Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Chair for
INFORMATICS I
Efficient Algorithms and
Knowledge-Based Systems

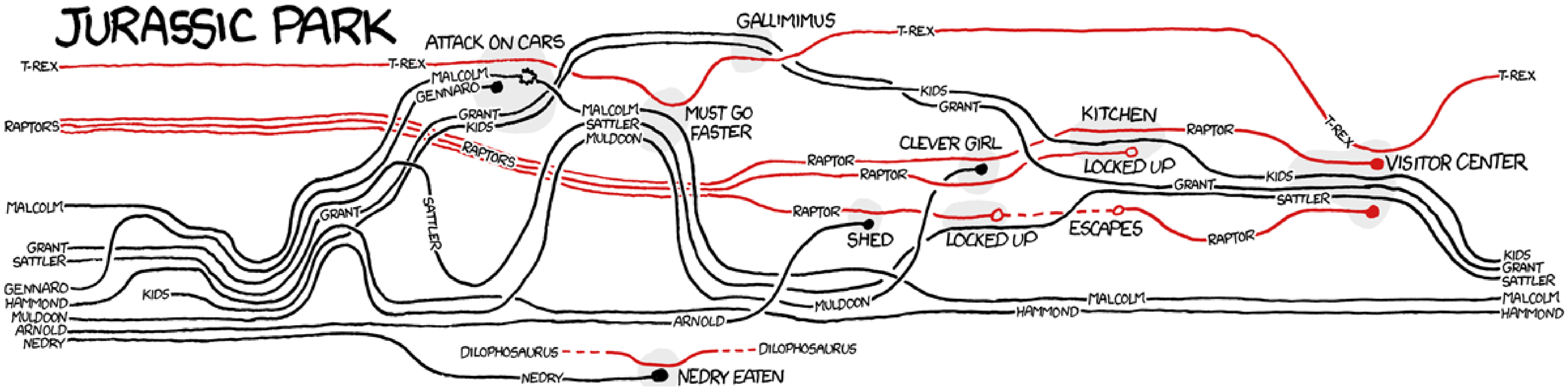
Institute for Informatics

Computing Storyline Visualizations with Few Block Crossings

Thomas C. van Dijk
Peter Markfelder

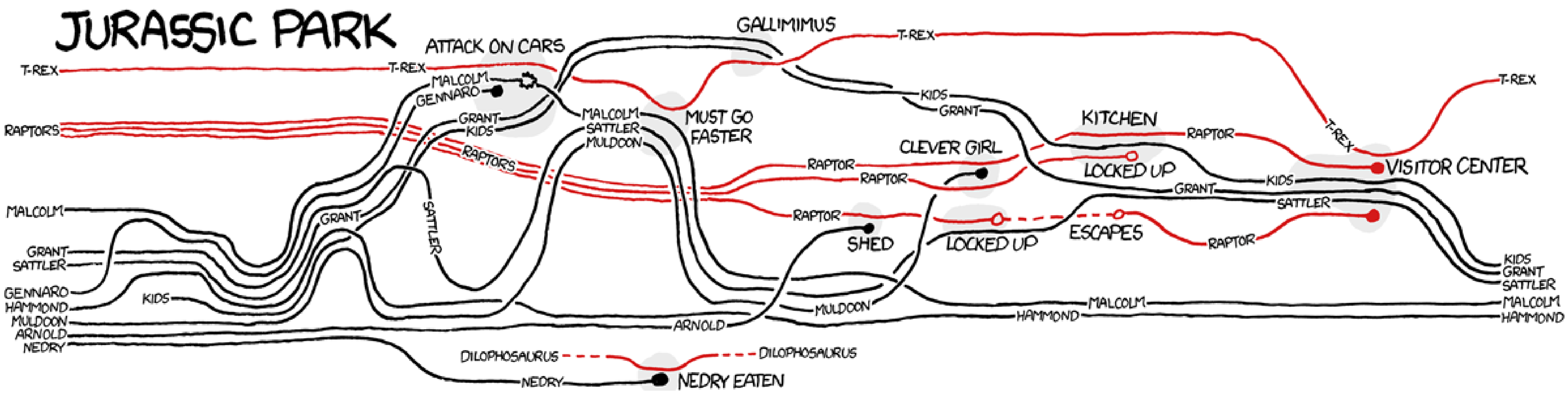
Fabian Lipp
Alexander Wolff

Storyline Visualizations



[https://xkcd.com/657/]

Storyline Visualizations



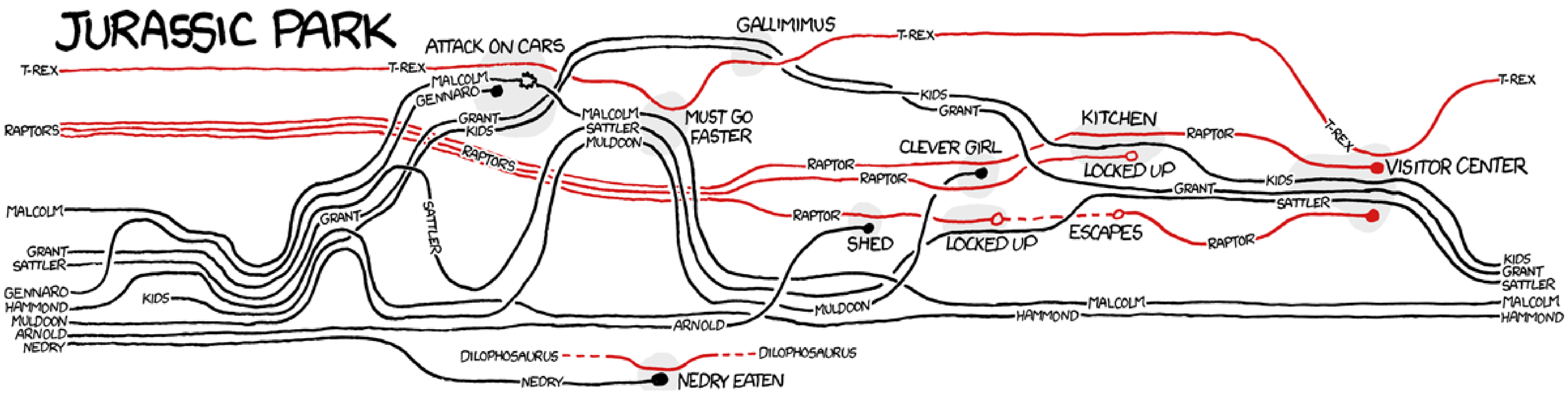
[https://xkcd.com/657/]

Design considerations for storyline drawings:

- Line wiggles
- White space gaps
- Line crossings

[Tanahashi & Ma, TVCG12]

Storyline Visualizations



[https://xkcd.com/657/]

Design considerations for storyline drawings:

- Line wiggles
- White space gaps
- Line crossings

[Tanahashi & Ma, TVCG12]

Related Work: Simple Crossings

Aim: Drawing with a minimum number of crossings

Related Work: Simple Crossings

Aim: Drawing with a minimum number of crossings

[Kostitsyna et al., GD'15]

- NP-hardness
- FPT for #characters
- upper and lower bounds for some cases with pairwise meetings

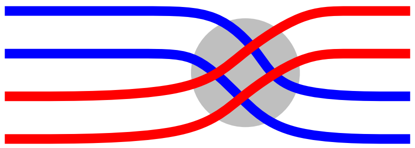
Related Work: Simple Crossings

Aim: Drawing with a minimum number of crossings

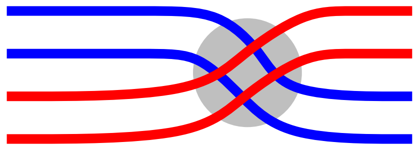
- NP-hardness [Kostitsyna et al., GD'15]
- FPT for #characters
- upper and lower bounds for some cases with pairwise meetings

- ILP solving the problem [Gronemann et al., GD'16]

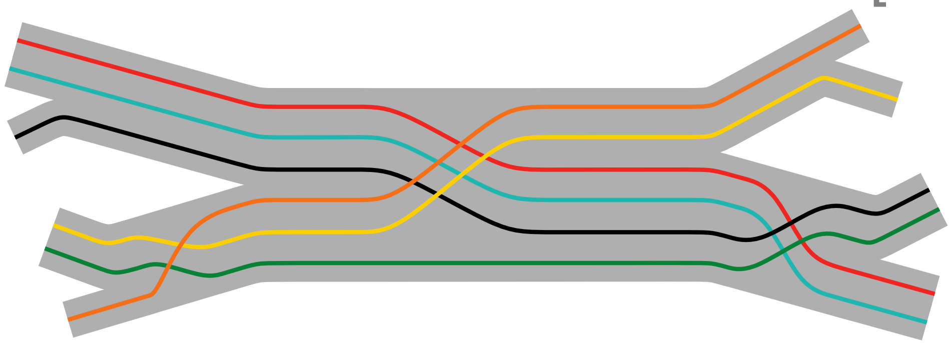
Related Work: Block Crossings



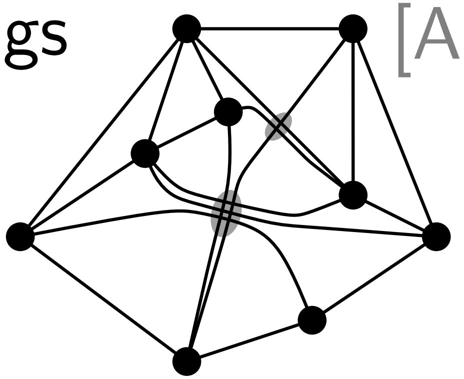
Related Work: Block Crossings



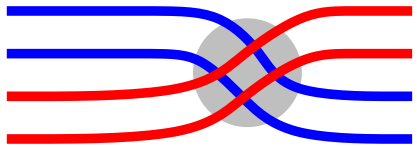
- Block crossings for metro lines [Fink, Pupyrev, Wolff; 2015]



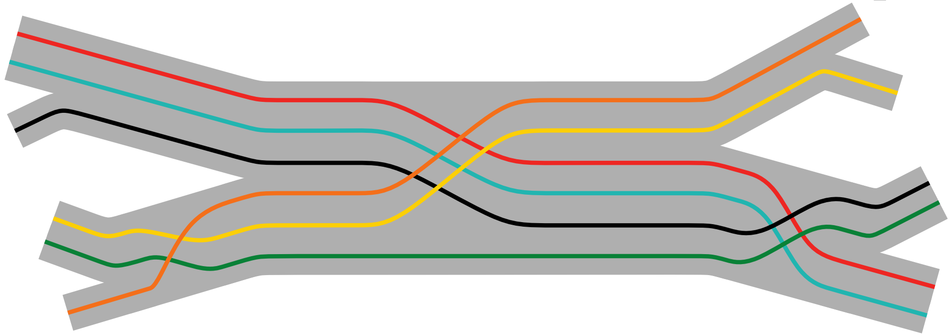
- Bundled Crossings [Alam, Fink, Pupyrev, GD 2016]
[Fink et al., LATIN 2016]



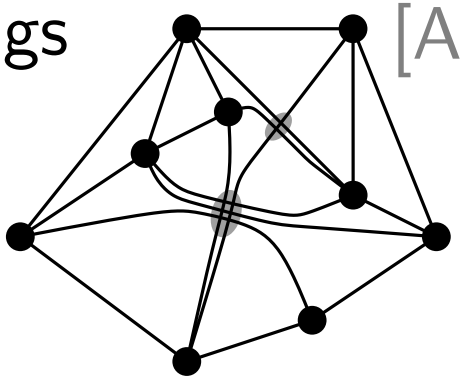
Related Work: Block Crossings



- Block crossings for metro lines [Fink, Pupyrev, Wolff; 2015]

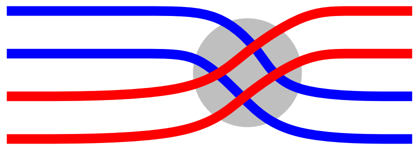


- Bundled Crossings [Alam, Fink, Pupyrev, GD 2016] [Fink et al., LATIN 2016]

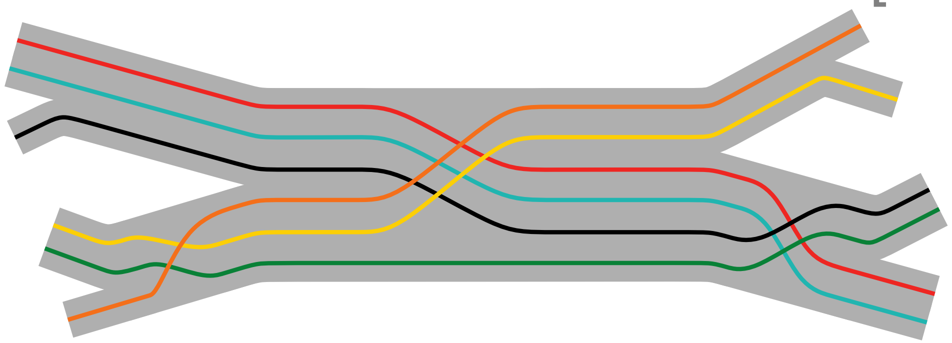


Our Aim: Drawing with a minimum number of *block* crossings

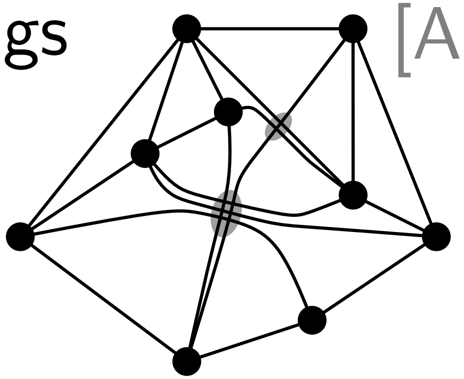
Related Work: Block Crossings



- Block crossings for metro lines [Fink, Pupyrev, Wolff; 2015]



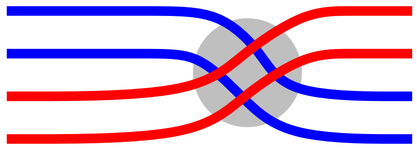
- Bundled Crossings [Alam, Fink, Pupyrev, GD 2016] [Fink et al., LATIN 2016]



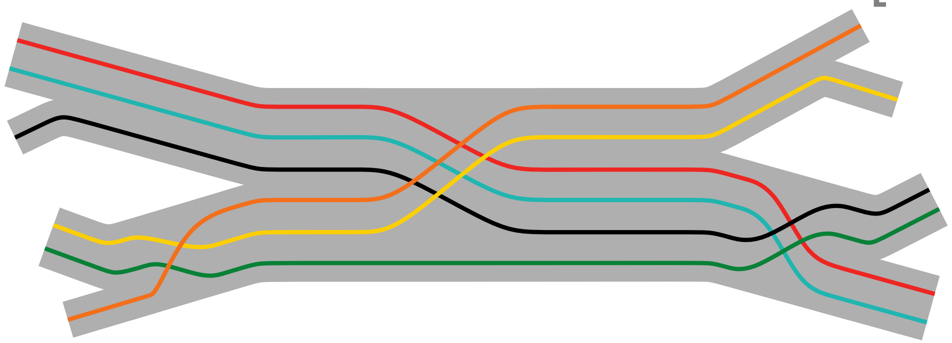
Our Aim: Drawing with a minimum number of *block* crossings

- Storyline Block Crossing Minimization
  - NP-hardness
  - Approximation algorithm
  - FPT [Van Dijk et al., GD 2016]

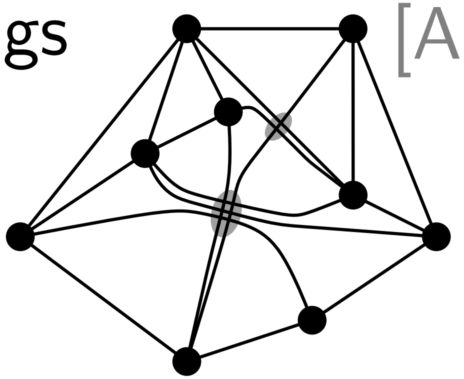
Related Work: Block Crossings



- Block crossings for metro lines [Fink, Pupyrev, Wolff; 2015]



- Bundled Crossings [Alam, Fink, Pupyrev, GD 2016]
  [Fink et al., LATIN 2016]

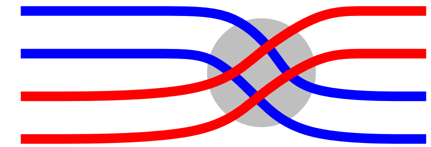


Our Aim: Drawing with a minimum number of *block* crossings

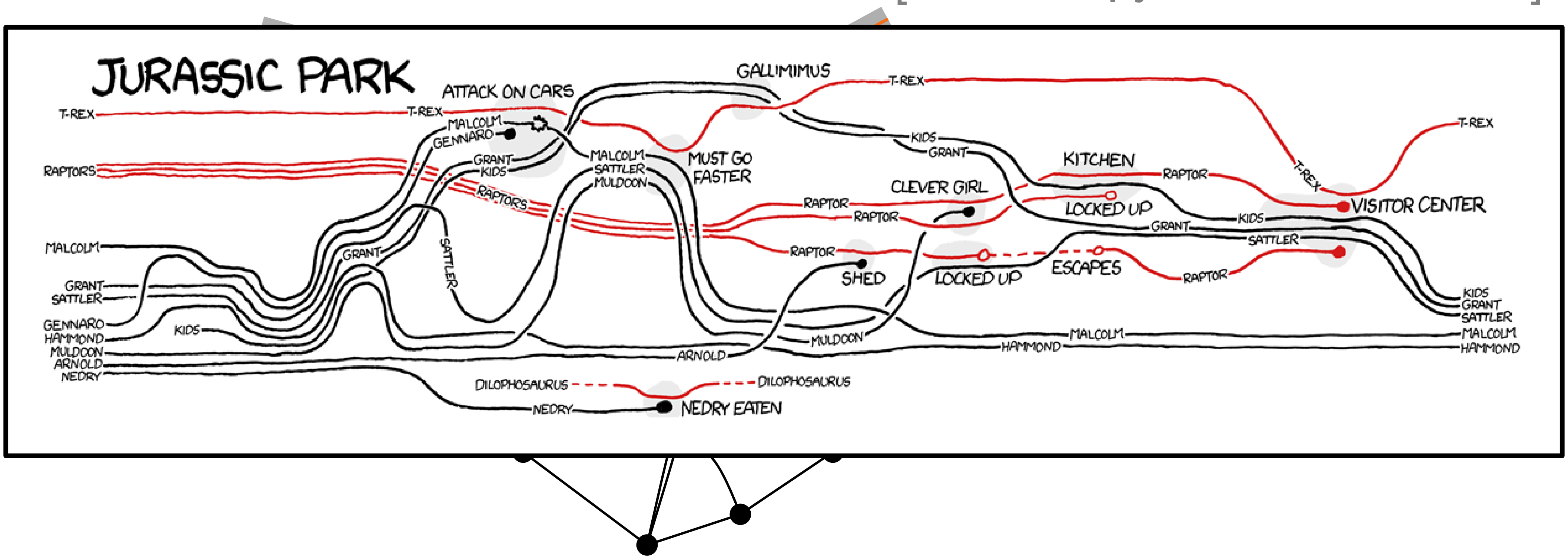
- Storyline Block Crossing Minimization
  - NP-hardness
  - Approximation algorithm
  - FPT [Van Dijk et al., GD 2016]

Now:
Solve SBCM via a SAT formulation

Related Work: Block Crossings



- Block crossings for metro lines [Fink, Pupyrev, Wolff; 2015]

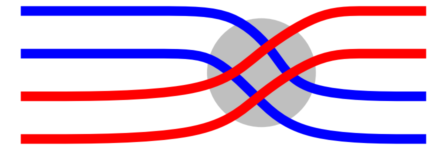


Our Aim: Drawing with a minimum number of *block* crossings

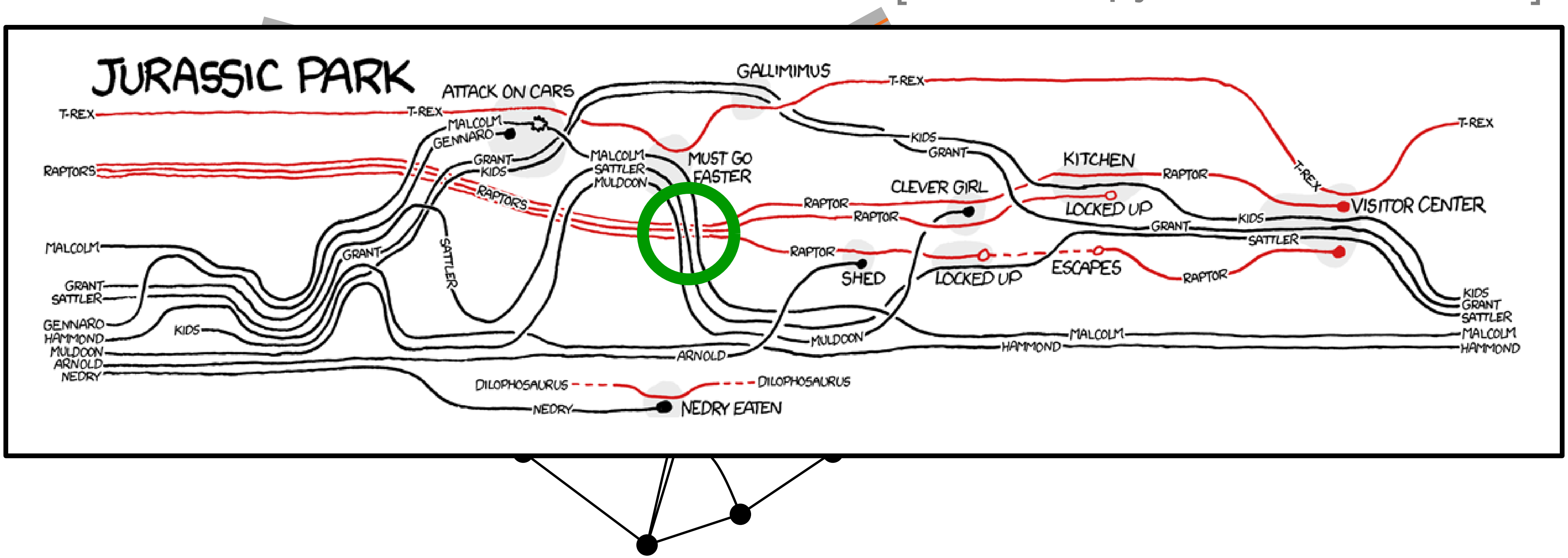
- Storyline Block Crossing Minimization
  - NP-hardness
  - Approximation algorithm
  - FPT [Van Dijk et al., GD 2016]

Now:
Solve SBCM via a SAT formulation

Related Work: Block Crossings



- Block crossings for metro lines [Fink, Pupyrev, Wolff; 2015]



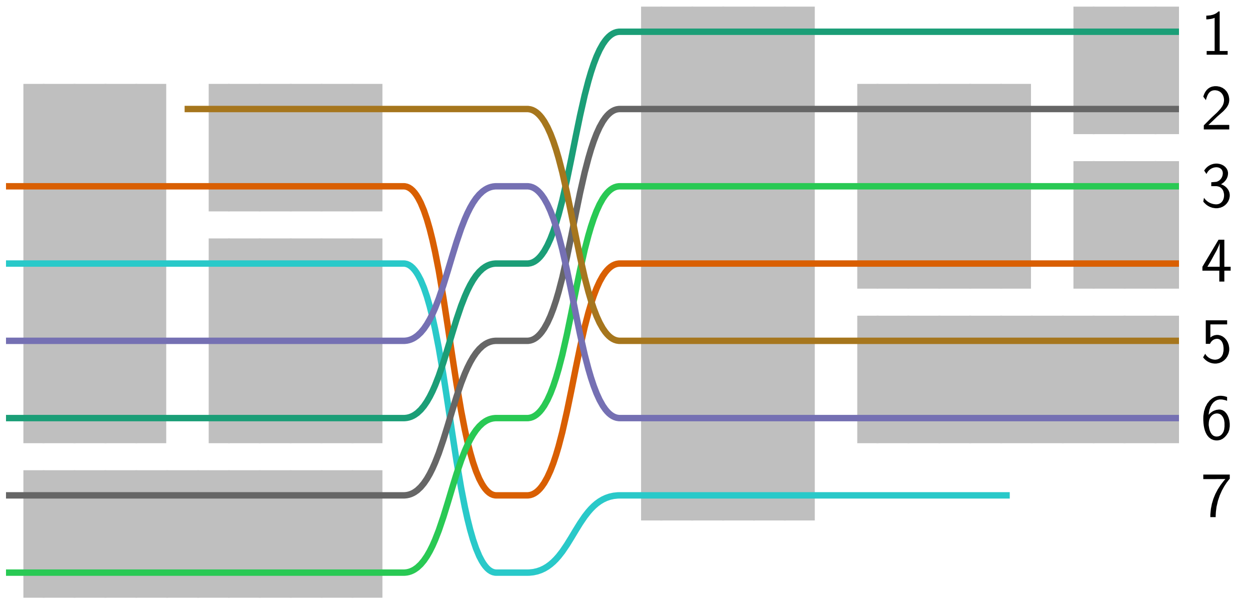
Our Aim: Drawing with a minimum number of *block* crossings

- Storyline Block Crossing Minimization
  - NP-hardness
  - Approximation algorithm
  - FPT [Van Dijk et al., GD 2016]

Now:
Solve SBCM via a SAT formulation

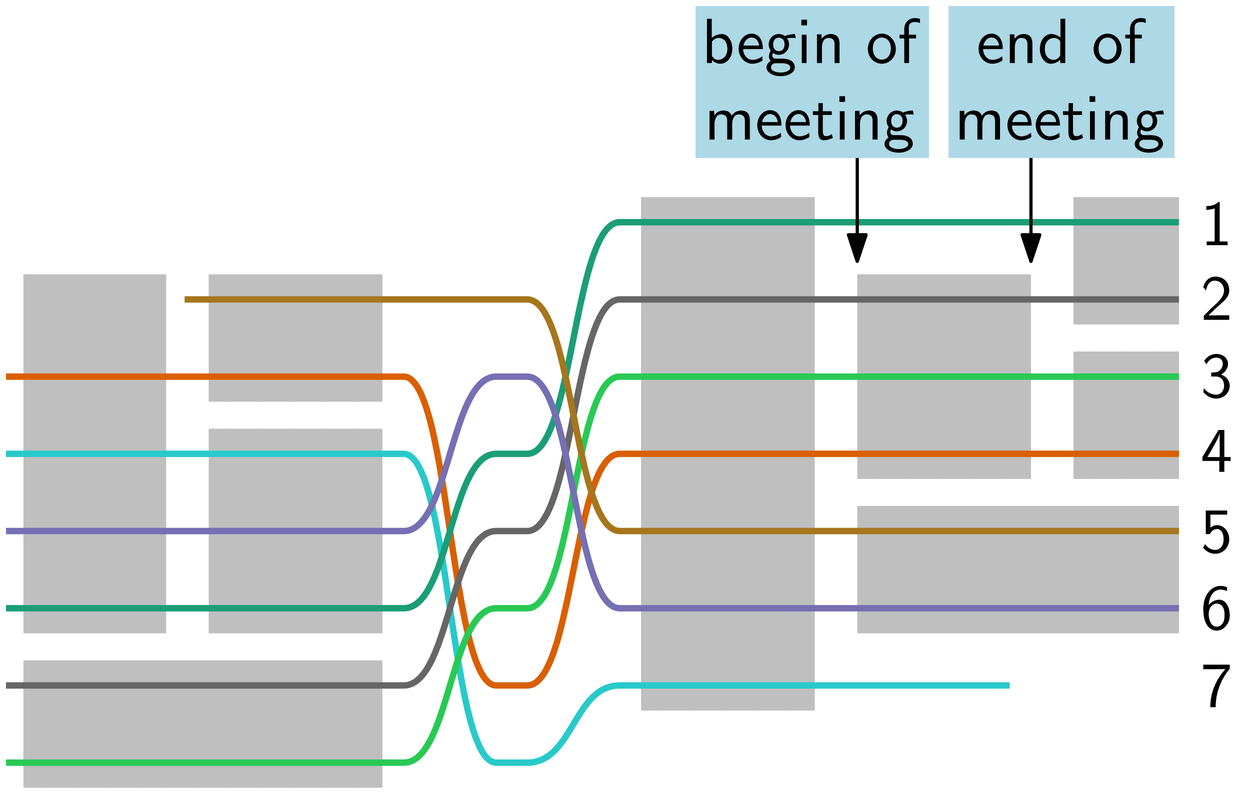
Problem Definition

Meeting: set of characters, start and end time



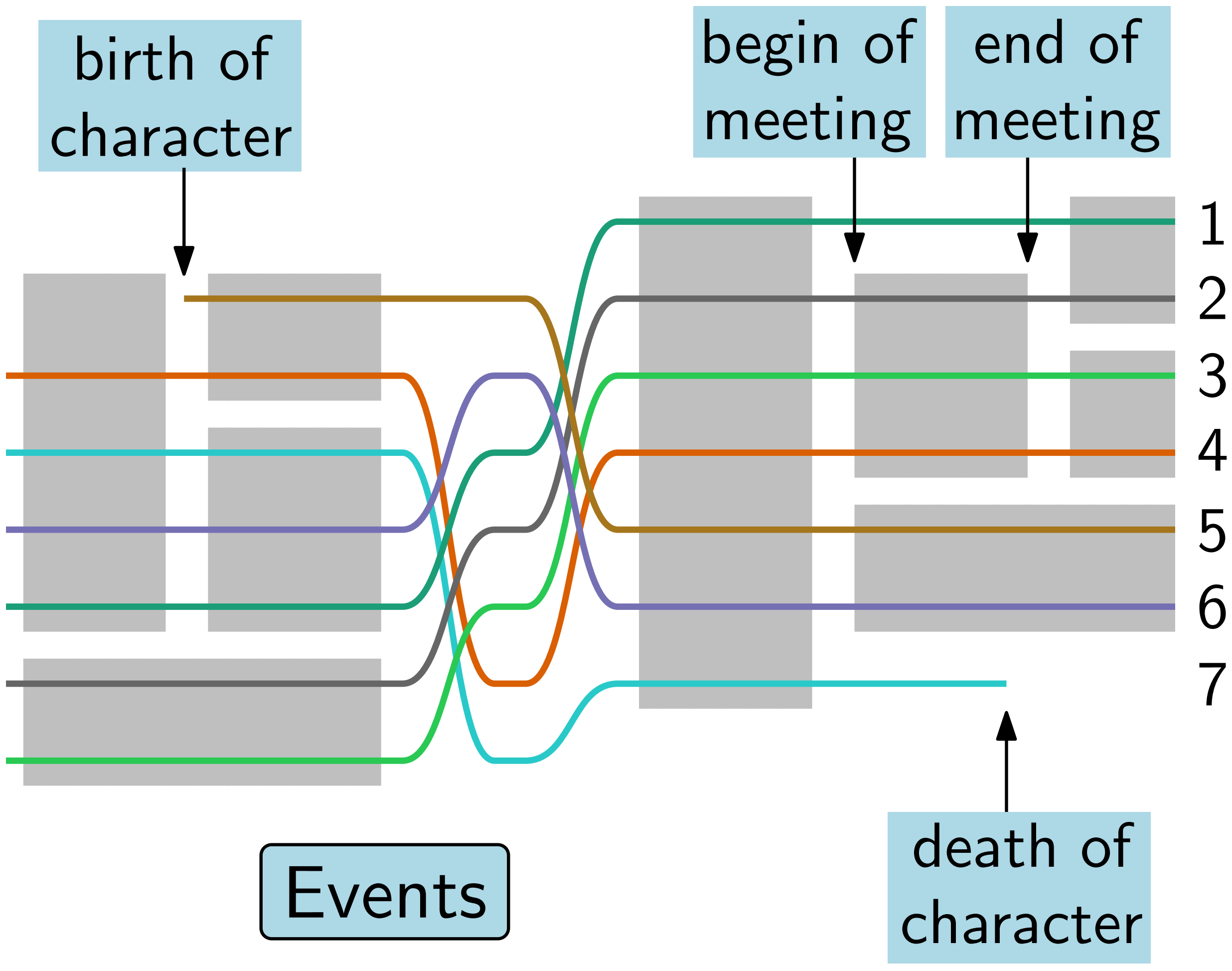
Problem Definition

Meeting: set of characters, start and end time



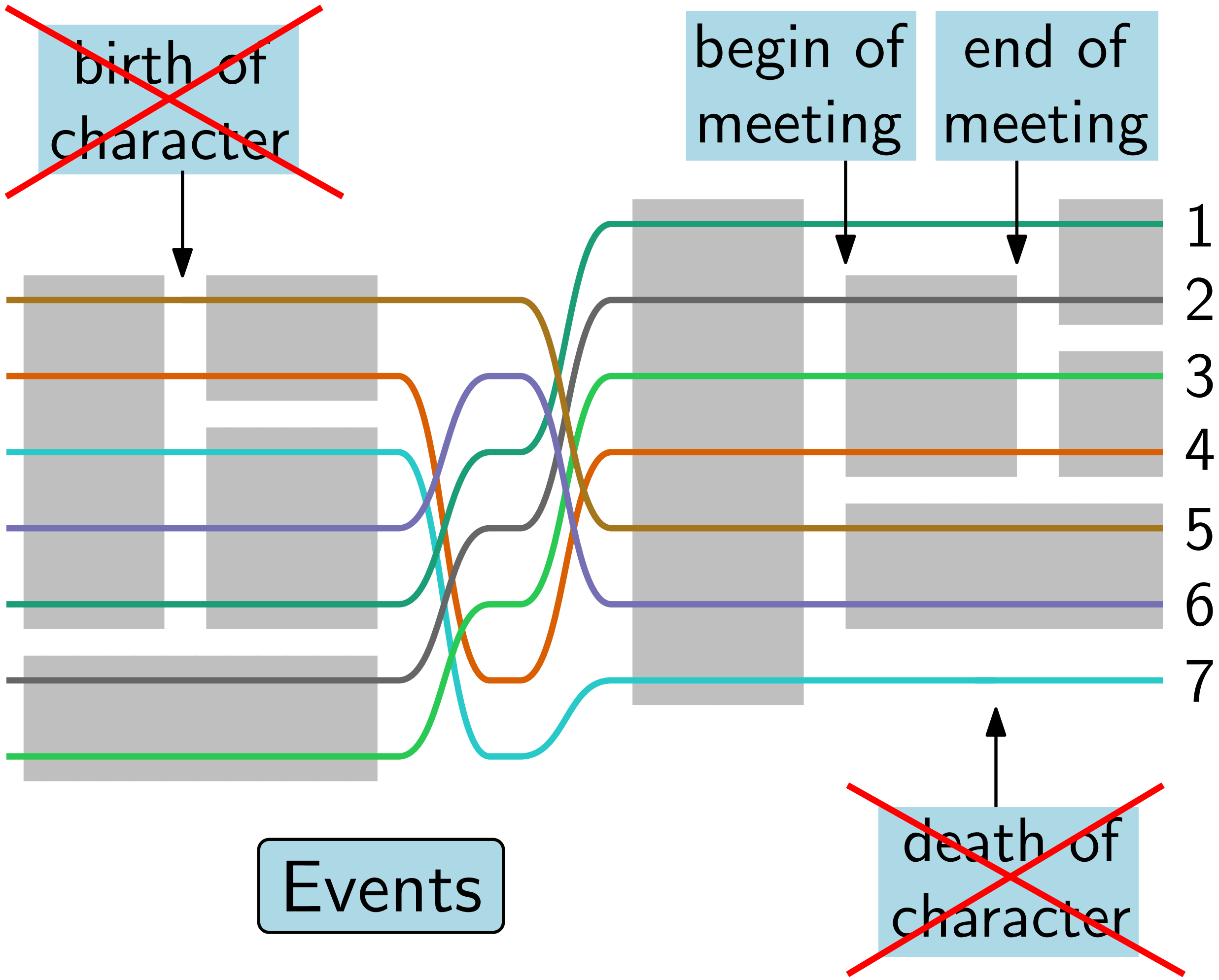
Problem Definition

Meeting: set of characters, start and end time



Problem Definition

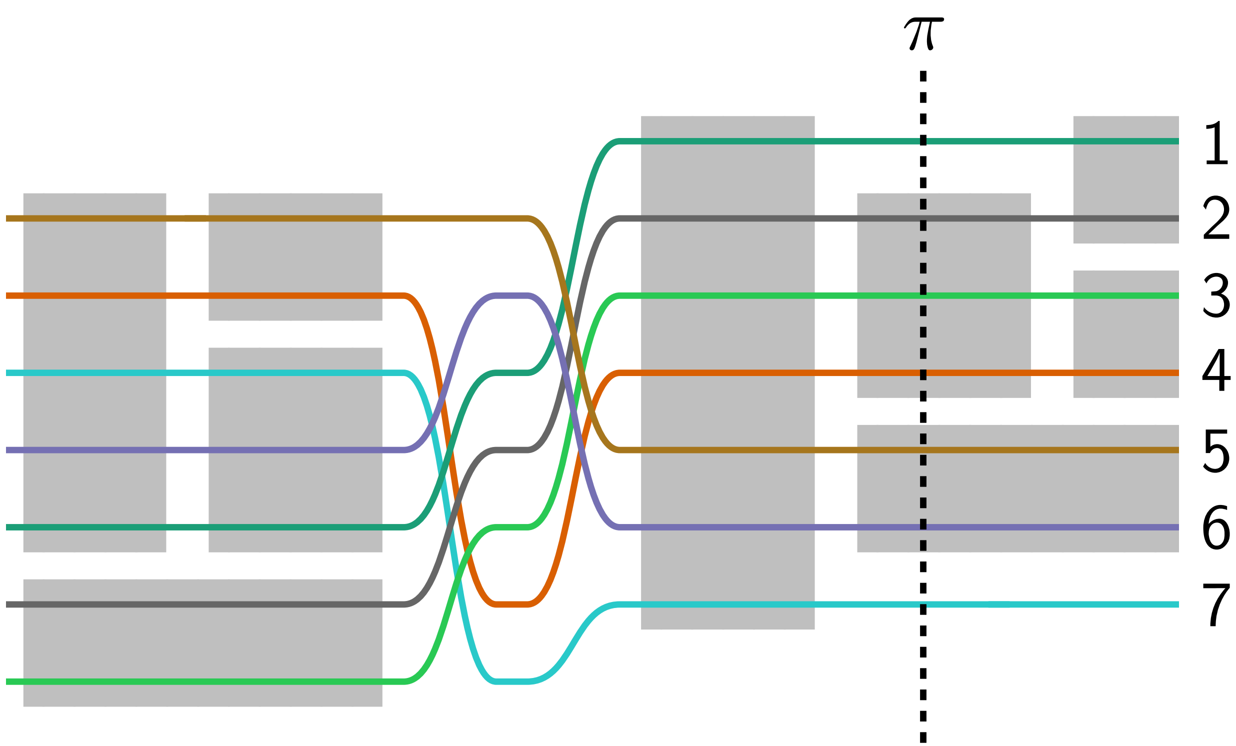
Meeting: set of characters, start and end time



No births and deaths in this talk – but in the paper.

Problem Definition

Meeting: set of characters, start and end time



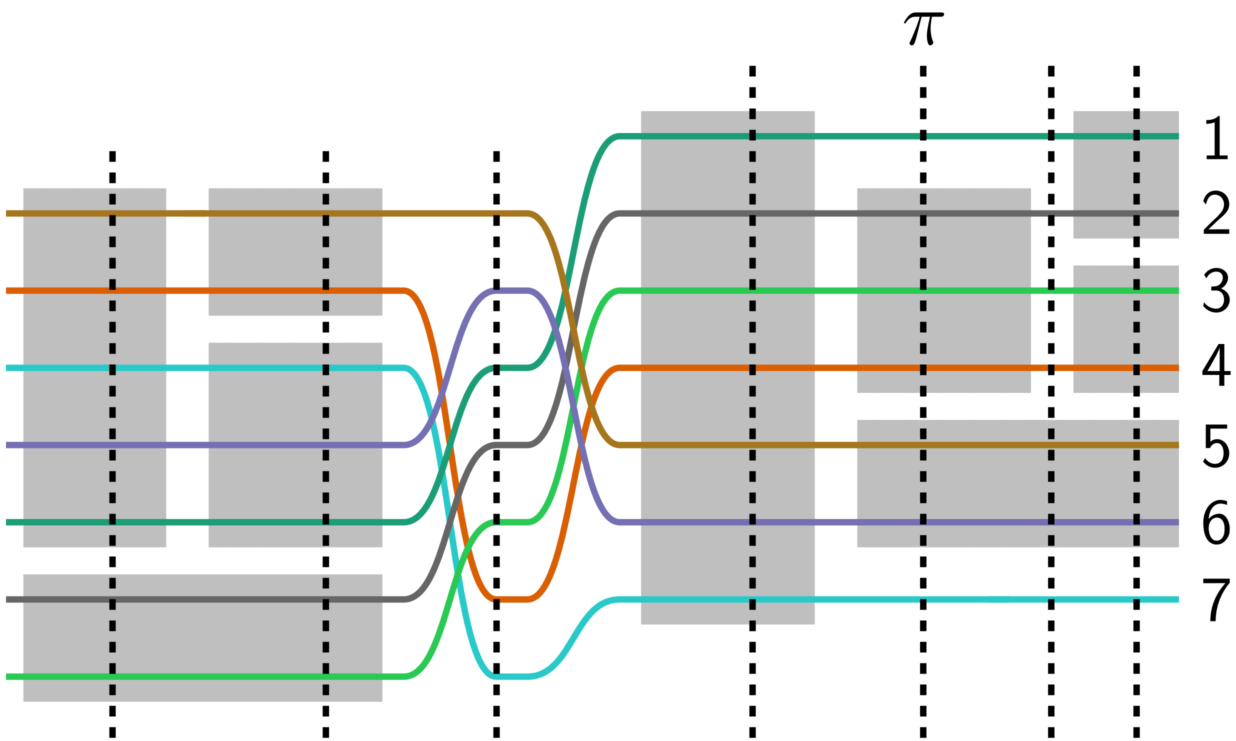
Permutation π supports meetings $\{2, 3, 4\}$ and $\{5, 6\}$

$=$

Sets $\{2, 3, 4\}$ and $\{5, 6\}$ each form a contiguous group in π

Problem Definition

Meeting: set of characters, start and end time



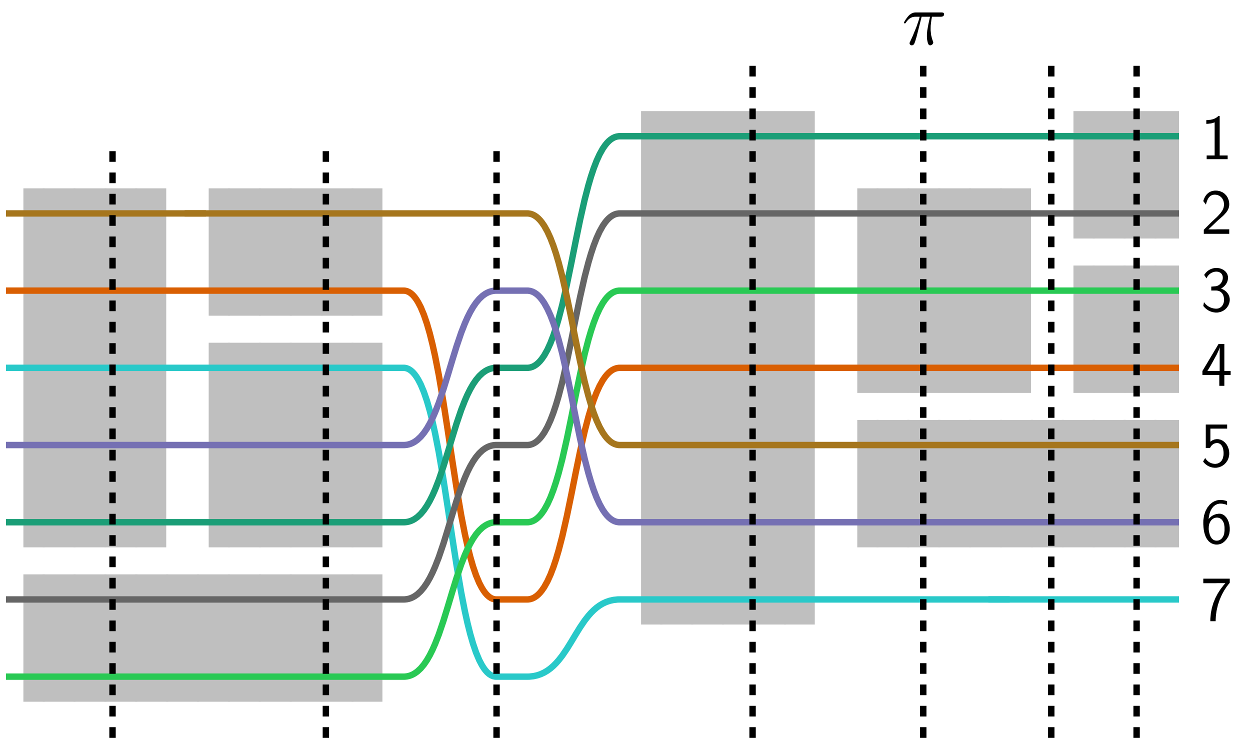
Permutation π supports meetings $\{2, 3, 4\}$ and $\{5, 6\}$

$=$

Sets $\{2, 3, 4\}$ and $\{5, 6\}$ each form a contiguous group in π

Problem Definition

Meeting: set of characters, start and end time



Permutation π supports meetings $\{2, 3, 4\}$ and $\{5, 6\}$

$=$

Sets $\{2, 3, 4\}$ and $\{5, 6\}$ each form a contiguous group in π

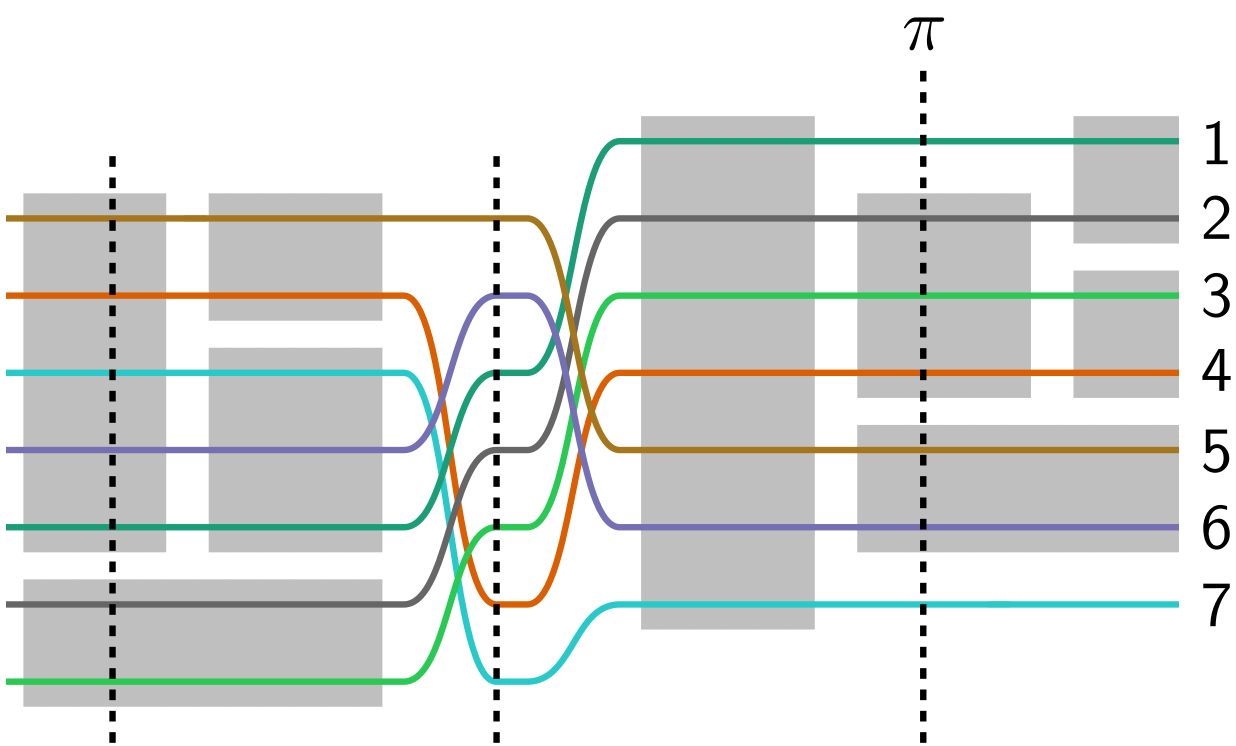
Find a **shortest** sequence of permutations such that . . .

. . . meetings are supported in the right order

. . . adjacent permutations differ by at most one block crossing

Problem Definition

Meeting: set of characters, start and end time



Permutation π supports meetings $\{2, 3, 4\}$ and $\{5, 6\}$

$=$

Sets $\{2, 3, 4\}$ and $\{5, 6\}$ each form a contiguous group in π

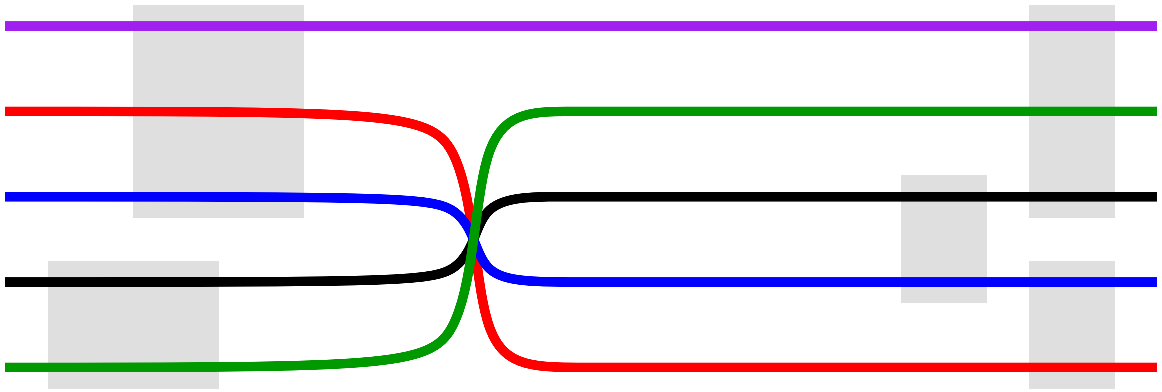
Find a **shortest** sequence of permutations such that . . .

. . . meetings are supported in the right order

. . . adjacent permutations differ by at most one block crossing

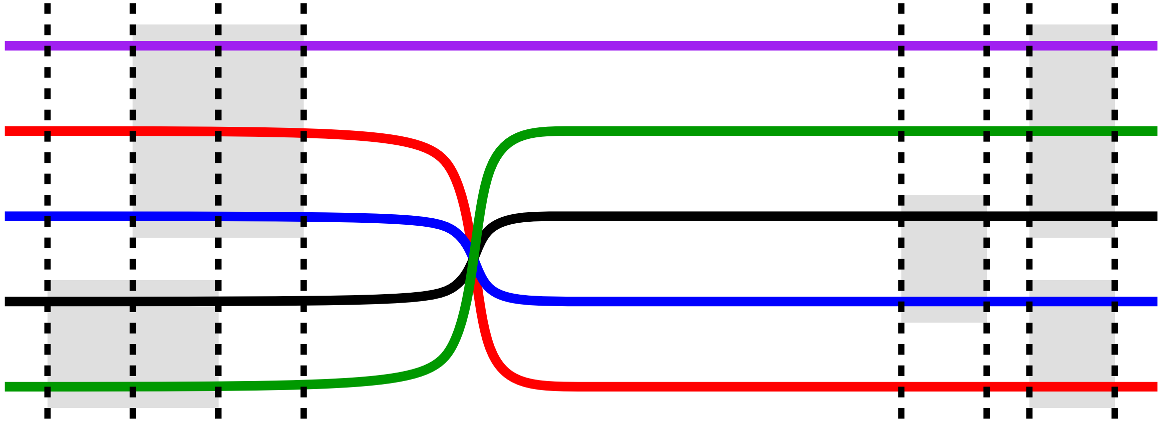
Approach of Gronemann et al.

Reduce to MLCM-TC (Multi-Layer Crossing Minimization Problem with Tree Constraints)



Approach of Gronemann et al.

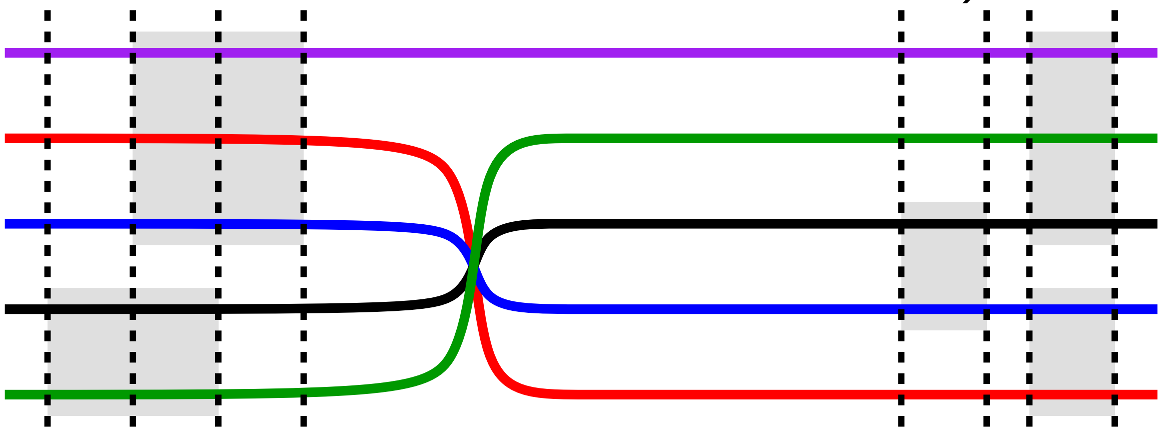
Reduce to MLCM-TC (Multi-Layer Crossing Minimization Problem with Tree Constraints)



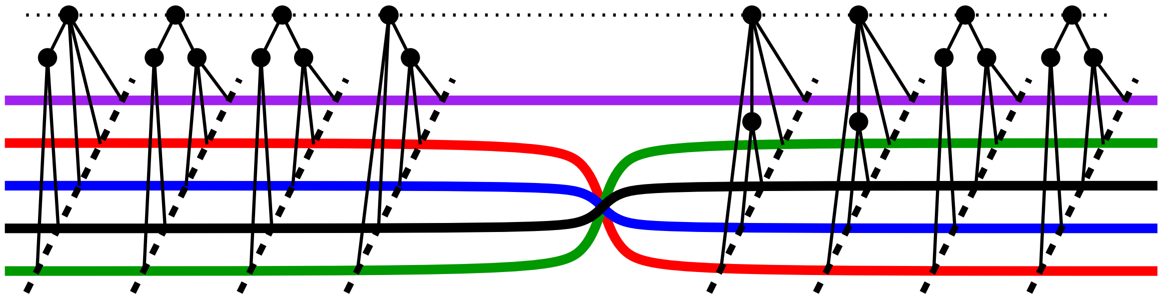
Each event corresponds to a layer

Approach of Gronemann et al.

Reduce to MLCM-TC (Multi-Layer Crossing Minimization Problem with Tree Constraints)



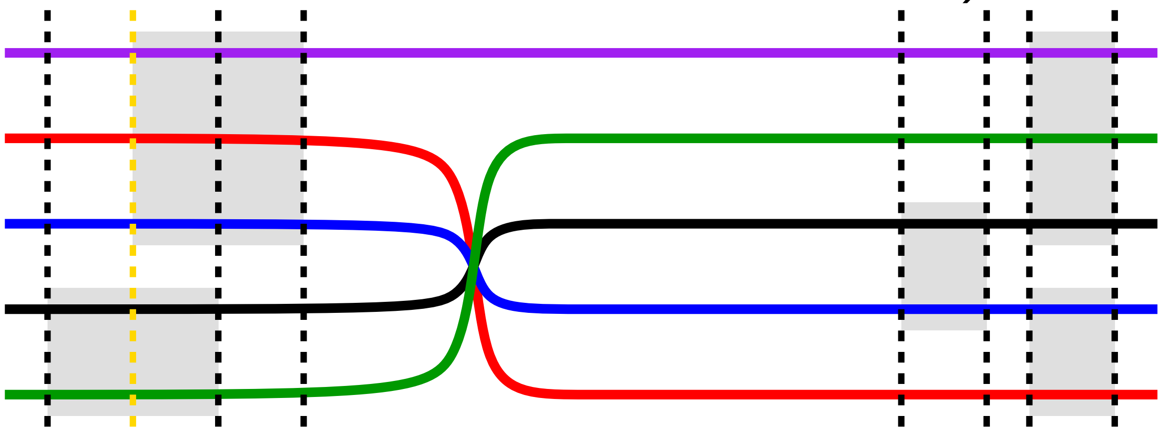
Each event corresponds to a layer



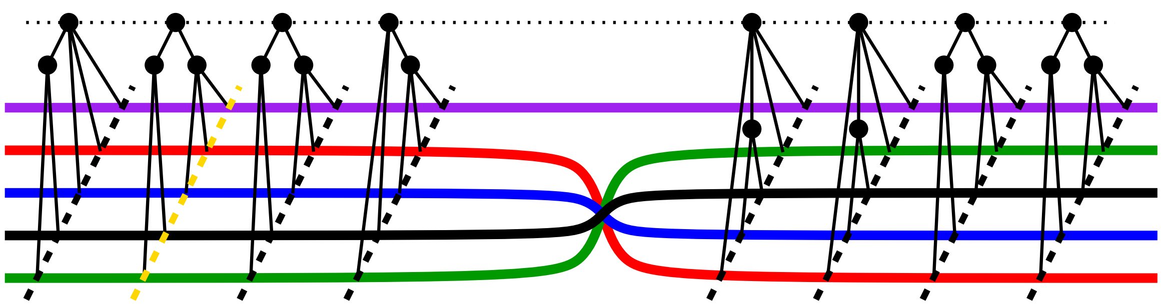
Trees describe neighborship between characters

Approach of Gronemann et al.

Reduce to MLCM-TC (Multi-Layer Crossing Minimization Problem with Tree Constraints)



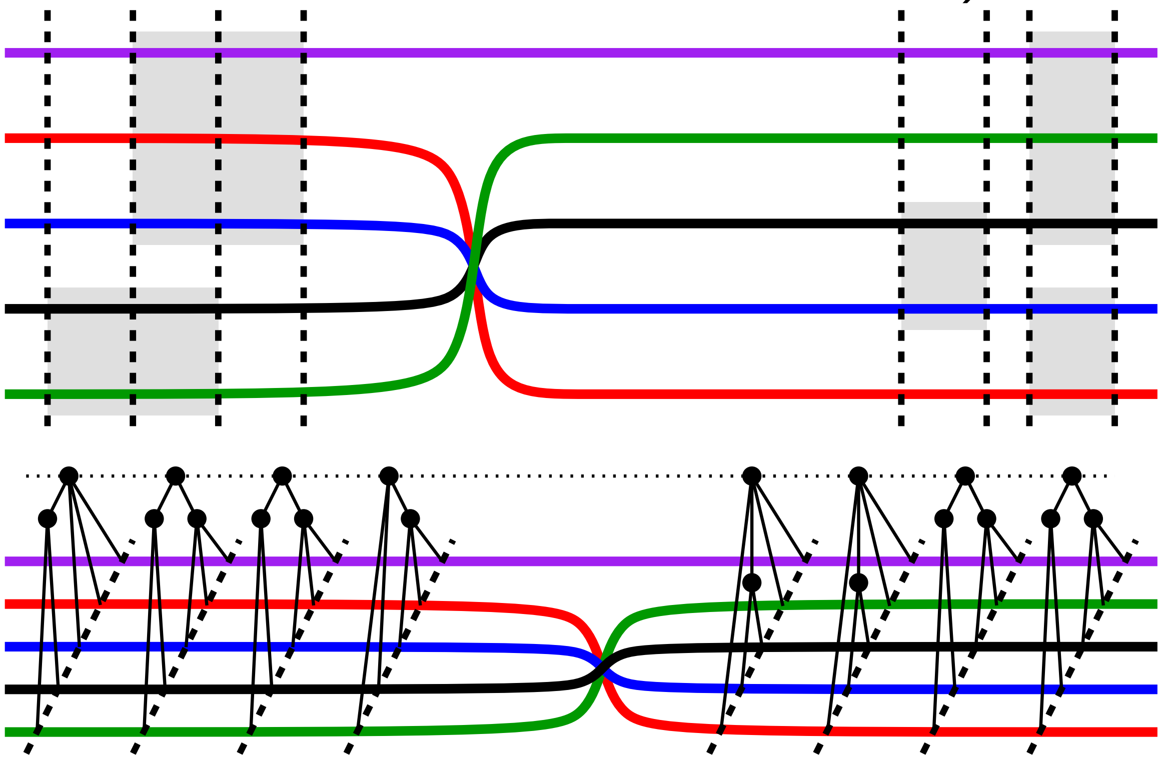
Each event corresponds to a layer



Trees describe neighborship between characters

Approach of Gronemann et al.

Reduce to MLCM-TC (Multi-Layer Crossing Minimization Problem with Tree Constraints)



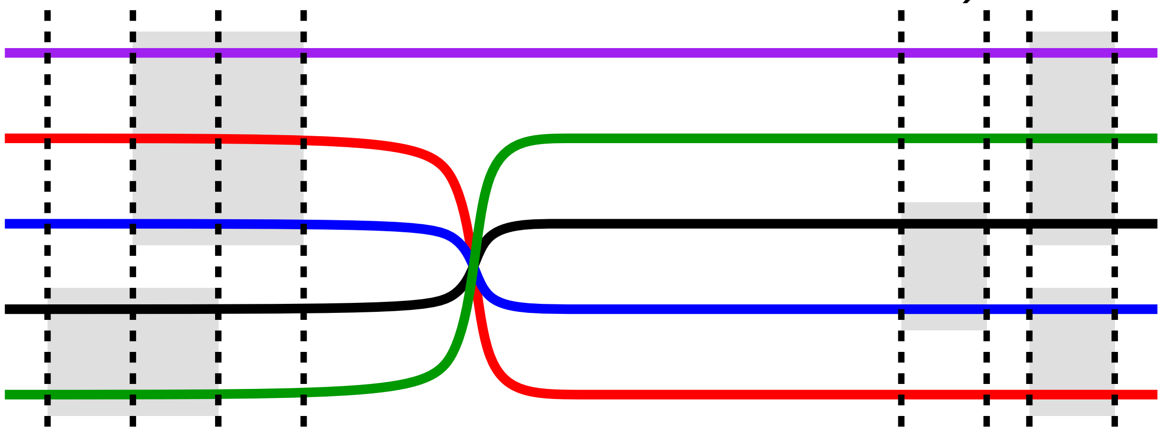
Each event corresponds to a layer

Trees describe neighborship between characters

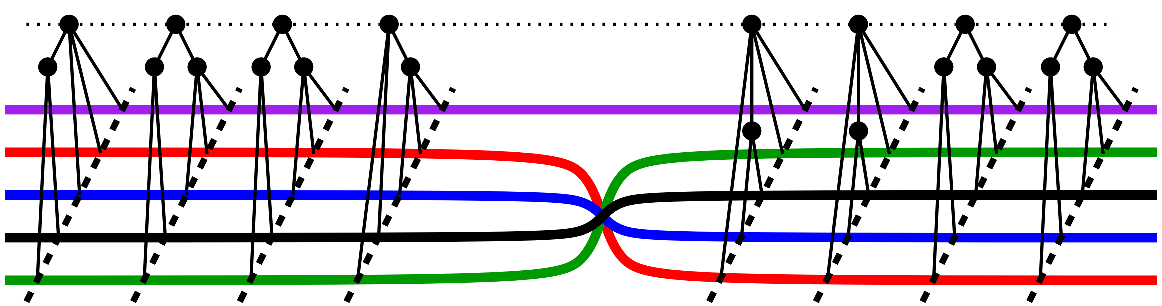
Find a permutation for each layer

Approach of Gronemann et al.

Reduce to MLCM-TC (Multi-Layer Crossing Minimization Problem with Tree Constraints)



Each event corresponds to a layer



Trees describe neighborship between characters

Find a permutation for each layer

Use ILP to solve MLCM-TC

Describing Permutations

Variables describing the order in each permutation: x_{ij}^{r}

layer

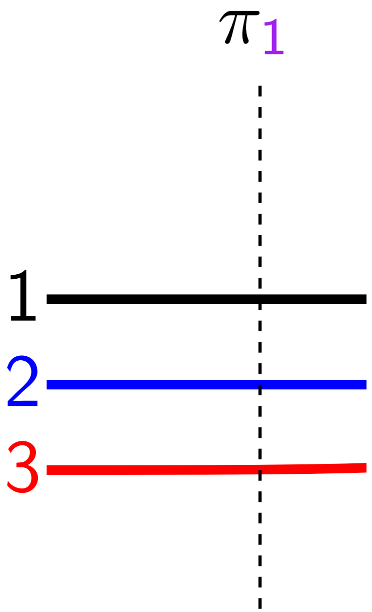
characters

Describing Permutations

Variables describing the order in each permutation: x_{ij}^{r}

layer

characters

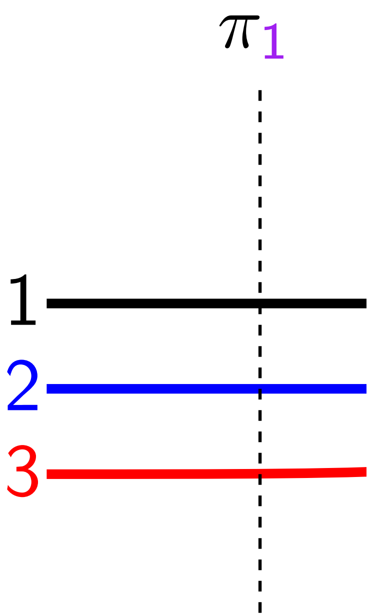


Describing Permutations

layer

Variables describing the order in each permutation: x_{ij}^{r}

characters



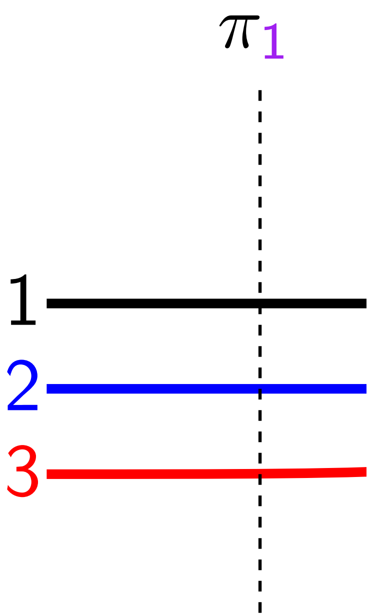
$$x_{12}^{1} = 1$$

Describing Permutations

Variables describing the order in each permutation: x_{ij}^{r}

layer

characters



$x_{12}^{1} = 1$

$x_{23}^{1} = 1$

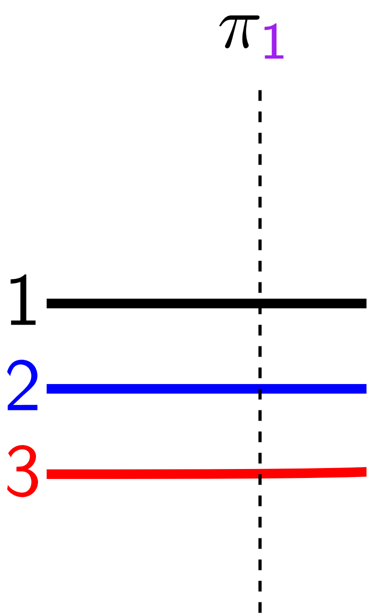
$x_{13}^{1} = 1$

Describing Permutations

layer

Variables describing the order in each permutation: x_{ij}^{r}

characters



$$x_{12}^{1} = 1 \quad x_{21}^{1} = 0$$

$$x_{23}^{1} = 1 \quad x_{32}^{1} = 0$$

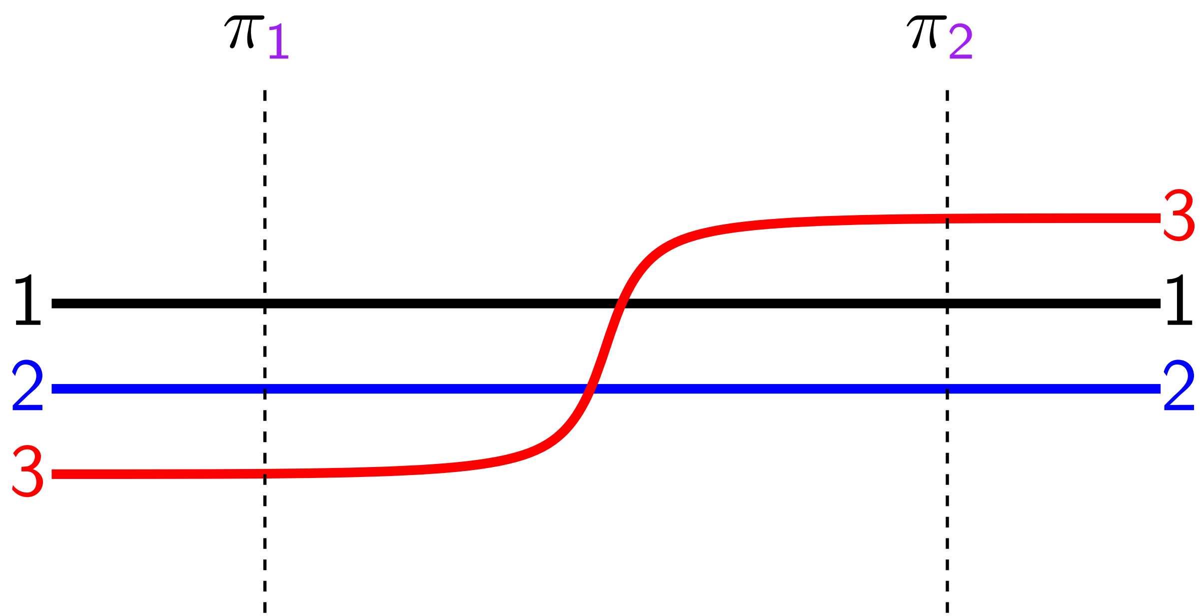
$$x_{13}^{1} = 1 \quad x_{31}^{1} = 0$$

Describing Permutations

Variables describing the order in each permutation: x_{ij}^{r}

layer

characters



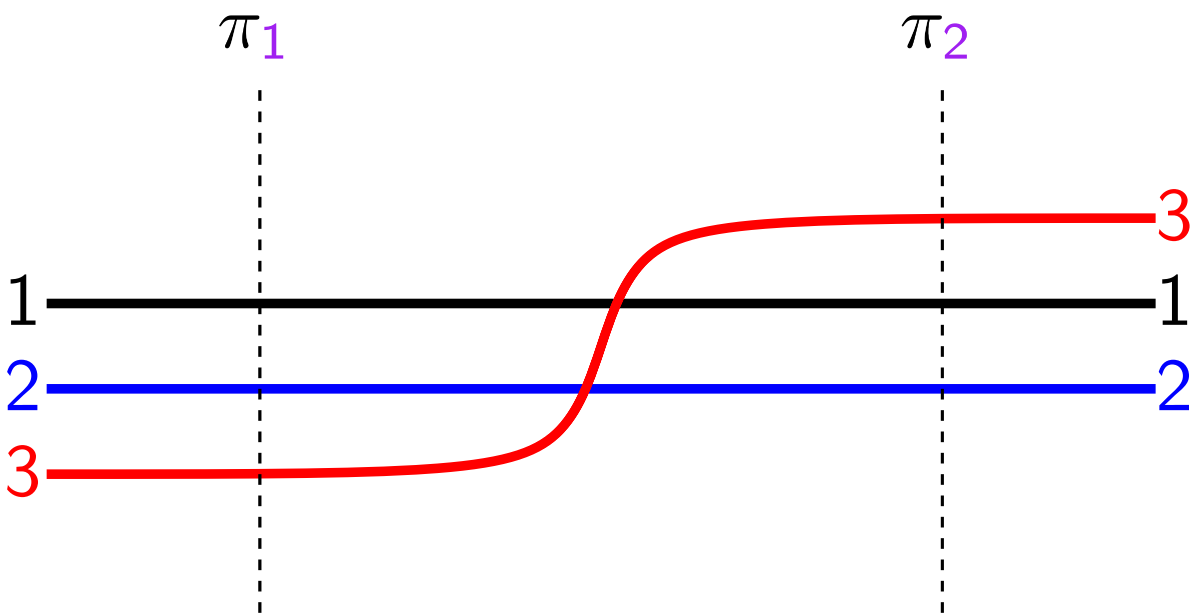
$$x_{12}^{1} = 1 \quad x_{21}^{1} = 0$$

$$x_{23}^{1} = 1 \quad x_{32}^{1} = 0$$

$$x_{13}^{1} = 1 \quad x_{31}^{1} = 0$$

Describing Permutations

Variables describing the order in each permutation: x_{ij}^{r} layer characters



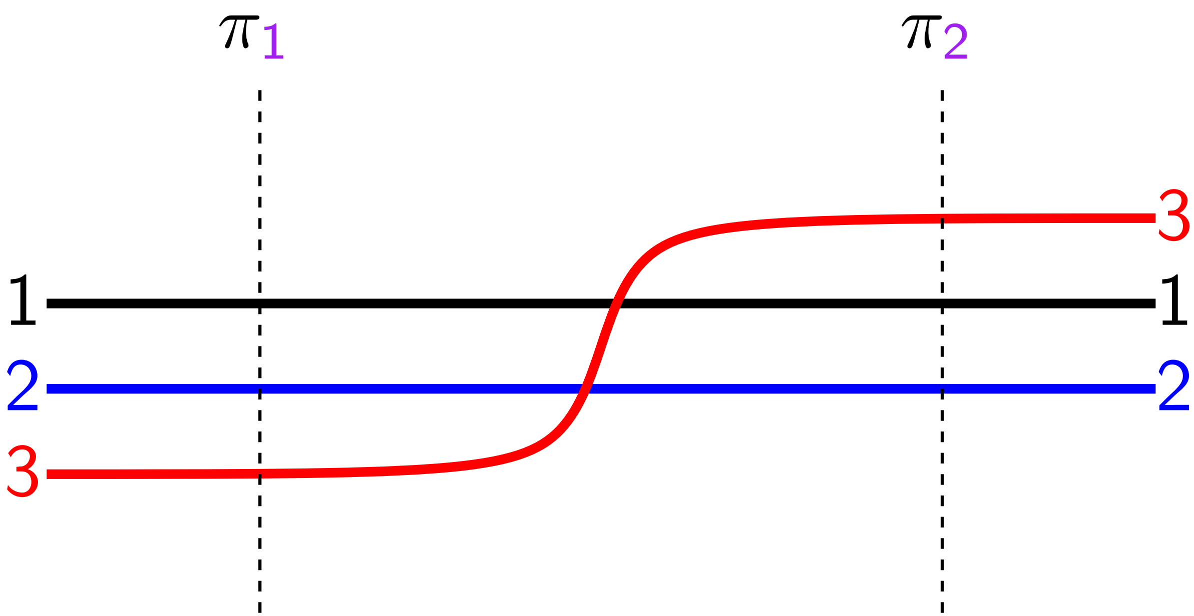
$$
\begin{array}{llll}
x_{12}^{1} = 1 & x_{21}^{1} = 0 & x_{12}^{2} = 1 & x_{21}^{2} = 0 \\
x_{23}^{1} = 1 & x_{32}^{1} = 0 & x_{23}^{2} = 0 & x_{32}^{2} = 1 \\
x_{13}^{1} = 1 & x_{31}^{1} = 0 & x_{13}^{2} = 0 & x_{31}^{2} = 1
\end{array}
$$

Describing Permutations

Variables describing the order in each permutation: x_{ij}^{r}

layer

characters



$x_{12}^{1} = 1$ $\quad x_{21}^{1} = 0$ $\qquad x_{12}^{2} = 1$ $\quad x_{21}^{2} = 0$

$x_{23}^{1} = 1$ $\quad x_{32}^{1} = 0$ $\qquad x_{23}^{2} = 0$ $\quad x_{32}^{2} = 1$

$x_{13}^{1} = 1$ $\quad x_{31}^{1} = 0$ $\qquad x_{13}^{2} = 0$ $\quad x_{31}^{2} = 1$

Antisymmetry:

$x_{ij}^{r} \Leftrightarrow \neg x_{ji}^{r}$

Transitivity:

$x_{ij}^{r} \vee x_{jk}^{r} \vee x_{ki}^{r}$

$\neg x_{ij}^{r} \vee \neg x_{jk}^{r} \vee \neg x_{ki}^{r}$

Counting *Block* Crossings

Easy to count pairwise crossings using x_{ij}^r:

i and j cross after layer r: $\chi_{ij}^r \Leftrightarrow (x_{ij}^r \neq x_{ij}^{r+1})$

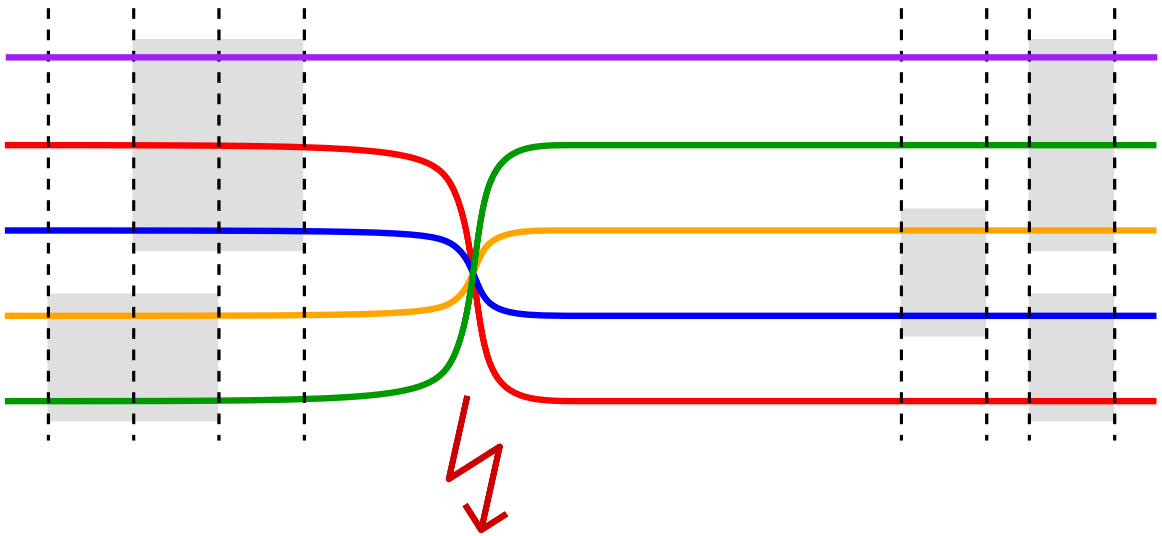
Counting *Block* Crossings

Easy to count pairwise crossings using x_{ij}^r:

i and j cross after layer r: $\chi_{ij}^r \Leftrightarrow (x_{ij}^r \neq x_{ij}^{r+1})$

Our approach for block crossings:

Allow at most one block crossing between two permutations



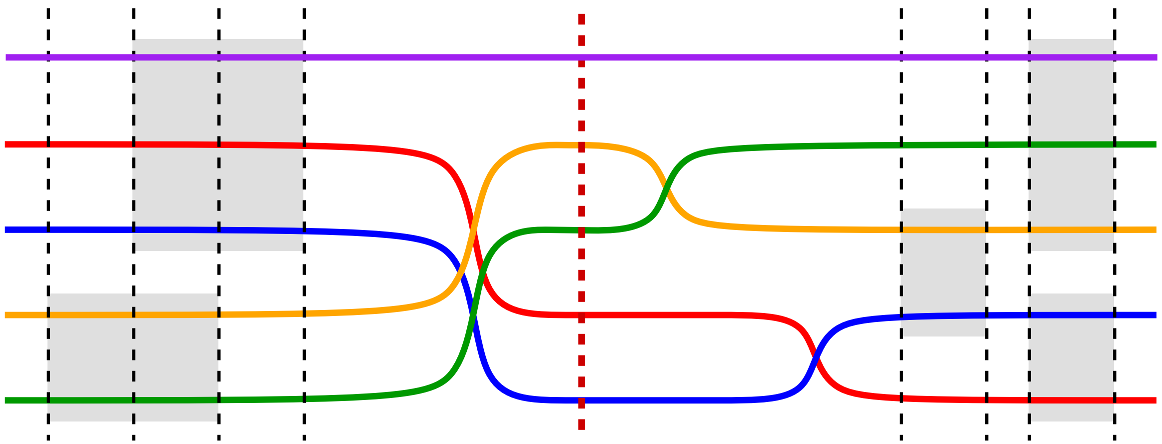
Counting *Block* Crossings

Easy to count pairwise crossings using x_{ij}^r:

i and j cross after layer r: $\chi_{ij}^r \Leftrightarrow (x_{ij}^r \neq x_{ij}^{r+1})$

Our approach for block crossings:

Allow at most one block crossing between two permutations



Add more permutations

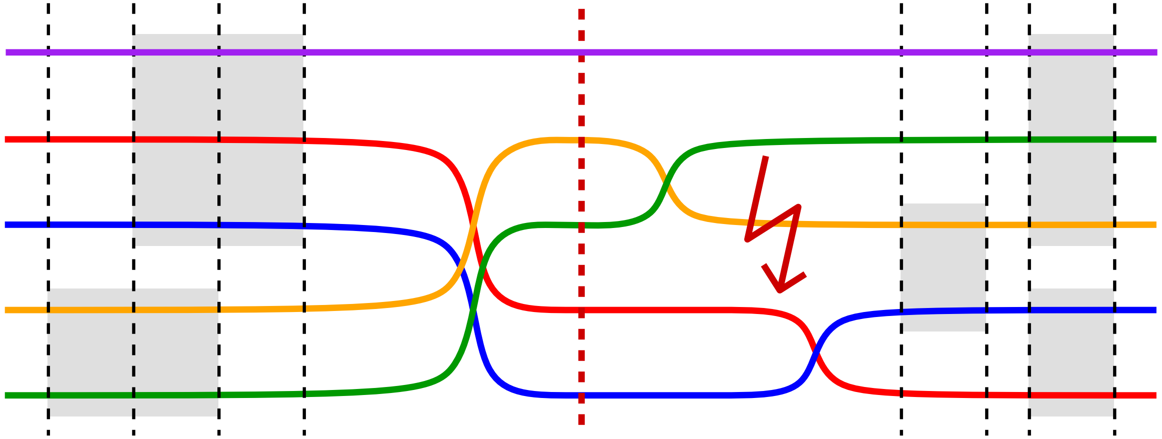
Counting *Block* Crossings

Easy to count pairwise crossings using x_{ij}^r:

i and j cross after layer r: $\chi_{ij}^r \Leftrightarrow (x_{ij}^r \neq x_{ij}^{r+1})$

Our approach for block crossings:

Allow at most one block crossing between two permutations



Add more permutations

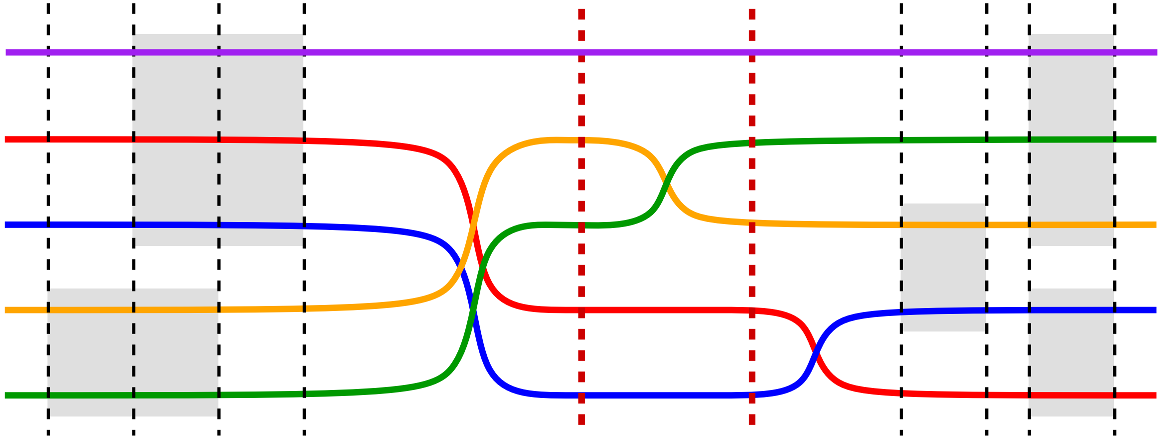
Counting *Block* Crossings

Easy to count pairwise crossings using x_{ij}^r:

i and j cross after layer r: $\chi_{ij}^r \Leftrightarrow (x_{ij}^r \neq x_{ij}^{r+1})$

Our approach for block crossings:

Allow at most one block crossing between two permutations



Add more permutations

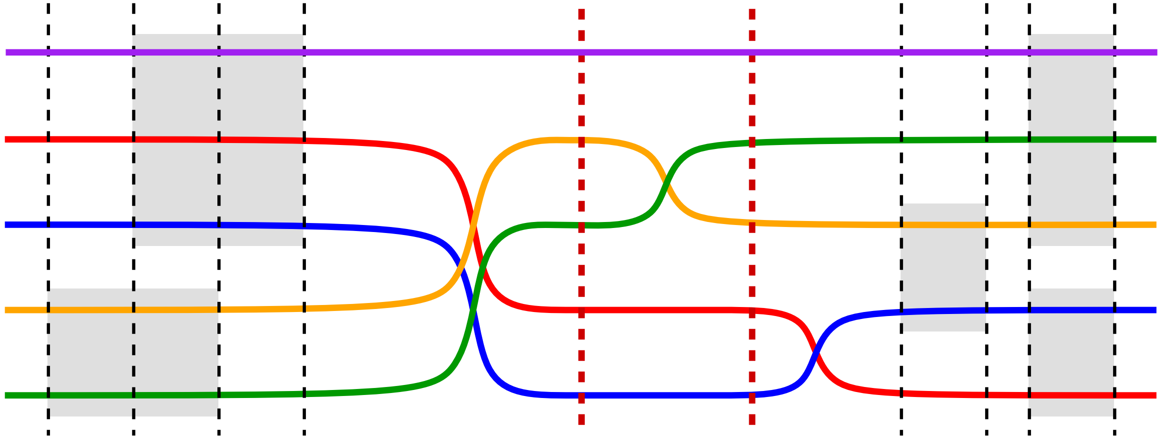
Counting *Block* Crossings

Easy to count pairwise crossings using x_{ij}^r:

i and j cross after layer r: $\chi_{ij}^r \Leftrightarrow (x_{ij}^r \neq x_{ij}^{r+1})$

Our approach for block crossings:

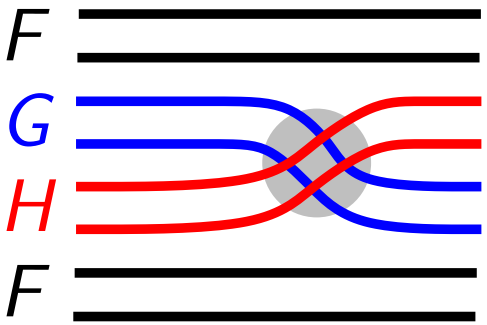
Allow at most one block crossing between two permutations



Add more permutations

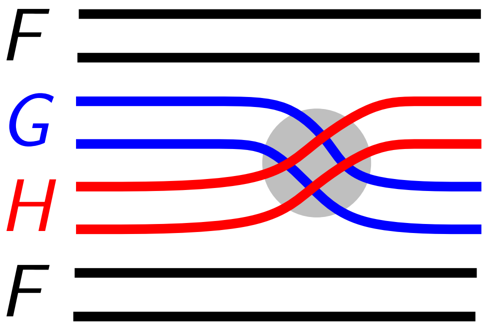
Prescribe maximum number of permutations

Describing Block Crossings



Blocks G and H cross

Describing Block Crossings

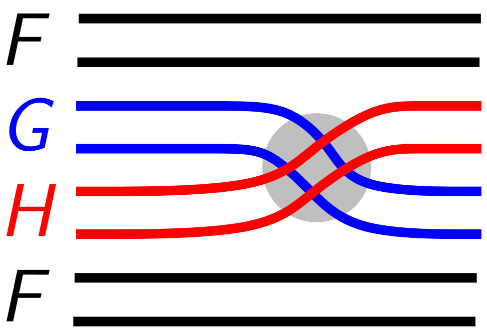


Blocks G and H cross

Constraints:

Exactly characters of G and H cross each other

Describing Block Crossings



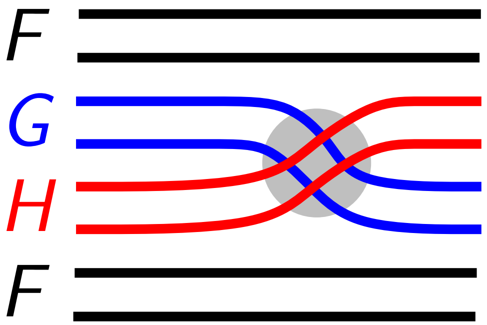
Blocks G and H cross

Constraints:

Exactly characters of G and H cross each other

$\chi_{ij}^r \Leftrightarrow g_i^r \wedge h_j^r$

Describing Block Crossings

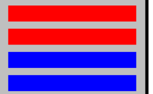


Blocks G and H cross

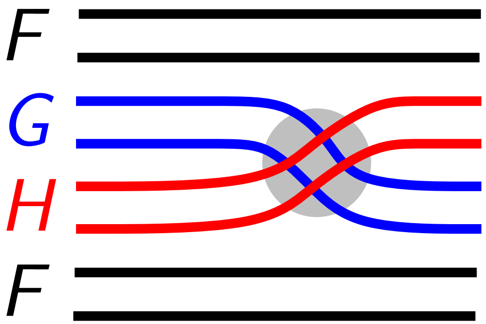
Constraints:

Exactly characters of G and H cross each other
$\chi_{ij}^{r} \Leftrightarrow g_i^r \wedge h_j^r$

G is above H



Describing Block Crossings



Blocks G and H cross

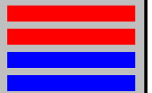
Constraints:

Exactly characters of G and H cross each other

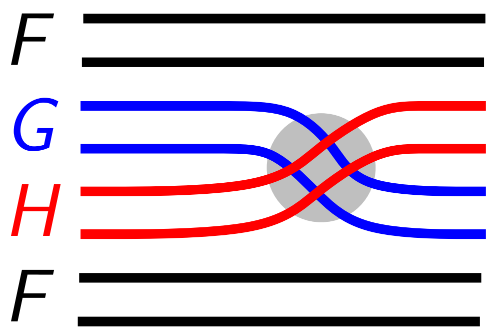
$\chi^r_{ij} \Leftrightarrow g^r_i \wedge h^r_j$

G is above H

$g^r_i \wedge h^r_j \Rightarrow$



Describing Block Crossings

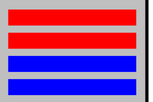


Blocks G and H cross

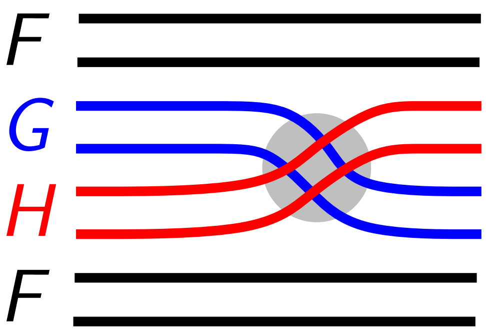
Constraints:

Exactly characters of G and H cross each other
$\chi_{ij}^{r} \Leftrightarrow g_i^r \wedge h_j^r$

G is above H
$g_i^r \wedge h_j^r \Rightarrow x_{ij}^r$



Describing Block Crossings



Blocks G and H cross

Constraints:

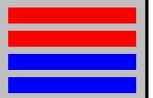
Exactly characters of G and H cross each other

$\chi^r_{ij} \Leftrightarrow g^r_i \wedge h^r_j$

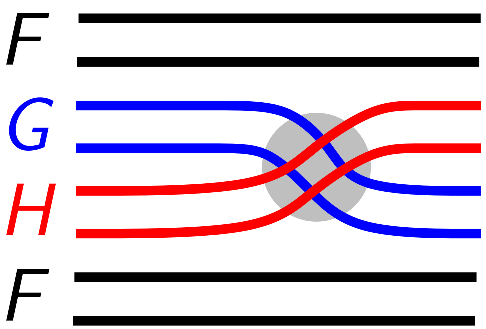
G and H are adjacent

G is above H

$g^r_i \wedge h^r_j \Rightarrow x^r_{ij}$



Describing Block Crossings



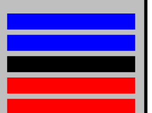
Blocks G and H cross

Constraints:

Exactly characters of G and H cross each other

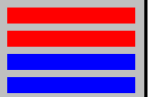
$\chi^r_{ij} \Leftrightarrow g^r_i \wedge h^r_j$

G and H are adjacent

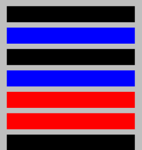


G is above H

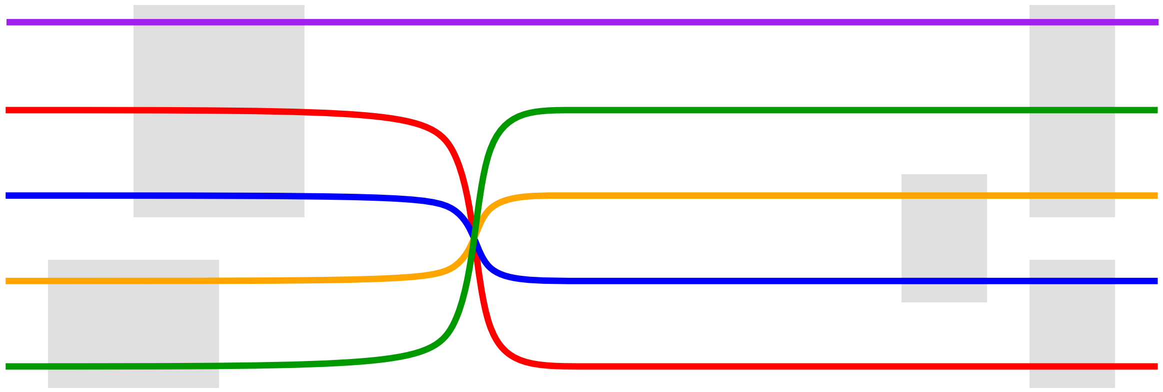
$g^r_i \wedge h^r_j \Rightarrow x^r_{ij}$



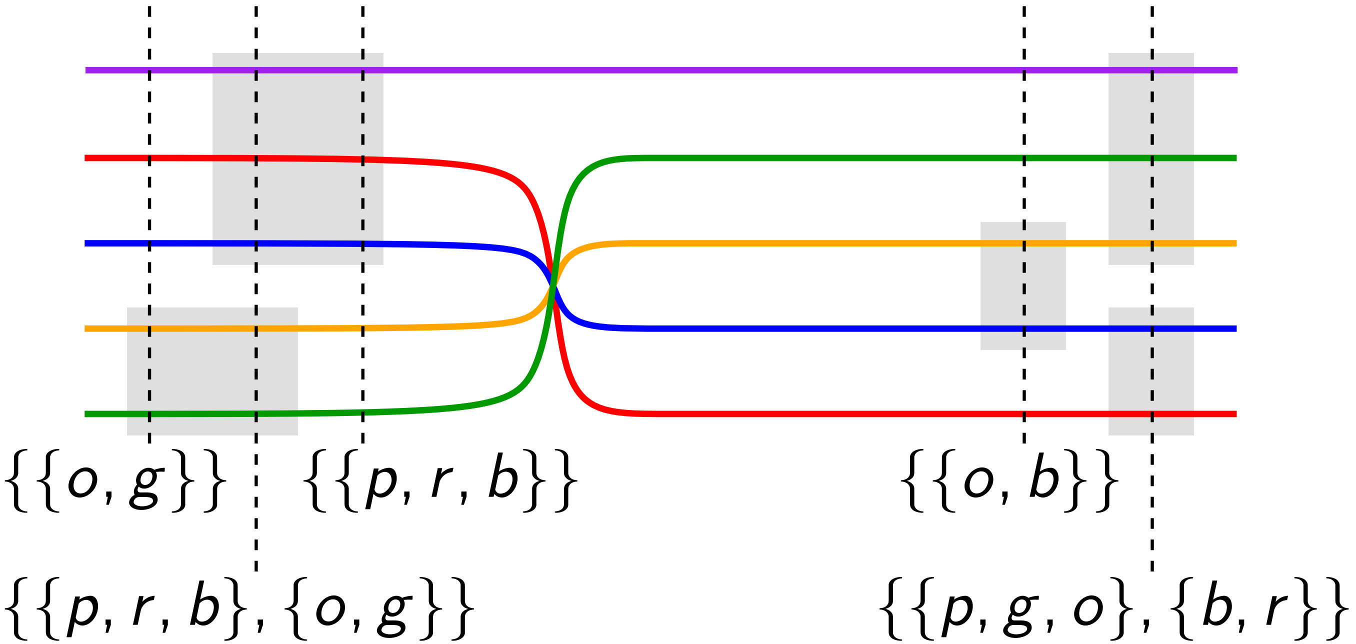
G and H are contiguous blocks



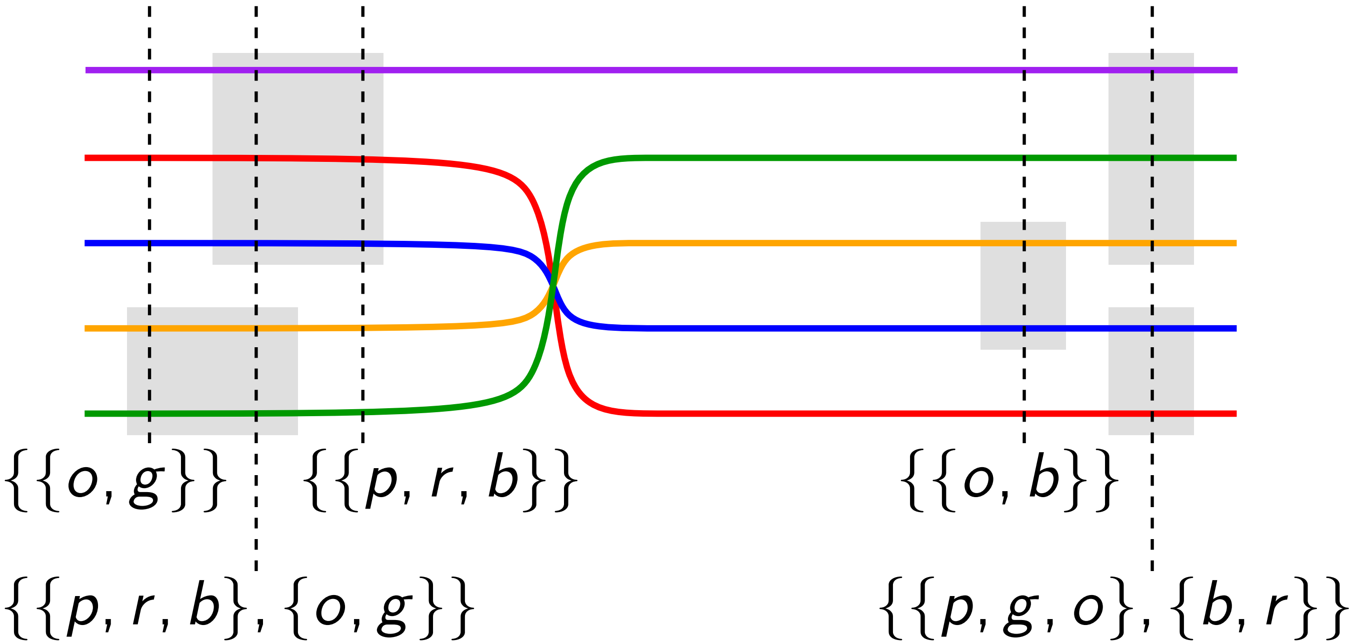
Meeting Groups

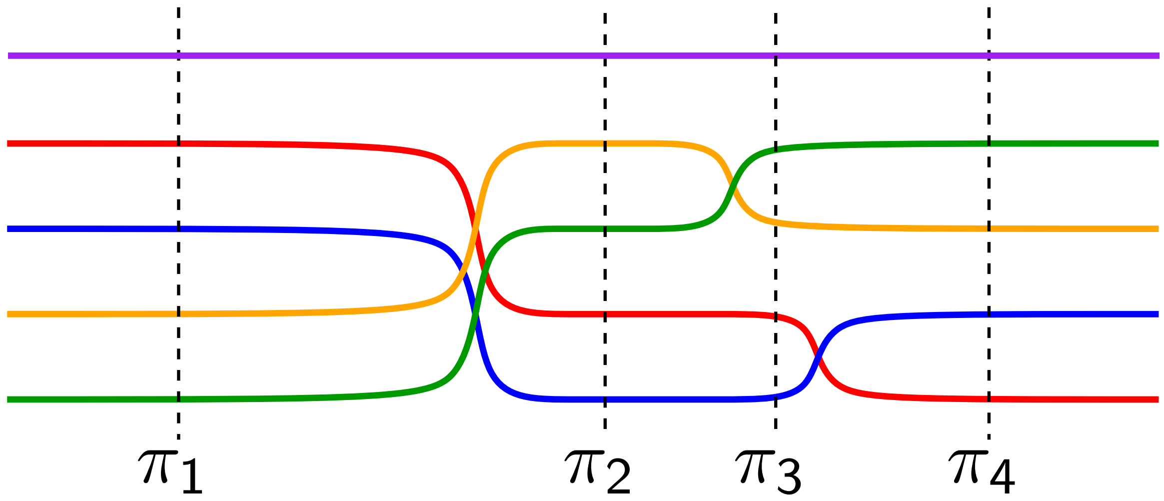


Meeting Groups

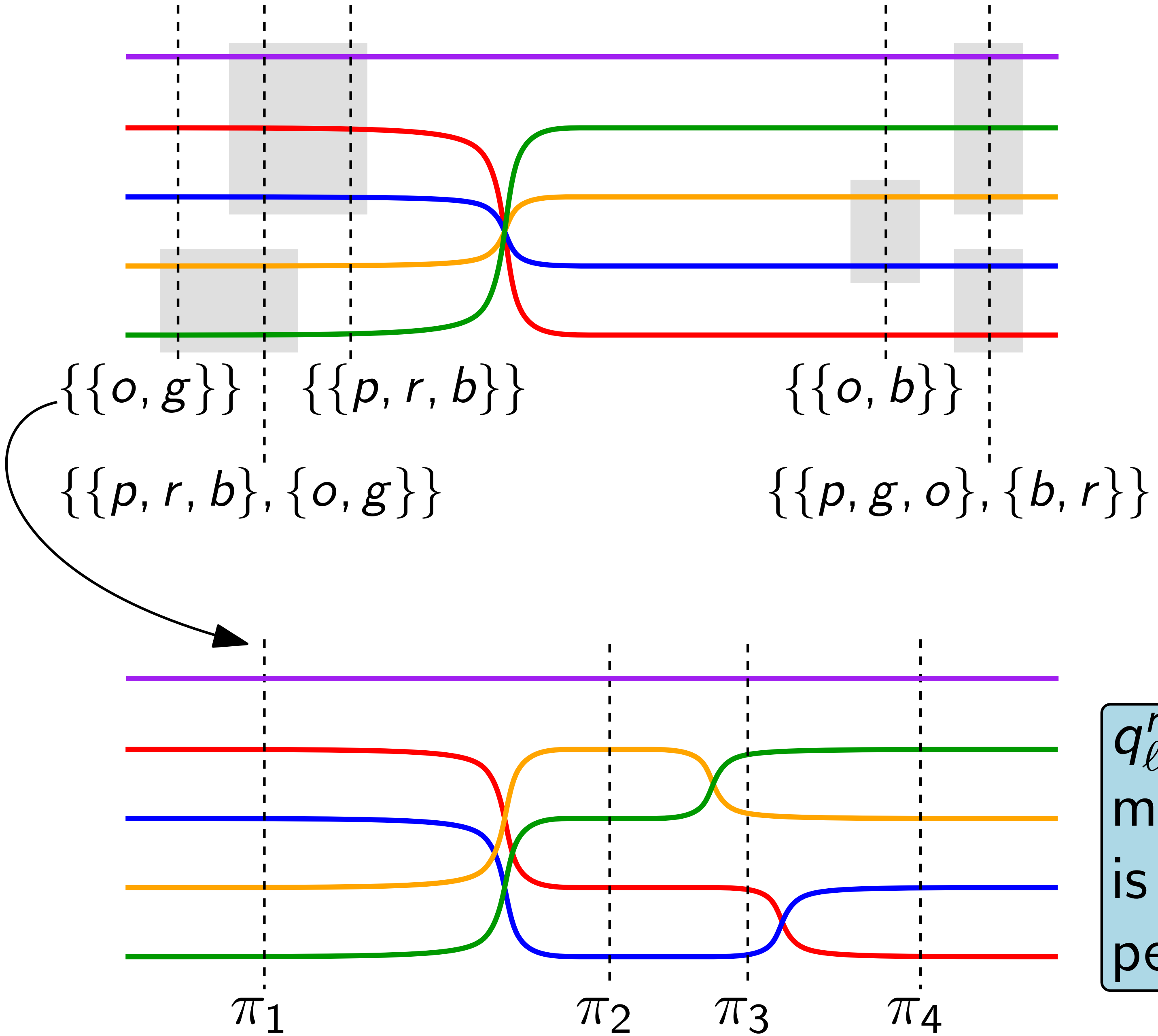


Meeting Groups



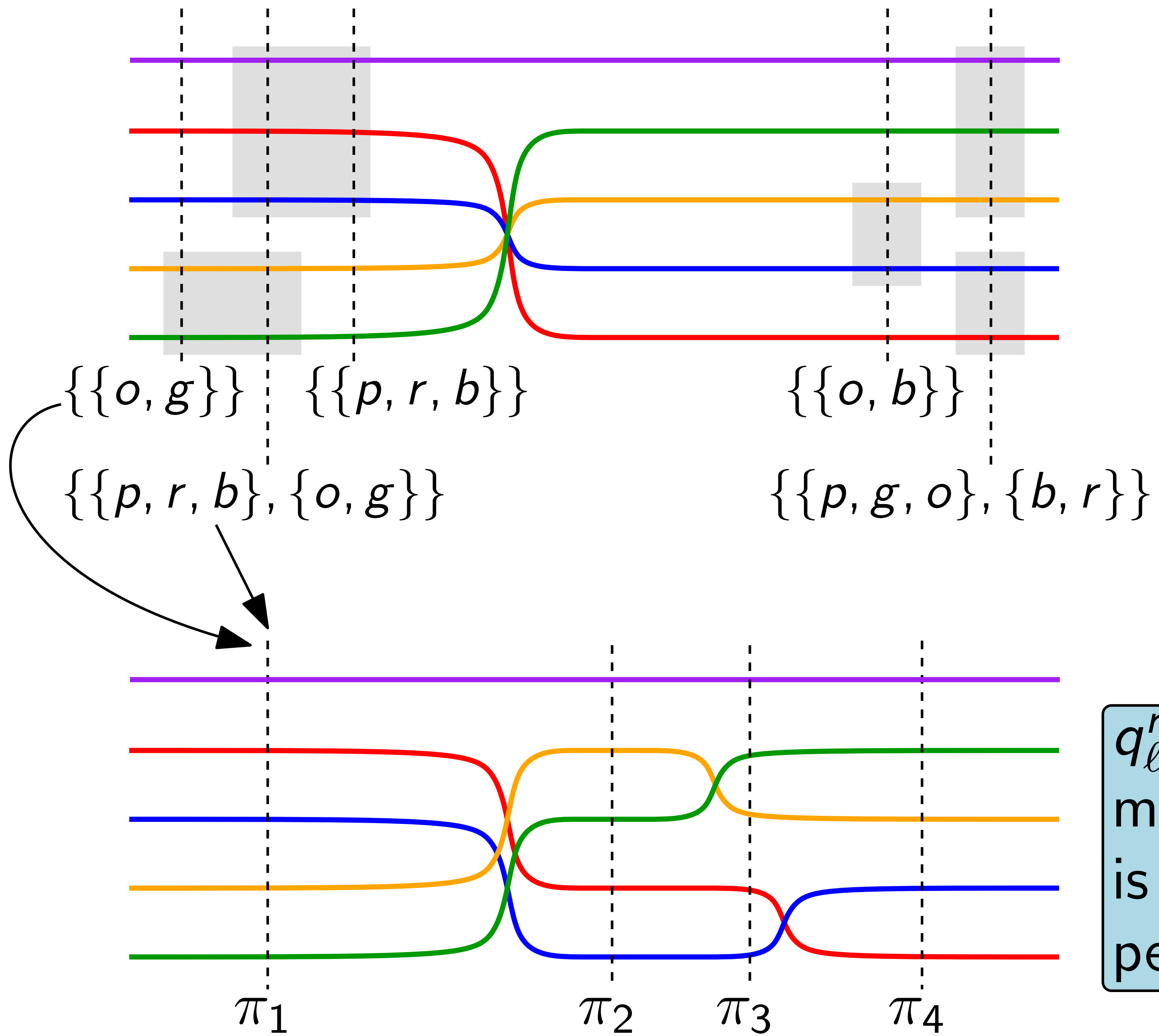


Meeting Groups



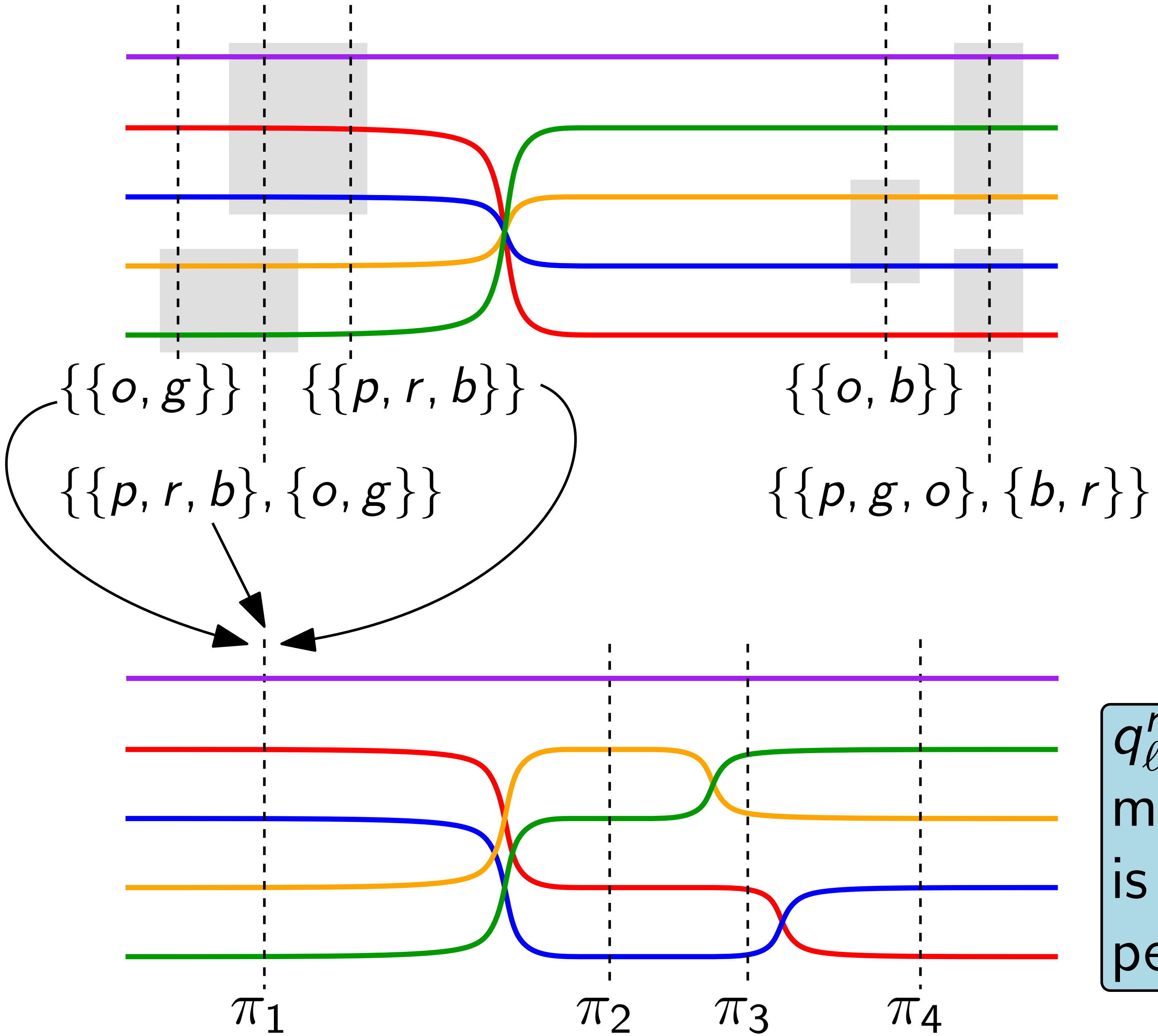
$q_\ell^r = 1 \Leftrightarrow$ meeting group ℓ is assigned to permutation π_r

Meeting Groups



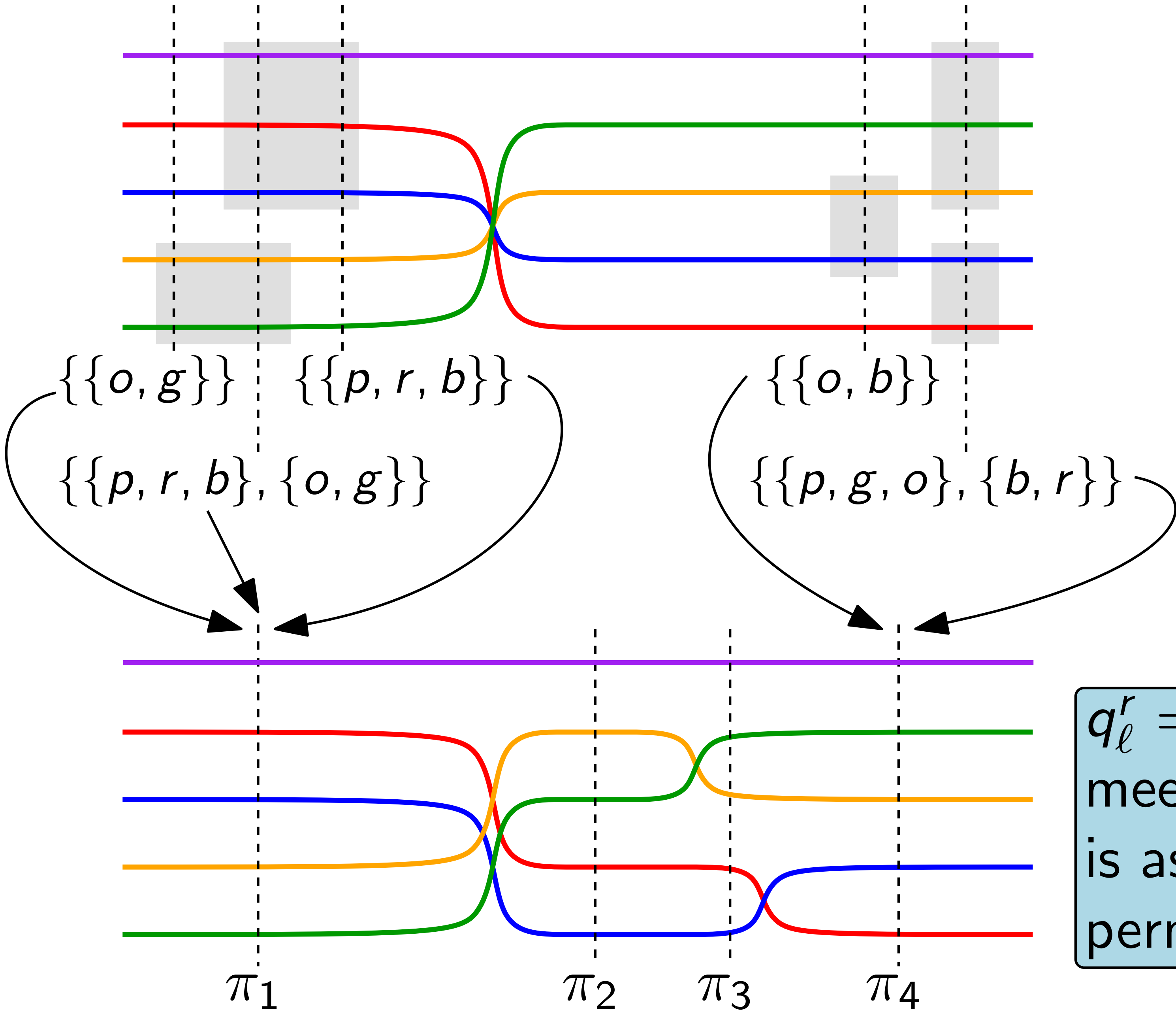
$q_\ell^r = 1 \Leftrightarrow$ meeting group ℓ is assigned to permutation π_r

Meeting Groups



$q_\ell^r = 1 \Leftrightarrow$ meeting group ℓ is assigned to permutation π_r

Meeting Groups



$q_\ell^r = 1 \Leftrightarrow$ meeting group ℓ is assigned to permutation π_r

Minimize the Number of Block Crossings

Choose a number of permutations λ and construct the clauses.

Minimize the Number of Block Crossings

Choose a number of permutations λ and construct the clauses.

There is a satisfying assignment
$\Leftrightarrow \exists$ solution for the SBCM instance using λ permutations.

Minimize the Number of Block Crossings

Choose a number of permutations λ and construct the clauses.

There is a satisfying assignment
$\Leftrightarrow$ $\exists$ solution for the SBCM instance using λ permutations.

$O(\lambda(\kappa^2 + \mu))$ variables, $O(\lambda\mu(\lambda + \kappa^3))$ clauses
λ: number of permutations
κ: number of characters
μ: number of meeting groups

Minimize the Number of Block Crossings

Choose a number of permutations λ and construct the clauses.

There is a satisfying assignment
$\Leftrightarrow$ $\exists$ solution for the SBCM instance using λ permutations.

$O(\lambda(\kappa^2 + \mu))$ variables, $O(\lambda\mu(\lambda + \kappa^3))$ clauses
λ: number of permutations
κ: number of characters
μ: number of meeting groups

Finding the optimum:

Repeatedly run the SAT solver with different values for λ (exponential search)

Experiments

FPT Breadth-first search a smarter state space; runtime: $O(k! \cdot k^3 \cdot n)$

- FPT are implemented in C++
- Concurrent meetings not implemented for FPT
- SAT clauses generated by Python and solved using MiniSat

Experiments

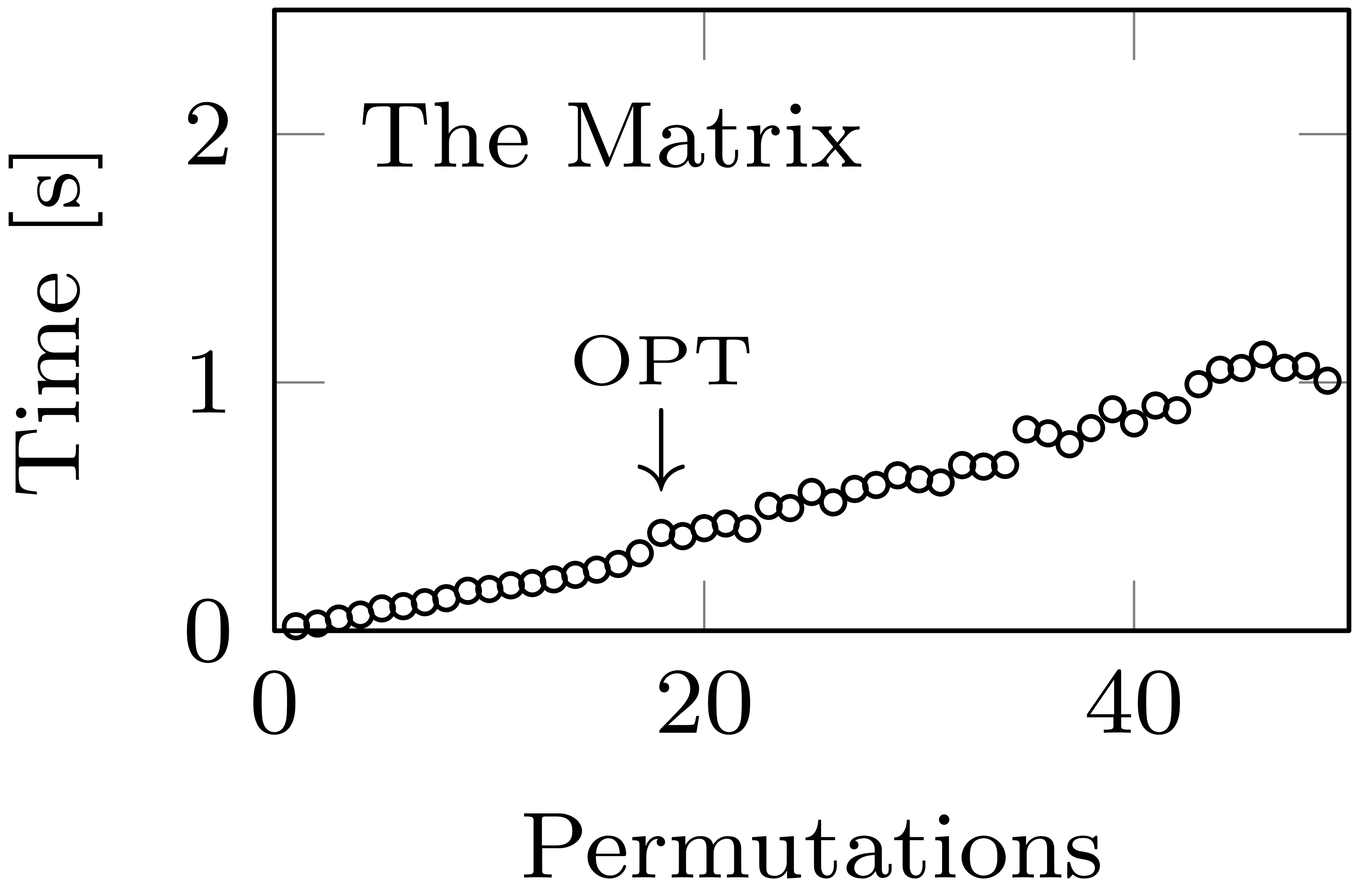
FPT Breadth-first search a smarter state space; runtime: $O(k! \cdot k^3 \cdot n)$

- FPT are implemented in C++
- Concurrent meetings not implemented for FPT
- SAT clauses generated by Python and solved using MiniSat

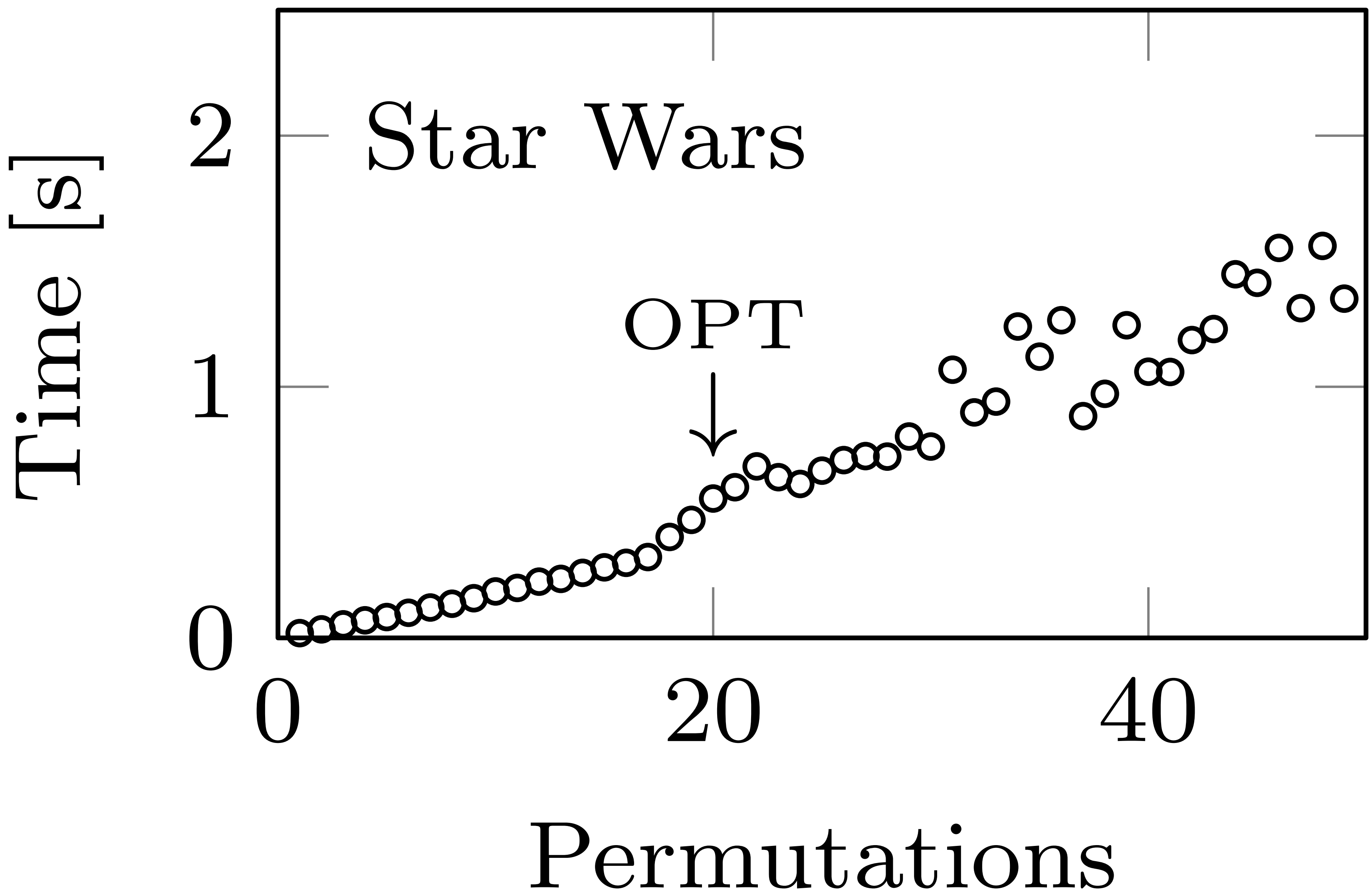
Test Data:

- Real-World instances (movies used by Gronemann et al.): The Matrix, Inception, Star Wars
- Random instances
- Random instances having a solution with few block crossings

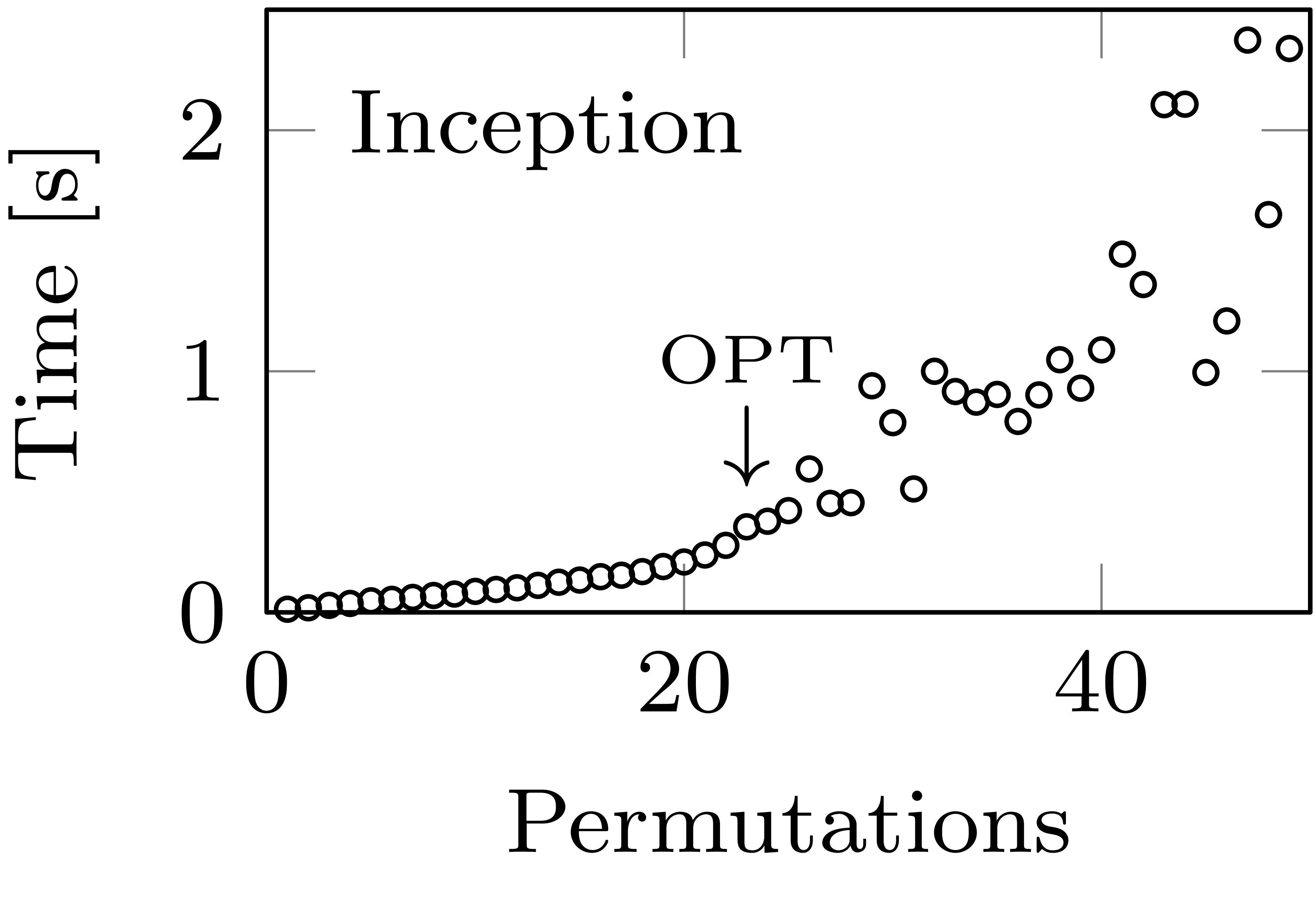
SAT: Runtime vs Number of Permutations



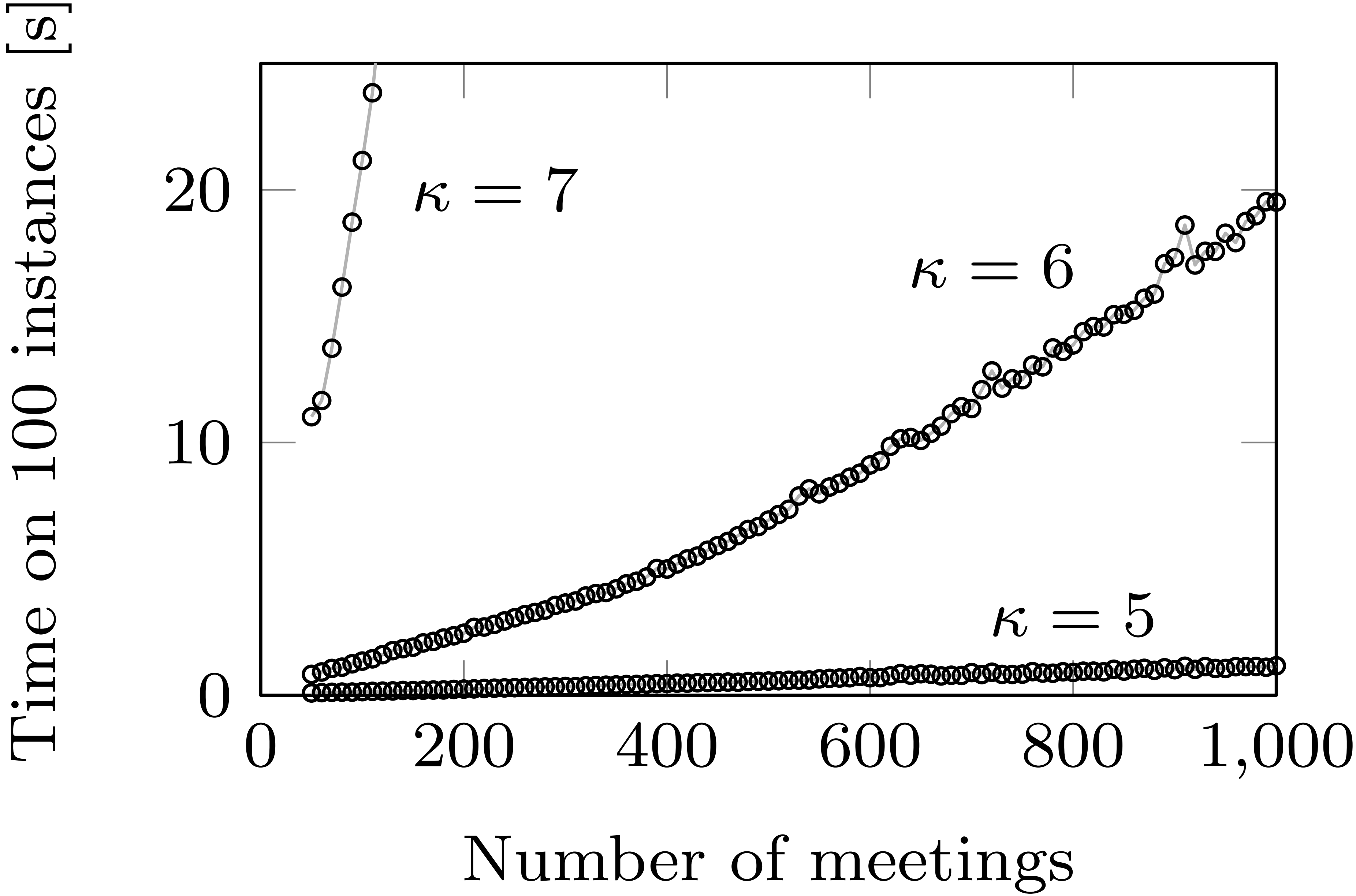
SAT: Runtime vs Number of Permutations



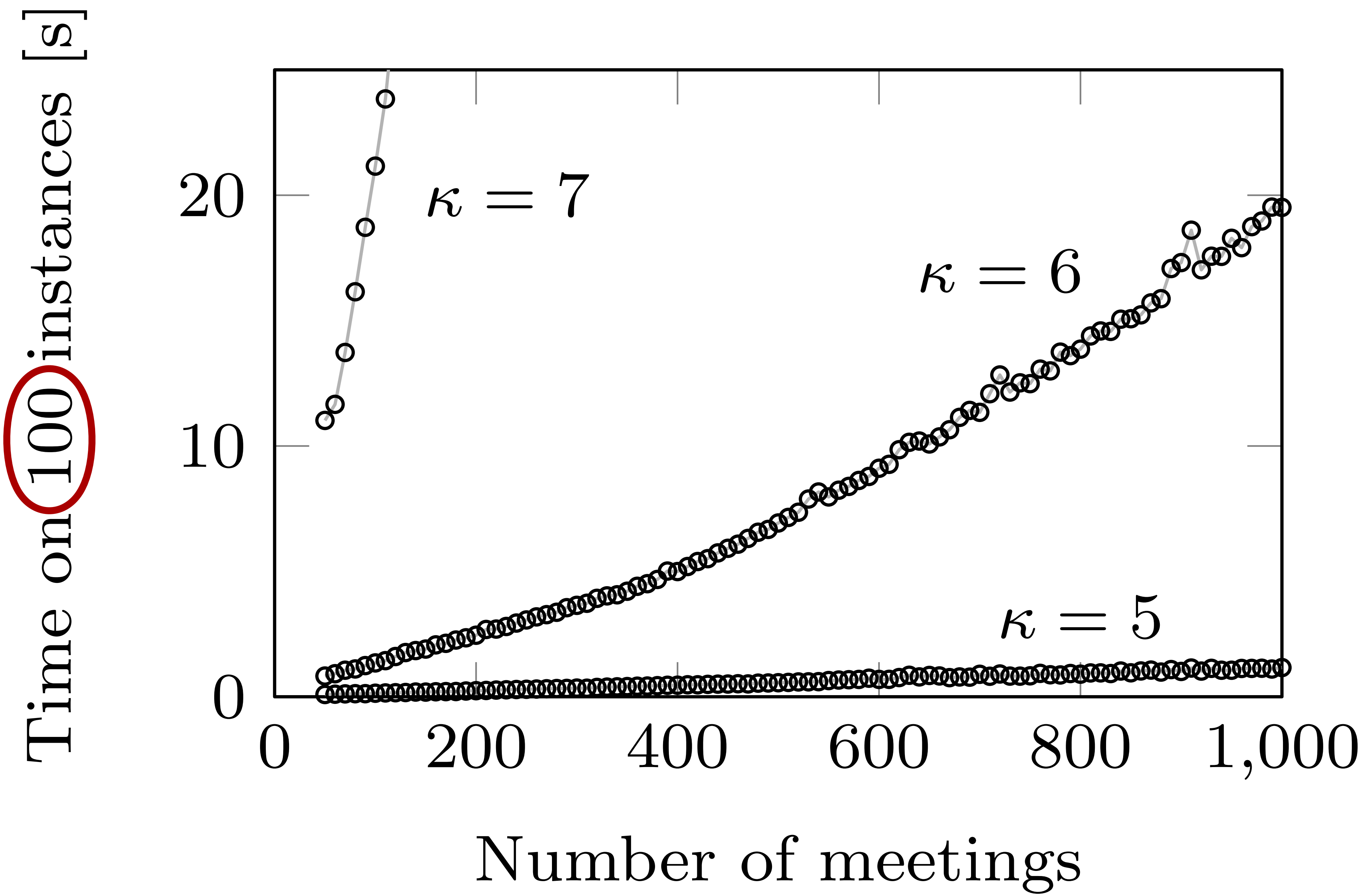
SAT: Runtime vs Number of Permutations



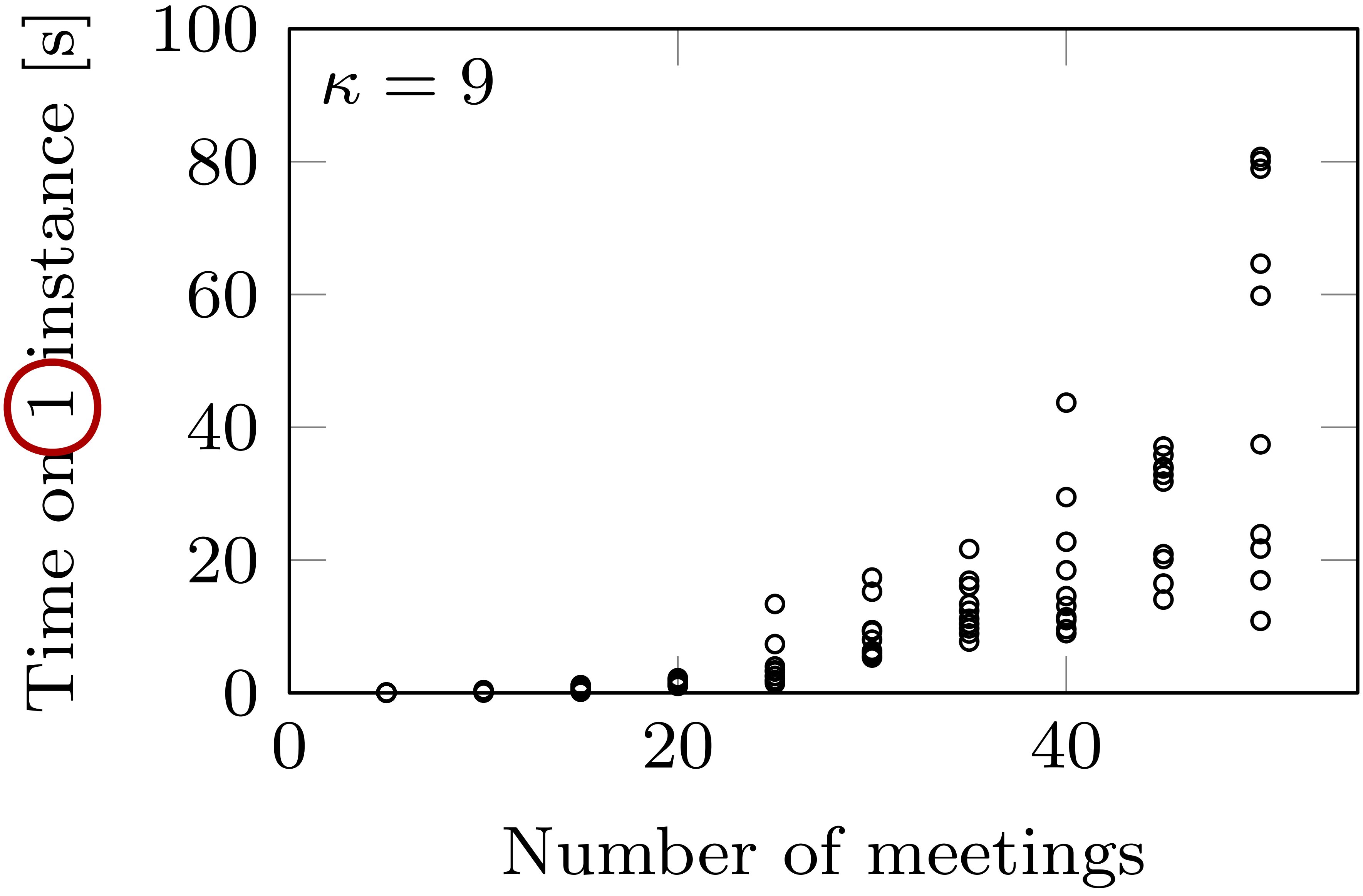
Uniform Random Instances: FPT



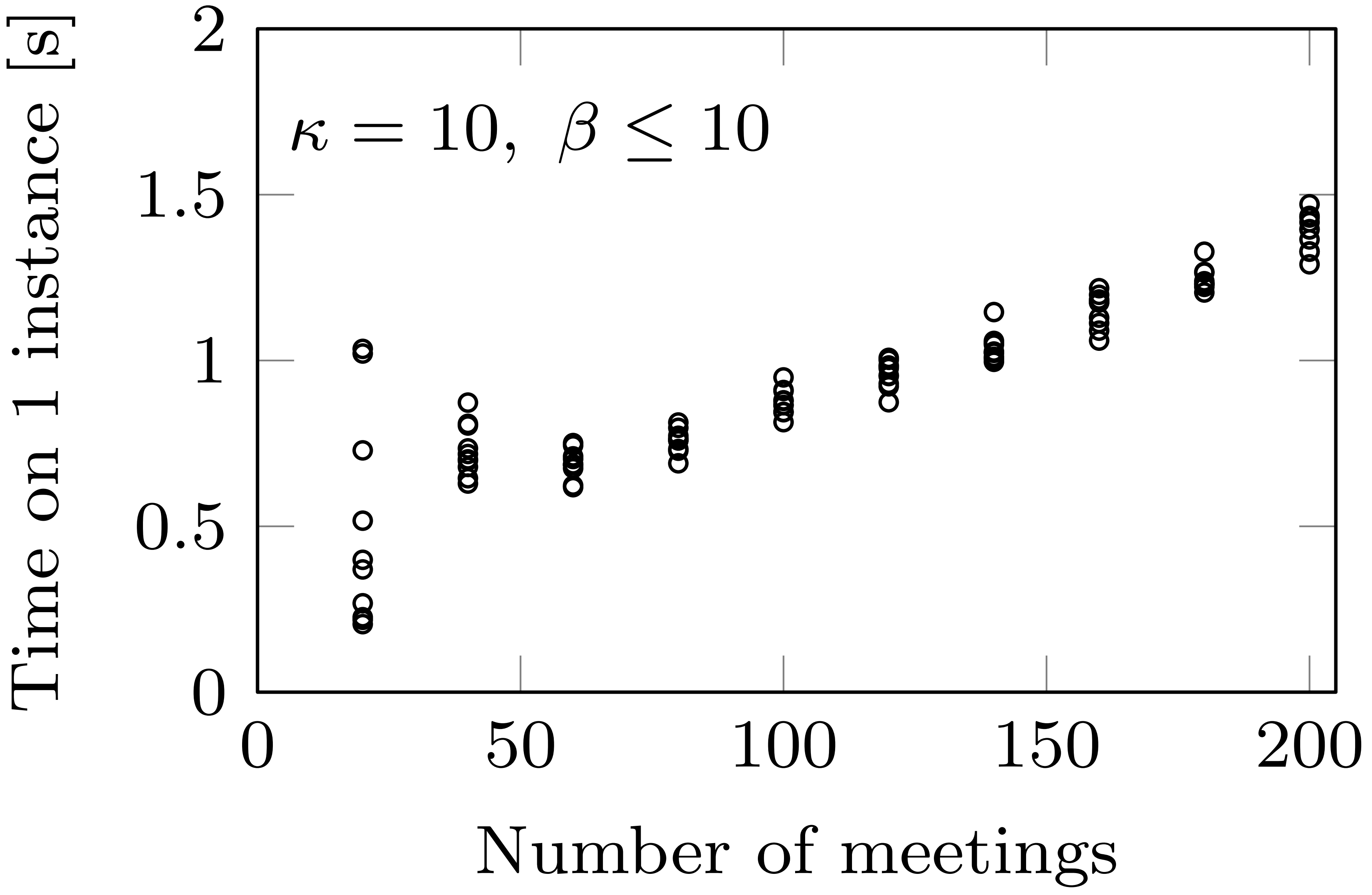
Uniform Random Instances: FPT



Uniform Random Instances: SAT



Small-OPT Random Instances: SAT



Results

Movie Instances:

	Our approach			Gronemann et al.		
Instance	cr	bc_{OPT}	Time [s]	cr_{OPT}	bc	Time [s]
Star Wars	54	10	3.77	39	18	0.99
The Matrix	21	4	2.86	12	8	0.77
Inception	51	12	1.54	35	20	2.02

Results

Movie Instances:

	Our approach			Gronemann et al.		
Instance	cr	bc_{OPT}	Time [s]	cr_{OPT}	bc	Time [s]
Star Wars	54	10	3.77	39	18	0.99
The Matrix	21	4	2.86	12	8	0.77
Inception	51	12	1.54	35	20	2.02

Results

Movie Instances:

	Our approach			Gronemann et al.		
Instance	cr	bc_{OPT}	Time [s]	cr_{OPT}	bc	Time [s]
Star Wars	54	10	3.77	39	18	0.99
The Matrix	21	4	2.86	12	8	0.77
Inception	51	12	1.54	35	20	2.02

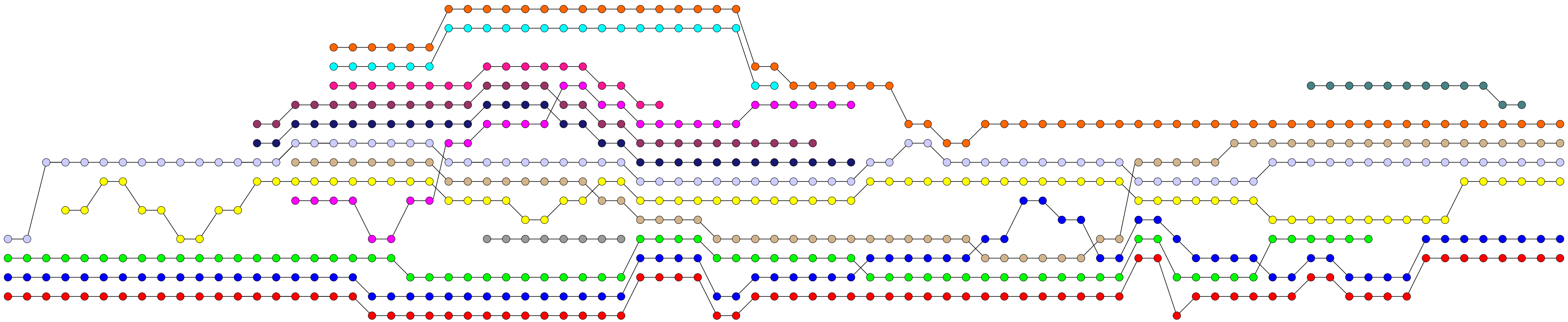
Results

Movie Instances:

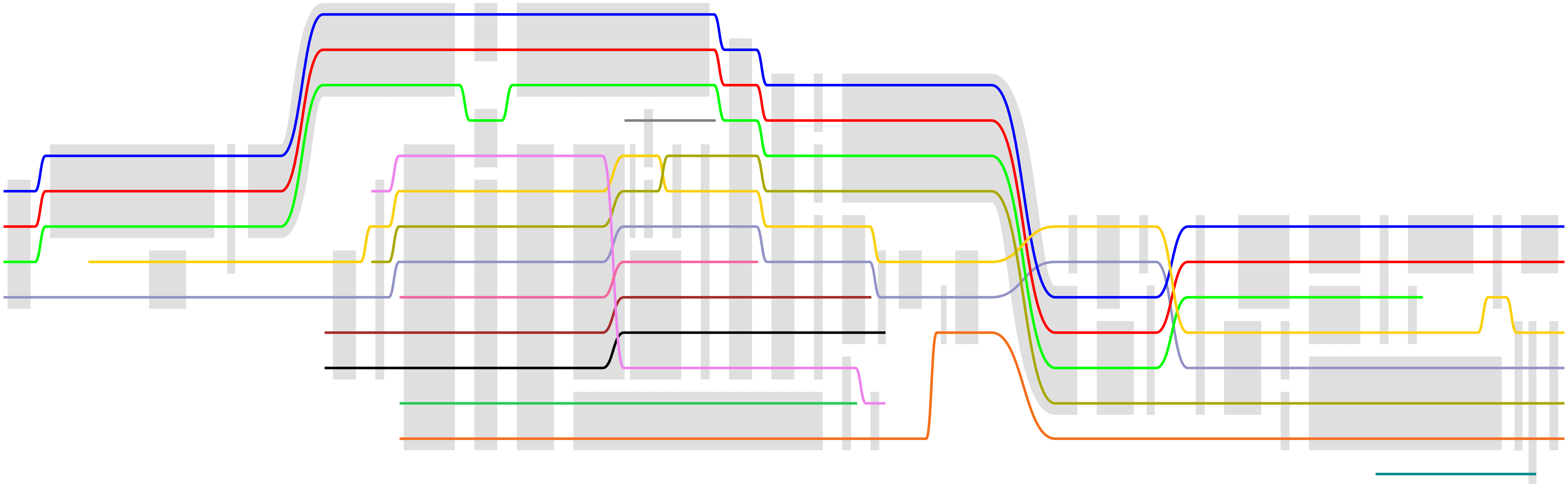
	Our approach			Gronemann et al.		
Instance	cr	bc_{OPT}	Time [s]	cr_{OPT}	bc	Time [s]
Star Wars	54	10	3.77	39	18	0.99
The Matrix	21	4	2.86	12	8	0.77
Inception	51	12	1.54	35	20	2.02

Example: The Matrix

Gronemann et al. 12 crossings / 8 block crossings

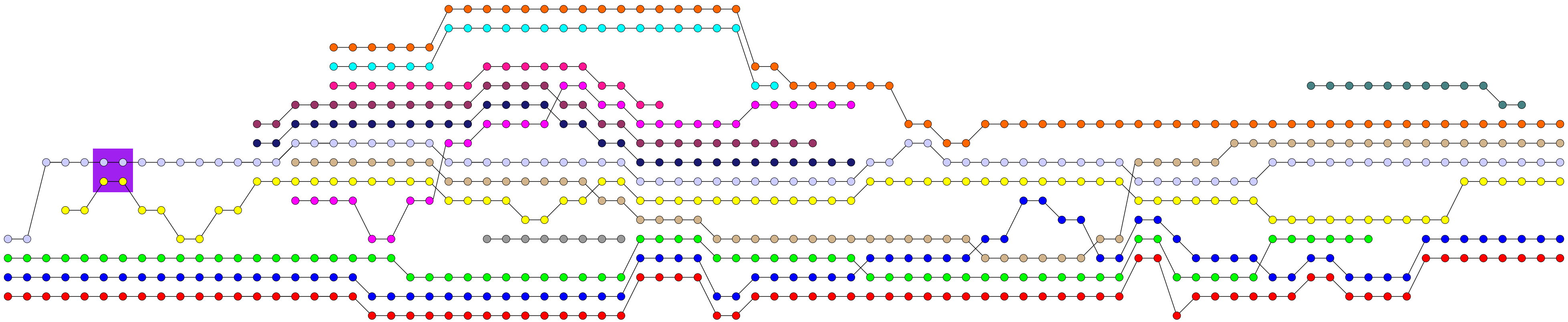


Our approach 21 crossings / 4 block crossings

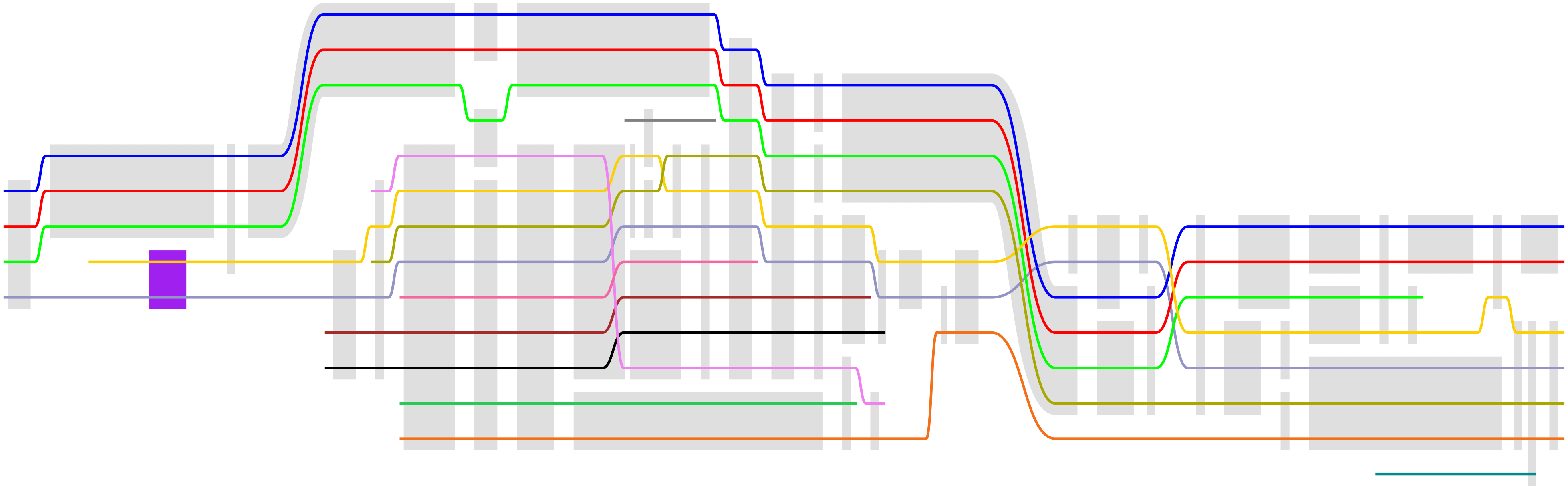


Example: The Matrix

Gronemann et al. 12 crossings / 8 block crossings

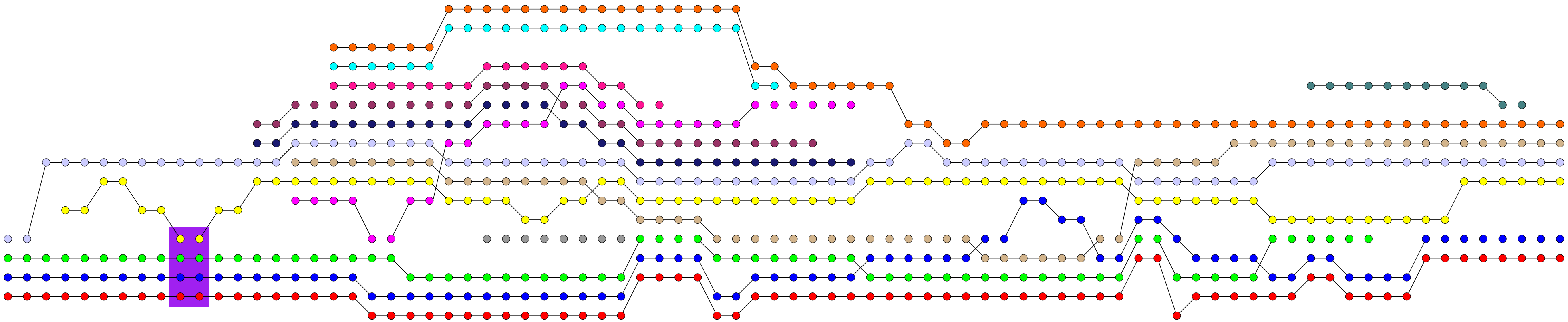


Our approach 21 crossings / 4 block crossings

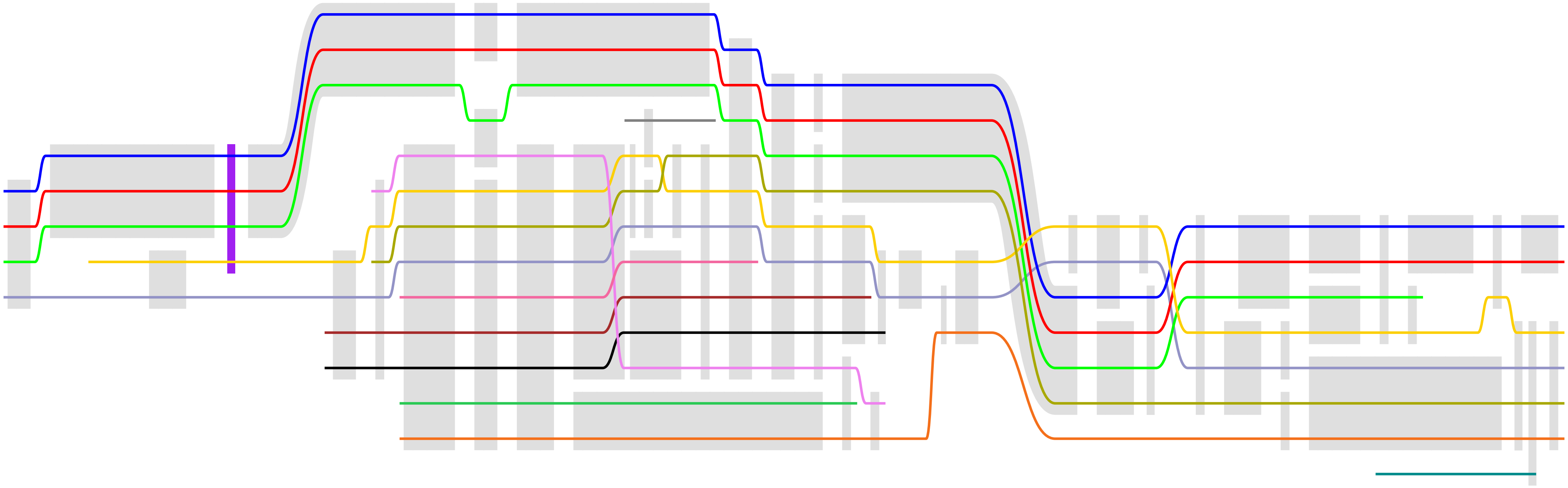


Example: The Matrix

Gronemann et al. 12 crossings / 8 block crossings

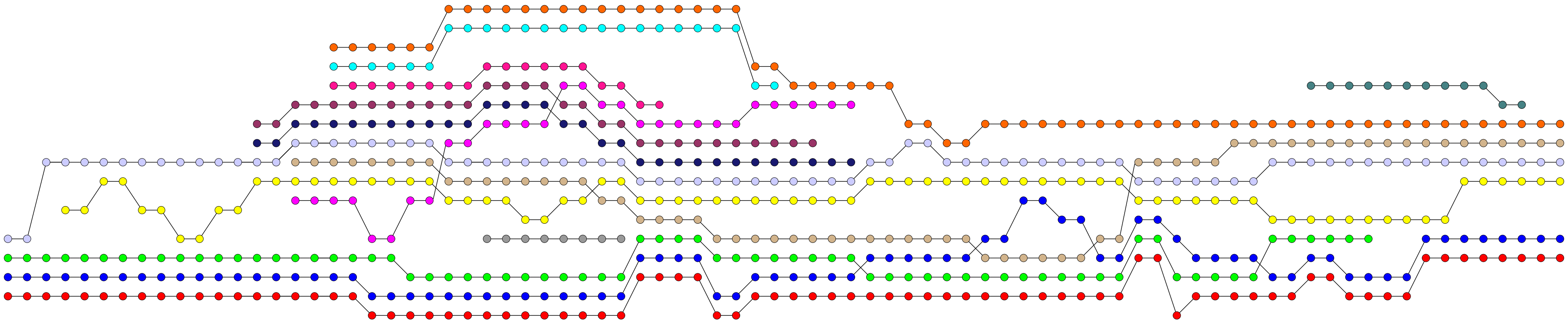


Our approach 21 crossings / 4 block crossings

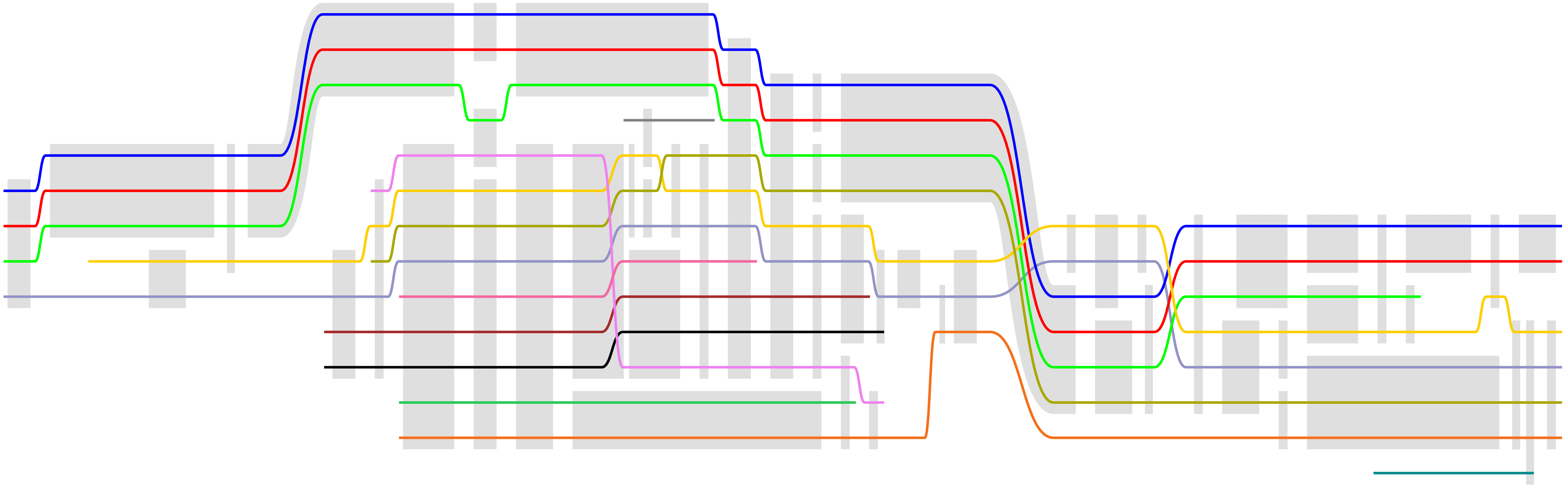


Example: The Matrix

Gronemann et al. 12 crossings / 8 block crossings



Our approach 21 crossings / 4 block crossings



Conclusion

- Our SAT approach is usable for real-world instances.
- Use SAT instead of ILP – turned out to be much faster!
- Source code is available online.

Conclusion

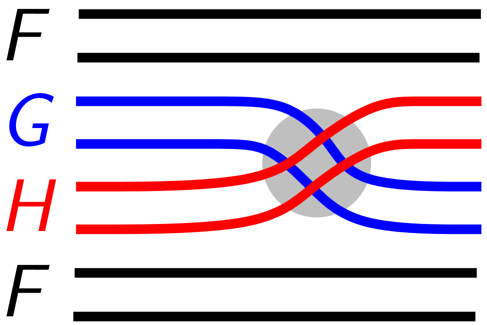
- Our SAT approach is usable for real-world instances.
- Use SAT instead of ILP – turned out to be much faster!
- Source code is available online.

Future work

- Try other (parallel) SAT solvers.
- Find more efficient way to model lifespans.
- Consider additional quality criteria of the drawing, e.g., minimize wiggles. [Fröschl & Nöllenburg, GD17]
- Perform a user study on the effect of block crossings, especially for storyline visualizations.

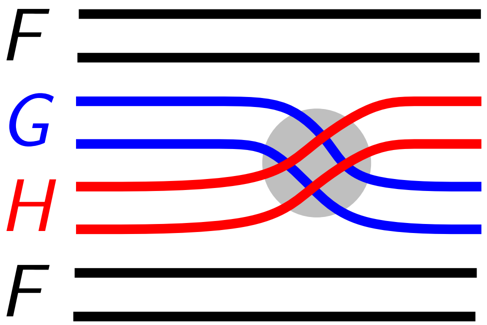
Appendix

Describing Block Crossings



Blocks G and H cross

Describing Block Crossings

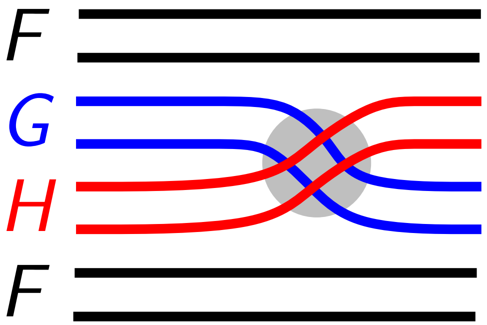


Blocks G and H cross

Constraints:

Exactly characters of G and H cross each other

Describing Block Crossings



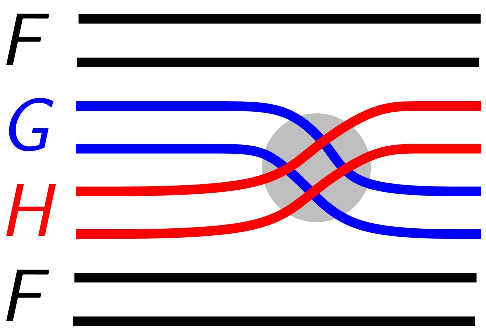
Blocks G and H cross

Constraints:

Exactly characters of G and H cross each other

$\chi^r_{ij} \Leftrightarrow g^r_i \wedge h^r_j$

Describing Block Crossings

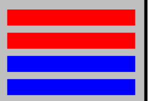


Blocks G and H cross

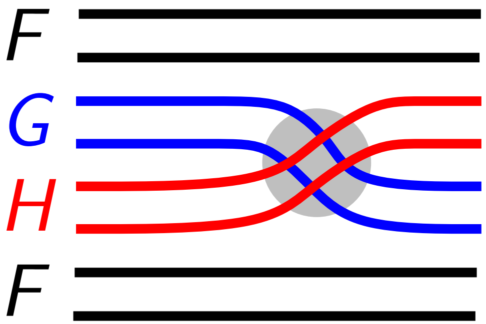
Constraints:

Exactly characters of G and H cross each other
$\chi_{ij}^r \Leftrightarrow g_i^r \wedge h_j^r$

G is above H



Describing Block Crossings



Blocks G and H cross

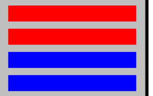
Constraints:

Exactly characters of G and H cross each other

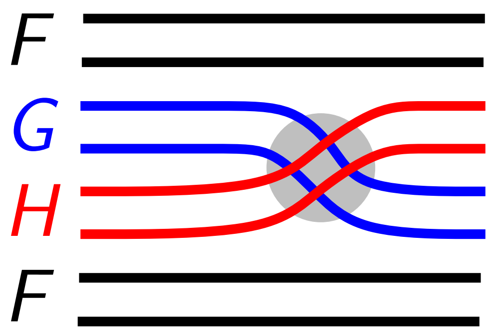
$$\chi_{ij}^r \Leftrightarrow g_i^r \wedge h_j^r$$

G is above H

$$g_i^r \wedge h_j^r \Rightarrow$$



Describing Block Crossings



Blocks G and H cross

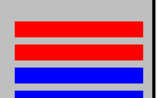
Constraints:

Exactly characters of G and H cross each other

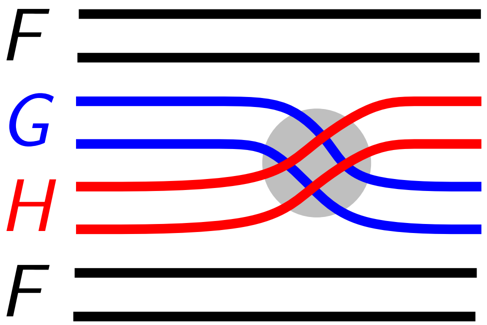
$\chi^r_{ij} \Leftrightarrow g^r_i \wedge h^r_j$

G is above H

$g^r_i \wedge h^r_j \Rightarrow x^r_{ij}$



Describing Block Crossings



Blocks G and H cross

Constraints:

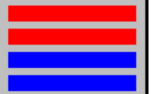
Exactly characters of G and H cross each other

$\chi^r_{ij} \Leftrightarrow g^r_i \wedge h^r_j$

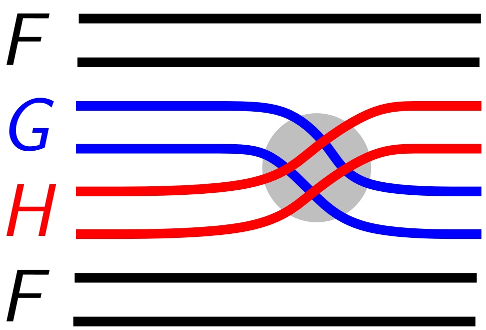
G and H are adjacent

G is above H

$g^r_i \wedge h^r_j \Rightarrow x^r_{ij}$



Describing Block Crossings



Blocks G and H cross

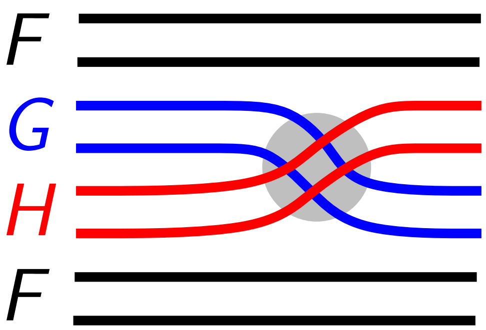
Constraints:

Exactly characters of G and H cross each other
$\chi_{ij}^r \Leftrightarrow g_i^r \wedge h_j^r$

G and H are adjacent
$x_{ij}^r \wedge x_{jk}^r \wedge g_i^r \wedge h_k^r \Rightarrow$

G is above H
$g_i^r \wedge h_j^r \Rightarrow x_{ij}^r$

Describing Block Crossings



Blocks G and H cross

Constraints:

Exactly characters of G and H cross each other

$\chi^r_{ij} \Leftrightarrow g^r_i \wedge h^r_j$

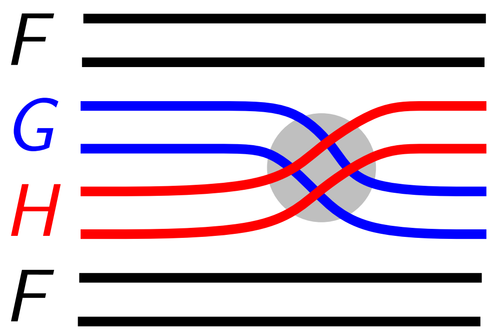
G and H are adjacent

$x^r_{ij} \wedge x^r_{jk} \wedge g^r_i \wedge h^r_k \Rightarrow \neg f^r_j$

G is above H

$g^r_i \wedge h^r_j \Rightarrow x^r_{ij}$

Describing Block Crossings



Blocks G and H cross

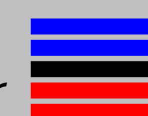
Constraints:

Exactly characters of G and H cross each other

$\chi^r_{ij} \Leftrightarrow g^r_i \wedge h^r_j$

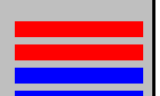
G and H are adjacent

$x^r_{ij} \wedge x^r_{jk} \wedge g^r_i \wedge h^r_k \Rightarrow \neg f^r_j$

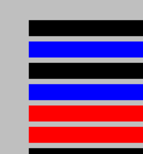


G is above H

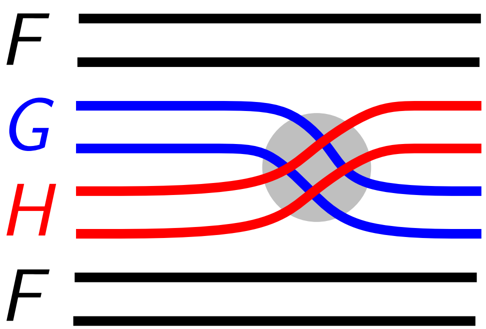
$g^r_i \wedge h^r_j \Rightarrow x^r_{ij}$



G and H are contiguous blocks



Describing Block Crossings



Blocks G and H cross

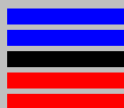
Constraints:

Exactly characters of G and H cross each other

$\chi_{ij}^r \Leftrightarrow g_i^r \wedge h_j^r$

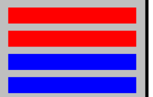
G and H are adjacent

$x_{ij}^r \wedge x_{jk}^r \wedge g_i^r \wedge h_k^r \Rightarrow \neg f_j^r$



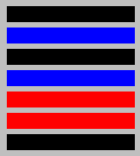
G is above H

$g_i^r \wedge h_j^r \Rightarrow x_{ij}^r$

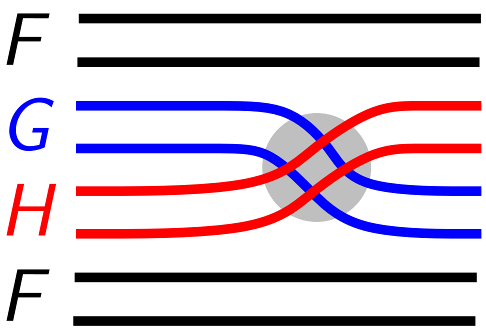


G and H are contiguous blocks

$x_{ij}^r \wedge x_{jk}^r \wedge g_i^r \wedge g_k^r \Rightarrow$



Describing Block Crossings



Blocks G and H cross

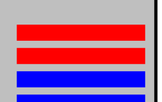
Constraints:

Exactly characters of G and H cross each other

$\chi_{ij}^{r} \Leftrightarrow g_i^r \wedge h_j^r$

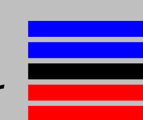
G is above H

$g_i^r \wedge h_j^r \Rightarrow x_{ij}^r$



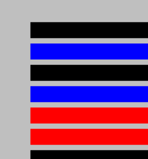
G and H are adjacent

$x_{ij}^r \wedge x_{jk}^r \wedge g_i^r \wedge h_k^r \Rightarrow \neg f_j^r$

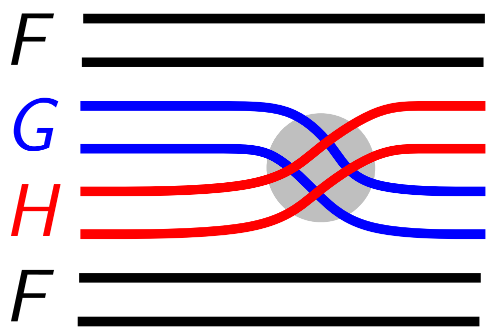


G and H are contiguous blocks

$x_{ij}^r \wedge x_{jk}^r \wedge g_i^r \wedge g_k^r \Rightarrow g_j^r$



Describing Block Crossings



Blocks G and H cross

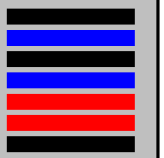
Constraints:

Exactly characters of G and H cross each other
$\chi^r_{ij} \Leftrightarrow g^r_i \wedge h^r_j$

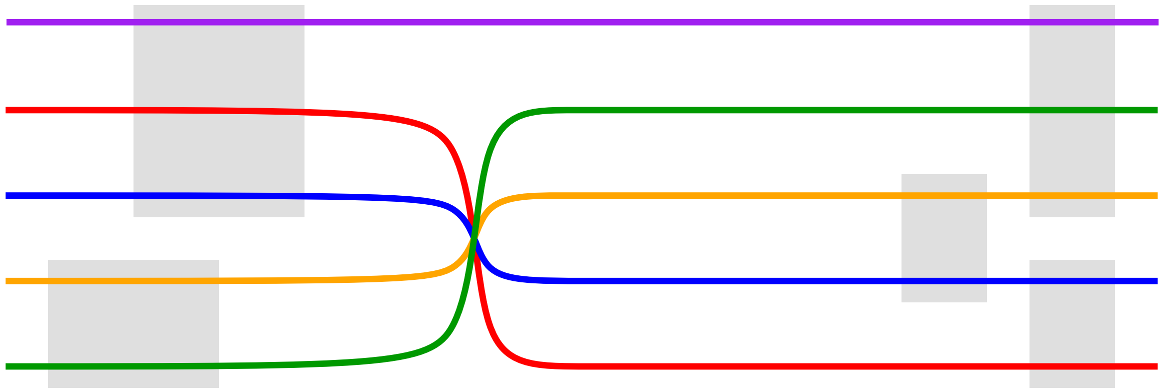
G and H are adjacent
$x^r_{ij} \wedge x^r_{jk} \wedge g^r_i \wedge h^r_k \Rightarrow \neg f^r_j$

G is above H
$g^r_i \wedge h^r_j \Rightarrow x^r_{ij}$

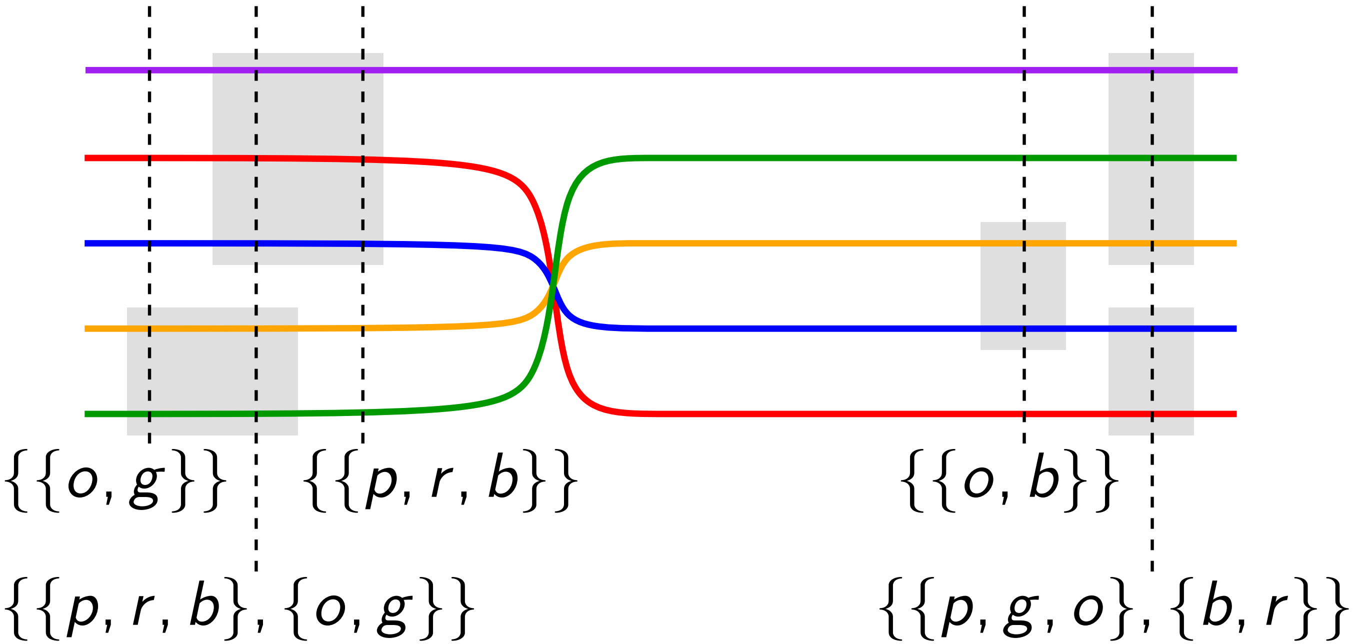
G and H are contiguous blocks
$x^r_{ij} \wedge x^r_{jk} \wedge g^r_i \wedge g^r_k \Rightarrow g^r_j$
$x^r_{ij} \wedge x^r_{jk} \wedge h^r_i \wedge h^r_k \Rightarrow h^r_j$



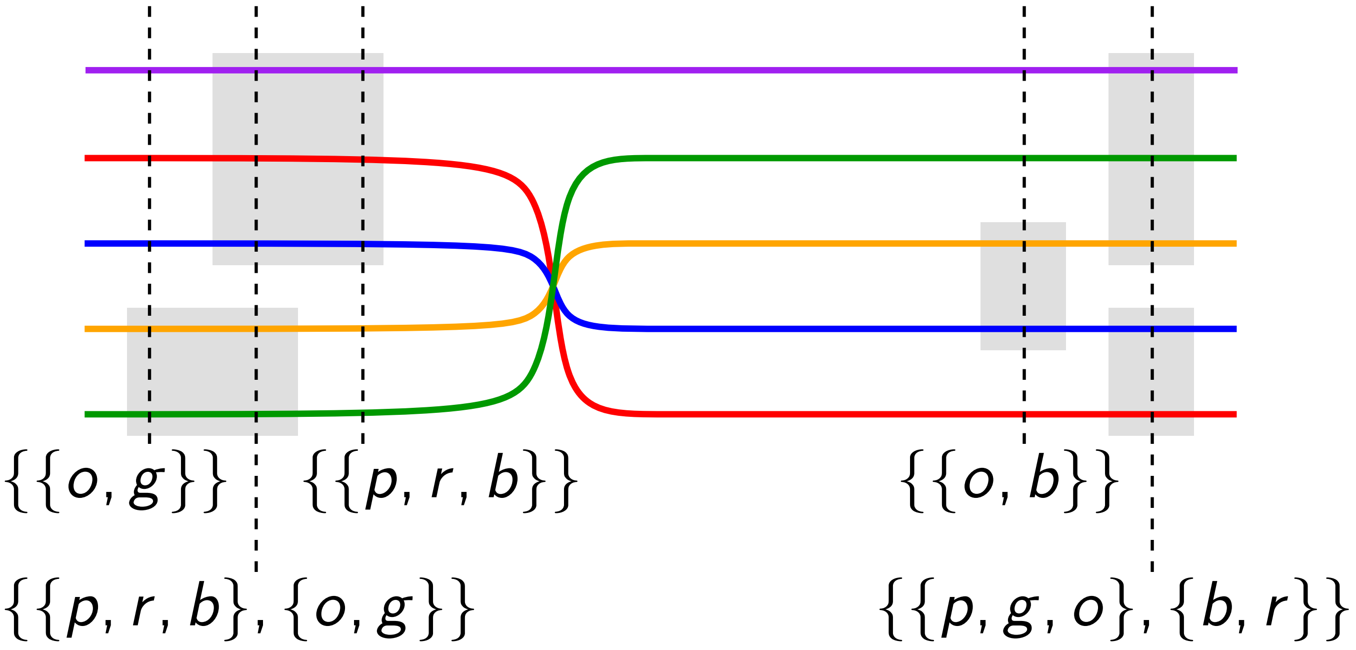
Meeting Groups

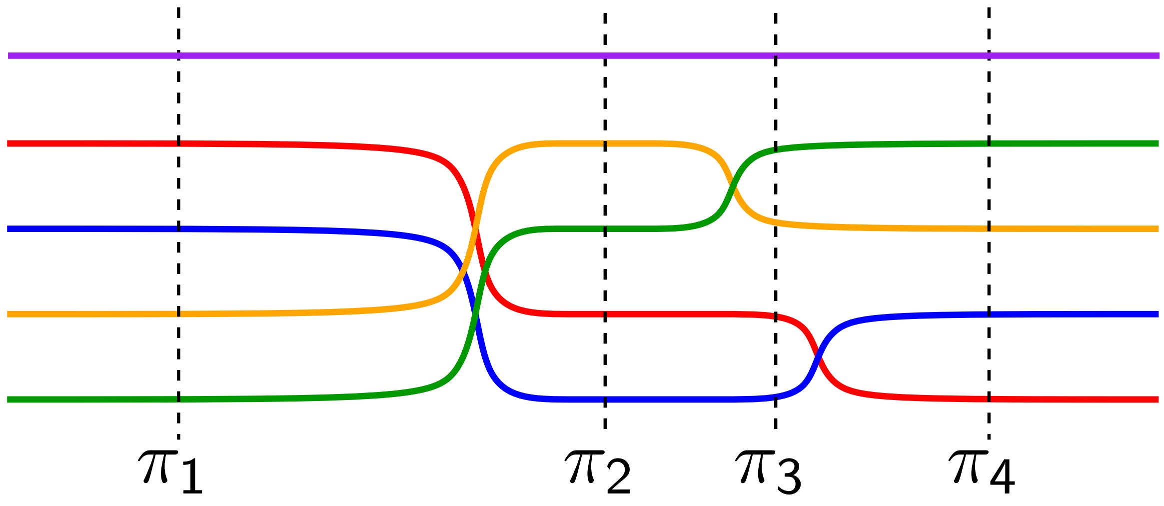


Meeting Groups

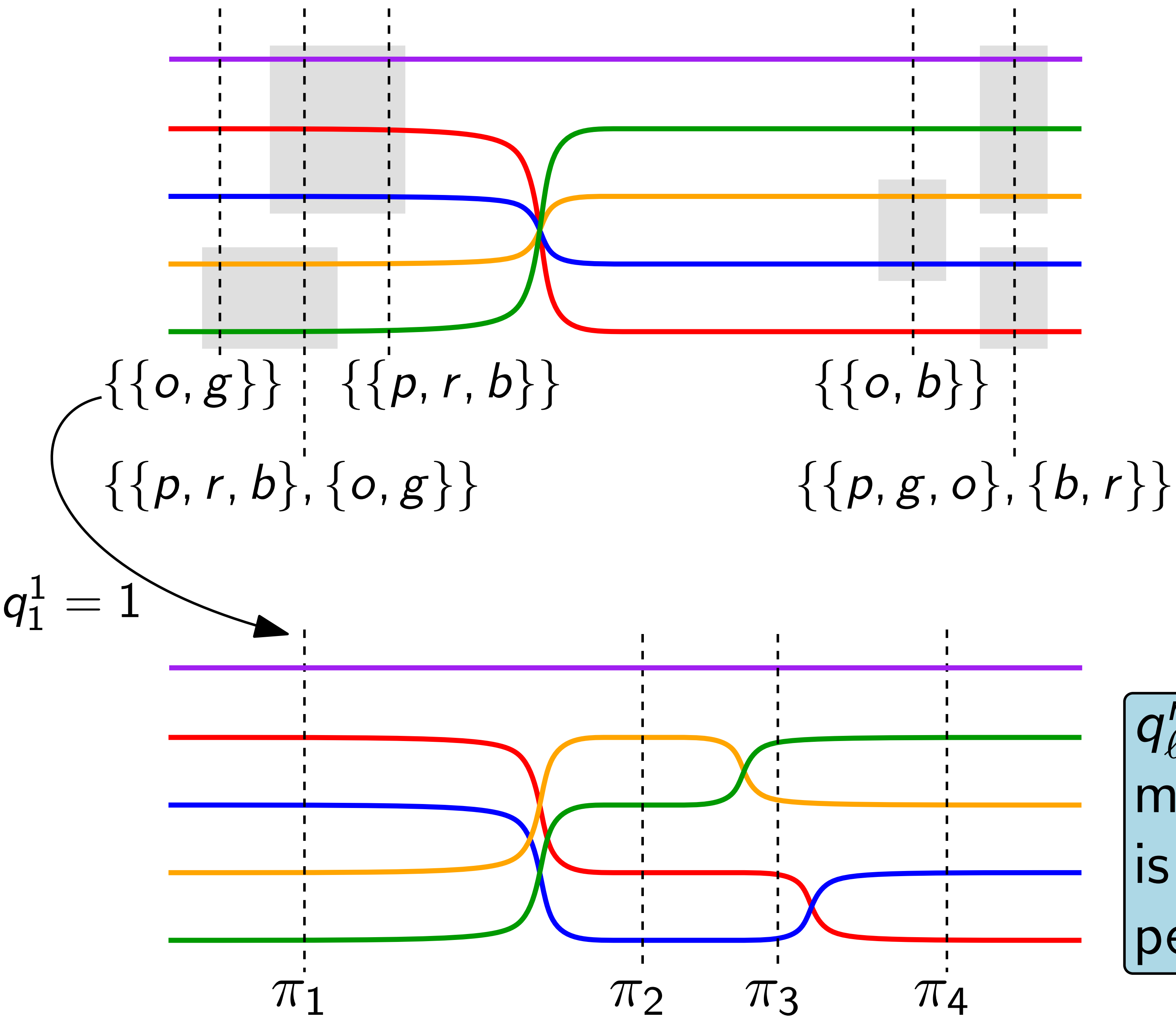


Meeting Groups



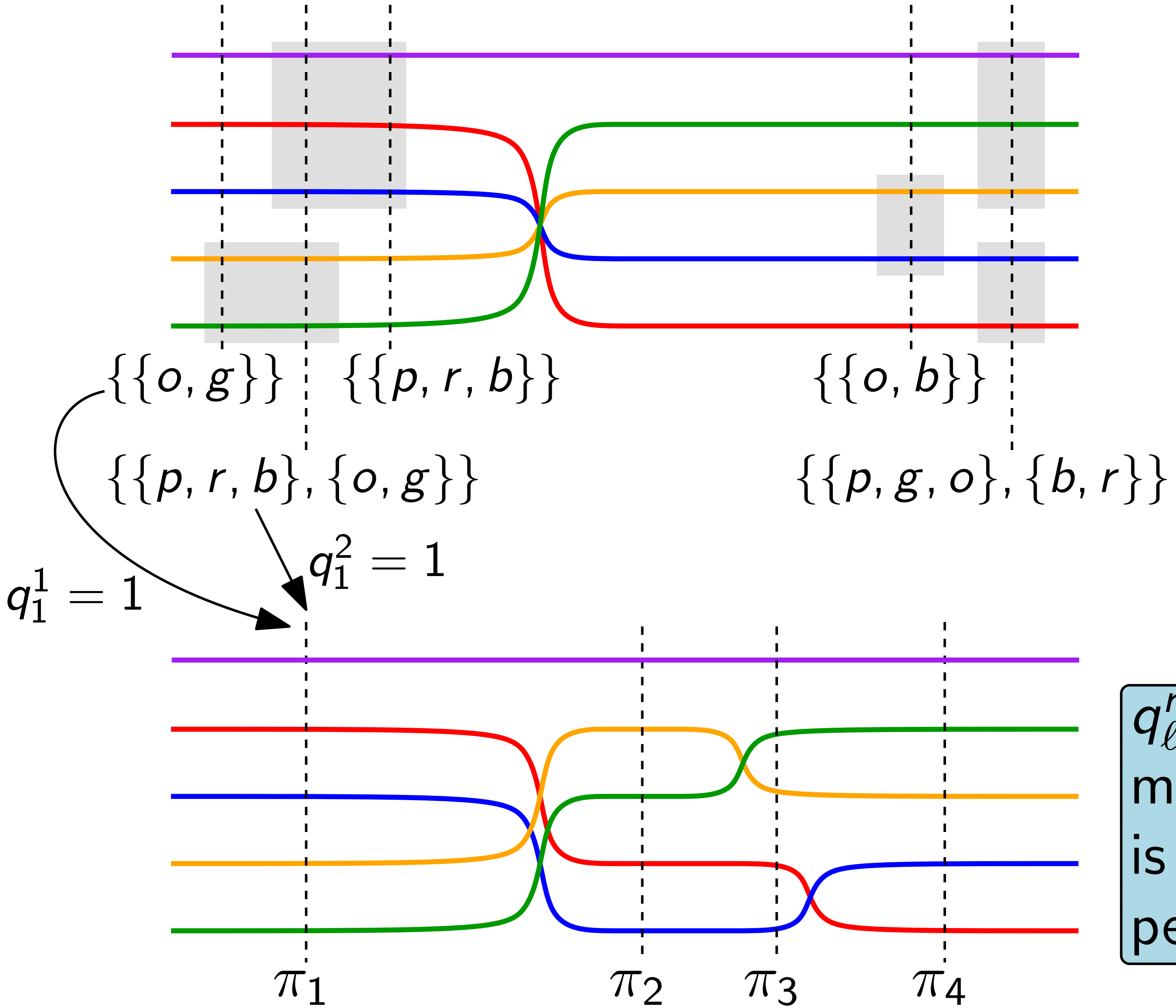


Meeting Groups



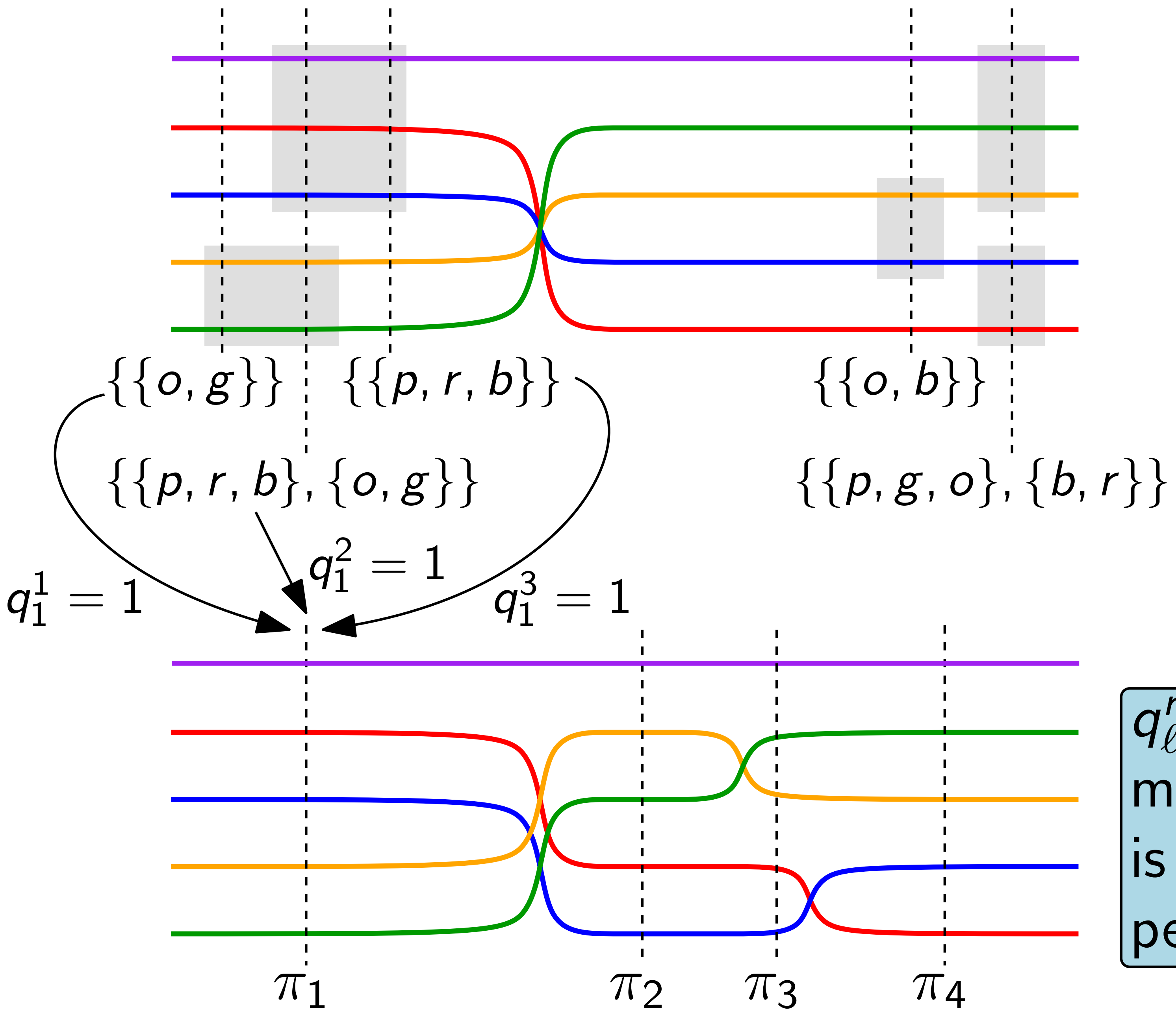
$q_\ell^r = 1 \Leftrightarrow$ meeting group ℓ is assigned to permutation π_r

Meeting Groups



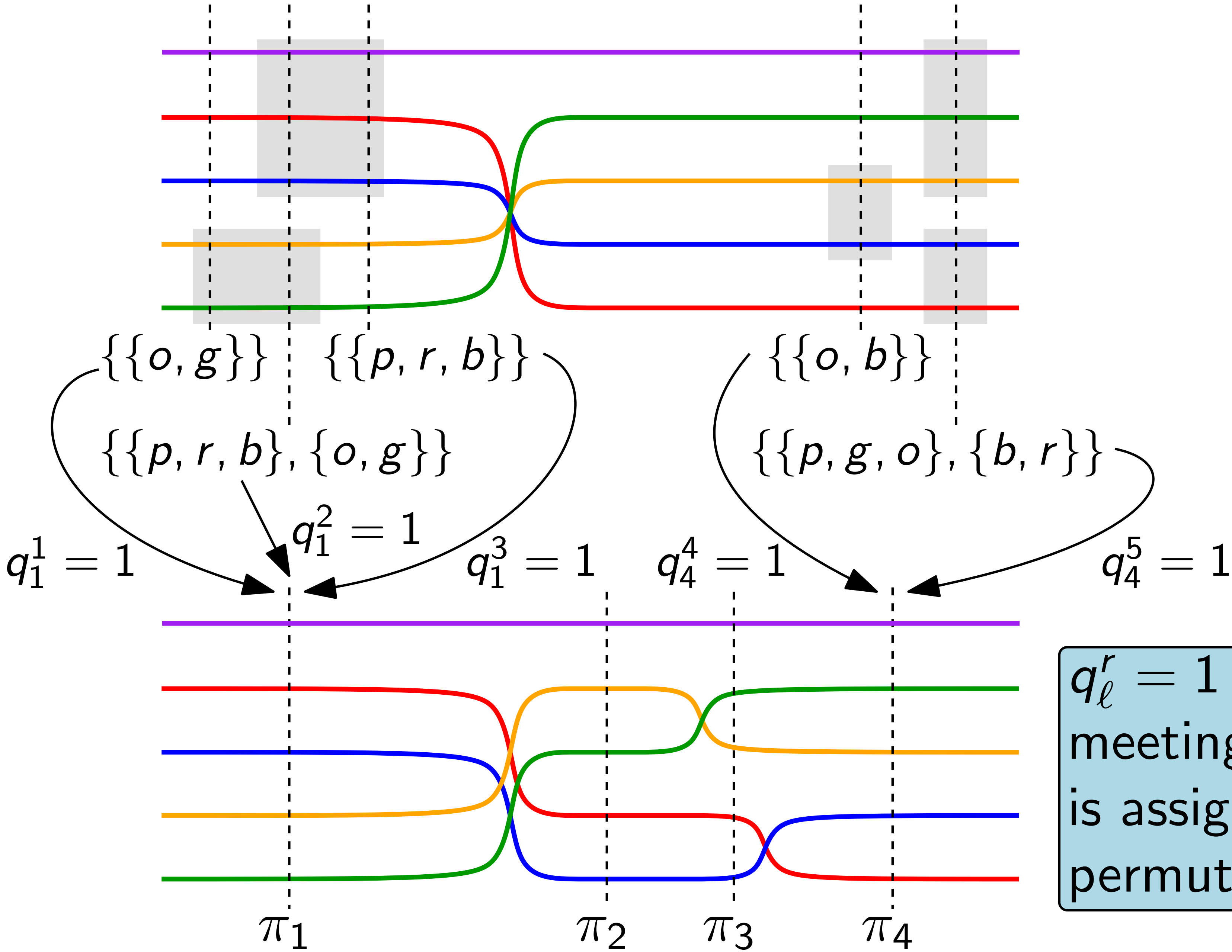
$q_\ell^r = 1 \Leftrightarrow$ meeting group ℓ is assigned to permutation π_r

Meeting Groups



$q_\ell^r = 1 \Leftrightarrow$ meeting group ℓ is assigned to permutation π_r

Meeting Groups



$q_\ell^r = 1 \Leftrightarrow$ meeting group ℓ is assigned to permutation π_r

Mapping the Meeting Groups

- Map the meeting groups in the right order:
- Map every meeting group exactly once:
- Force meeting characters to be next to each other:

Mapping the Meeting Groups

- Map the meeting groups in the right order:
  $q_\ell^r \Rightarrow \bigvee_{u=1}^{r} q_{\ell-1}^u$
- Map every meeting group exactly once:
- Force meeting characters to be next to each other:

Mapping the Meeting Groups

- Map the meeting groups in the right order:
  $q_\ell^r \Rightarrow \bigvee_{u=1}^{r} q_{\ell-1}^u$
- Map every meeting group exactly once:
  $\bigvee_{r=1}^{\lambda} q_\ell^r$ and $\neg(q_\ell^r \wedge q_\ell^p)$
- Force meeting characters to be next to each other:

Mapping the Meeting Groups

- Map the meeting groups in the right order:
  $q_\ell^r \Rightarrow \bigvee_{u=1}^{r} q_{\ell-1}^{u}$

- Map every meeting group exactly once:
  $\bigvee_{r=1}^{\lambda} q_\ell^r$ and $\neg(q_\ell^r \wedge q_\ell^p)$

- Force meeting characters to be next to each other:
  If i and k are part of the same meeting in meeting group ℓ and j is not: $q_\ell^r \Rightarrow (x_{ij}^r = x_{kj}^r)$