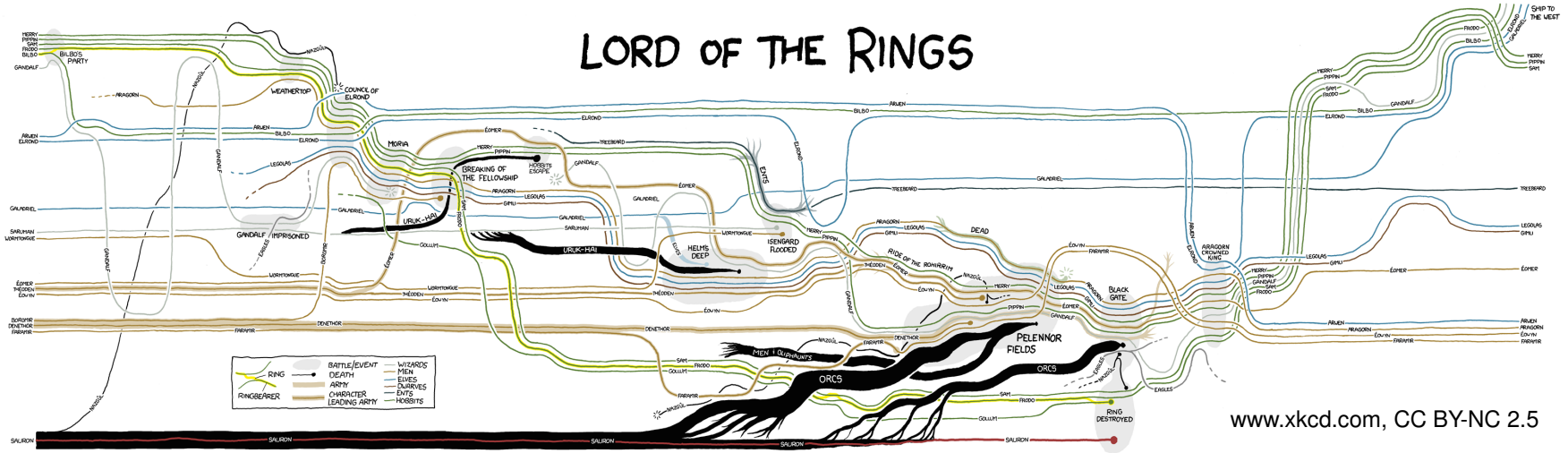
On Minimizing Crossings in Storyline Visualizations

Irina Kostitsyna Martin Nöllenburg Valentin Polishchuk
André Schulz **Darren Strash**

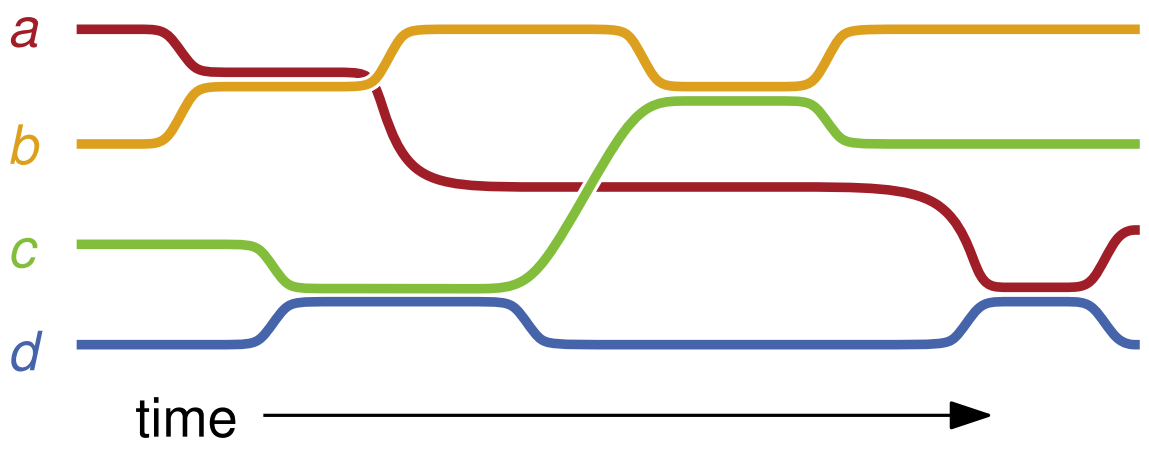
Institute of Theoretical Informatics – Algorithmics



Storyline Visualizations

- Input: A **story** (e.g., movie, play, etc.): set of *n* characters and their interactions over time (*m meetings*)
- Output: Visualization of character interactions

- *x*-axis $\rightarrow$ time
- Characters $\rightarrow$ curves monotone w.r.t time (no time travel)
- Curves converge during an interaction, and diverge otherwise



I. Kostitsyna, M. Nöllenburg, V. Polishchuk, A. Schulz, *D. Strash*:
On Minimizing Crossings in Storyline Visualizations

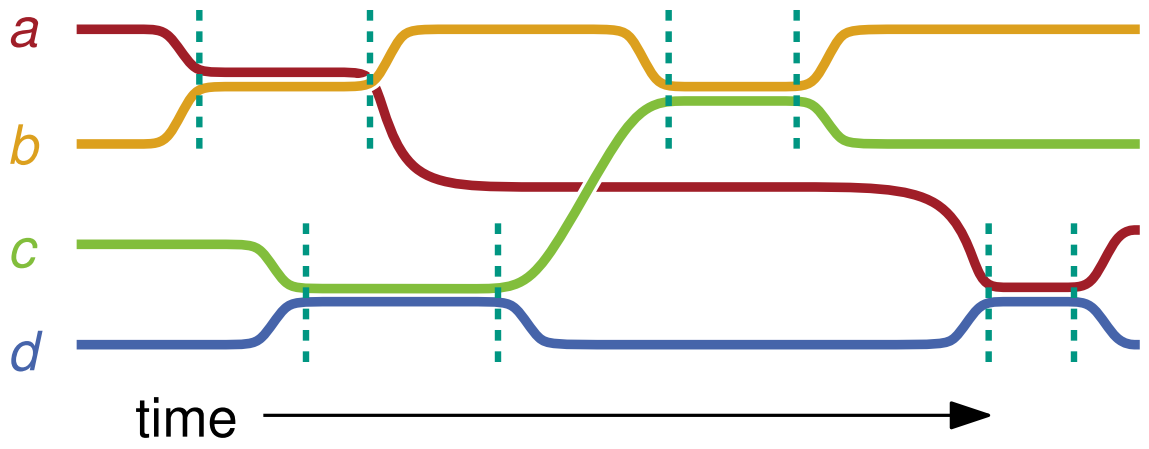
Institute of Theoretical Informatics
Algorithmics

Storyline Visualizations

- Input: A **story** (e.g., movie, play, etc.): set of *n* characters and their interactions over time (*m meetings*)
- Output: Visualization of character interactions

- *x*-axis $\rightarrow$ time
- Characters $\rightarrow$ curves monotone w.r.t time (no time travel)
- Curves converge during an interaction, and diverge otherwise

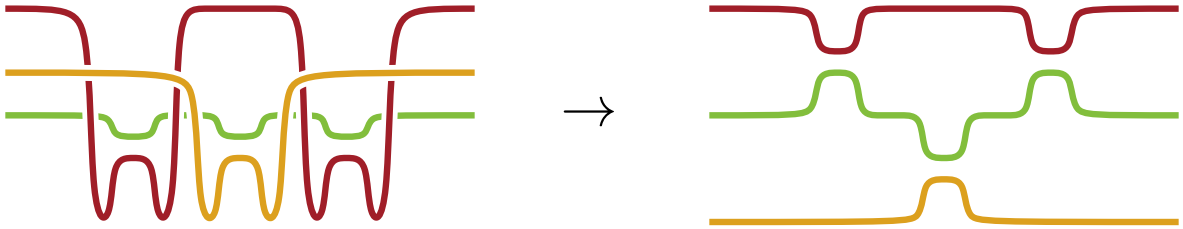
Meetings have start and end times



I. Kostitsyna, M. Nöllenburg, V. Polishchuk, A. Schulz, *D. Strash*:
On Minimizing Crossings in Storyline Visualizations
Institute of Theoretical Informatics
Algorithmics

Previous Results

- Draw pretty pictures → **minimize crossings** between curves
- **NP-hard** in general → reduction from BIPARTITE CROSSING NUMBER



In practice:

- Layered graph drawing → try permutations of curve ordering [Sugiyama et al. '81]
- Heuristics to minimize crossings, wiggles, and gaps [Tanahashi et al. '12, Muelder et al. '13]

Towards a Theoretical Understanding of Storylines

- Almost no existing theoretical results!
- Many interesting questions...
- Among them:

Can we bound the number of crossings?

Fixed-parameter tractable (FPT) for realistic inputs?

Towards a Theoretical Understanding of Storylines

- Almost no existing theoretical results!
- Many interesting questions...
- Among them:

Can we bound the number of crossings? **Yes!**

We show: matching upper and lower bounds for a special case

Fixed-parameter tractable (FPT) for realistic inputs? **Yes!**

We show: $\rightarrow$ FPT on # characters k

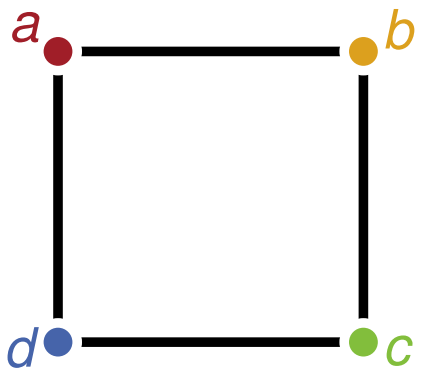
Pairwise One-Time Meetings

- We consider a special case:
  - meetings are restricted to **two characters**
  - these characters meet **only once**

Event graph: characters $\rightarrow$ vertices, meetings $\rightarrow$ edges

pairwise one-time meetings

event graph



I. Kostitsyna, M. Nöllenburg, V. Polishchuk, A. Schulz, *D. Strash*:
On Minimizing Crossings in Storyline Visualizations

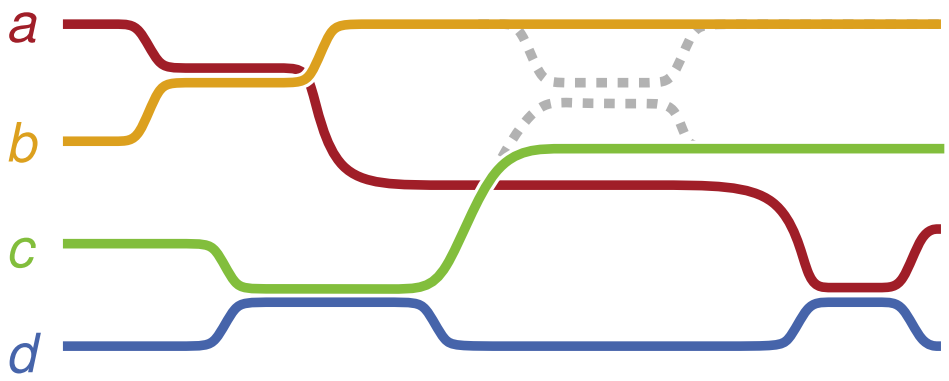
Institute of Theoretical Informatics
Algorithmics

Pairwise One-Time Meetings

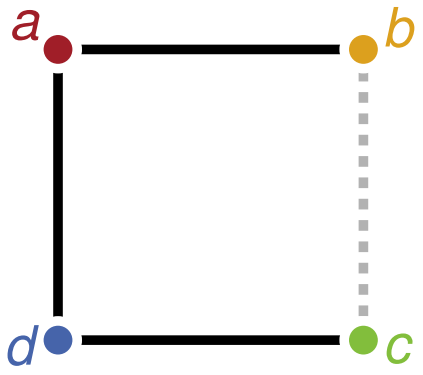
- We consider a special case:
  - meetings are restricted to **two characters**
  - these characters meet **only once**

Event graph: characters $\rightarrow$ vertices, meetings $\rightarrow$ edges

pairwise one-time meetings



event graph



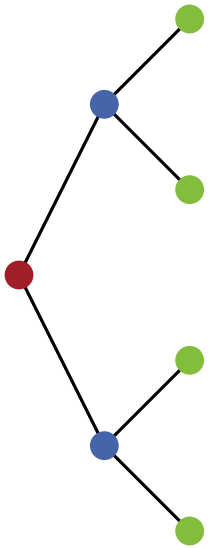
We further restrict to the case where the event graph is a tree.

I. Kostitsyna, M. Nöllenburg, V. Polishchuk, A. Schulz, *D. Strash*:

Institute of Theoretical Informatics
Algorithmics

Algorithm for $O(n \log n)$ Crossings

- Intuition: Full binary tree can be drawn with $O(n \log n)$ crossings
- Achieve the same bound for arbitrary trees using a *heavy path decomposition*

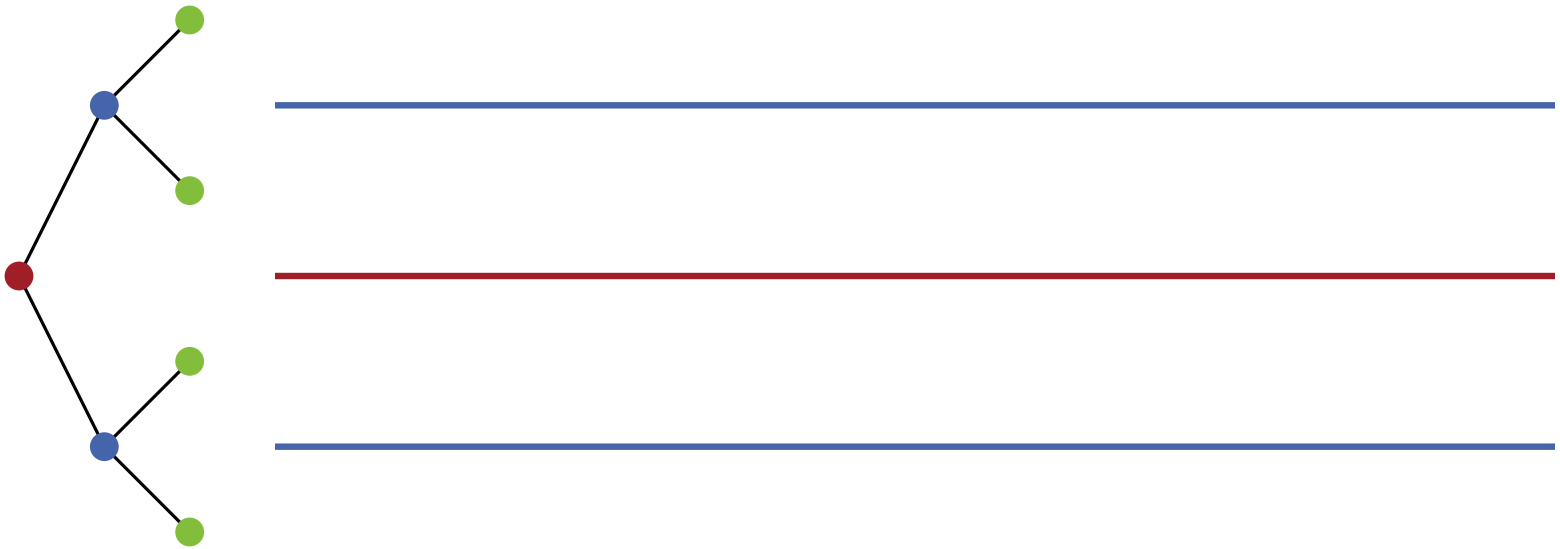


Algorithm for $O(n \log n)$ Crossings

- Intuition: Full binary tree can be drawn with $O(n \log n)$ crossings
- Achieve the same bound for arbitrary trees using a *heavy path decomposition*

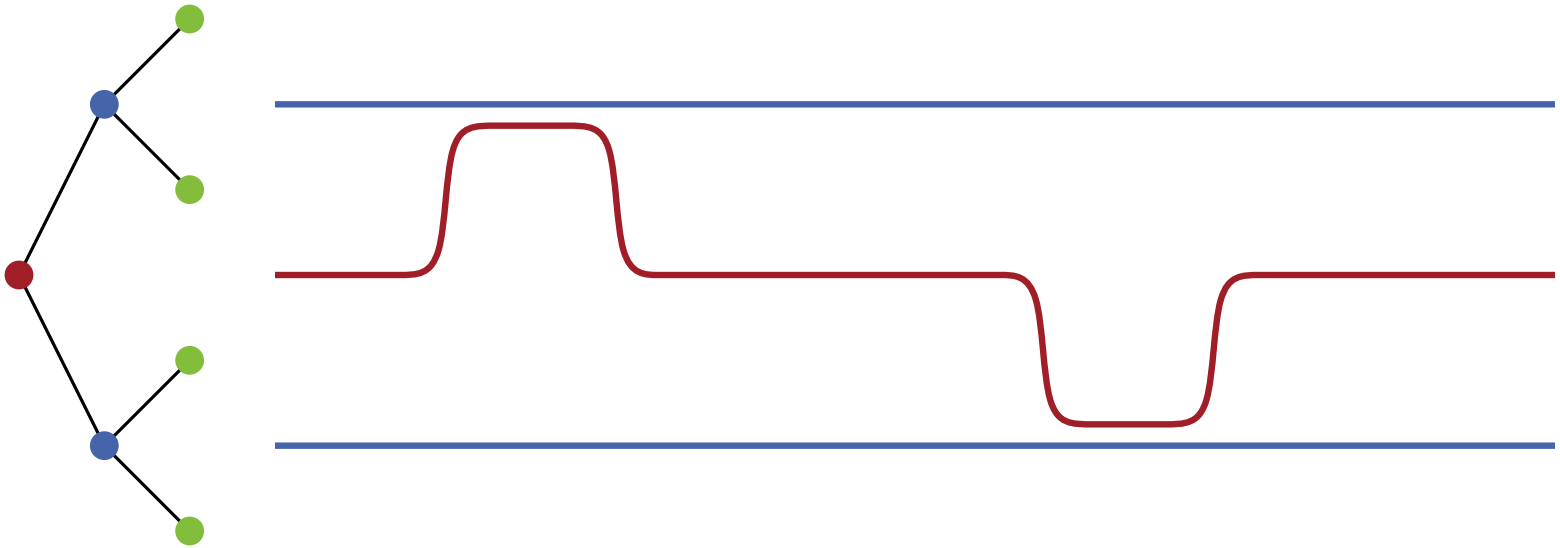
Algorithm for $O(n \log n)$ Crossings

- Intuition: Full binary tree can be drawn with $O(n \log n)$ crossings
- Achieve the same bound for arbitrary trees using a *heavy path decomposition*



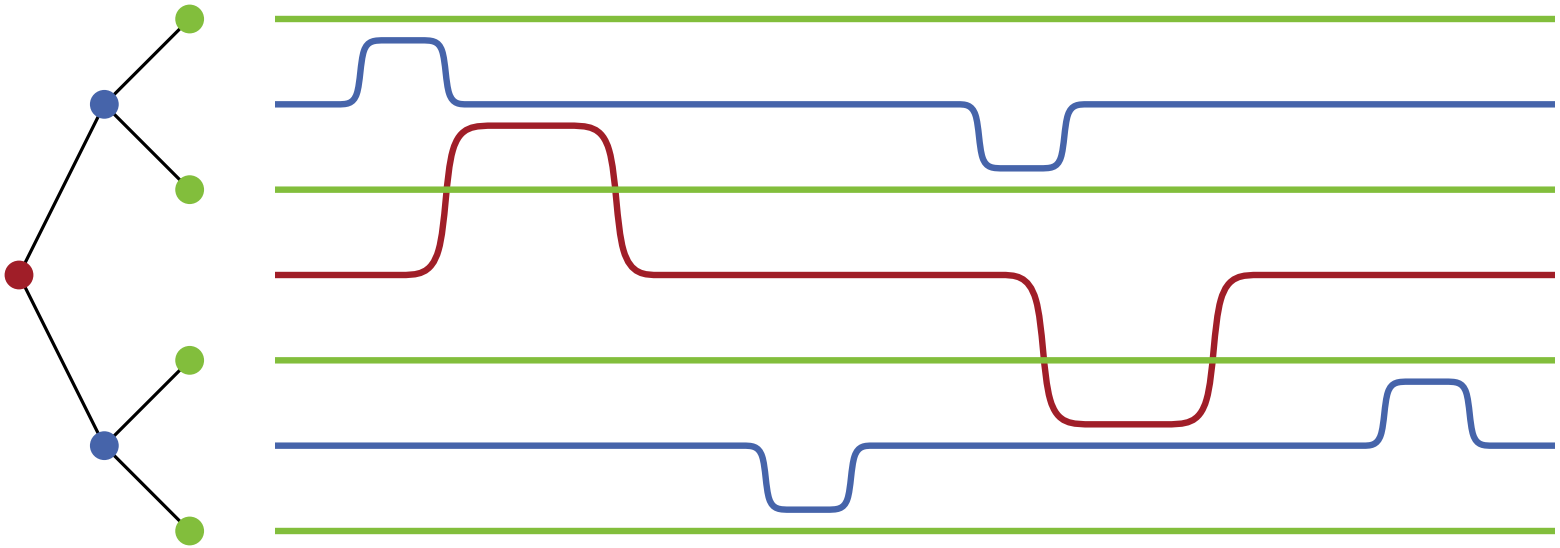
Algorithm for $O(n \log n)$ Crossings

- Intuition: Full binary tree can be drawn with $O(n \log n)$ crossings
- Achieve the same bound for arbitrary trees using a *heavy path decomposition*



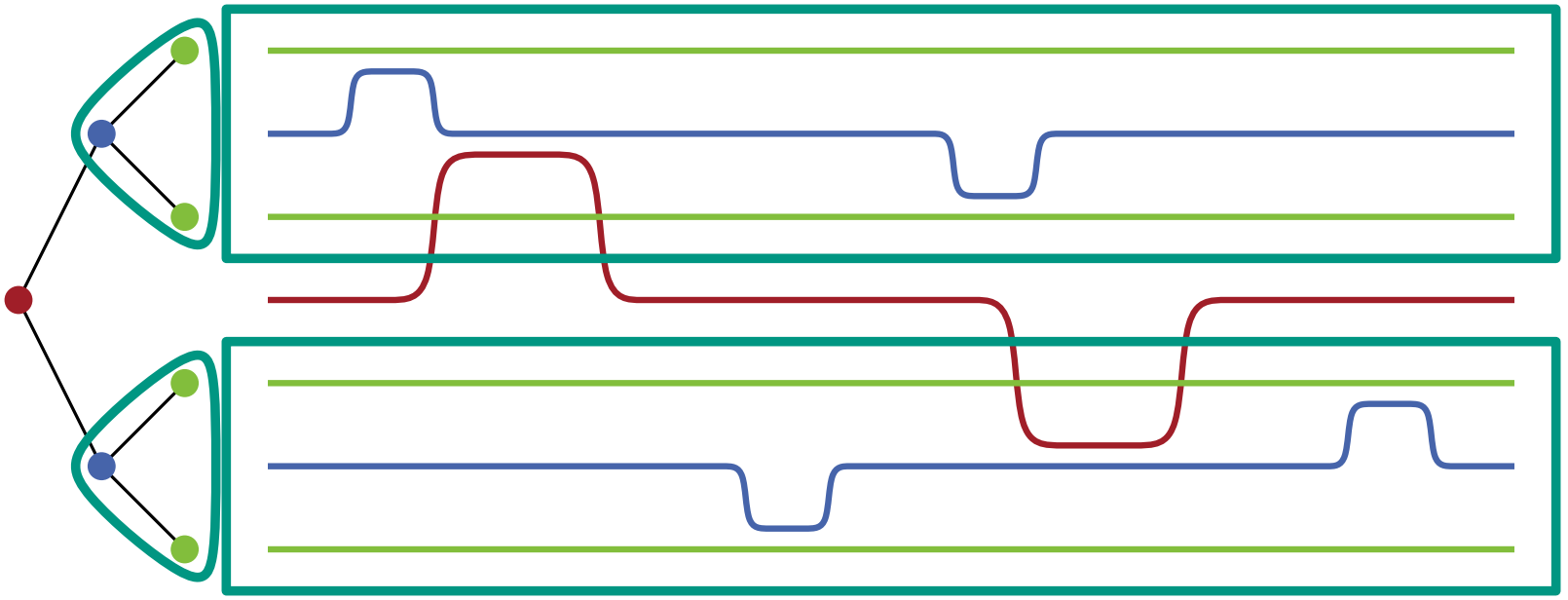
Algorithm for $O(n \log n)$ Crossings

- Intuition: Full binary tree can be drawn with $O(n \log n)$ crossings
- Achieve the same bound for arbitrary trees using a *heavy path decomposition*



Algorithm for $O(n \log n)$ Crossings

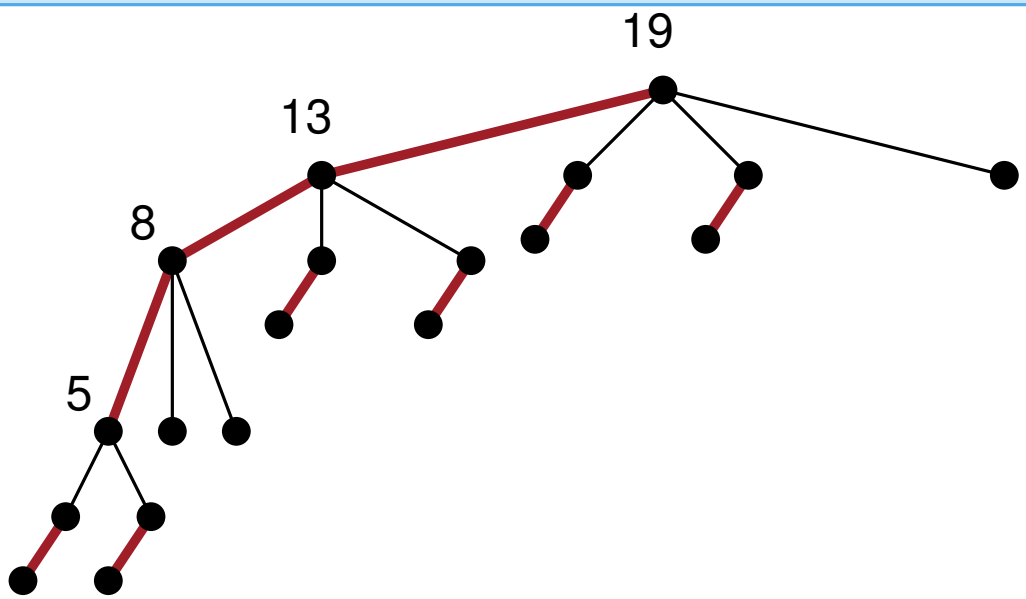
- Intuition: Full binary tree can be drawn with $O(n \log n)$ crossings
- Achieve the same bound for arbitrary trees using a *heavy path decomposition*



Observation: Build bottom-up $\rightarrow$ draw subtree and connect with root.

Heavy-Path Decomposition

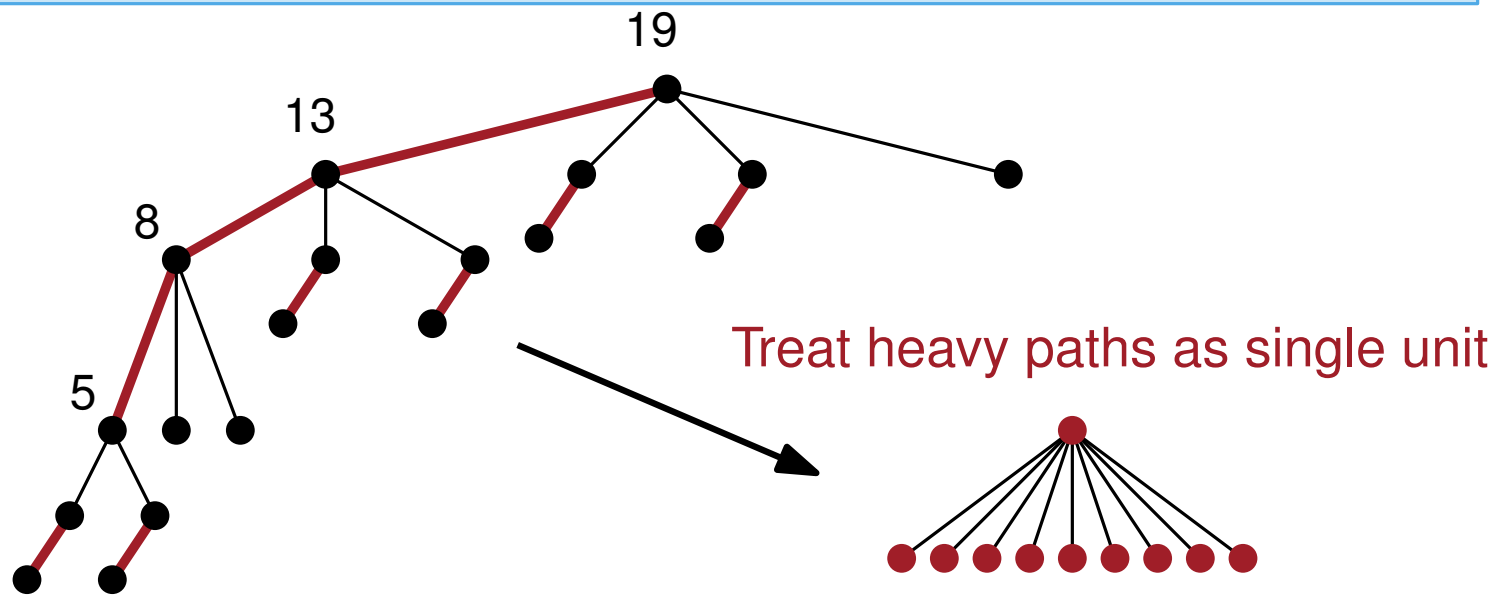
- Heavy edge := $|\text{child subtree}| > 1/2|\text{parent subtree}|$
- Light edge := otherwise
- Key property: $O(\log n)$ light edges on any root-leaf path



I. Kostitsyna, M. Nöllenburg, V. Polishchuk, A. Schulz, *D. Strash*:
On Minimizing Crossings in Storyline Visualizations
Institute of Theoretical Informatics
Algorithmics

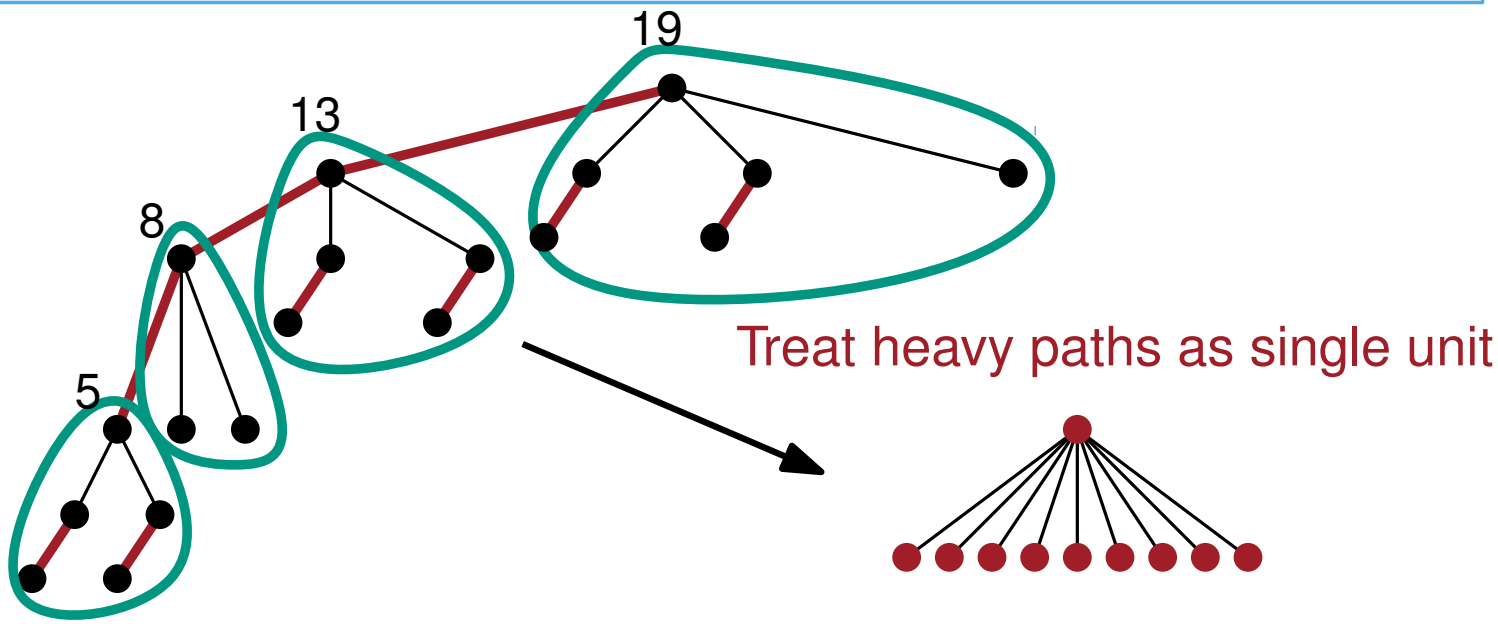
Heavy-Path Decomposition

- Heavy edge := $|\text{child subtree}| > 1/2|\text{parent subtree}|$
- Light edge := otherwise
- Key property: $O(\log n)$ light edges on any root-leaf path



Heavy-Path Decomposition

- Heavy edge := $|\text{child subtree}| > 1/2|\text{parent subtree}|$
- Light edge := otherwise
- Key property: $O(\log n)$ light edges on any root-leaf path



Key idea: Draw light subtrees, then connect roots

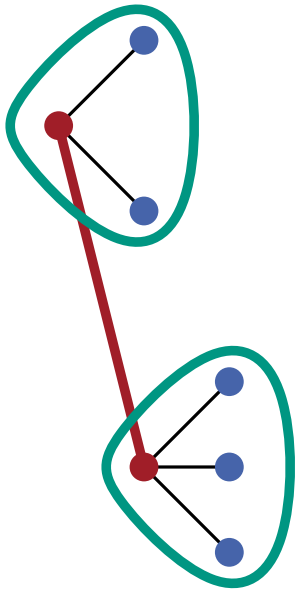
I. Kostitsyna, M. Nöllenburg, V. Polishchuk, A. Schulz, *D. Strash*:

Institute of Theoretical Informatics
Algorithmics

Drawing Tree Event Graphs

Karlsruhe Institute of Technology

Draw each light subtree in an axis-aligned rectangle



Drawing Tree Event Graphs

Draw each light subtree in an axis-aligned rectangle

Order light children vertically by start time of meeting with root

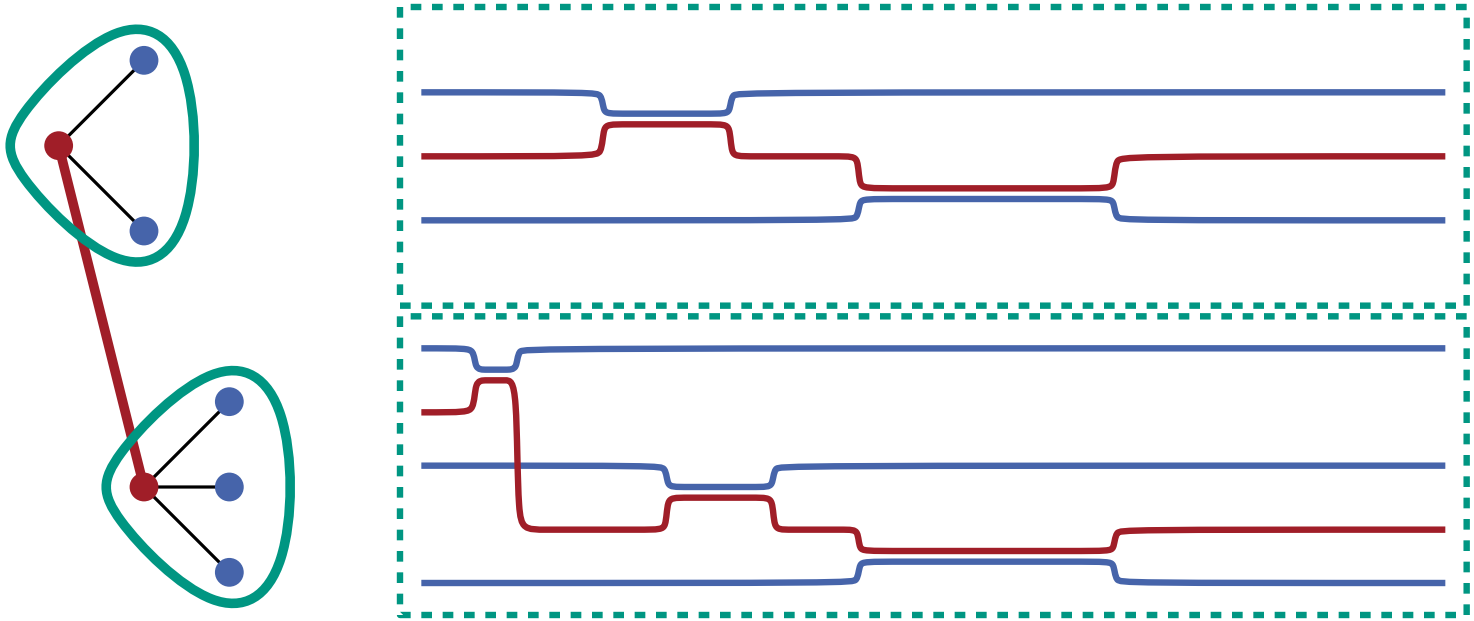
I. Kostitsyna, M. Nöllenburg, V. Polishchuk, A. Schulz, *D. Strash*:
On Minimizing Crossings in Storyline Visualizations

Institute of Theoretical Informatics
Algorithmics

Drawing Tree Event Graphs

Draw each light subtree in an axis-aligned rectangle

Order light children vertically by start time of meeting with root



I. Kostitsyna, M. Nöllenburg, V. Polishchuk, A. Schulz, *D. Strash*:
On Minimizing Crossings in Storyline Visualizations

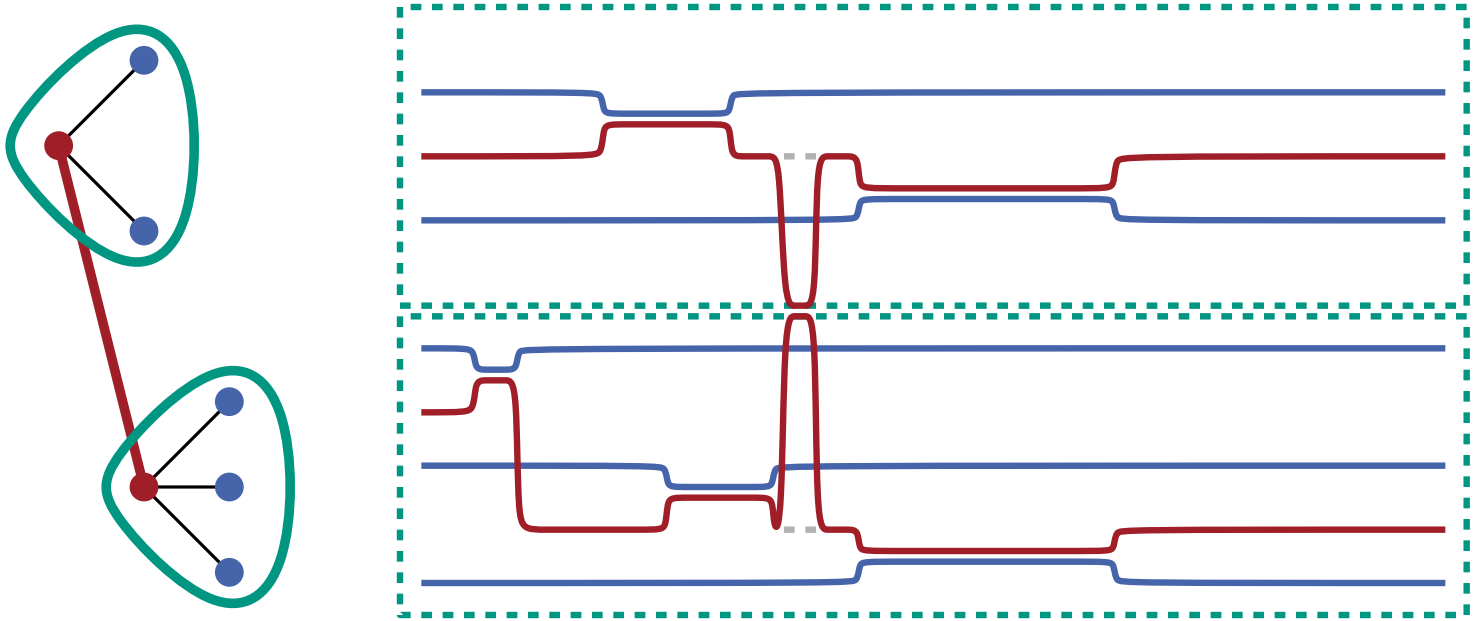
Institute of Theoretical Informatics
Algorithmics

Drawing Tree Event Graphs

Draw each light subtree in an axis-aligned rectangle

Order light children vertically by start time of meeting with root

Introduce “detours”: connect roots on heavy path



I. Kostitsyna, M. Nöllenburg, V. Polishchuk, A. Schulz, *D. Strash*:
On Minimizing Crossings in Storyline Visualizations

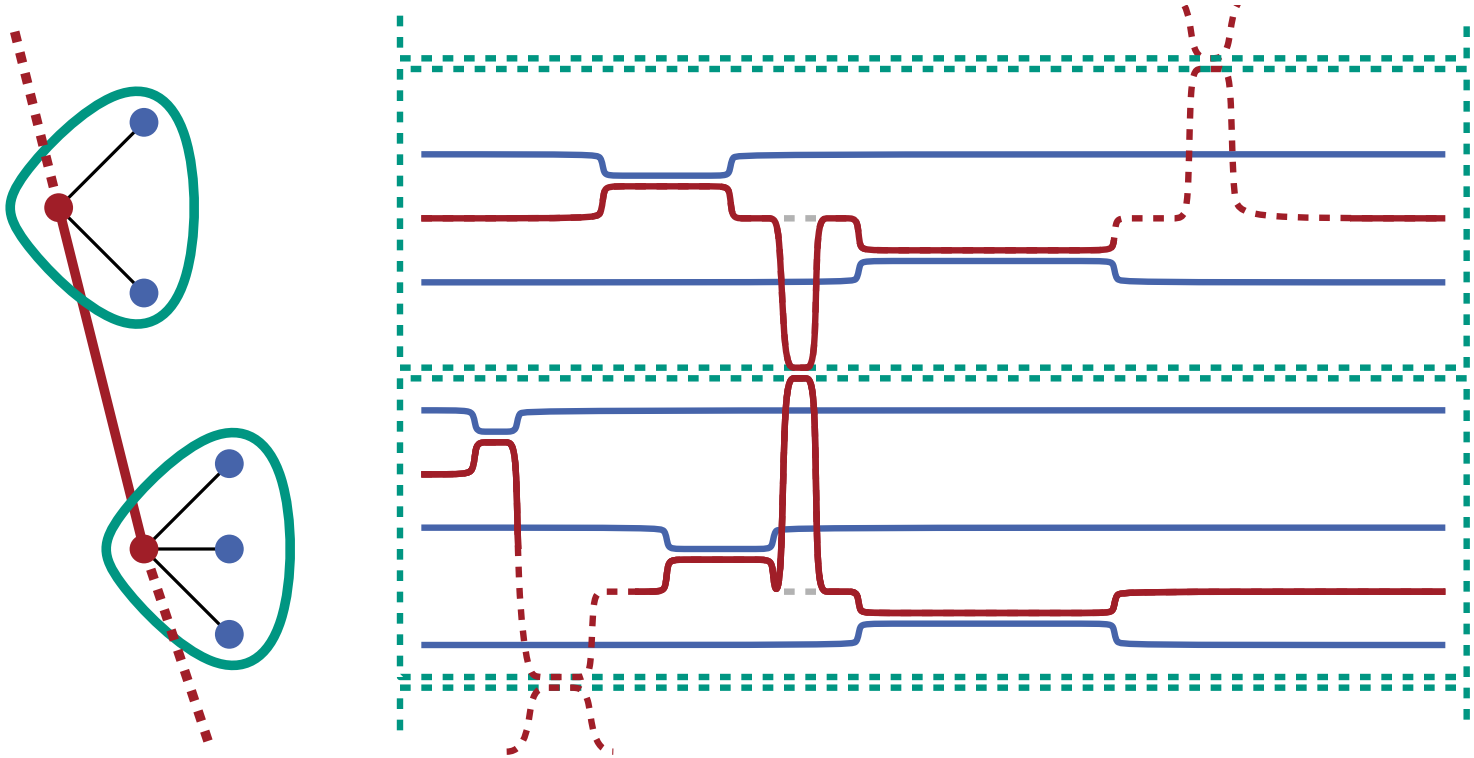
Institute of Theoretical Informatics
Algorithmics

Drawing Tree Event Graphs

Draw each light subtree in an axis-aligned rectangle

Order light children vertically by start time of meeting with root

Introduce “detours”: connect roots on heavy path



I. Kostitsyna, M. Nöllenburg, V. Polishchuk, A. Schulz, *D. Strash*:
On Minimizing Crossings in Storyline Visualizations

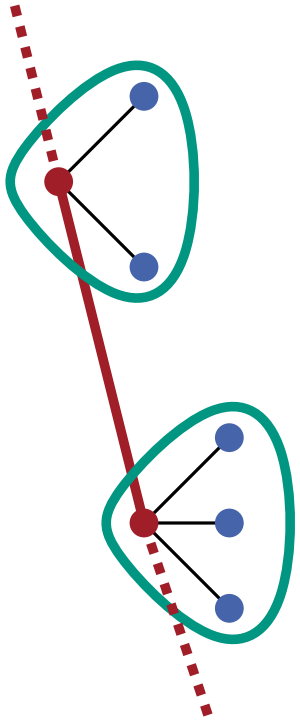
Institute of Theoretical Informatics
Algorithmics

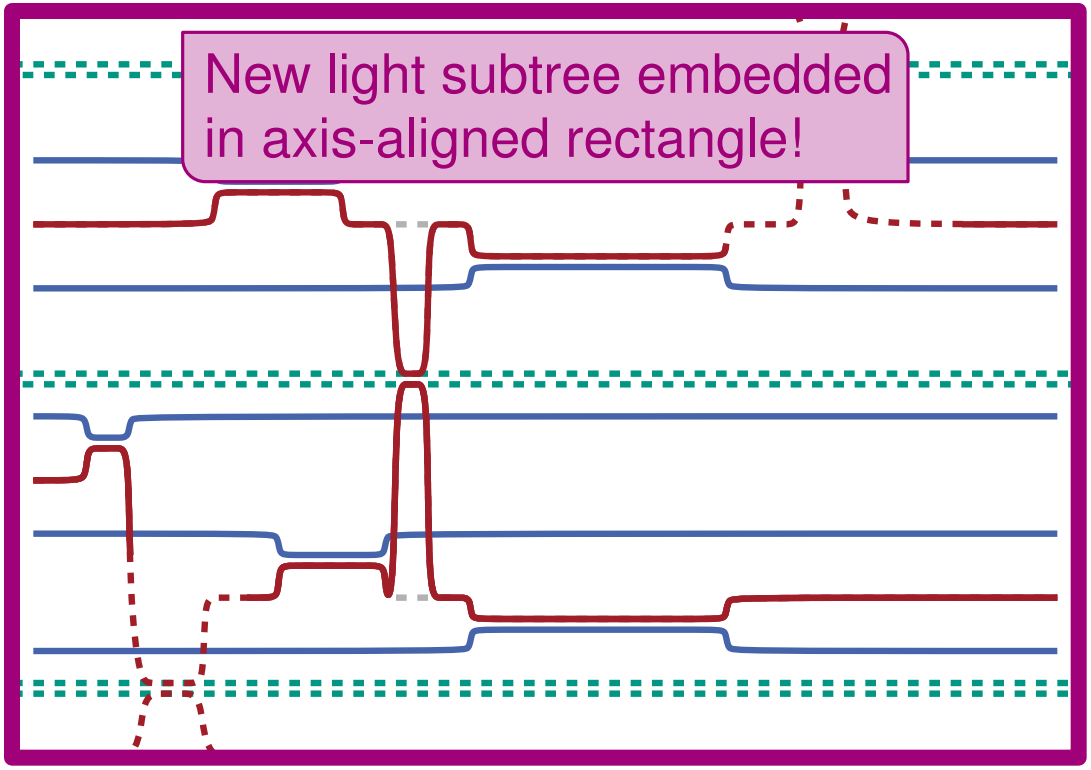
Drawing Tree Event Graphs

Draw each light subtree in an axis-aligned rectangle

Order light children vertically by start time of meeting with root

Introduce “detours”: connect roots on heavy path



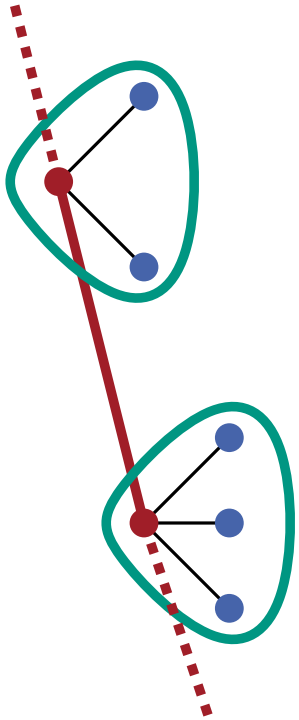


Drawing Tree Event Graphs

Draw each light subtree in an axis-aligned rectangle

Order light children vertically by start time of meeting with root

Introduce “detours”: connect roots on heavy path



New light subtree embedded in axis-aligned rectangle!

Recurrence:

$$N(T) \leq \sum_{\text{light subtrees } L} N(L) + 5n$$

light subtree crossings — root crossings

$$\rightarrow N(T) = O(n \log n)$$

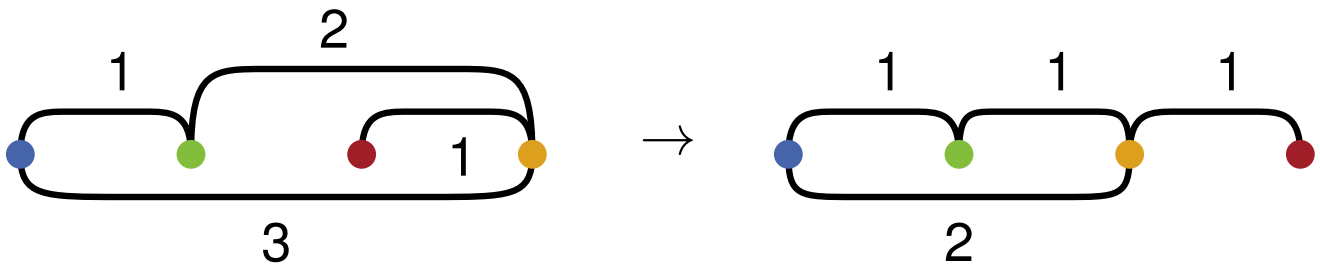
depth of recurrence is $O(\log n)$

I. Kostitsyna, M. Nöllenburg, V. Polishchuk, A. Schulz, *D. Strash*:
On Minimizing Crossings in Storyline Visualizations

Institute of Theoretical Informatics
Algorithmics

Lower Bound for All Inputs

- Use length L^* of *optimal linear ordering* of a graph
- Embed vertices on the line (unique integers) to minimize total edge length

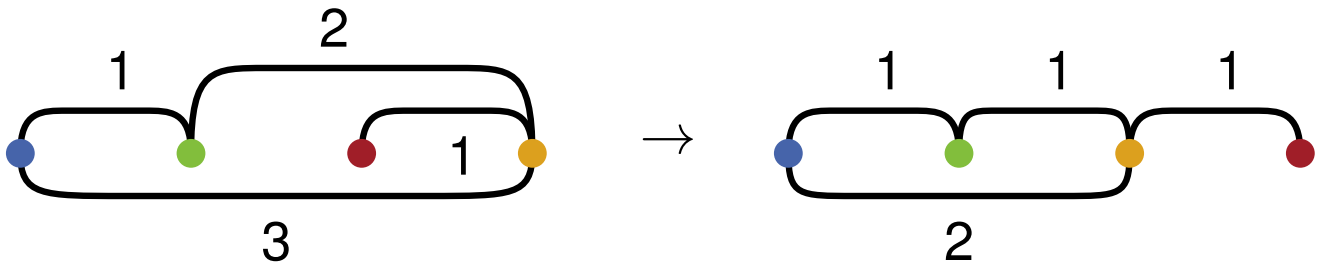


total edge length = 7

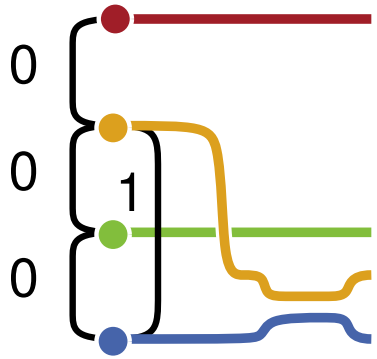
$L^* = 5$

Lower Bound for All Inputs

- Use length L^* of *optimal linear ordering* of a graph
- Embed vertices on the line (unique integers) to minimize total edge length



total edge length = 7 $L^* = 5$



Our problem: Optimal cost $= L^* -$ # edges

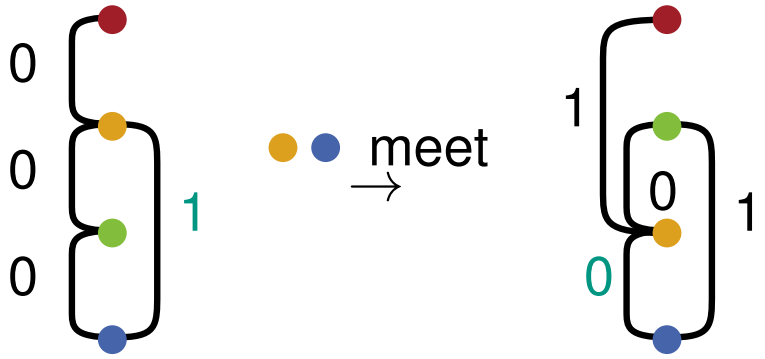
Edge $\rightarrow$ meeting

Crossings for 2 vertices to meet $\rightarrow$ edge length - 1

I. Kostitsyna, M. Nöllenburg, V. Polishchuk, A. Schulz, *D. Strash*:
On Minimizing Crossings in Storyline Visualizations
Institute of Theoretical Informatics
Algorithmics

Lower Bound for All Inputs

Claim: total # of crossings $\geq \frac{L^* - \#\text{edges}}{2\Delta}$



L^* **changes:**

New cost for ● ● to meet $\rightarrow$ 0

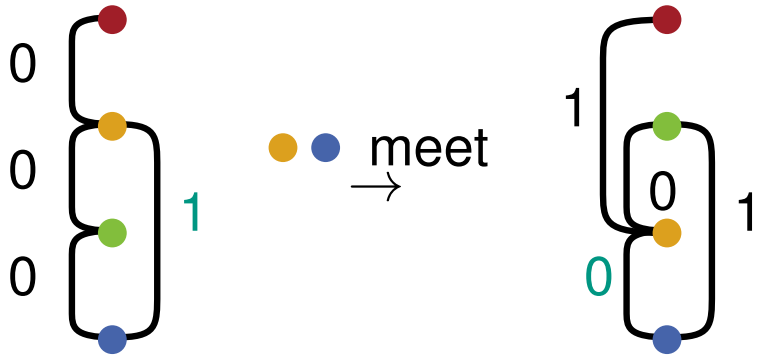
Other costs increase by:
$\leq$ # crossings $\times 2\Delta$

I. Kostitsyna, M. Nöllenburg, V. Polishchuk, A. Schulz, *D. Strash*:
On Minimizing Crossings in Storyline Visualizations

Institute of Theoretical Informatics
Algorithmics

Lower Bound for All Inputs

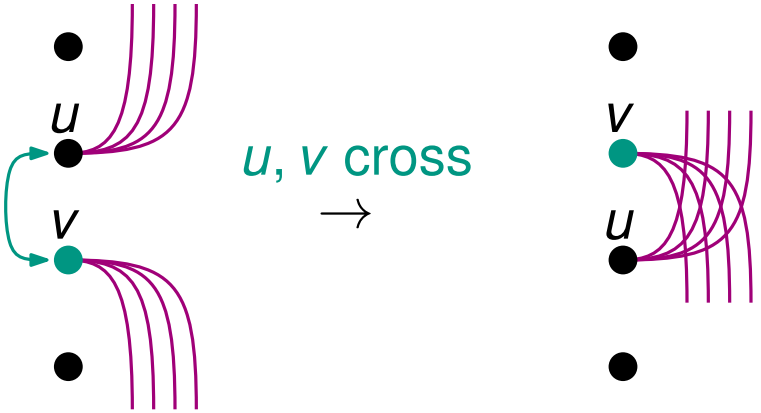
Claim: total # of crossings $\geq \frac{L^* - \#\text{edges}}{2\Delta}$



L^* **changes:**

New cost for ● ● to meet $\rightarrow$ 0

Other costs increase by:
$\leq$ # crossings $\times 2\Delta$



Each edge has cost increase of 1

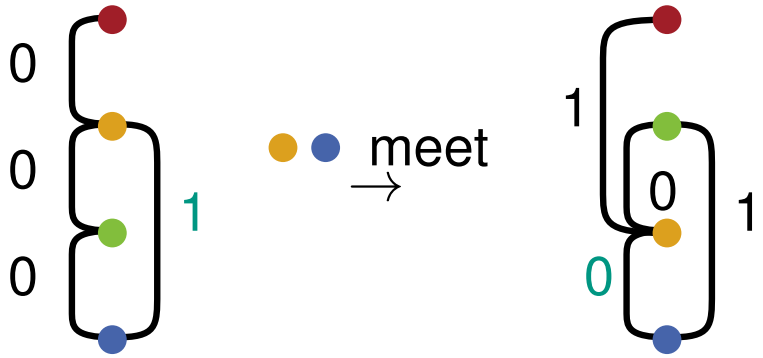
Increase by $\leq d(u) + d(v) \leq 2\Delta$

I. Kostitsyna, M. Nöllenburg, V. Polishchuk, A. Schulz, *D. Strash*:
On Minimizing Crossings in Storyline Visualizations
Institute of Theoretical Informatics
Algorithmics

Lower Bound for All Inputs

Karlsruhe Institute of Technology

Claim: total # of crossings $\geq \frac{L^* - \#\text{edges}}{2\Delta}$



L^* **changes:**

New cost for ● ● to meet $\rightarrow$ 0

Other costs increase by:
$\leq$ # crossings $\times 2\Delta$

---

total # of crossings $\times 2\Delta \geq$ total cost increase

$\geq$ original total cost $\geq L^* -$ #edges

total # of crossings $\geq \frac{L^* - \#\text{edges}}{2\Delta}$

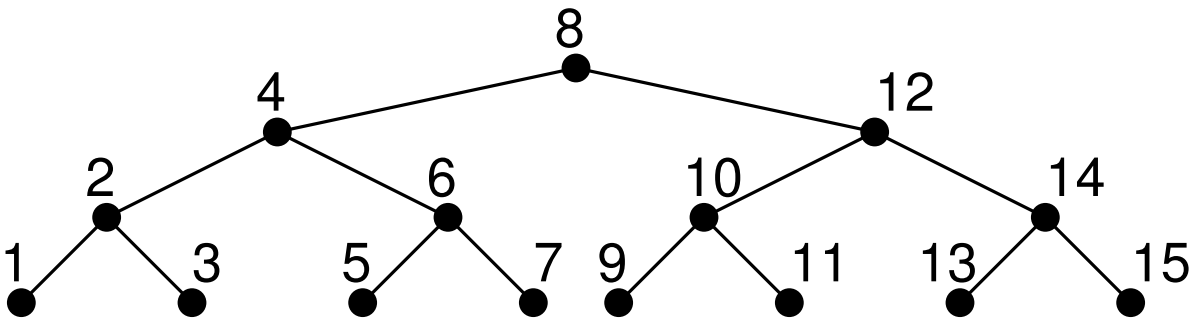
e of 1

$\leq 2\Delta$

Lower Bound for Tree Event Graphs

Trees:

- Optimal linear ordering = minimum valuation
- Minimized for full binary tree



$L^* = \Omega(n \log n)$ [Chung '78], $\Delta = 3$

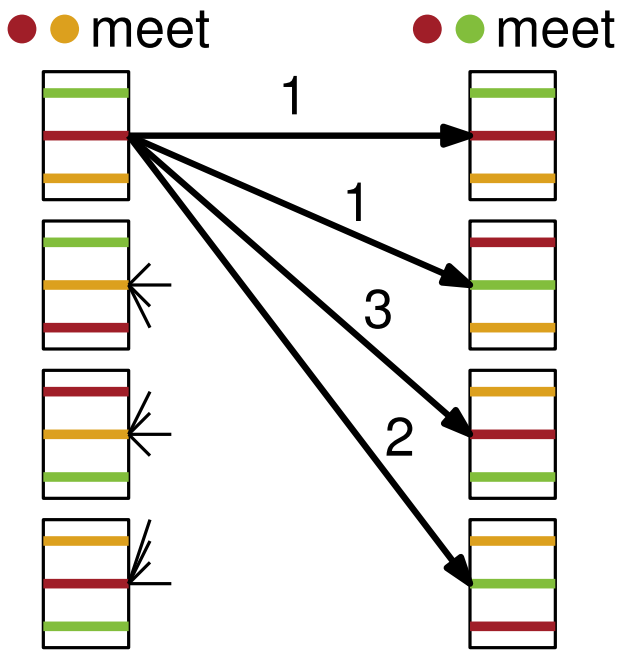
$\#\text{ crossings} = \Omega(\frac{\Omega(n \log n) - m}{2 * 3}) = \Omega(n \log n)$

FPT for All Inputs

We transform to shortest path problem

- vertex $\rightarrow$ valid vertical ordering of curves at meeting start time
- edge $\rightarrow$ transformation between orderings by swaps (weight = min # crossings)

characters

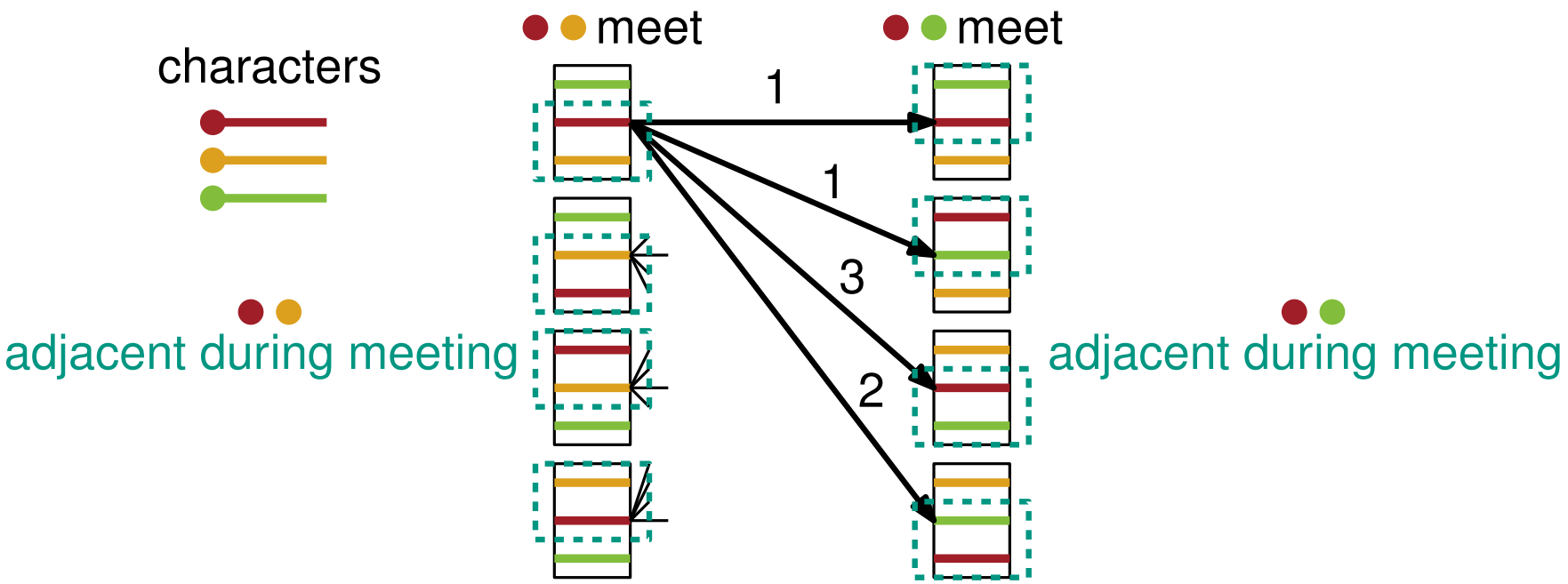


I. Kostitsyna, M. Nöllenburg, V. Polishchuk, A. Schulz, *D. Strash*:
On Minimizing Crossings in Storyline Visualizations
Institute of Theoretical Informatics
Algorithmics

FPT for All Inputs

We transform to shortest path problem

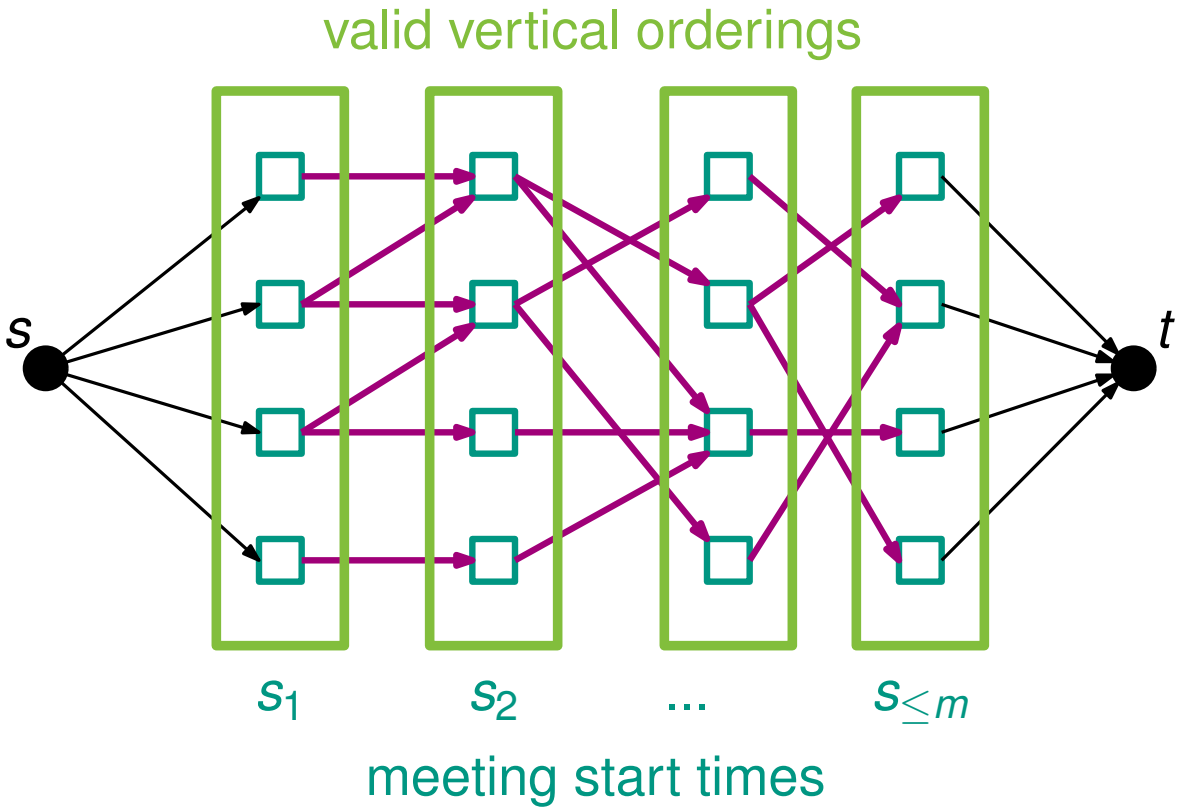
- vertex $\rightarrow$ valid vertical ordering of curves at meeting start time
- edge $\rightarrow$ transformation between orderings by swaps (weight = min # crossings)



I. Kostitsyna, M. Nöllenburg, V. Polishchuk, A. Schulz, *D. Strash*:
On Minimizing Crossings in Storyline Visualizations
Institute of Theoretical Informatics
Algorithmics

FPT for All Inputs

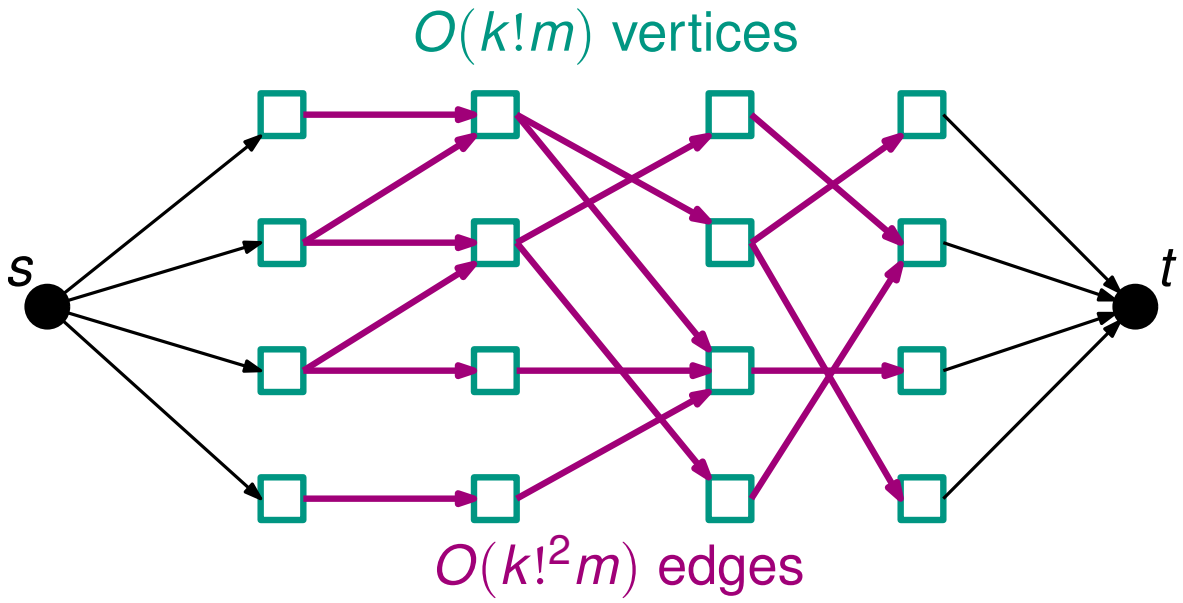
- Evaluate all vertical orderings with one search



FPT for Crossing Minimization

Parameter $\rightarrow k$ characters

- precompute edge weights between all $k!^2$ pairs in time $O(k!^2 k \log k)$ with **merge sort**
- find shortest $s \rightarrow t$ path in **linear time**



Total running time

$O(k!^2 k \log k + k!^2 m)$

Questions?

- Is the FPT algorithm efficient in practice for small k (e.g., $k \leq 6$)?
- Is there a polynomial time exact algorithm for tree event graphs?
- How about other graph classes? (e.g., small arboricity, unicyclic graphs, cactus graphs)
- Are there sparse event graphs that require $\Omega(n^2)$ crossings?
- What about minimizing the number of bends/wiggles?

www.xkcd.com, CC BY-NC 2.5