Maximum Independent Set in 2-Direction Outer-Segment Graphs

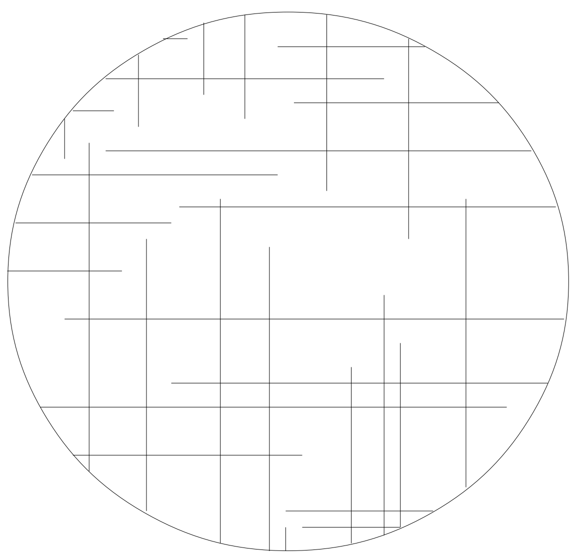
Holger Flier, Matúš Mihalák, Peter Widmayer,
Anna Zych

ETH Zürich

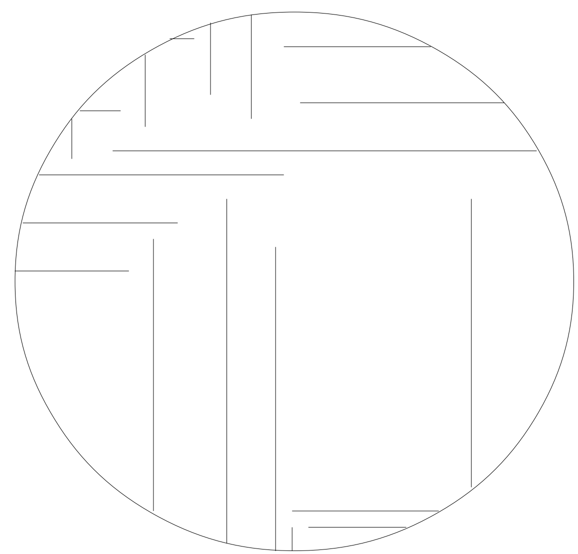
The problem: Maximum Independent Set of Segments

aligned horizontally or vertically inside a disk
with one endpoint on the boundary

Input: A set of segments as above



Output: A subset of segments pairwise disjoint



The problem reduces to Maximum Independent Set (MIS) Problem in corresponding intersection graphs

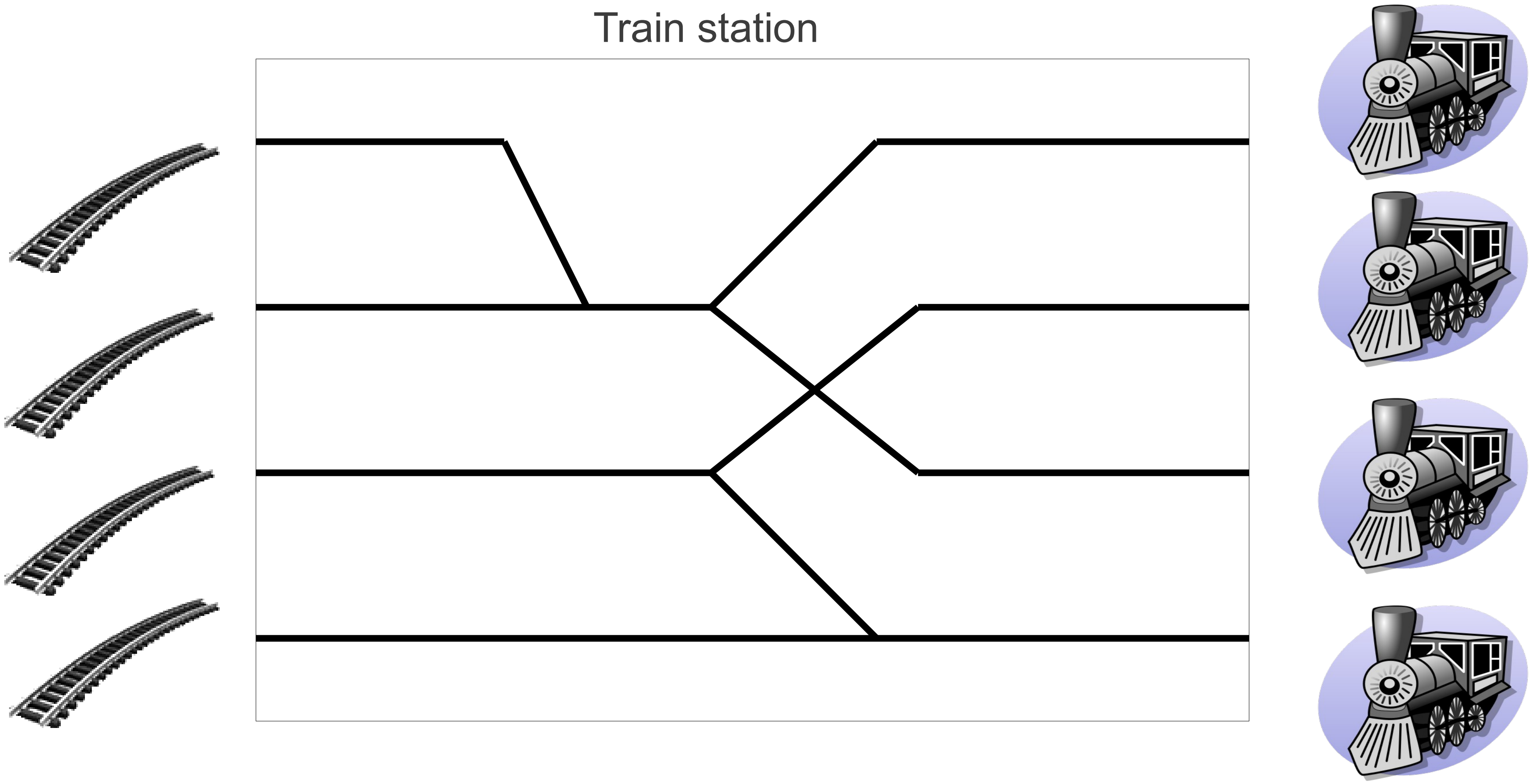
Our result

- MIS (of segments) is polynomial for segments in 2 directions with one endpoint fixed on a boundary of a disk

Related results [Kratochvíl, Nešetřil 1990]

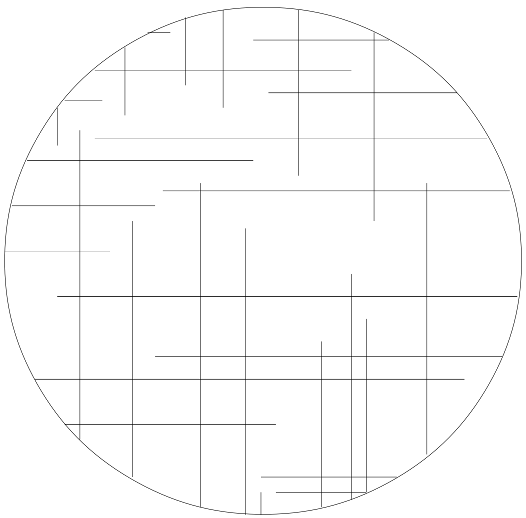
- MIS is NP-hard for segments in the plane
  - aligned in 2 directions...
  - ...or in 3 directions but no two segments in one direction intersect

Why this problem?



MIS is open for outer-string graphs

Easy vs. NP-hard

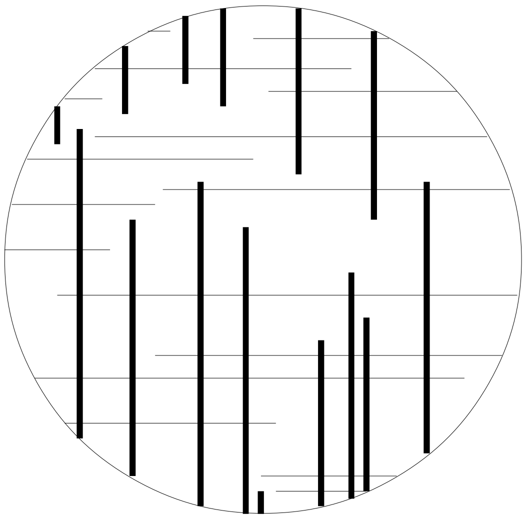


Easy instance:
    opposite segments do not intersect

The intersection graph is bipartite:

$$G=(V, E): V=A\cup B, E\subseteq A\times B$$

Easy vs. NP-hard



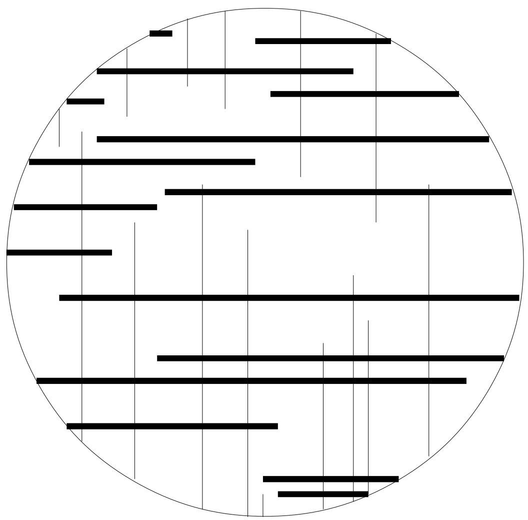
Easy instance:
opposite segments do not intersect

The intersection graph is bipartite:

$$G=(V,E): V=A\cup B, E\subseteq A\times B$$

A: vertical segments

Easy vs. NP-hard

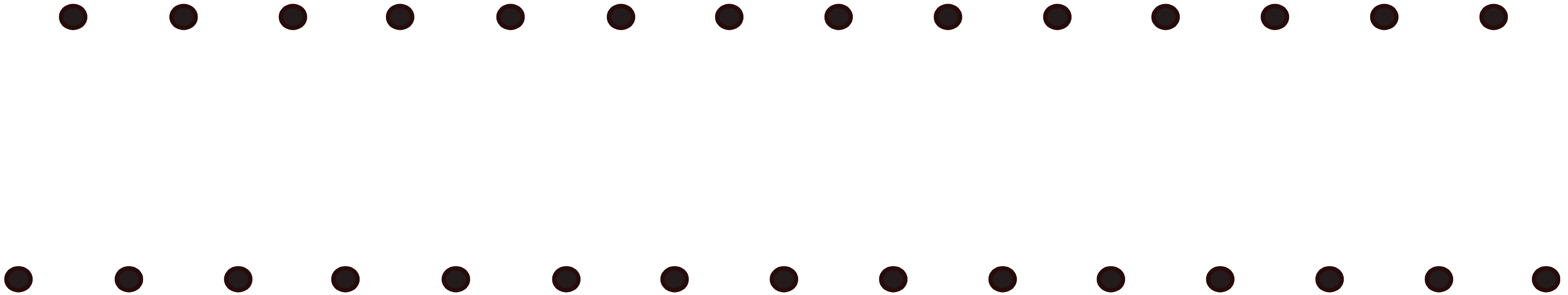


Easy instance:
opposite segments do not intersect

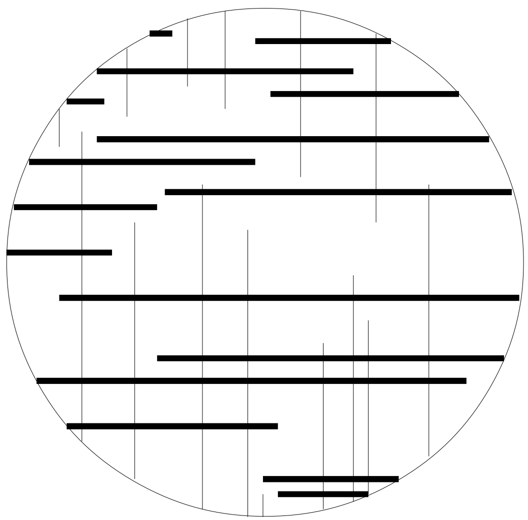
The intersection graph is bipartite:

$$G=(V, E): V=A\cup B, E\subseteq A\times B$$

A: vertical segments
B: horizontal segments



Easy vs. NP-hard

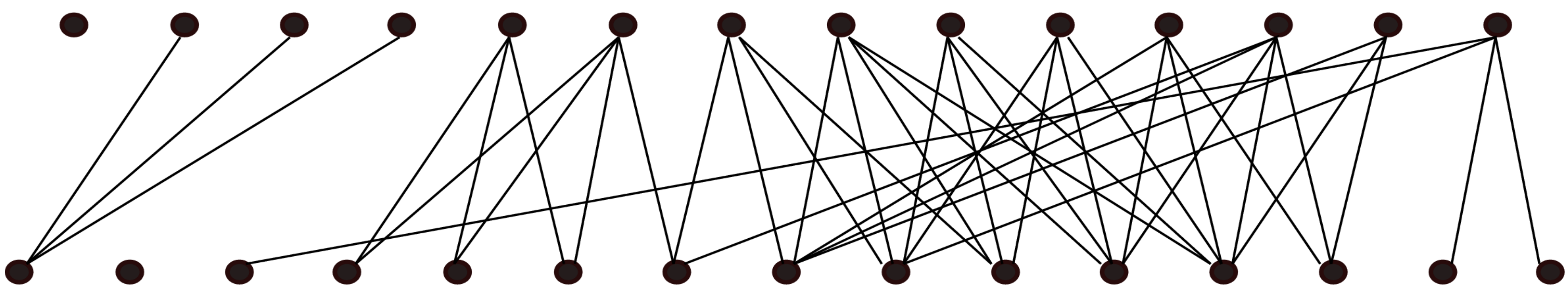


Easy (bipartite) instance:
opposite segments do not intersect

The intersection graph is bipartite:

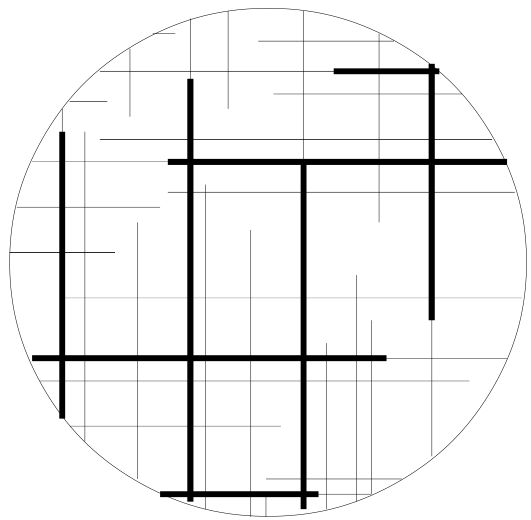
$$G=(V,E): V=A\cup B, E\subseteq A\times B$$

A: vertical segments
B: horizontal segments
E: edges connect intersecting pairs:
vertical with horizontal



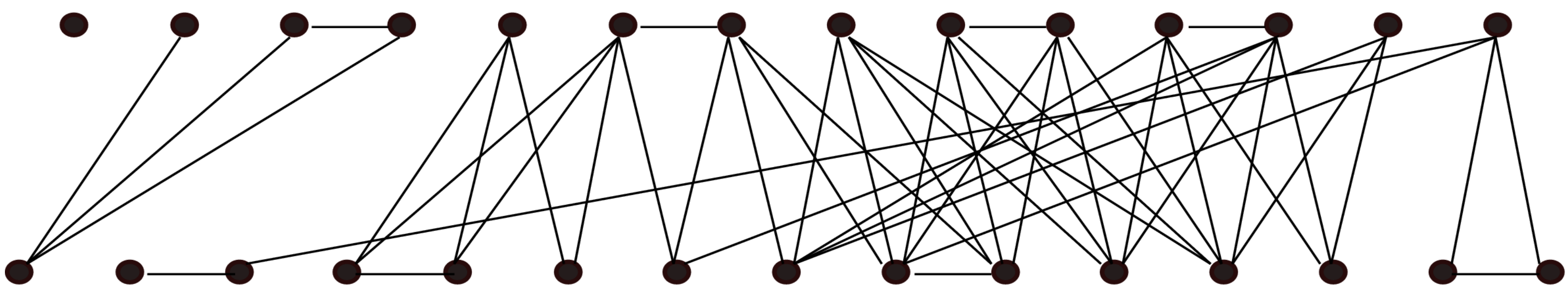
Independent set in bipartite graphs is polynomial!

Easy vs. NP-hard



Not easy instance:
opposite segments intersect

The intersection graph is bipartite
plus two matchings:



Independent set is NP-hard in such graphs!

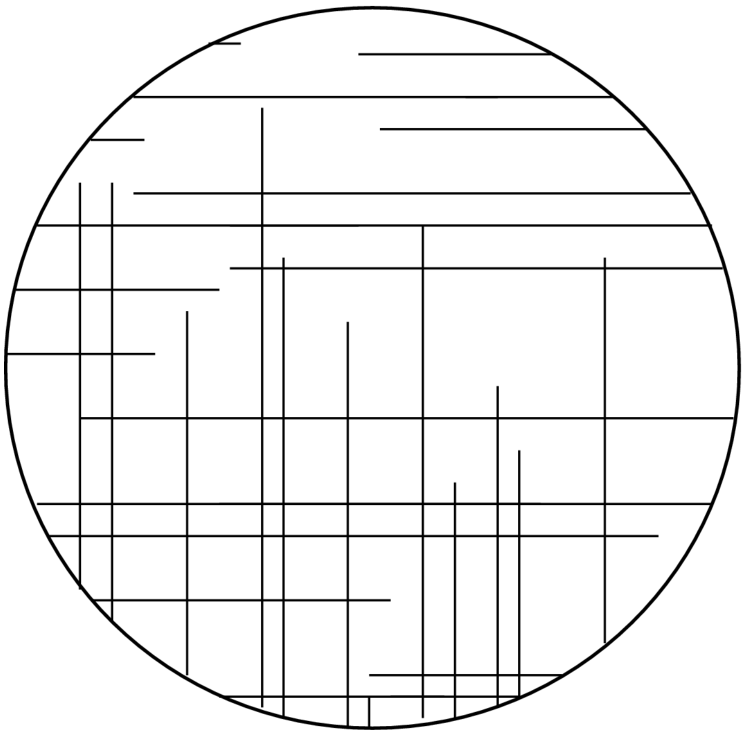
Easy vs. NP-hard

Easy instance:
no horizontal segments (one direction)

The intersection graph is a **matching**:
also a bipartite instance

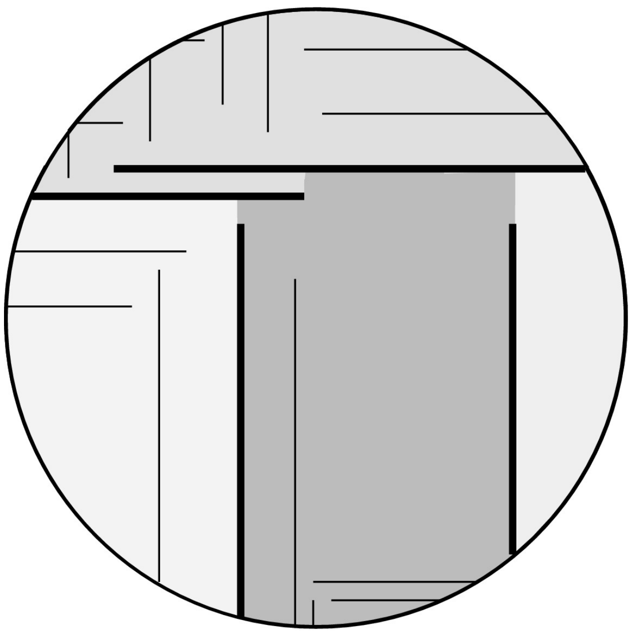
3 side instances

no top segments



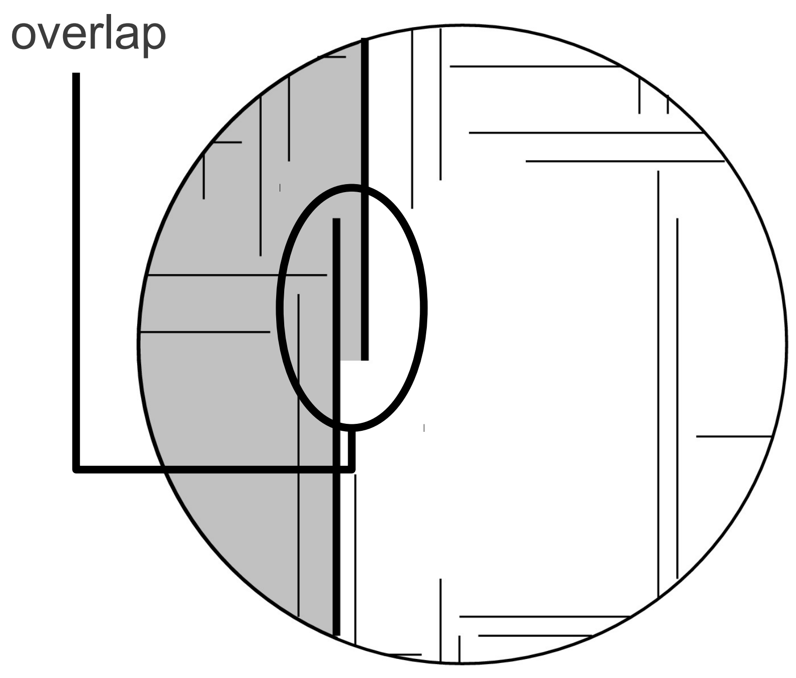
Not trivial:
solvable in polynomial time
with dynamic programming
(still to come)

General case: Decomposition to simpler instances

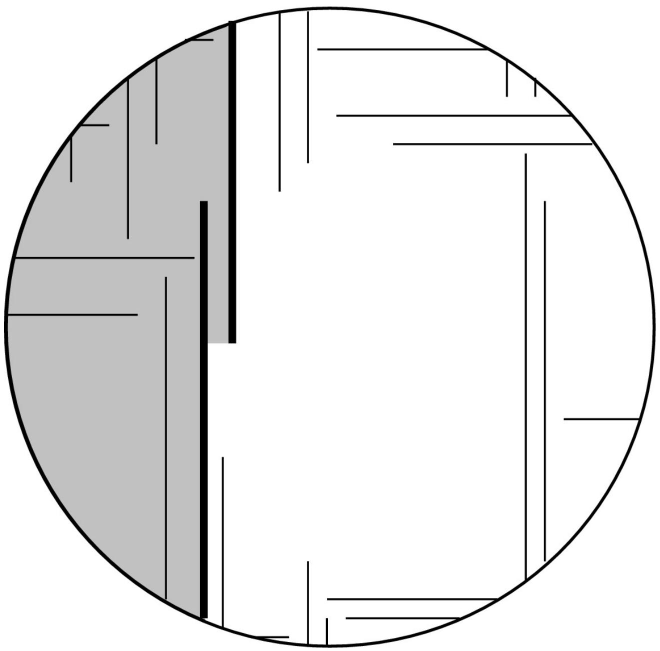


- Decompose the input instance into few 3 side instances
- Decomposition is determined by a constant number of segments
- Exhaustively search for the decomposition segments and solve 3 side instances

Getting rid of overlaps



Getting rid of overlaps

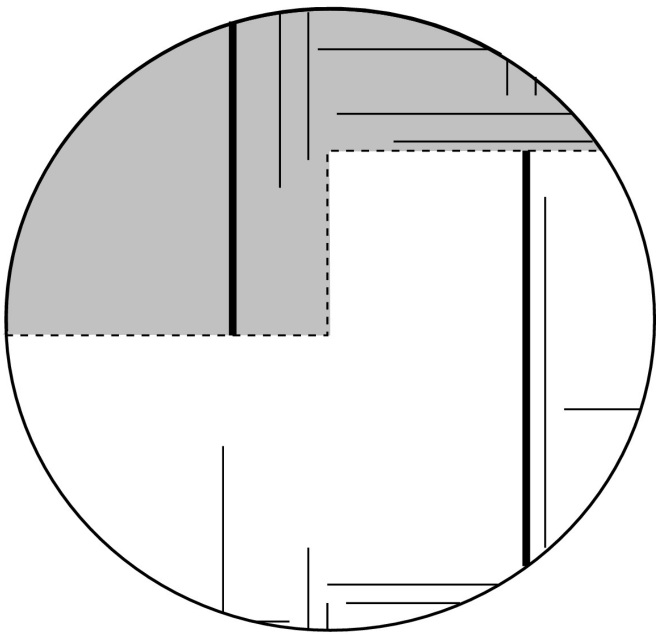


Instance of interest:

- No overlapping pair of left segments in optimum
- No overlapping pair of right segments in optimum

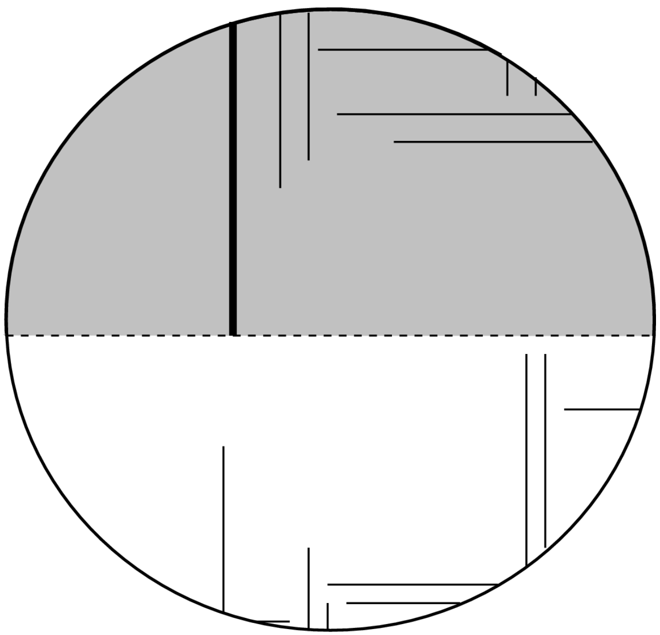
This partition is determined by four segments of the optimum

Completing the partition into 3 side instances



Case 1: an overlapping pair of a left and a right segment

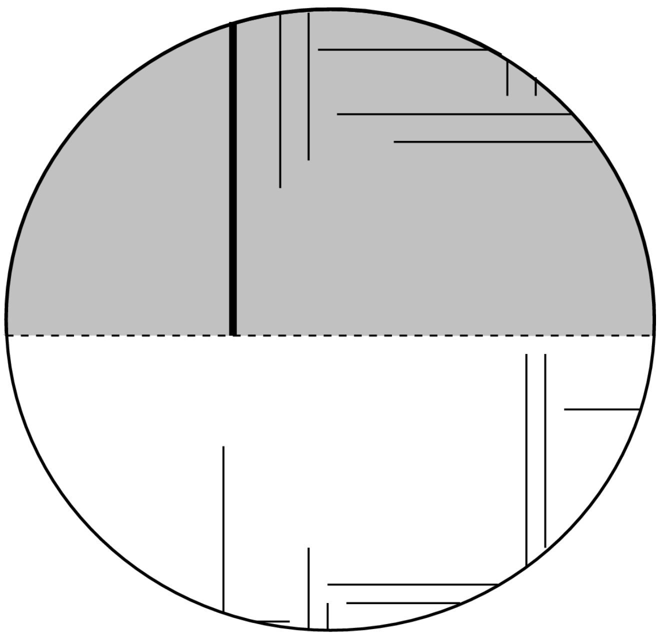
Completing the partition into 3 side instances



Case 1: an overlapping pair of
a left and a right segment √

Case 2: no overlapping pair

Completing the partition into 3 side instances

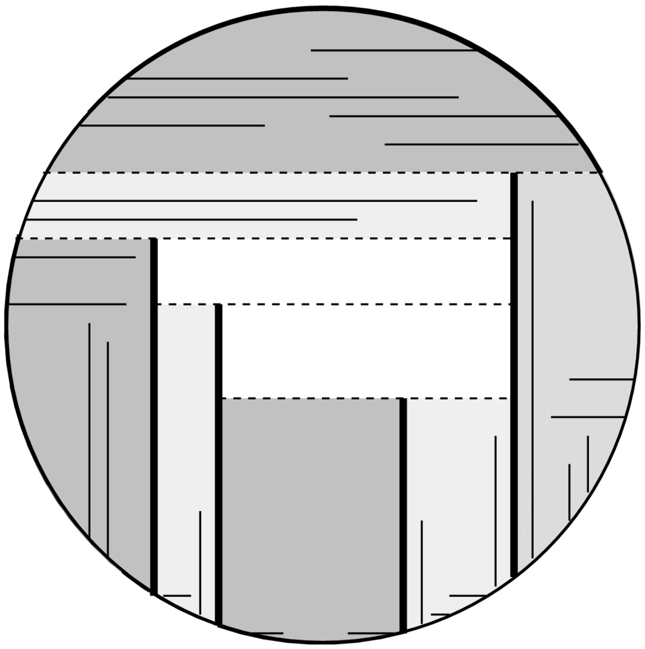


Case 1: an overlapping pair of a left and a right segment √

Case 2: no overlapping pair √

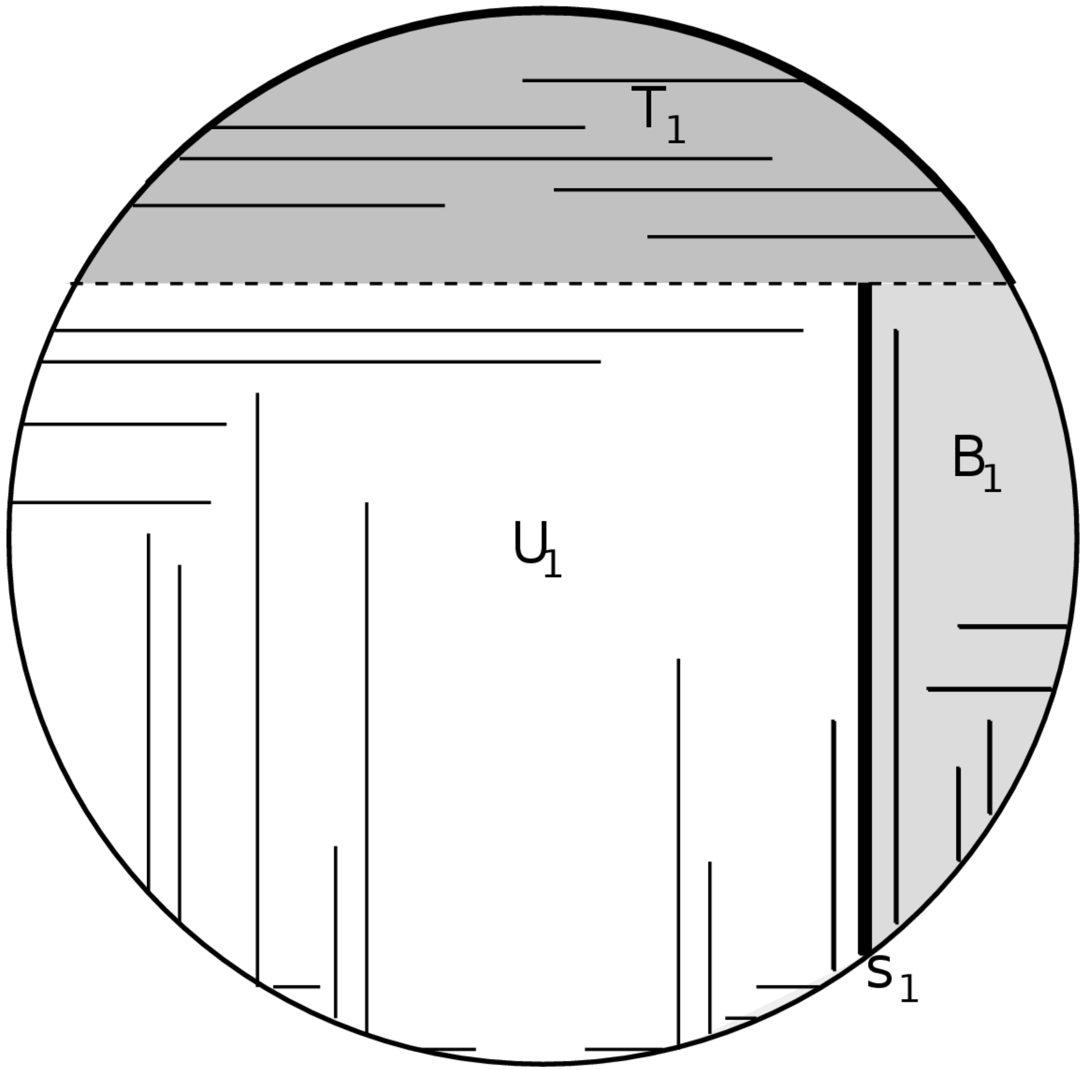
The overall partition into 3 side instances is determined by at most six segments

The algorithm for 3 side instances

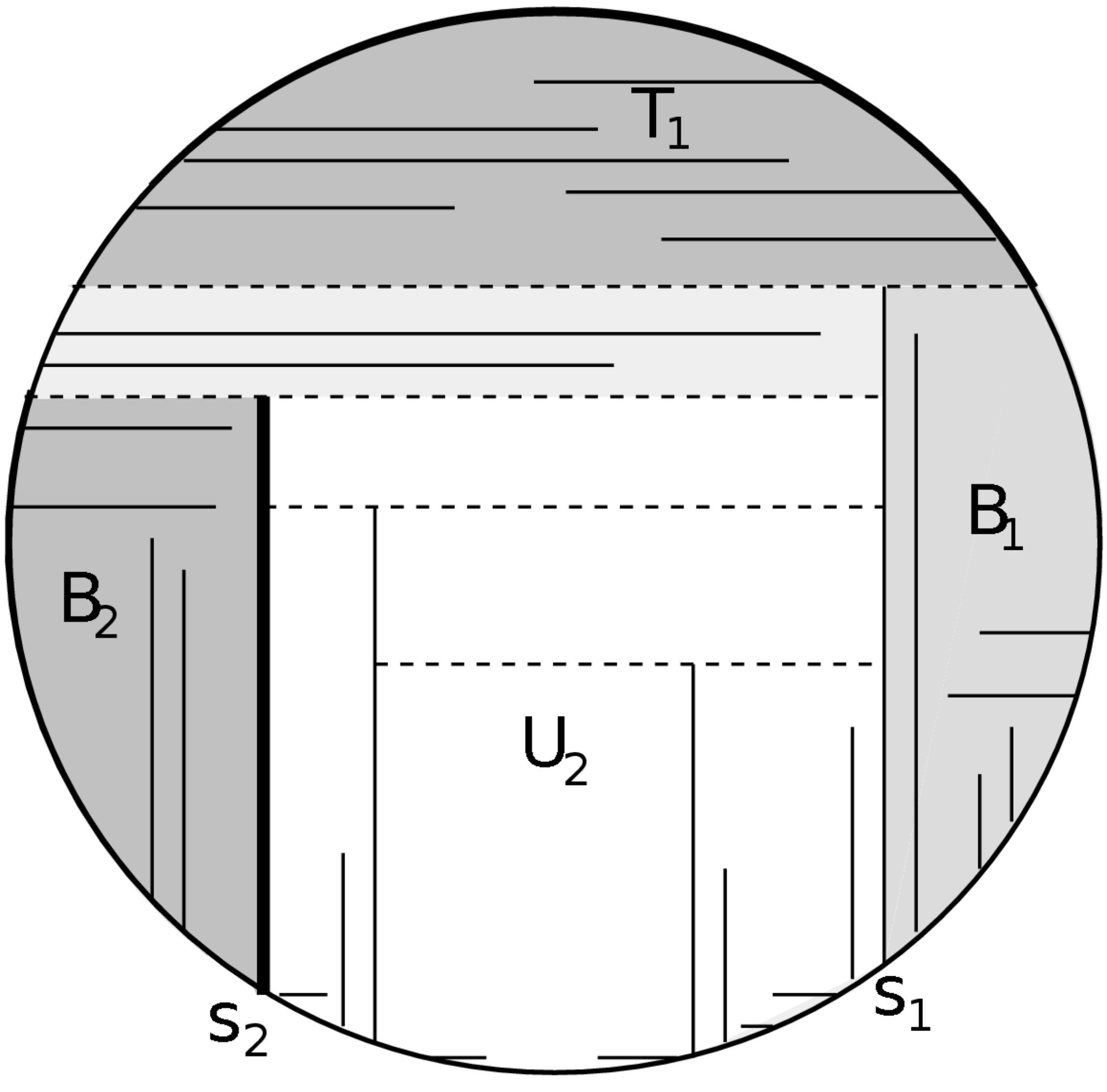


- Bottom side segments of optimum partition the disk into regions:
  - optimum is the sum of optima within the regions
  - instances within the regions are bipartite
- Dynamic programming:
  - to compute a solution for a region, use partial solutions computed for regions contained in it

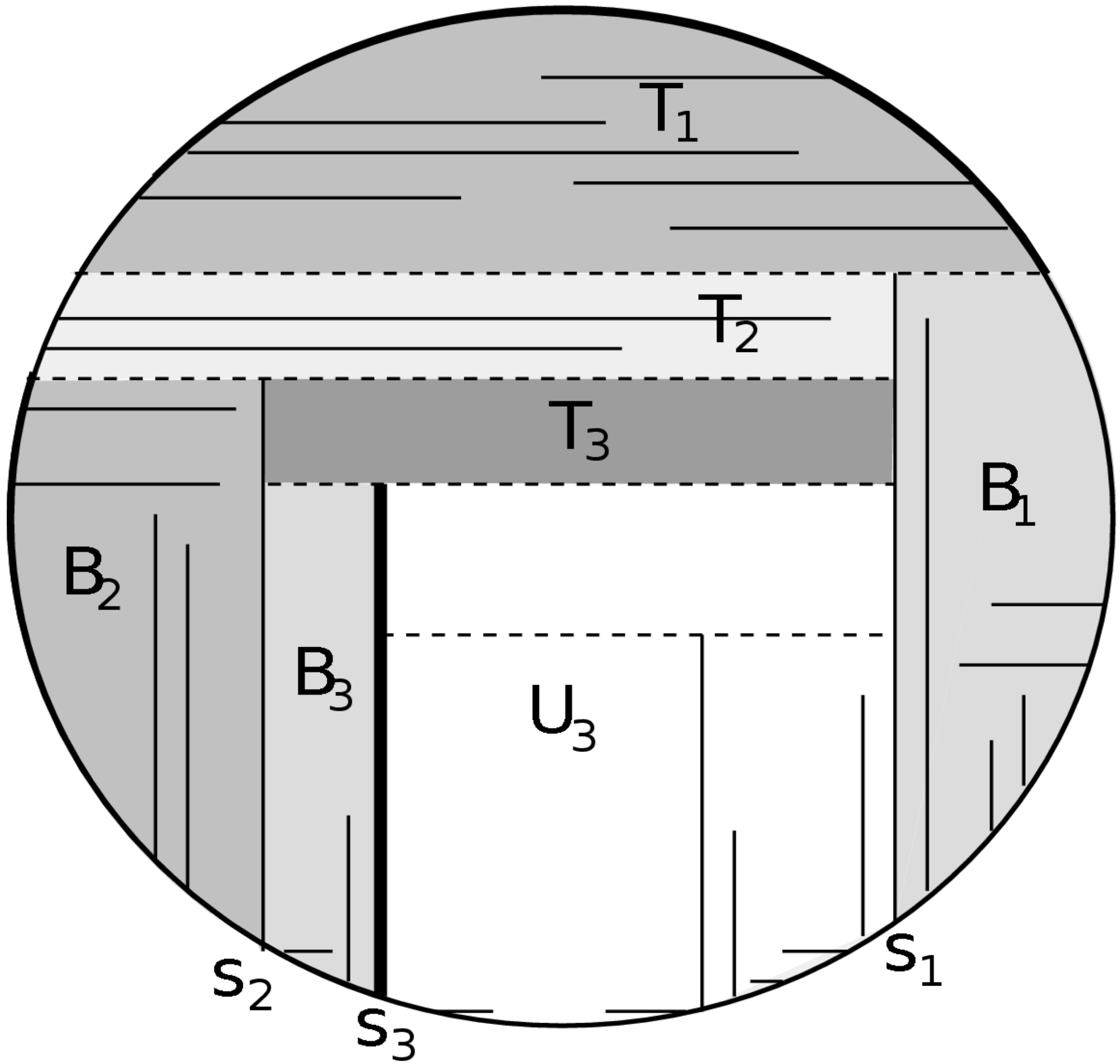
Decomposing 3 side instances



Decomposing 3 side instances

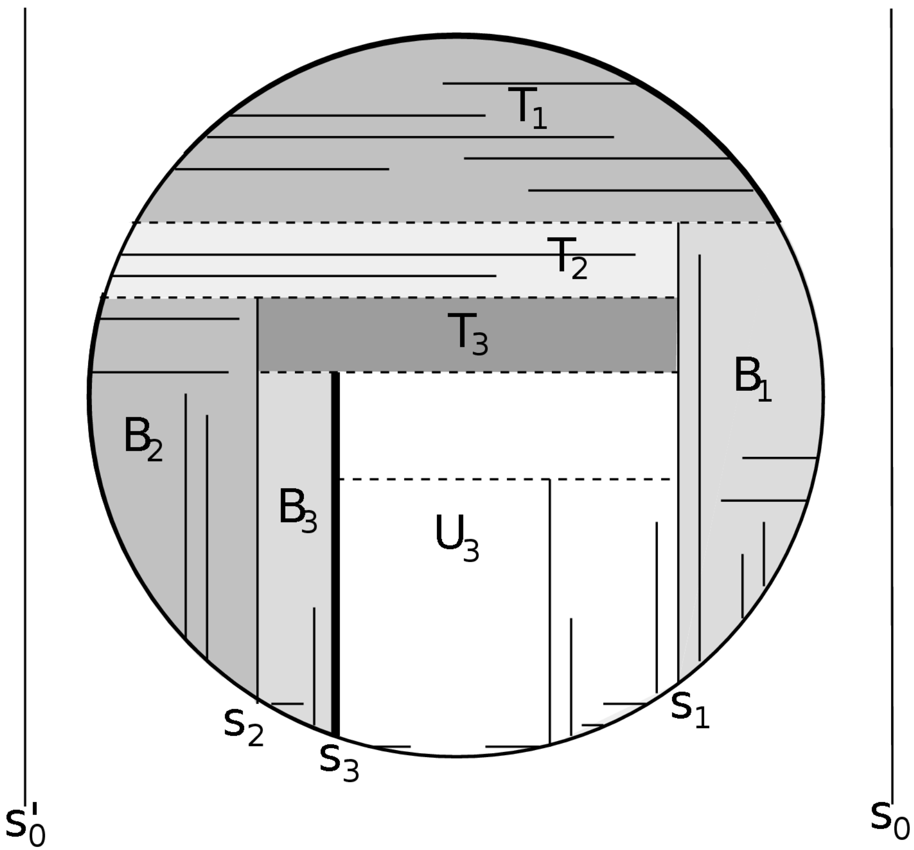


Decomposing 3 side instances



Dynamic Programming

$MIS[U_i]$ is a maximum independent set in $D \setminus U_i$



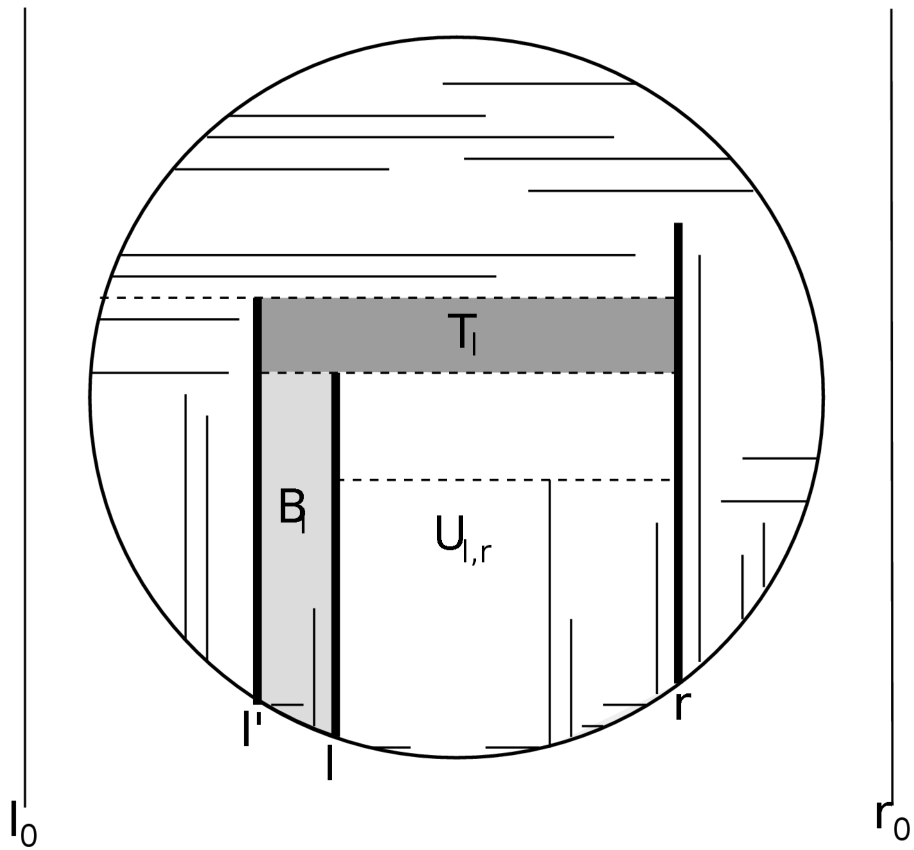
Properties:

$U_{i-1} = U_i \cup B_i \cup T_i$

U_i is determined uniquely by a left and a right segment

The sweeping process processes the left segments from left to right and the right segments from right to left

Dynamic Programming



Properties:

$U_{i-1} = U_i \cup B_i \cup T_i$

$U_{l,r}$ is determined uniquely By s_l(left) and s_r(right segment)

The sweeping process processes the left segments from left to right and the right segments from right to left

$MIS[U_{l,r}]$ is a maximum independent set in $D \setminus U_{l,r}$:

$$MIS[U_{0,0}] = 0$$
$$MIS[U_{l,r}] = max_{l' < l : s_{l'} \geq s_l} (MIS[U_{l',r}] + OPT(T_l) + OPT(B_l))$$

Open Problems

- MIS in outer-segment graphs for segments aligned in more than 2 directions
- MIS in outer-string graphs