Graph Classes with Structured Neighbourhoods

Rémy Belmonte   Martin Vatshelle

University of Bergen

June 22, 2011

Outline

- Why are Interval graphs easy?

Outline

- Why are Interval graphs easy?
- Representative-size.

Outline

- Why are Interval graphs easy?
- Representative-size.
- Graph classes with bounded representative-size.

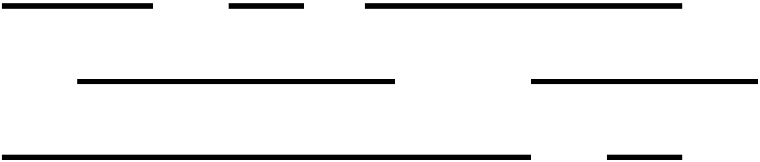
Outline

- Why are Interval graphs easy?
- Representative-size.
- Graph classes with bounded representative-size.
- Application of our results.

Outline

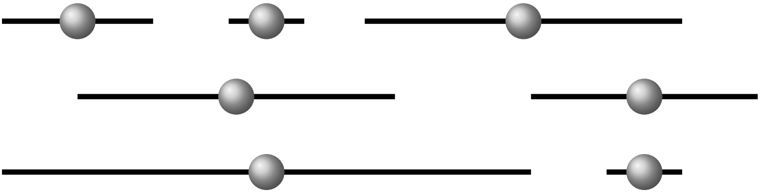
- Why are Interval graphs easy?
- Representative-size.
- Graph classes with bounded representative-size.
- Application of our results.
- Conclusion and future research.

Interval graphs



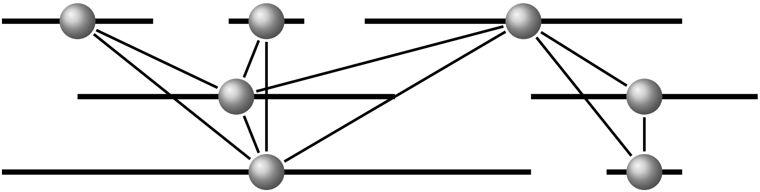
- A collection of intervals of the real line.

Interval graphs



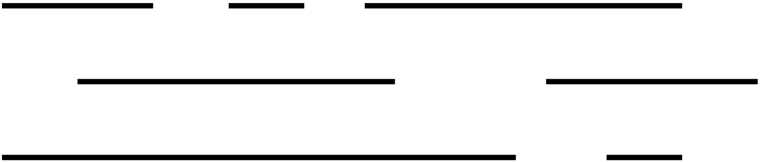
- A collection of intervals of the real line.
- Represent each interval by a node.

Interval graphs



- A collection of intervals of the real line.
- Represent each interval by a node.
- If intervals intersect, their nodes are neighbours.

Dynamic programming on Interval graphs



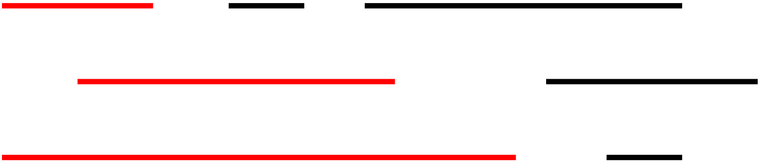
Dynamic programming on Interval graphs

- Go through intervals from left to right.

Dynamic programming on Interval graphs

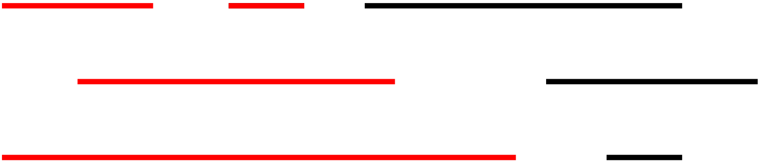
- Go through intervals from left to right.

Dynamic programming on Interval graphs



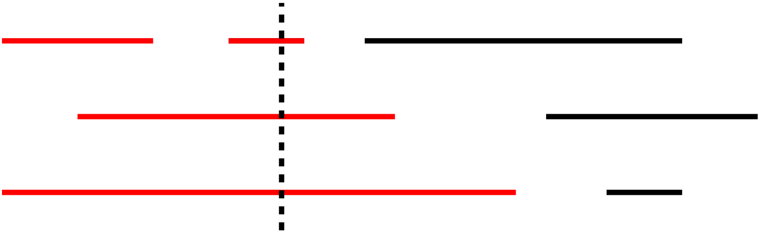
- Go through intervals from left to right.

Dynamic programming on Interval graphs



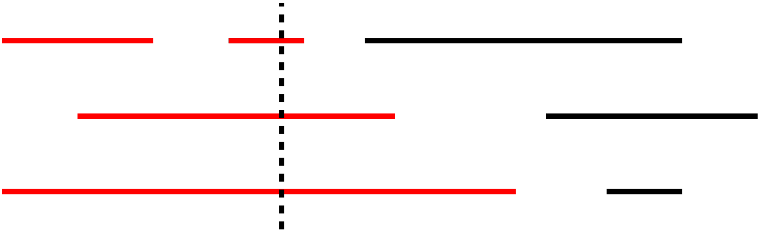
- Go through intervals from left to right.

Dynamic programming on Interval graphs



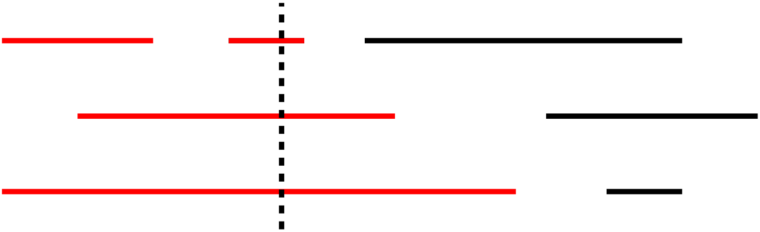
- Go through intervals from left to right.
- Keep all the possibly optimal partial solutions.

Dynamic programming on Interval graphs



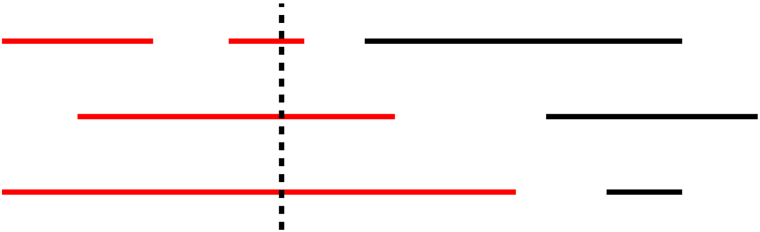
- Go through intervals from left to right.
- Keep all the possibly optimal partial solutions.
- Two solutions will be equivalent if they have the same neighbourhood across the cut.

Dynamic programming on Interval graphs



- Go through intervals from left to right.
- Keep all the possibly optimal partial solutions.
- Two solutions will be equivalent if they have the same neighbourhood across the cut.
- There is at most *n* different neighbourhoods.

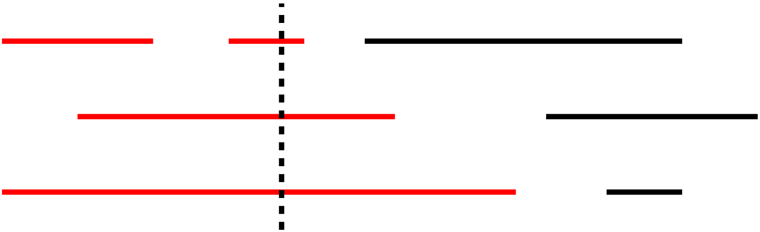
Representative-size



Definition (for a set)

Consider a cut A, B of a graph, and $S \subseteq A$, then the representative-size of S with respect to A is the minimum size of $S' \subseteq S$ such that $N(S) \cap B = N(S') \cap B$.

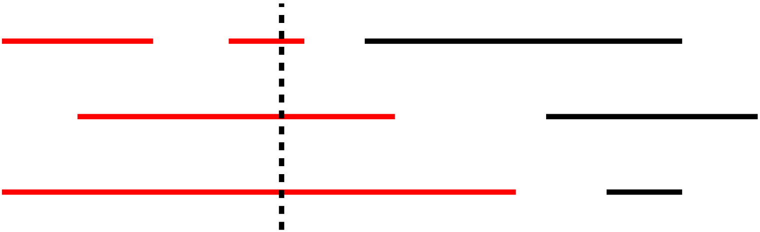
Representative-size



Definition (for a cut)

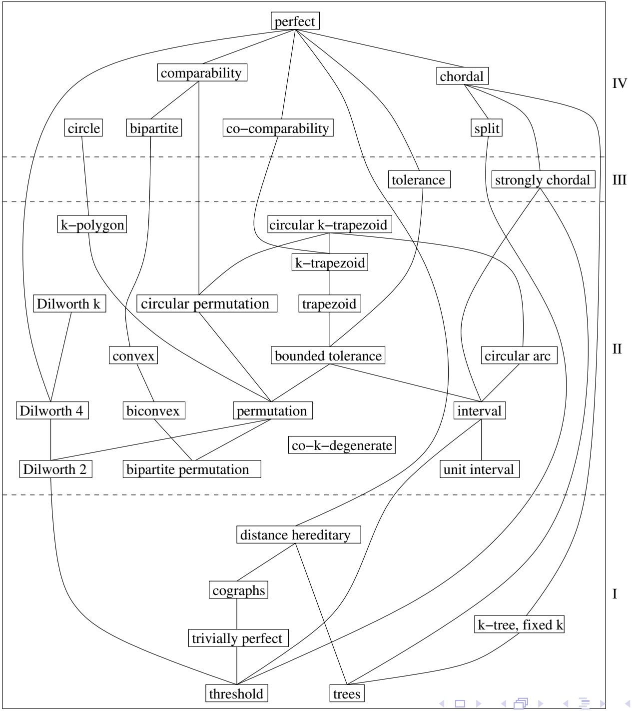
Consider a cut (A, B) of a graph, then the representative-size of (A, B) is the maximum representative-size over all $S \subseteq A$ with respect to A.

Representative-size



Definition (for a graph, linear version)

A graph $G = (V, E)$ has representative-size r if one can order the vertices such that for every i the representative-size of $(A_i, V \setminus A_i)$ is at most r, where A_i is the set containing the i first vertices.


perfect
comparability
chordal
IV
circle
bipartite
co−comparability
split
tolerance
strongly chordal
III
k−polygon
circular k−trapezoid
k−trapezoid
Dilworth k
circular permutation
trapezoid
convex
bounded tolerance
circular arc
II
Dilworth 4
biconvex
permutation
interval
co−k−degenerate
Dilworth 2
bipartite permutation
unit interval
distance hereditary
cographs
I
k−tree, fixed k
trivially perfect
threshold
trees

Applications of our results

- The graph width parameter Boolean-width was introduced by Bui-Xuan, Telle and Vatshelle [IWPEC 2009]

Applications of our results

- The graph width parameter Boolean-width was introduced by [BTV'09]
- representative-size $r \Rightarrow$ boolean-width $\leq r * log(n)$

Applications of our results

- The graph width parameter Boolean-width was introduced by [BTV'09]
- representative-size $r \Rightarrow$ boolean-width $\leq r * log(n)$
- INDEPENDENT SET and DOMINATING SET can be solved in $c^{boolean-width(G)} * poly(n)$ time [BTV'09]

Applications of our results

- The graph width parameter Boolean-width was introduced by [BTV'09]
- representative-size $r \Rightarrow$ boolean-width $\leq r * log(n)$
- INDEPENDENT SET and DOMINATING SET can be solved in $c^{boolean-width(G)} * poly(n)$ time [BTV'09]

Theorem

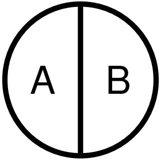
INDEPENDENT SET *and* DOMINATING SET *can be solved in polynomial time on graphs of bounded representative-size.*

VERTEX PARTITIONING PROBLEMS

The VERTEX PARTITIONING problems was introduced by Telle and Proskurowski. [JDM 1997]

Vertex Partitioning Problems

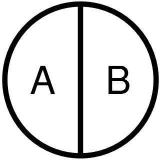
The Vertex Partitioning problems by [TP'97]



Independent set:

VERTEX PARTITIONING PROBLEMS

The VERTEX PARTITIONING problems by [TP'97]

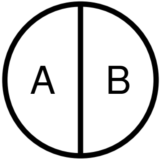


INDEPENDENT SET:

- We ask for a partition into (A, B) maximizing $|A|$.

VERTEX PARTITIONING PROBLEMS

The VERTEX PARTITIONING problems by [TP'97]

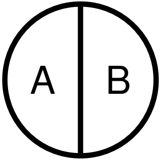


INDEPENDENT SET:

- We ask for a partition into (A, B) maximizing $|A|$.
- Such that every node in A has 0 neighbours in A.

VERTEX PARTITIONING PROBLEMS

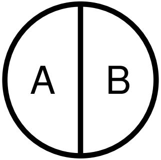
The VERTEX PARTITIONING problems by [TP'97]



DOMINATING SET:

Vertex Partitioning Problems

The Vertex Partitioning problems by [TP'97]

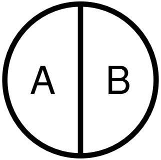


Dominating set:

- We ask for a partition into (A, B) minimizing $|A|$.

VERTEX PARTITIONING PROBLEMS

The VERTEX PARTITIONING problems by [TP'97]

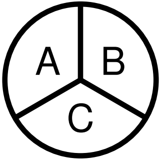


DOMINATING SET:

- We ask for a partition into (A, B) minimizing $|A|$.
- Every node in B has at least 1 neighbour in A.

Vertex Partitioning Problems

The Vertex Partitioning problems by [TP'97]



3-coloring:

- Every node in A has 0 neighbour in A.
- Every node in B has 0 neighbour in B.
- Every node in C has 0 neighbour in C.

VERTEX PARTITIONING PROBLEMS

The VERTEX PARTITIONING problems by [TP'97]

In general:

VERTEX PARTITIONING PROBLEMS

The VERTEX PARTITIONING problems by [TP'97]

In general:

- We ask for a partition into p parts maximizing or minimizing one part.

Vertex Partitioning Problems

The Vertex Partitioning problems by [TP'97]

In general:

- We ask for a partition into p parts maximizing or minimizing one part.
- For each ordered pair of parts we may put a requirement on the number of neighbours in the other part.

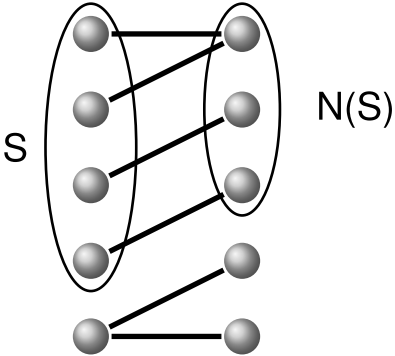
VERTEX PARTITIONING PROBLEMS

The VERTEX PARTITIONING problems by [TP'97]

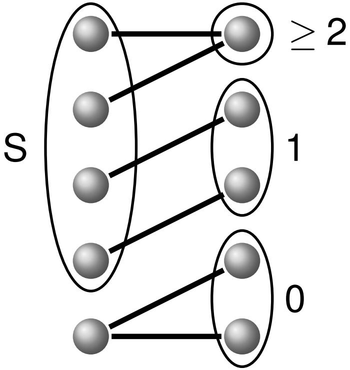
In general:

- We ask for a partition into p parts maximizing or minimizing one part.
- For each ordered pair of parts we may put a requirement on the number of neighbours in the other part.
- Requirements can be any finite or co-finite set.

d-neighbourhood



d-neighbourhood



d-neighbourhood

BTV'09 use the "number of d-neighbourhoods", denoted $UN_d(G)$, which is related to representative-size.

d-neighbourhood

BTV'09 use the "number of d-neighbourhoods", denoted $UN_d(G)$, which is related to representative-size.

Theorem

$$UN_d \leq n^{d \cdot r}$$

Vertex Partitioning Problems

Theorem (ABTRRV, WG2010)

Every vertex partitioning problem can be solved in time $O^*(UN_d^{3 \cdot p})$

Vertex Partitioning Problems

Theorem (ABTRRV, WG2010)

Every vertex partitioning problem can be solved in time $O^*(UN_d^{3 \cdot p})$

Corollary

Every vertex partitioning problem can be solved in time $O^*(n^{3 \cdot d \cdot r \cdot p})$

Summary of results

The problems:

INDEPENDENT SET, DOMINATING SET, MAXIMUM INDUCED MATCHING, PERFECT CODE, k-COLOURING, H-COVER/HOMOMORPHISM/ROLE ASSIGNMENT.

Are polynomial on:

Interval graphs, circular arc graphs, permutation graphs, trapezoid graphs, Dilworth-k graphs, convex graphs ...

Open problems

- What is the representative-size of strongly chordal and tolerance graphs?
- Is there a graph-class with unbounded representative-size where ever VERTEX PARTITIONING problem is polynomial?
- Can we compute the representative-size?
- Can the same ideas be used on directed graphs?

Thank you!