More applications of the d-neighbor equivalence: acyclic and connectivity constraints

Benjamin Bergougnoux[⋆], Mamadou Kanté[†]

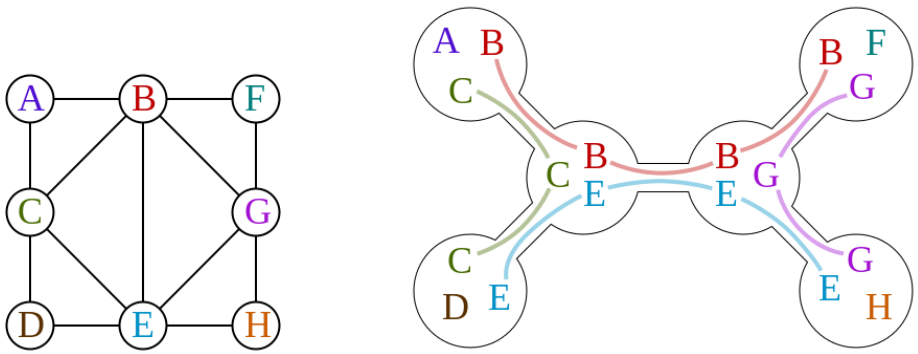
[⋆] IRIF, CNRS, Université Paris Diderot, France

[†] LIMOS, CNRS, Université Clermont Auvergne, France

IBS, Daejong, July 23 2019

The famous: tree-width

The most studied graph parameter.



- Measure how close a graph is to the stucture of a tree.
- $2^{O(\mathsf{tw})} \cdot n$ time algorithms for many NP-hard problems.
- Can be approximated in time $2^{O(\mathsf{tw})} \cdot n$.

Limits and alternatives

Tree-width is unbounded in any dense graph class.

→ Tree-width of a clique $= n - 1$.

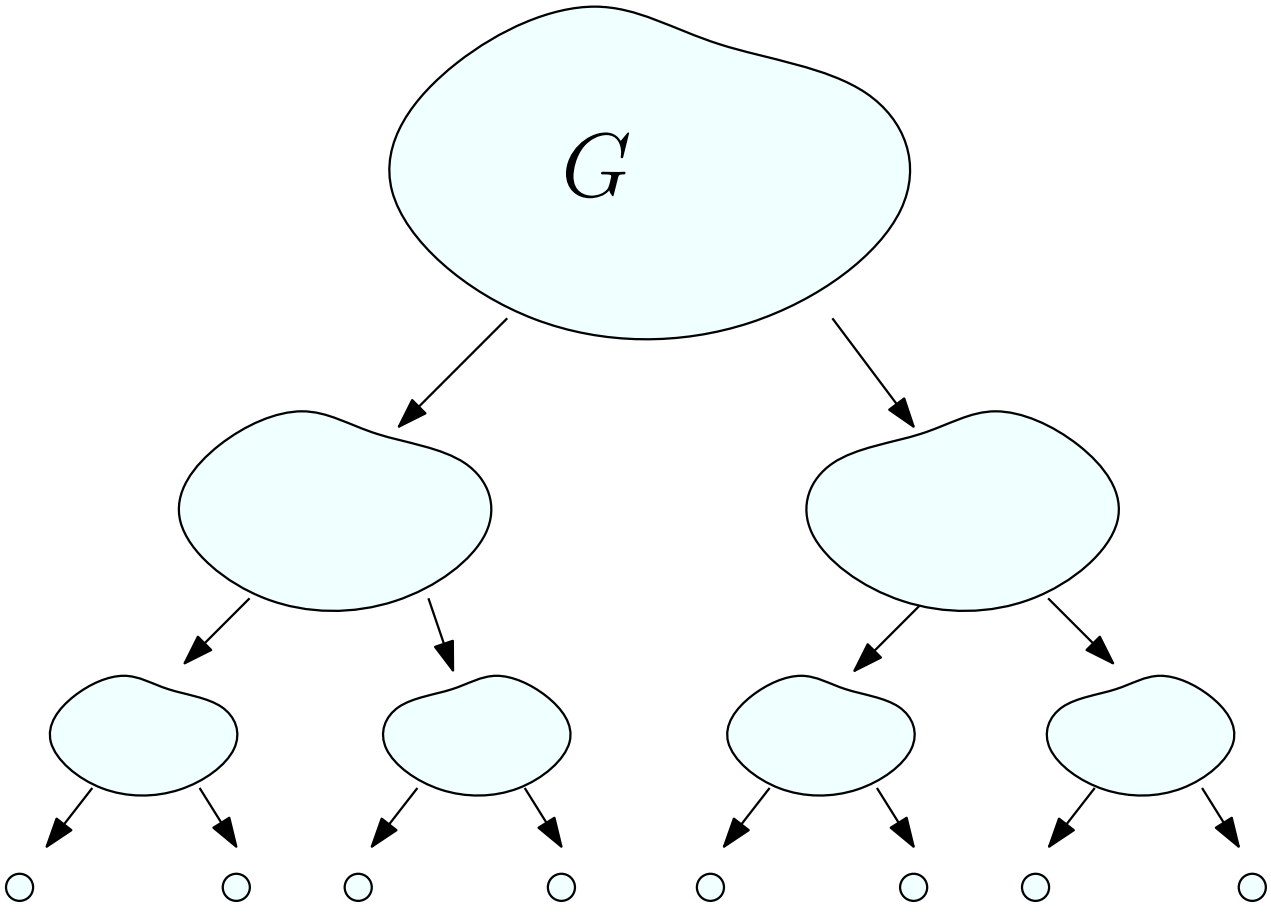
- Dense graphes can be simple.
- Many NP-hard problems are tractable in on some dense graph classes.
- We can explain this with other width measures.

  → clique-width, rank-width, mim width.

Divide a graph

Divide a graph

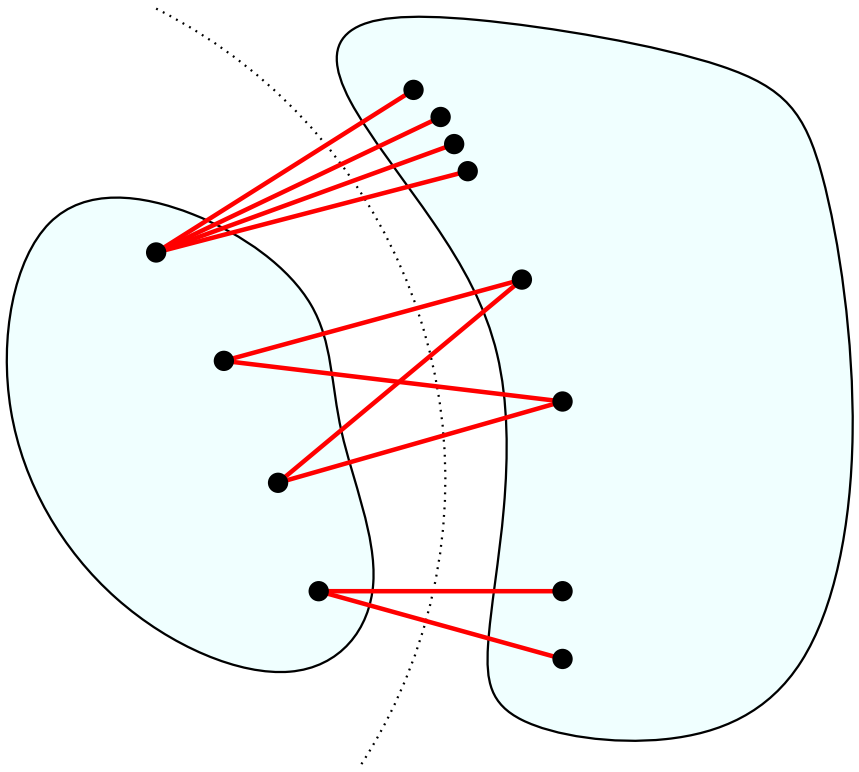
Recursively decompose your graph...



We recursively cut the vertex set in two

Divide a graph

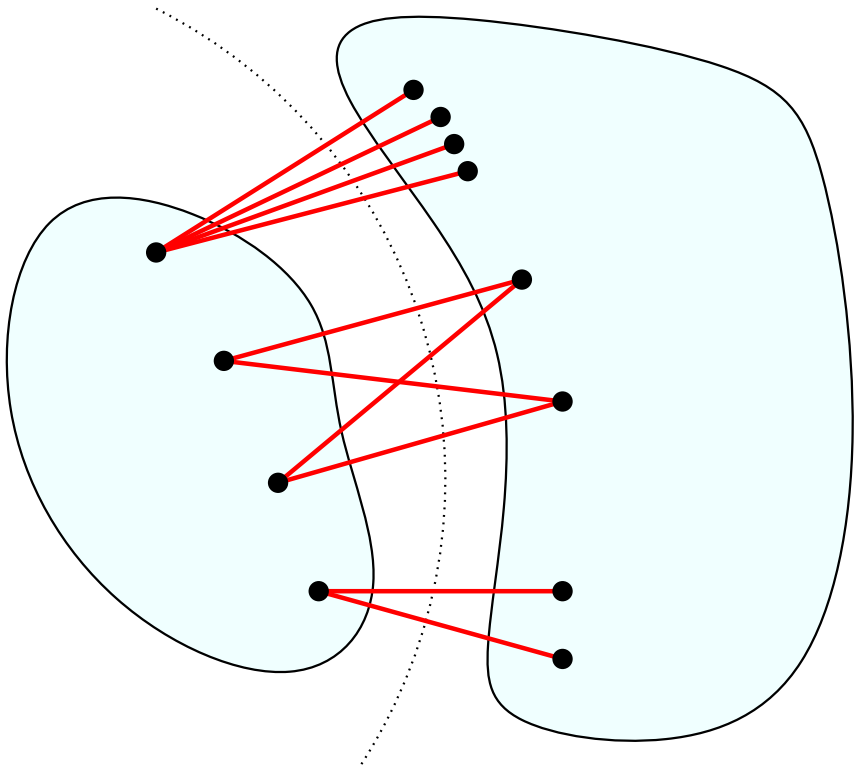
Recursively decompose your graph... into simple cuts.



Different notions of simplicity = different width measures.

Divide a graph

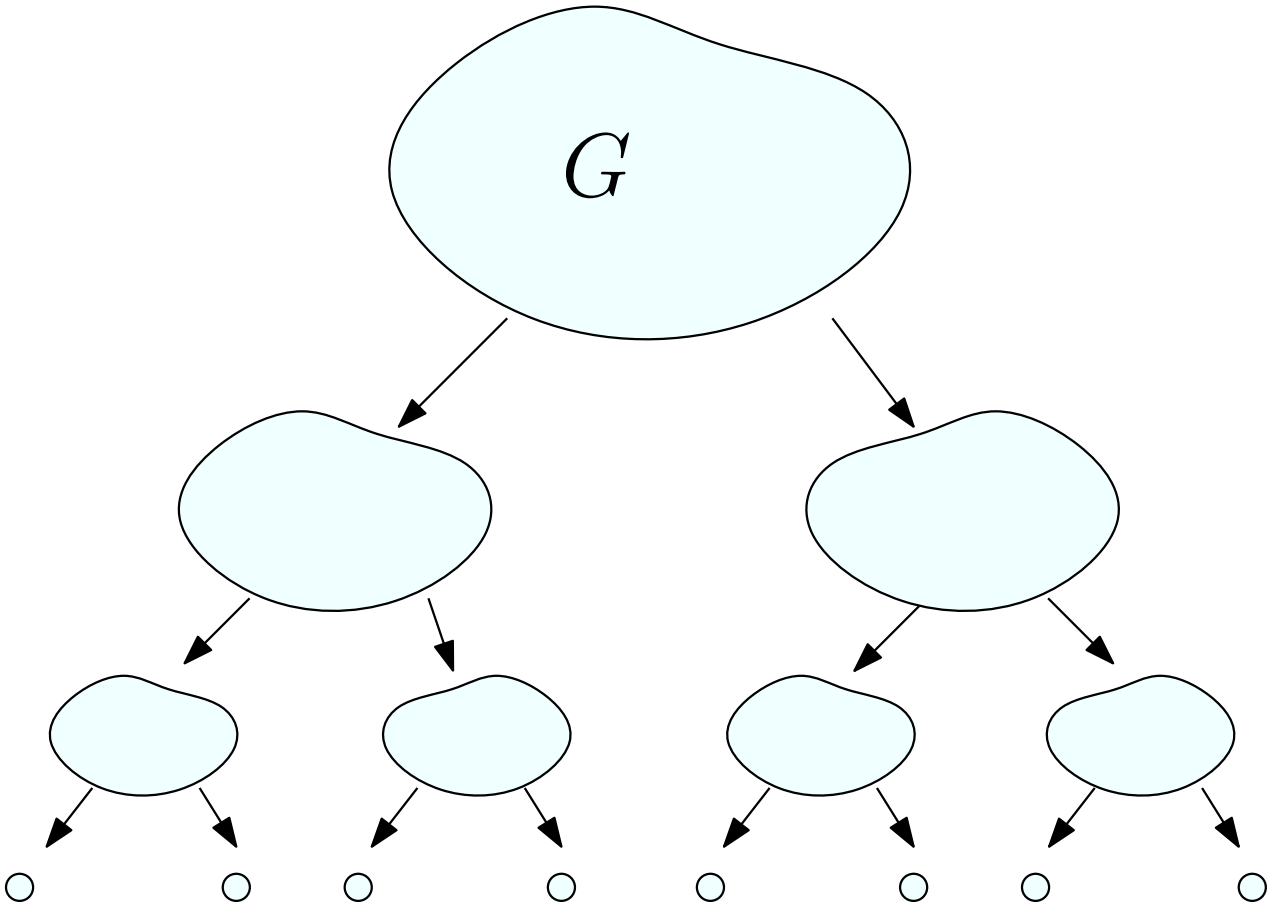
We describe the simplicity of a cut with a function f: cut $\to \mathbb{N}$.



Exemple: f(cut) = number of vertices having at least one neighbor in the rest of the graph.

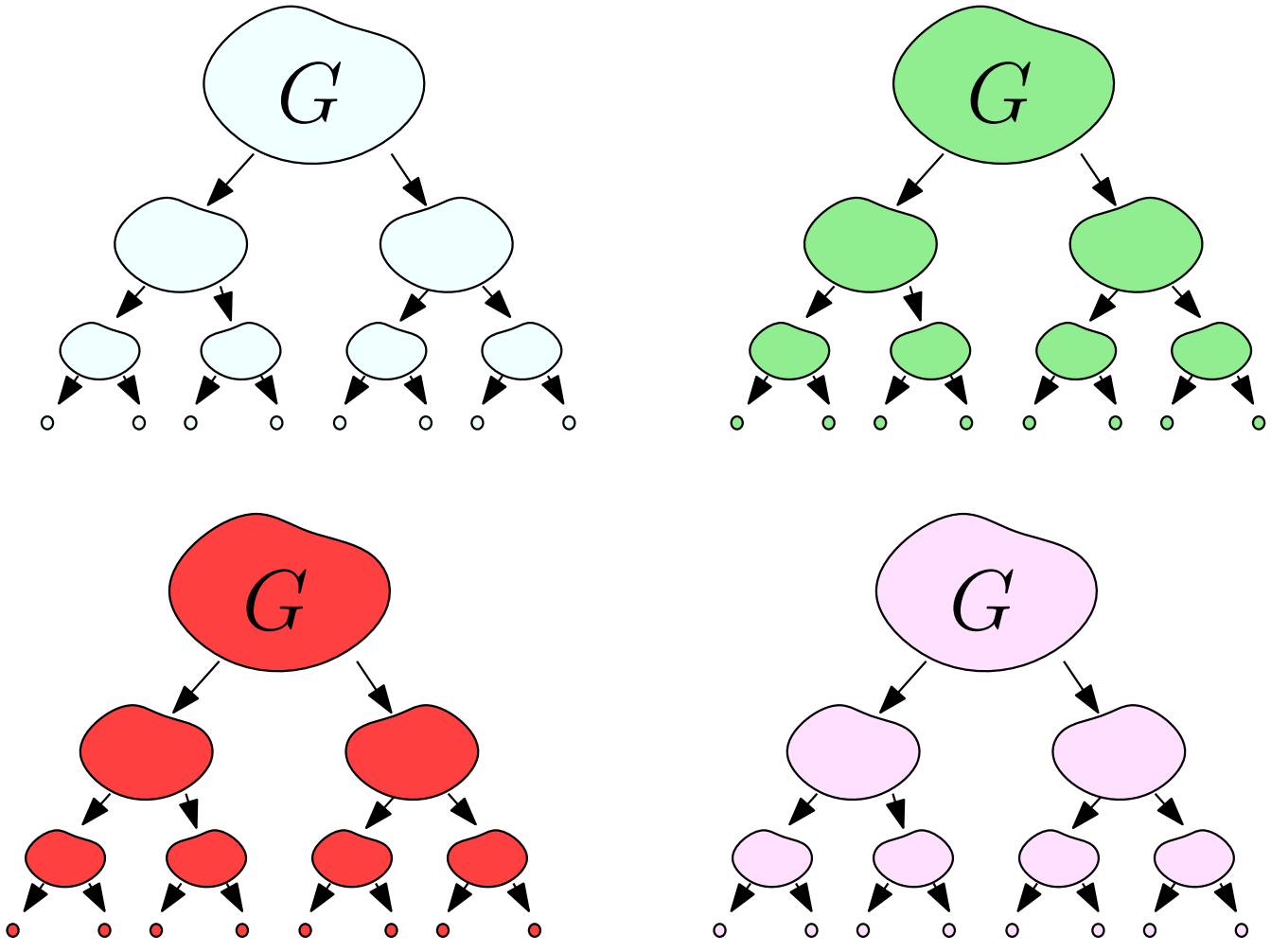
Divide a graph

Width of a decomposition D := $\max$ f(cut) among the cuts of D.



Divide a graph

Width of a graph G := min width of the decompositions of G.

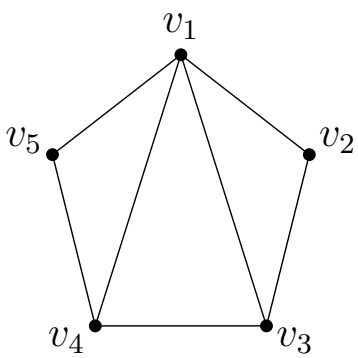


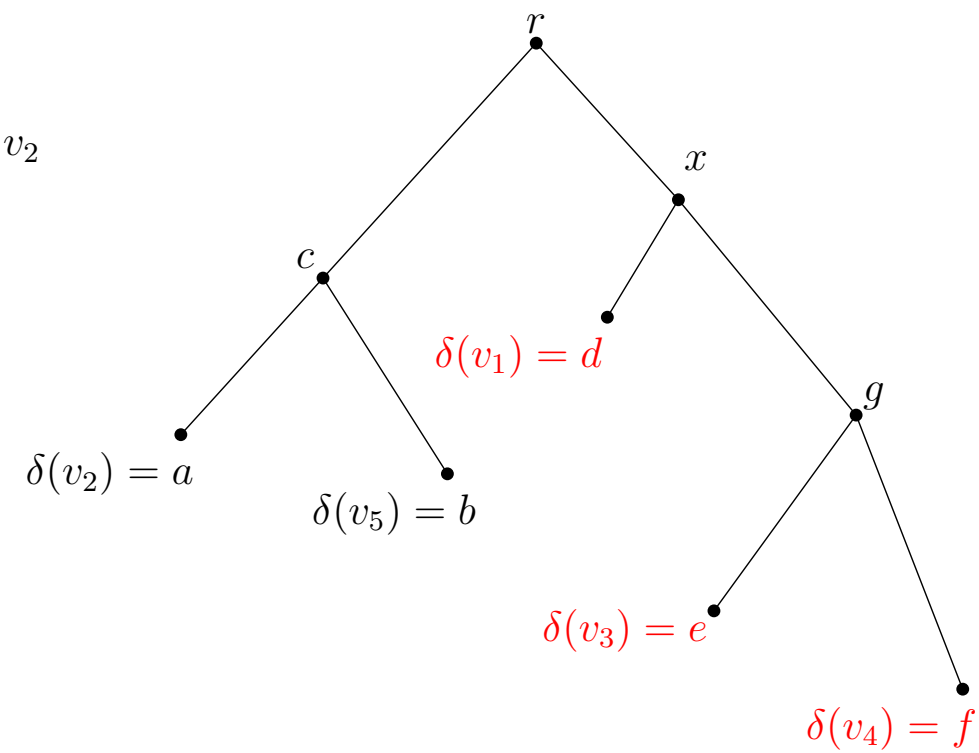
A formalism for dividing

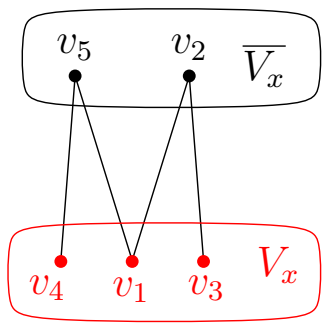
A rooted layout of a graph G is a pair (T, δ) with

- T a binary tree,
- δ a bijection between the leaves of T and the vertices of G.

Each node x of T is associated with a vertex set $V_x \subseteq V(G)$:



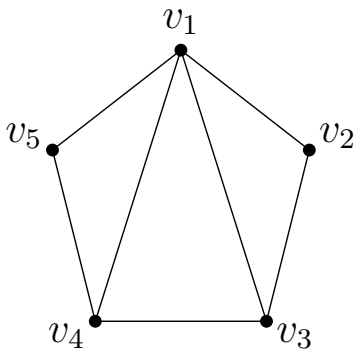


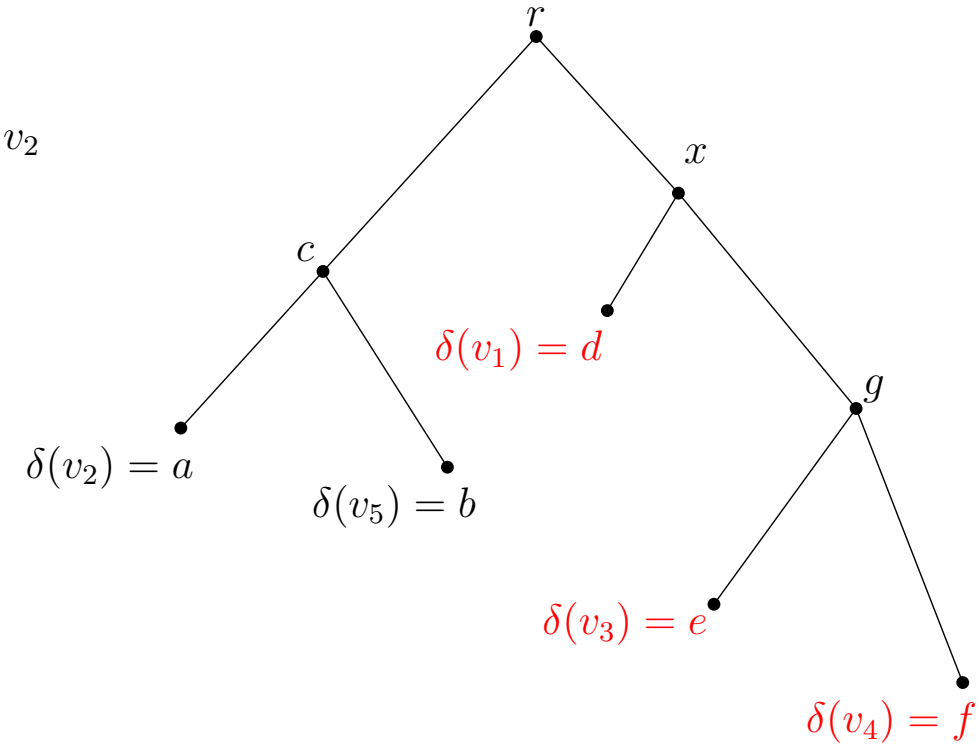


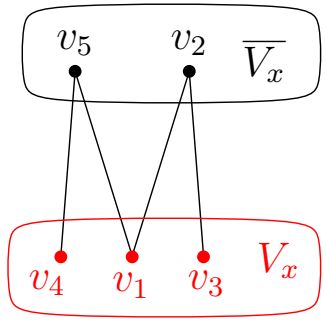
f-width

Given a function $\mathsf{f} : 2^{V(G)} \to \mathbb{N}$, we define the f-width of

- a rooted layout (T, δ): $\mathsf{f}(T, \delta) := \max_{x \in V(T)} \mathsf{f}(V_x)$,
- a graph G: $\mathsf{f}(G) := \min \mathsf{f}(\mathcal{L})$ among all rooted layouts $\mathcal{L}$ of G.

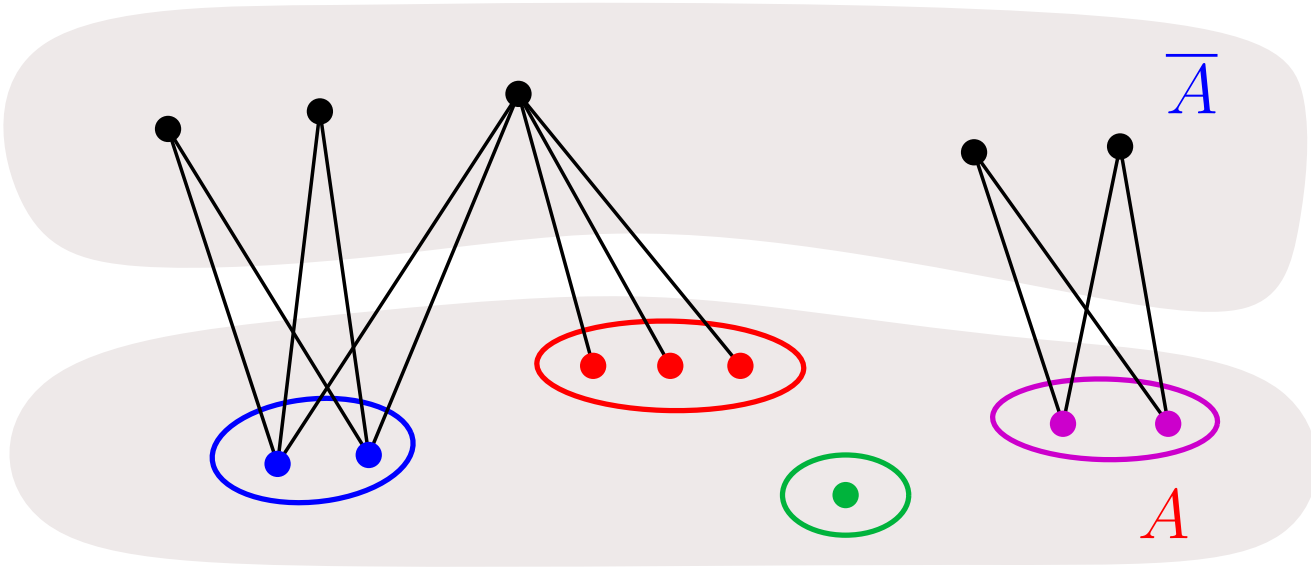






Module-width

Defined from the function $\mathsf{mw}(A) := |\{N(v) \cap \overline{A} : v \in A\}|$.



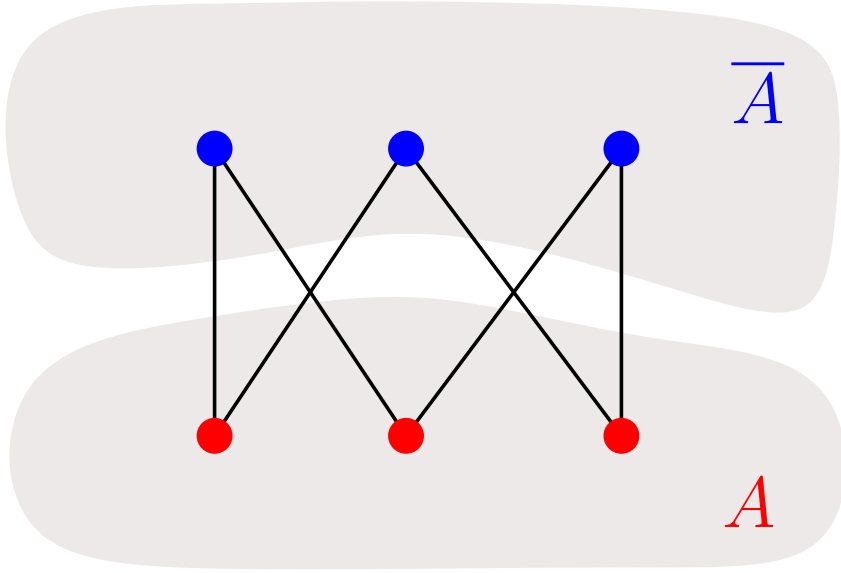
Linearly equivalent to clique-width:

[Rao 2006]

For all graphs G, we have $\mathsf{mw}(G) \leq \mathsf{cw}(G) \leq 2\mathsf{mw}(G)$.

Rank-width

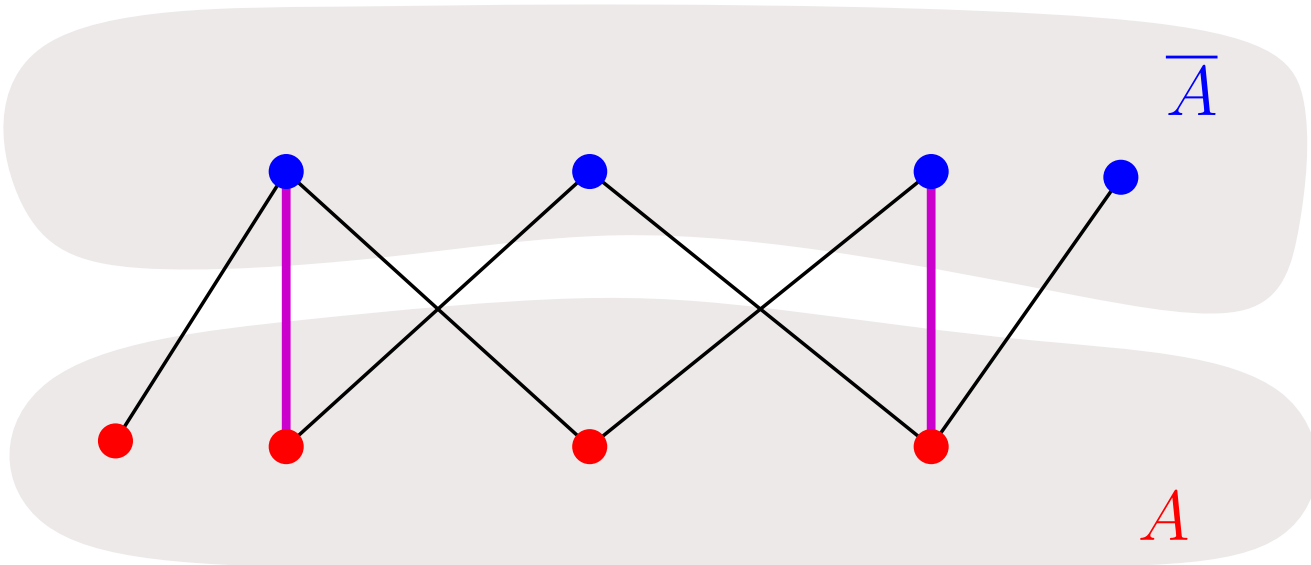
Defined from the function $\mathrm{rw}(A)$:= the rank of adjacency matrix between A and $\overline{A}$ over $GF(2)$.



$$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

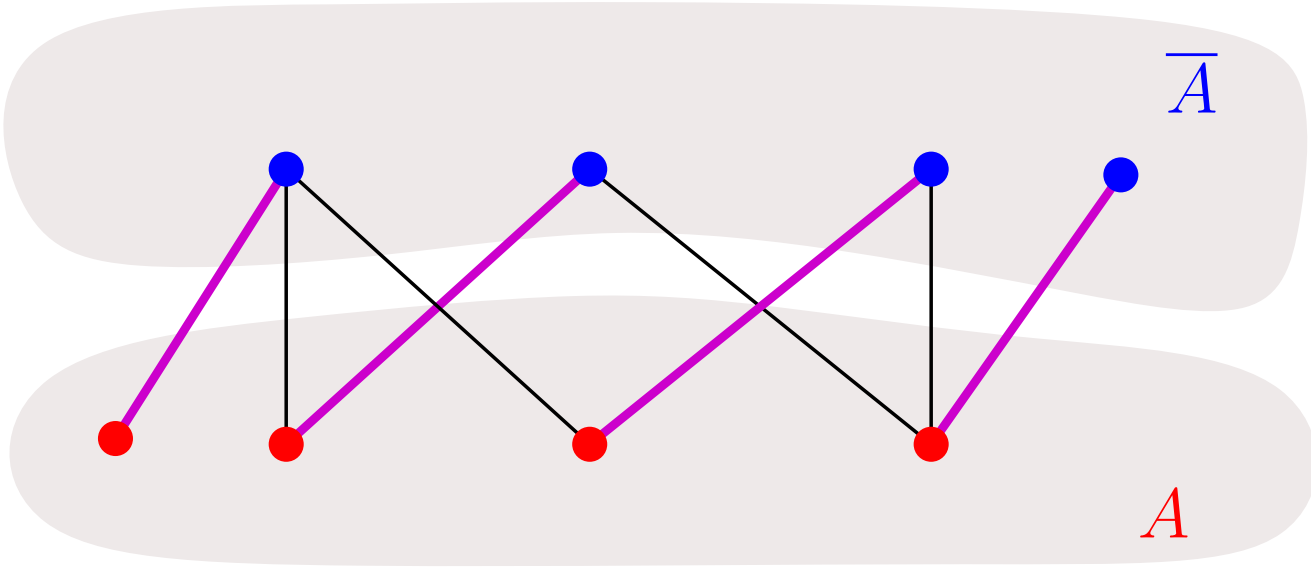
Maximum Induced Matching width (mim-width)

Defined from the function mim(A) := the size of a maximum induced matching in the bipartite graph between A and $\overline{A}$.



Maximum matching width

Defined from the function $\mathsf{mmw}(A) := |\{N(v) \cap \overline{A} : v \in A\}|$.



Linearly equivalent to tree-width:

[Vatshelle 2012]

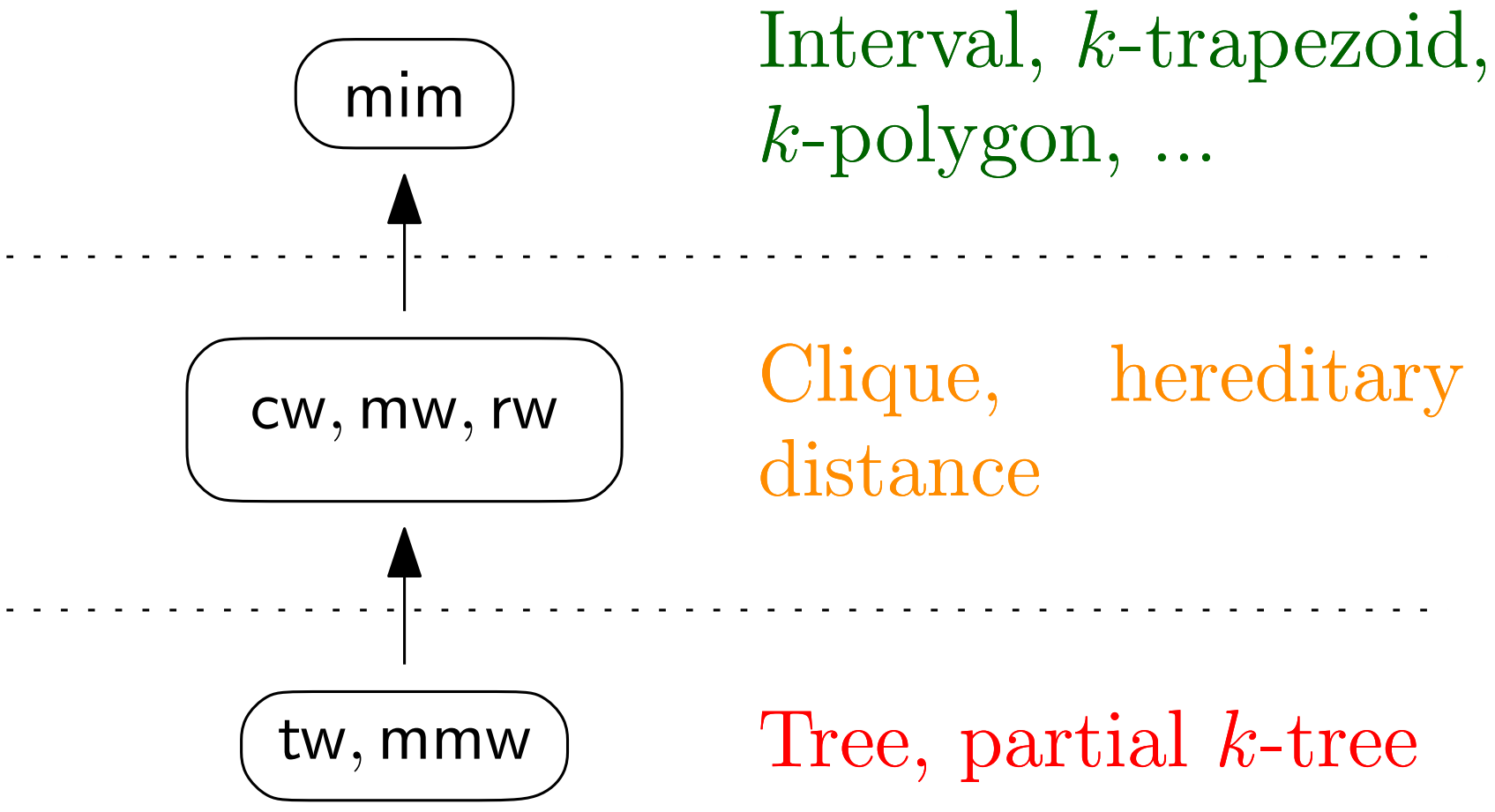
For all graphs G, we have $\frac{1}{3}\mathsf{tw}(G) \leq \mathsf{mmw}(G) \leq \mathsf{tw}(G)$.

How to compare them?

There are three aspects to consider:

- The generality/modeling power.
- The complexity of computing a decomposition of optimal/approximated width.
- Algorithmic applications.

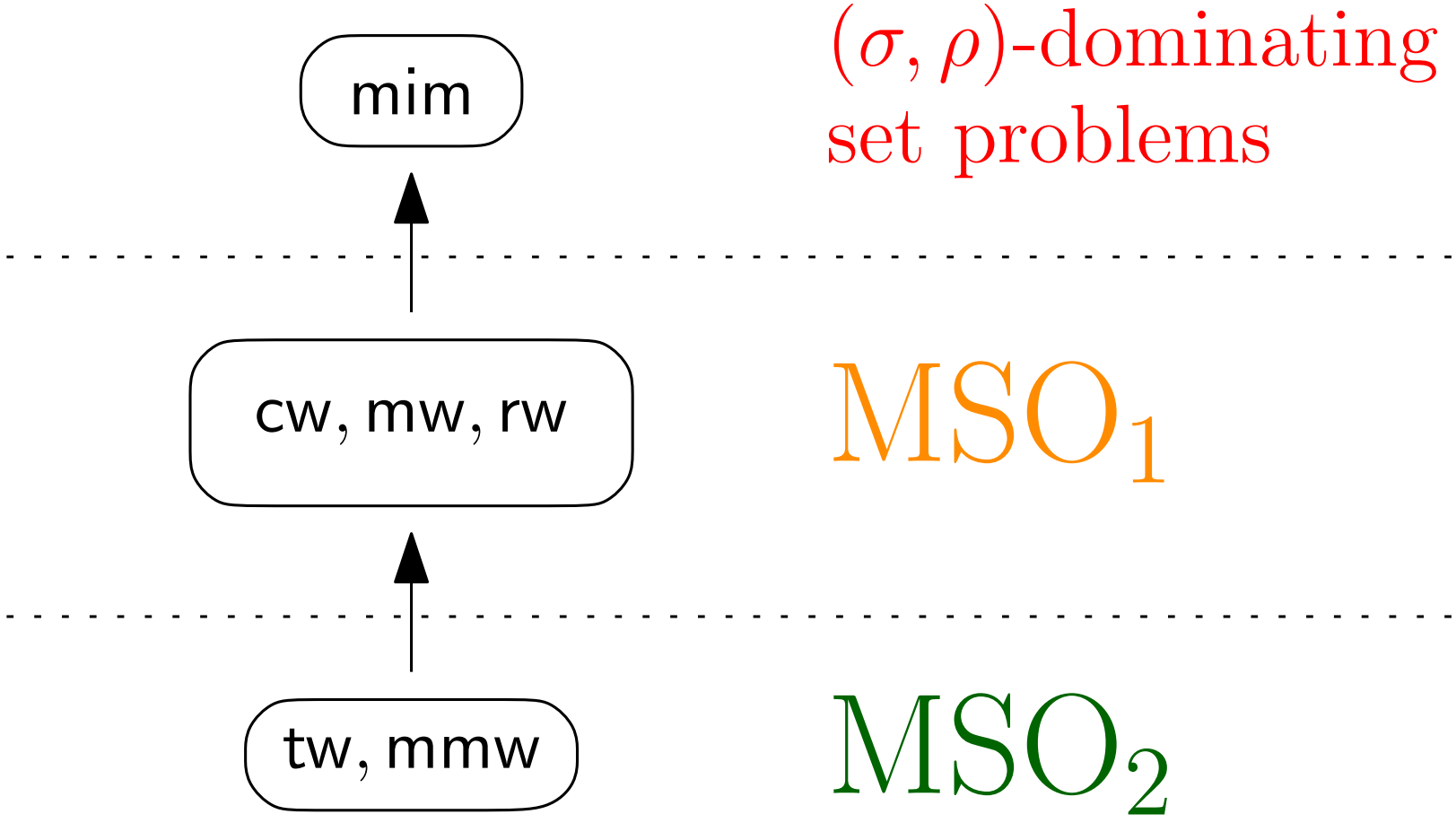
Generality


mim
cw, mw, rw
tw, mmw
Interval, k-trapezoid, k-polygon, ...
Clique, hereditary distance
Tree, partial k-tree

(B) Computation complexity

- NP-hard for all these widths measures.
- Efficient algorithms for tree-width and rank-width.
  → Running time: $2^{O(k)} \cdot n^{O(1)}$.
- Tough open questions for clique-width and mim-width.

(C) Algorithmic applications

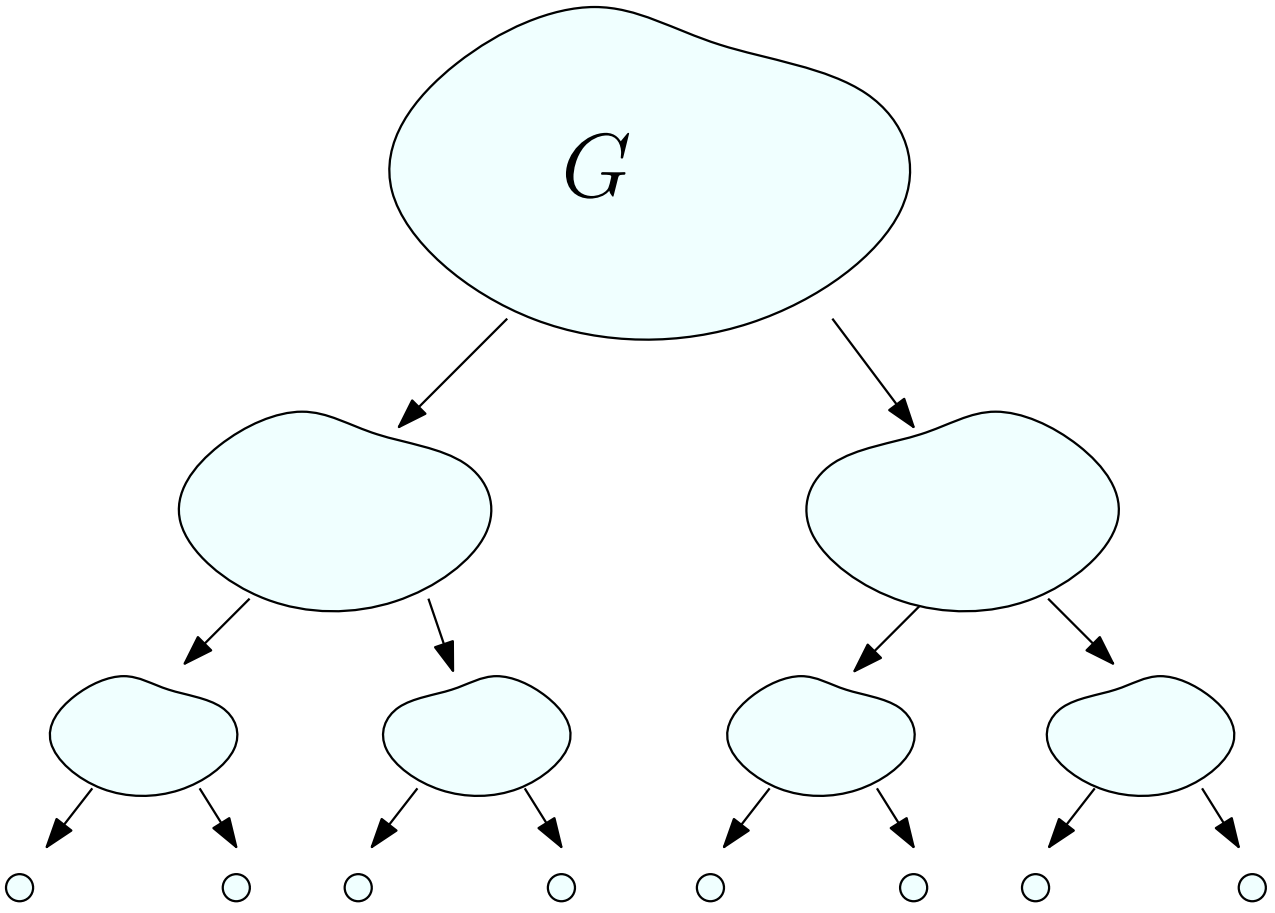


Conquer

Intuition: conquer

At each encountered cut: we solve a sub-problem.

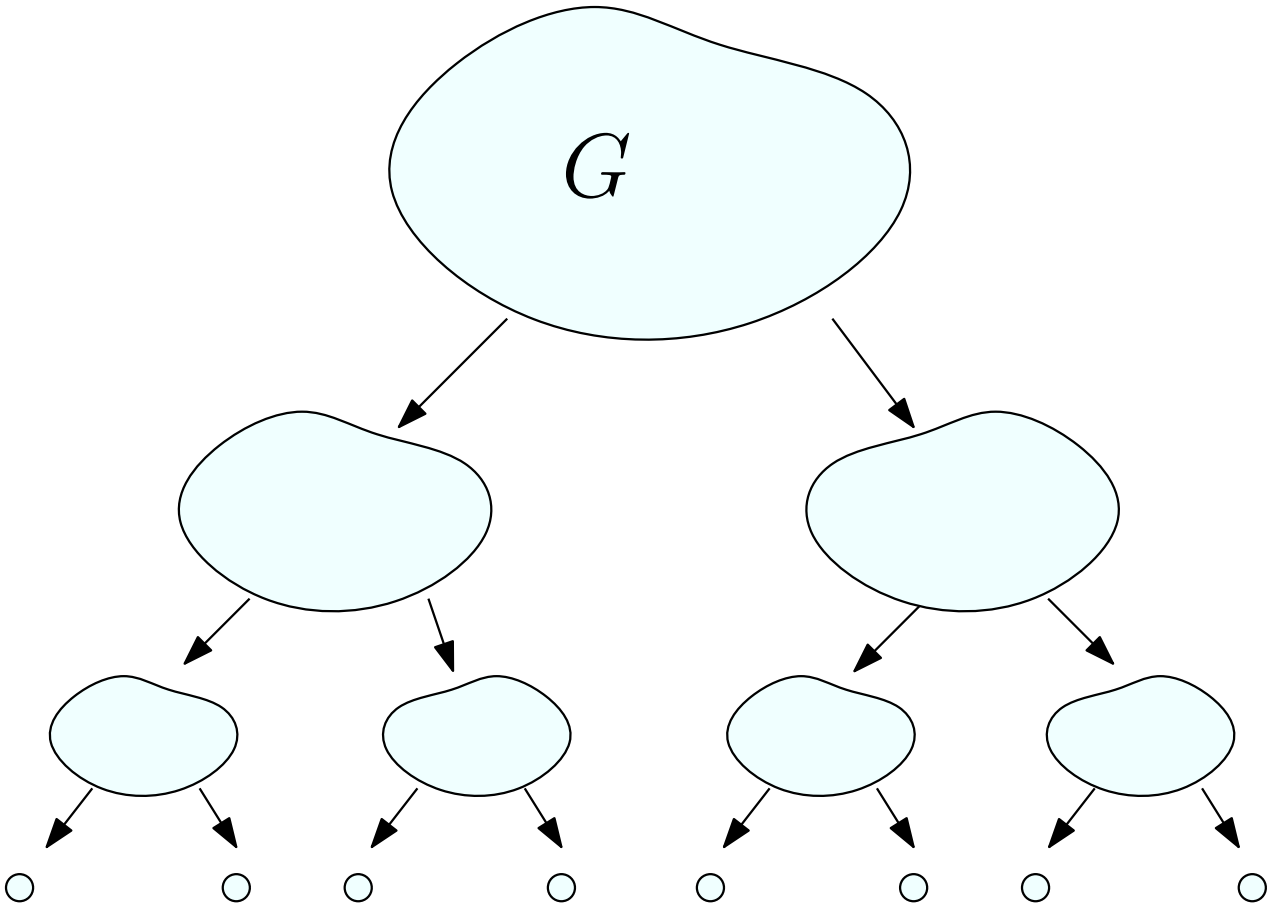
→ Computing a set of partial solutions.



Intuition: conquer

At each encountered cut: we solve a sub-problem.

→ Computing a set of partial solutions.

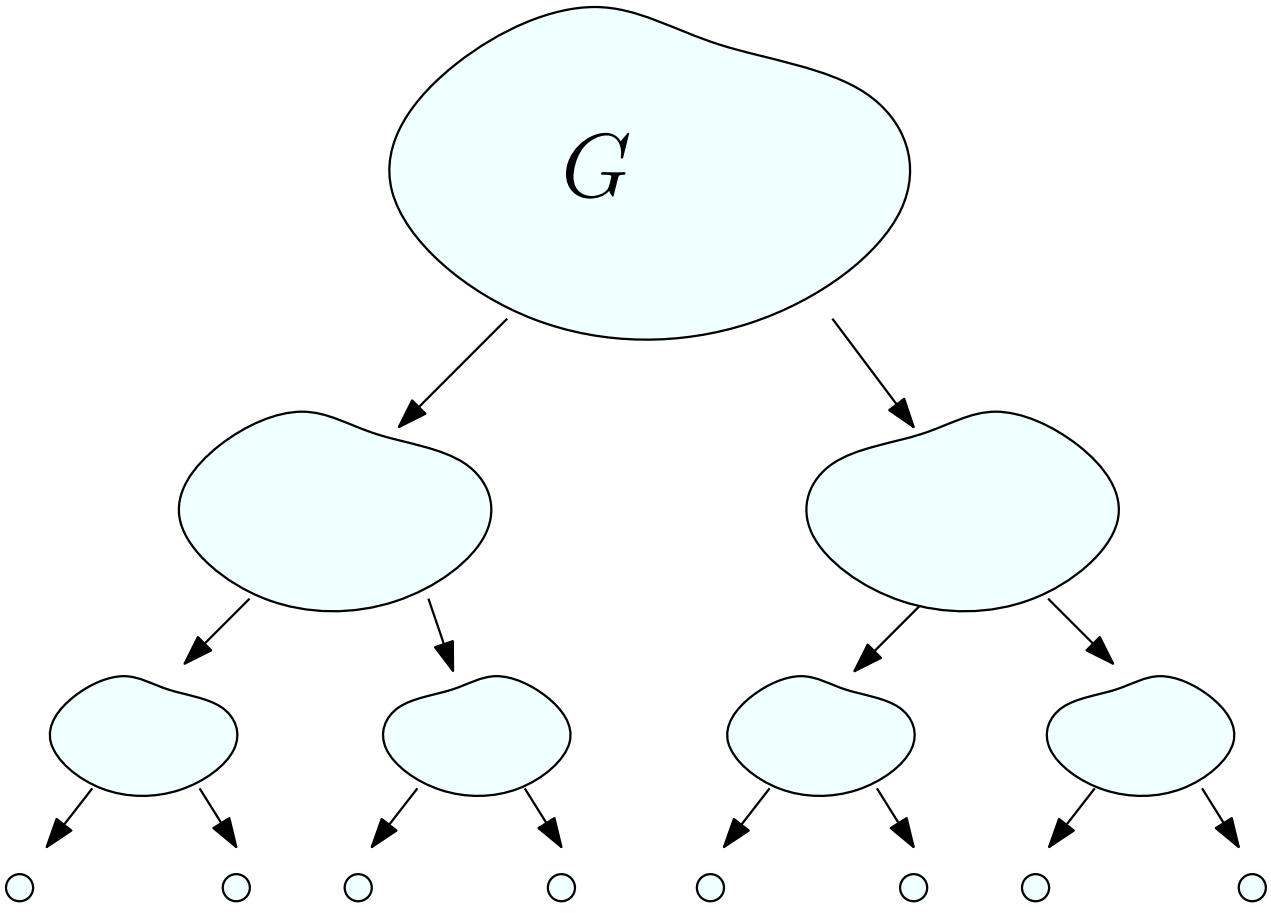


At the root: we obtain a solution.

Intuition: conquer

At each encountered cut: we solve a sub-problem.

→ Computing a set of partial solutions.

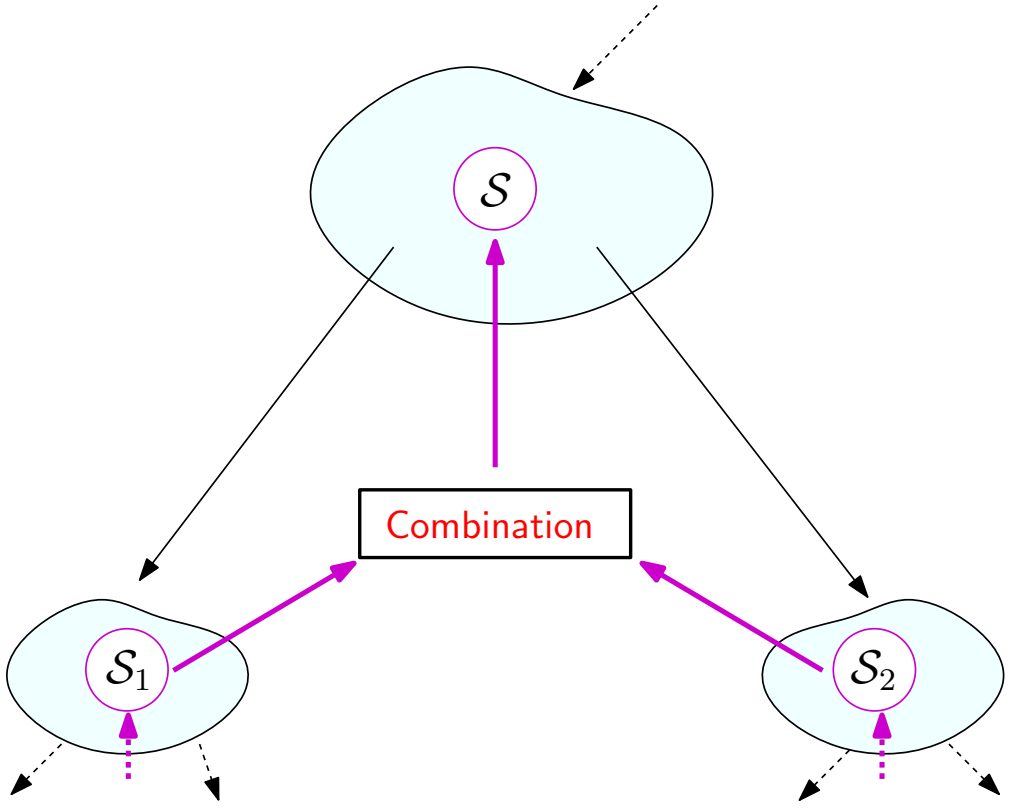


The computation is trivial for leaves.

Intuition: conquer

To compute a set of partial solutions of an internal cut:

→ We combine the sets of partial solutions of its children.

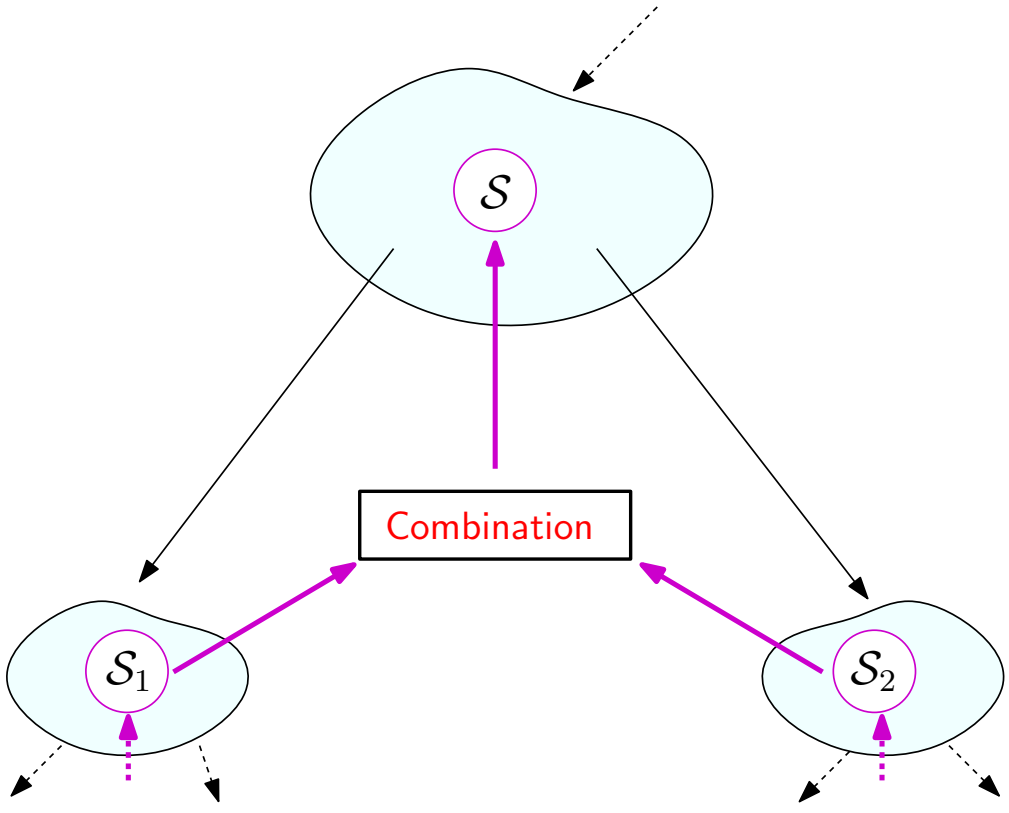


It is not trivial but rarely complicated.

Intuition: conquer

To compute a set of partial solutions of an internal cut:

→ We combine the sets of partial solutions of its children.

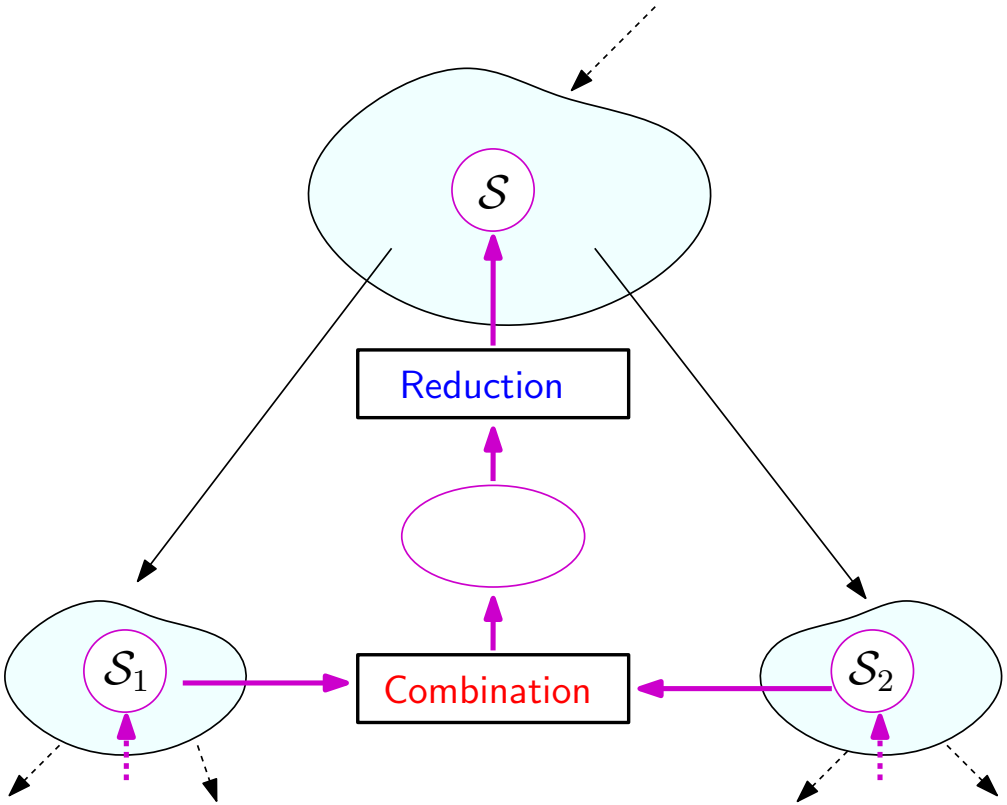


In general: running time $= O(|\mathcal{S}_1| \cdot |\mathcal{S}_2|)$.

Key step: reduction

Combining increase the size of the manipulated sets.

→ We need to reduce the size of the combination.



Key step: reduction

We need to upper-bound the number of partial solutions we need to keep.

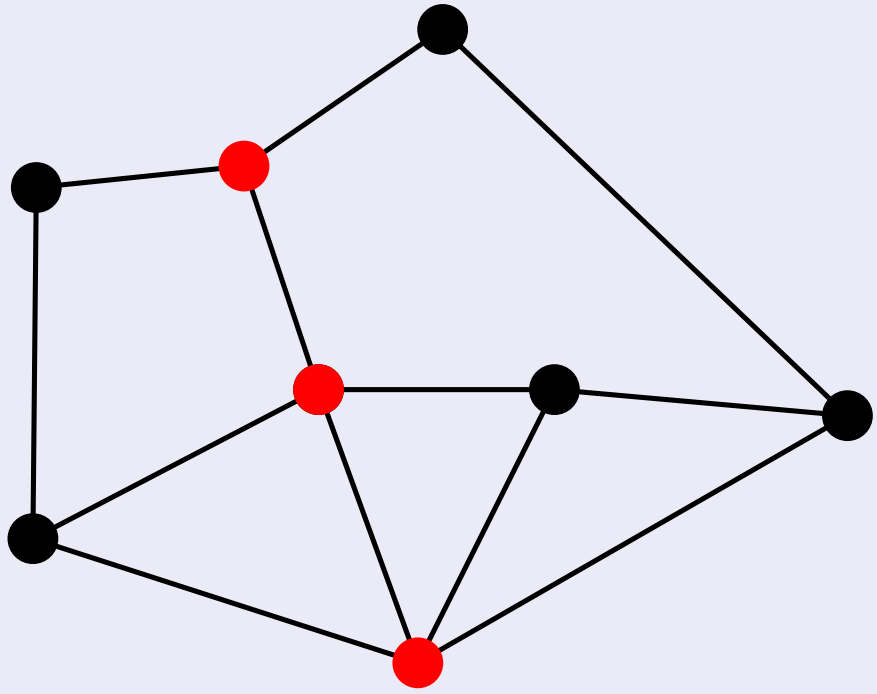
- If the upper bound is $f(k) \cdot n^{g(k)} \Rightarrow$ an XP algorithm.
- If the upper bound is $f(k) \cdot n^{O(1)} \Rightarrow$ an FPT algorithm.

Connected Dominating set and 1-neighbor width

Connected Dominating set and 1-neighbor width

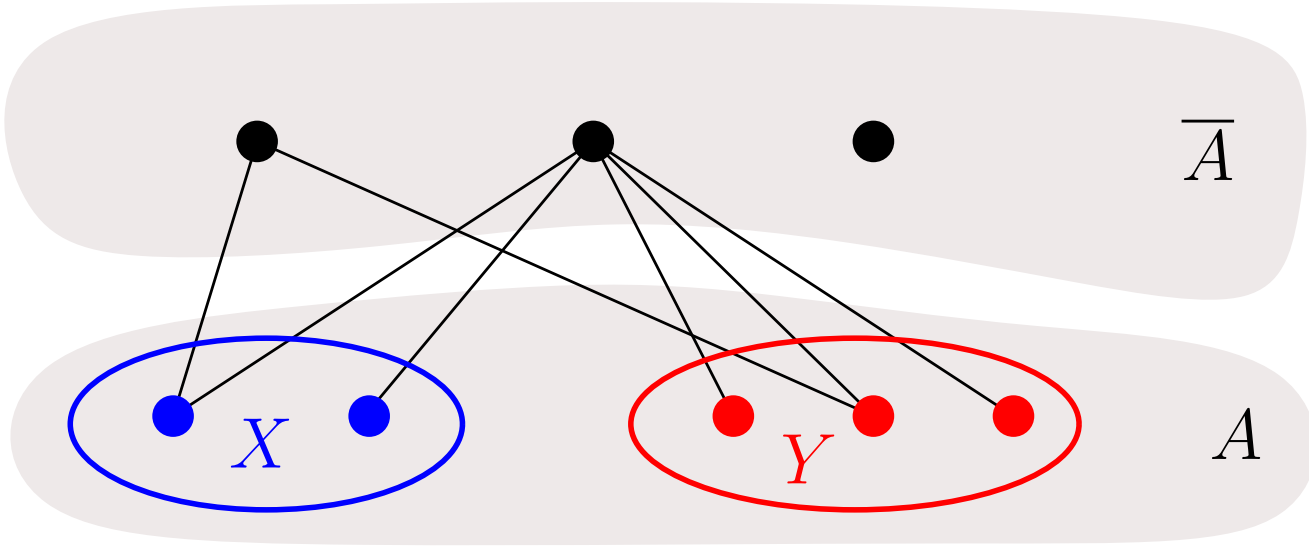
Connected dominating set

Finding a vertex set D of minimum weight which dominates all the vertices and which induces a connected graph.



1-neighbor equivalence relation

$X, Y \subseteq A$ are 1-neighbor equivalent in A if $N(X) \cap \overline{A} = N(Y) \cap \overline{A}$.



1-neighbor width

Defined from the function $\mathrm{nec}_1(A)$:= the number of equivalence classes.

Results

Theorem [Bui-Xuan, Telle and Vatshelle, 2013]

Dominating set is solvable in time $\mathsf{nec}_1(\mathcal{L})^{O(1)} \cdot n^{O(1)}$.

Theorem [B. and Kanté 2018]

Connected dominating set is solvable in time $\mathsf{nec}_1(\mathcal{L})^{O(1)} \cdot n^{O(1)}$.

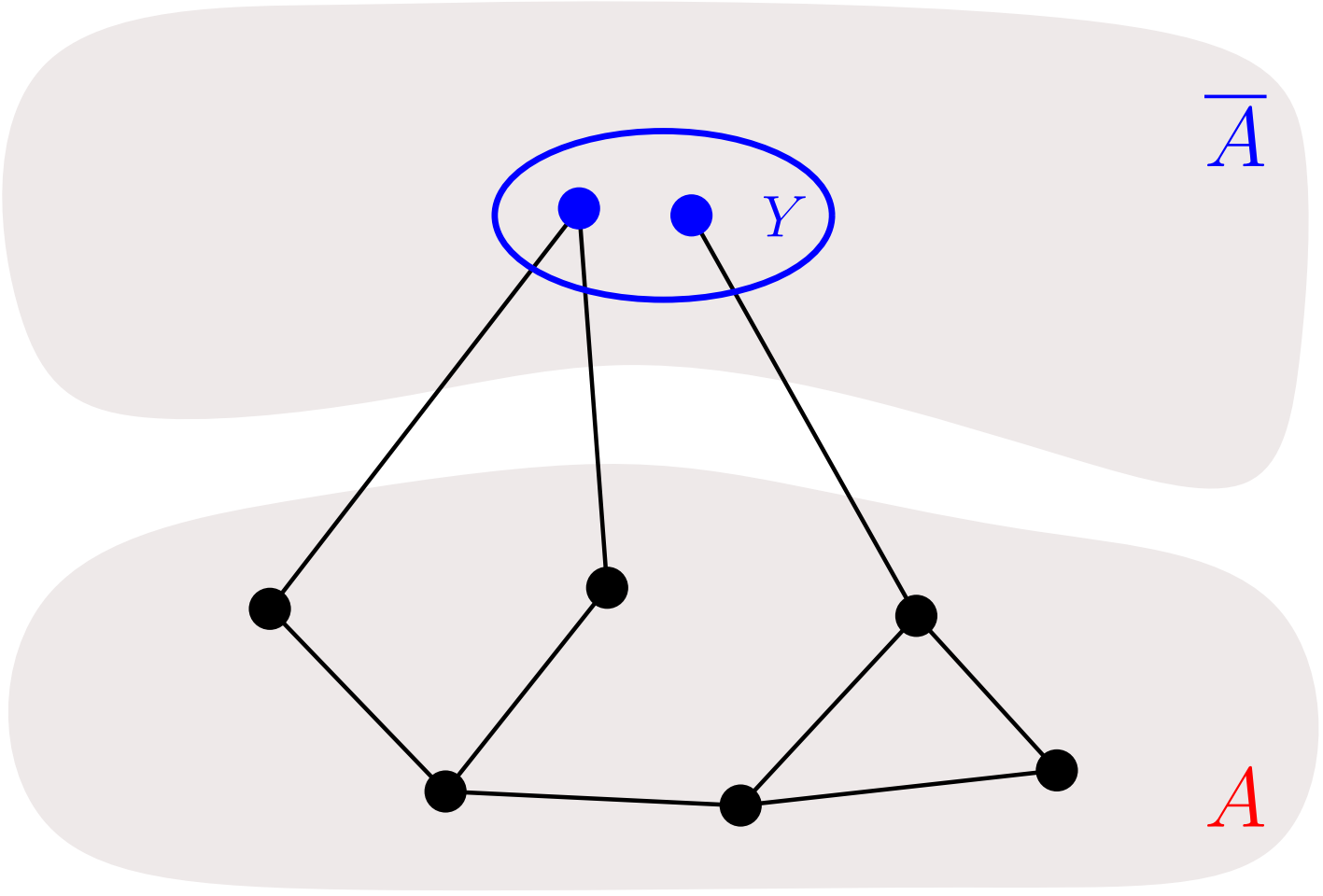
Consequences on the other width measures:

Lemme

A $\mathsf{nec}_1(\mathcal{L})^{O(1)} \cdot n^{O(1)}$ time algorithm $\Rightarrow$ we can solve the problem in time: $2^{O(\mathsf{tw})} \cdot n^{O(1)}$, $2^{O(\mathsf{cw})} \cdot n^{O(1)}$, $2^{O(\mathsf{rw}^2)} \cdot n^{O(1)}$ and $n^{O(\mathsf{mim})}$.

Dealing with domination

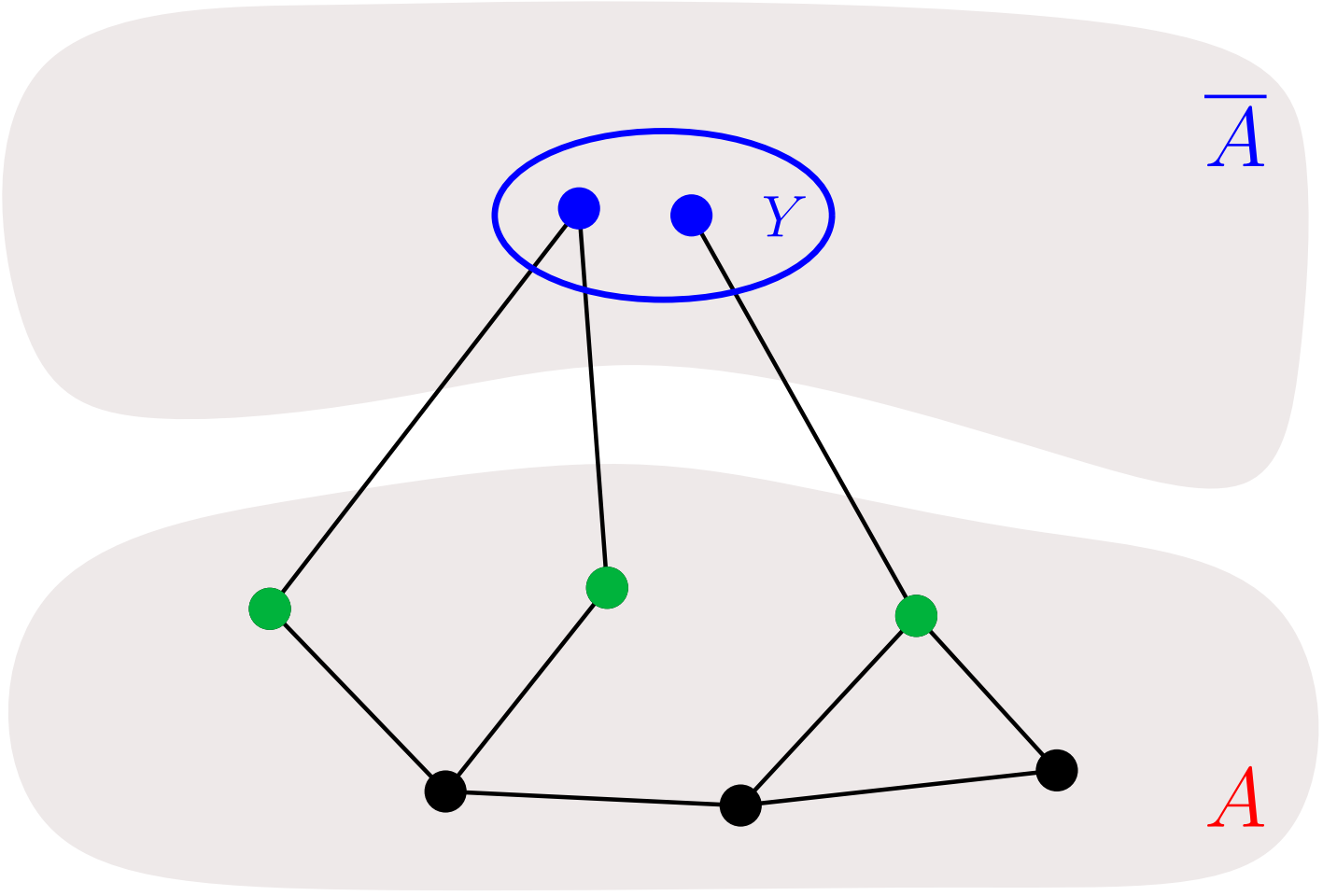
We compute a set of partial solutions for each equivalence class R' of the 1-neighbor equivalence over $\overline{A}$.



Dealing with domination

The sets $Y \in R'$ have the same neighborhood in A.

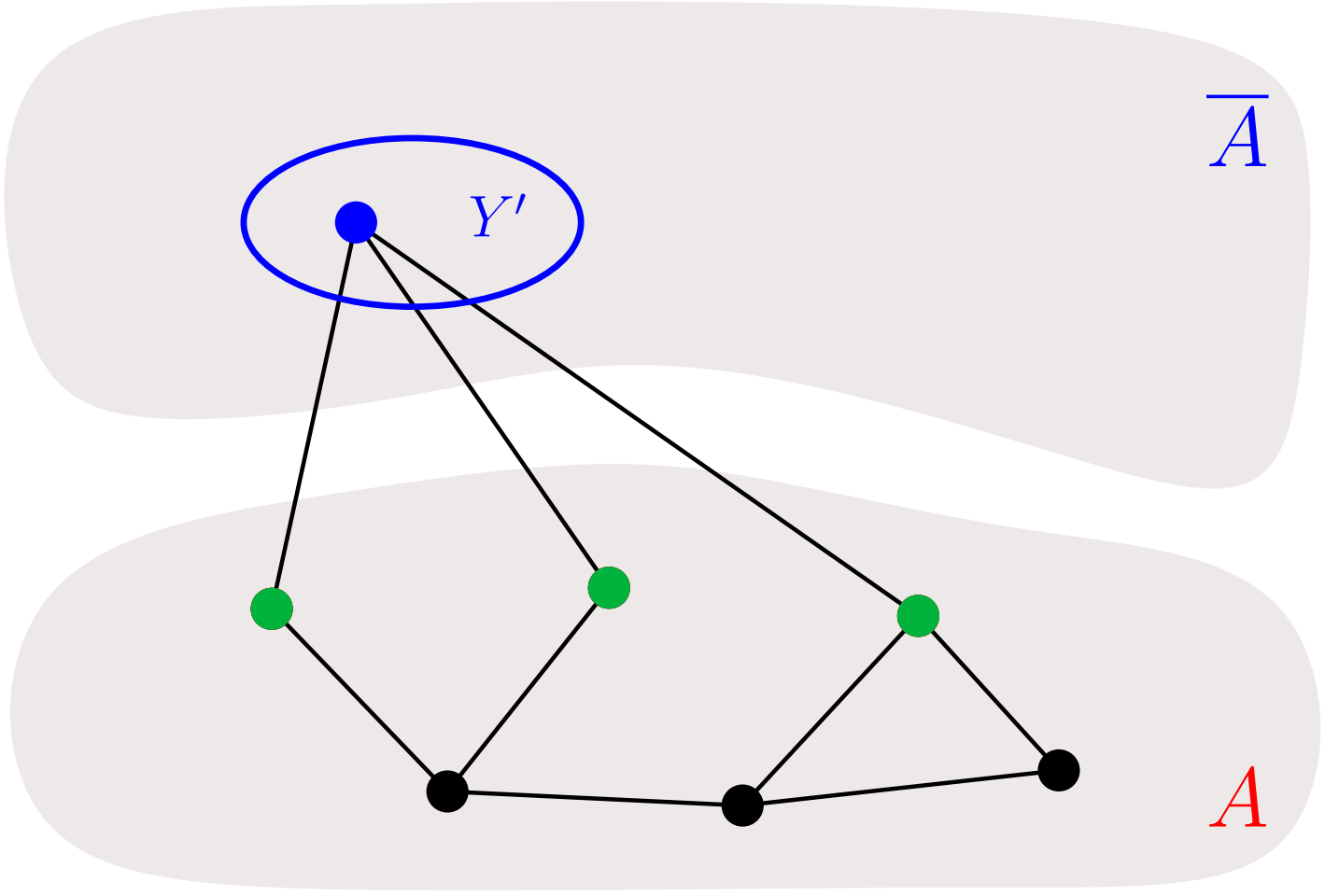
$\rightarrow$ We know which vertices in A will be dominated.



Dealing with domination

The sets $Y \in R'$ have the same neighborhood in A.

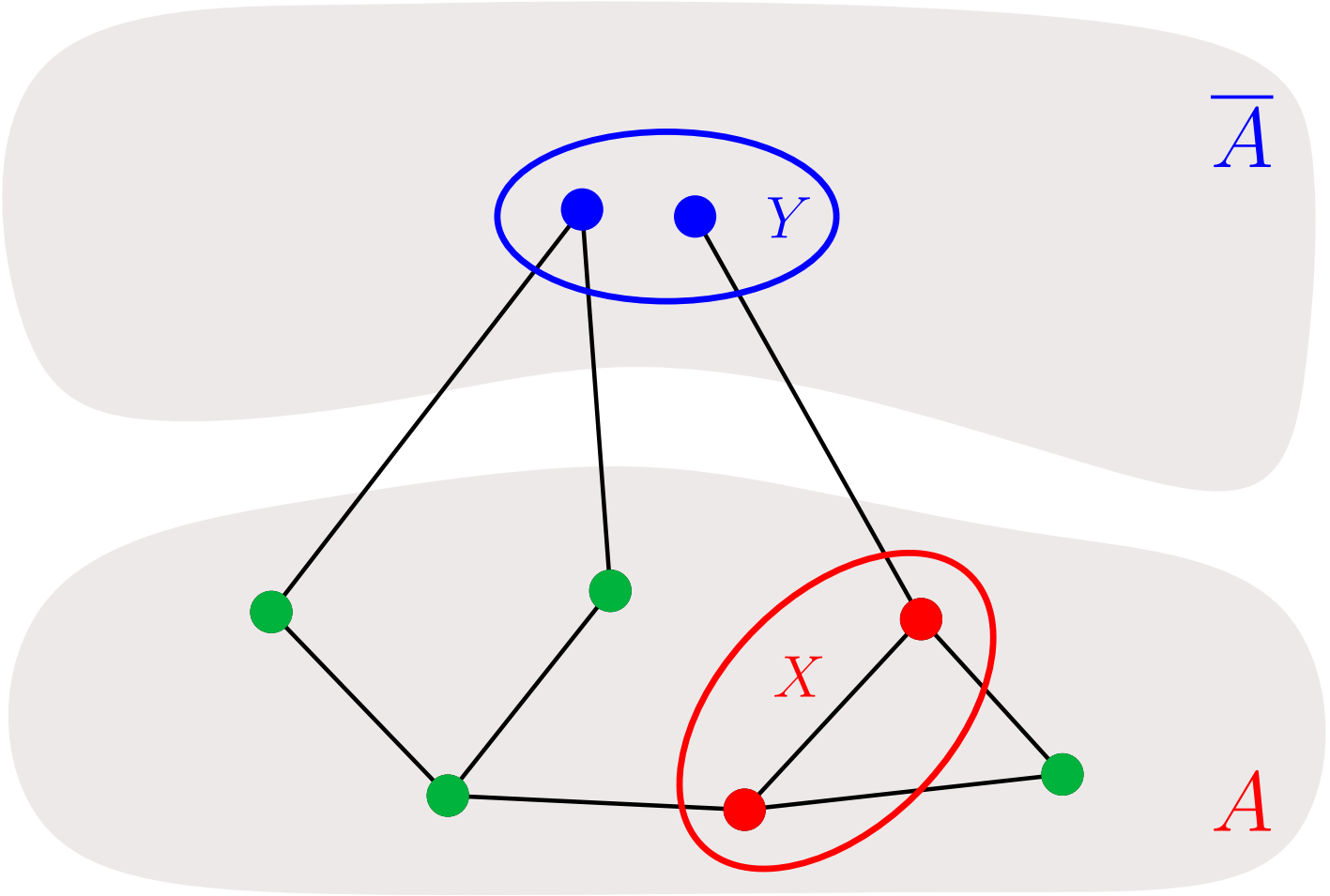
$\rightarrow$ We know which vertices in A will be dominated.



Dealing with domination

Partial solutions associated with R' :

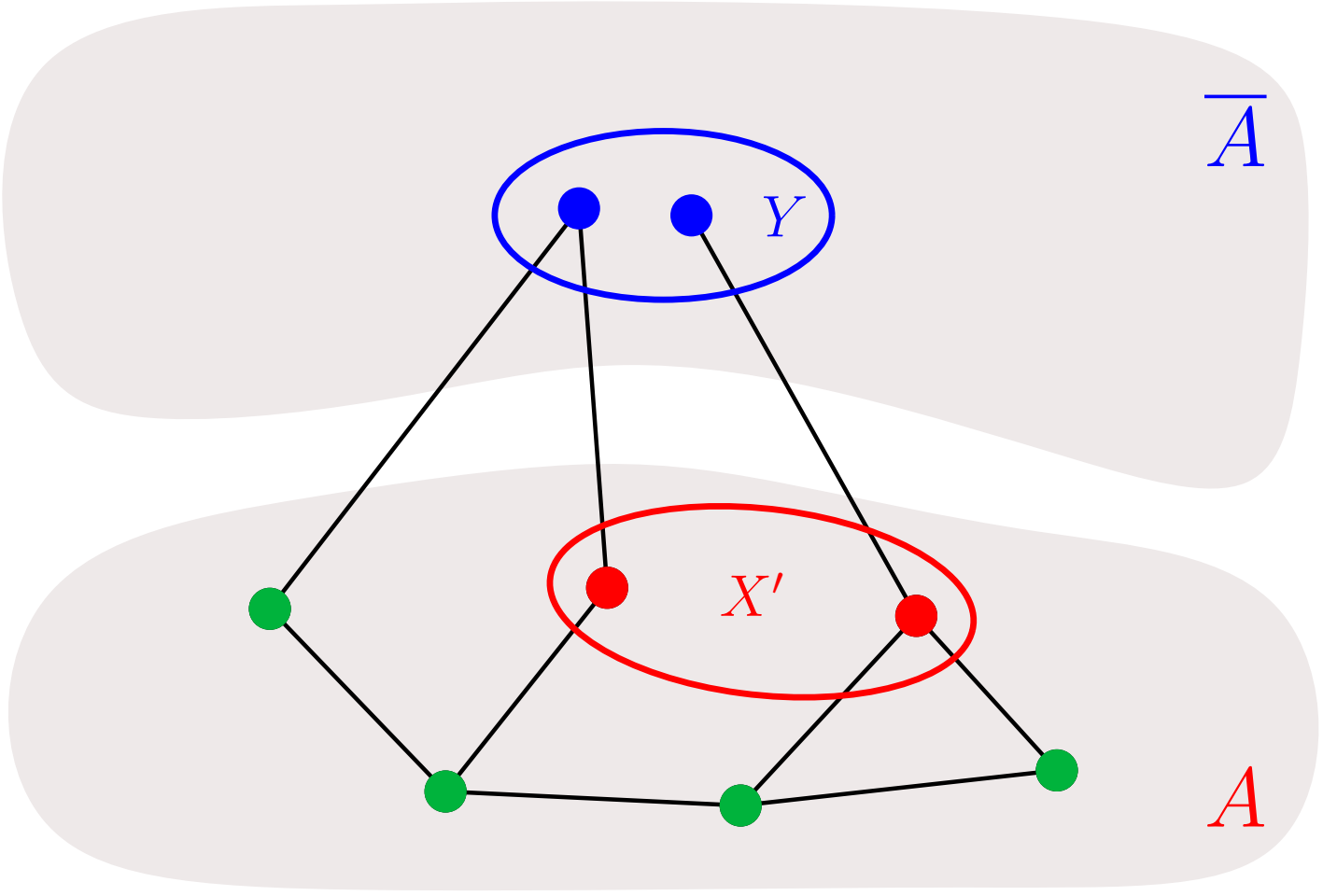
$\rightarrow X \subseteq A$ such that $X \cup Y$ dominates A, for (all) $Y \in R'$



Dealing with domination

Partial solutions associated with R' :

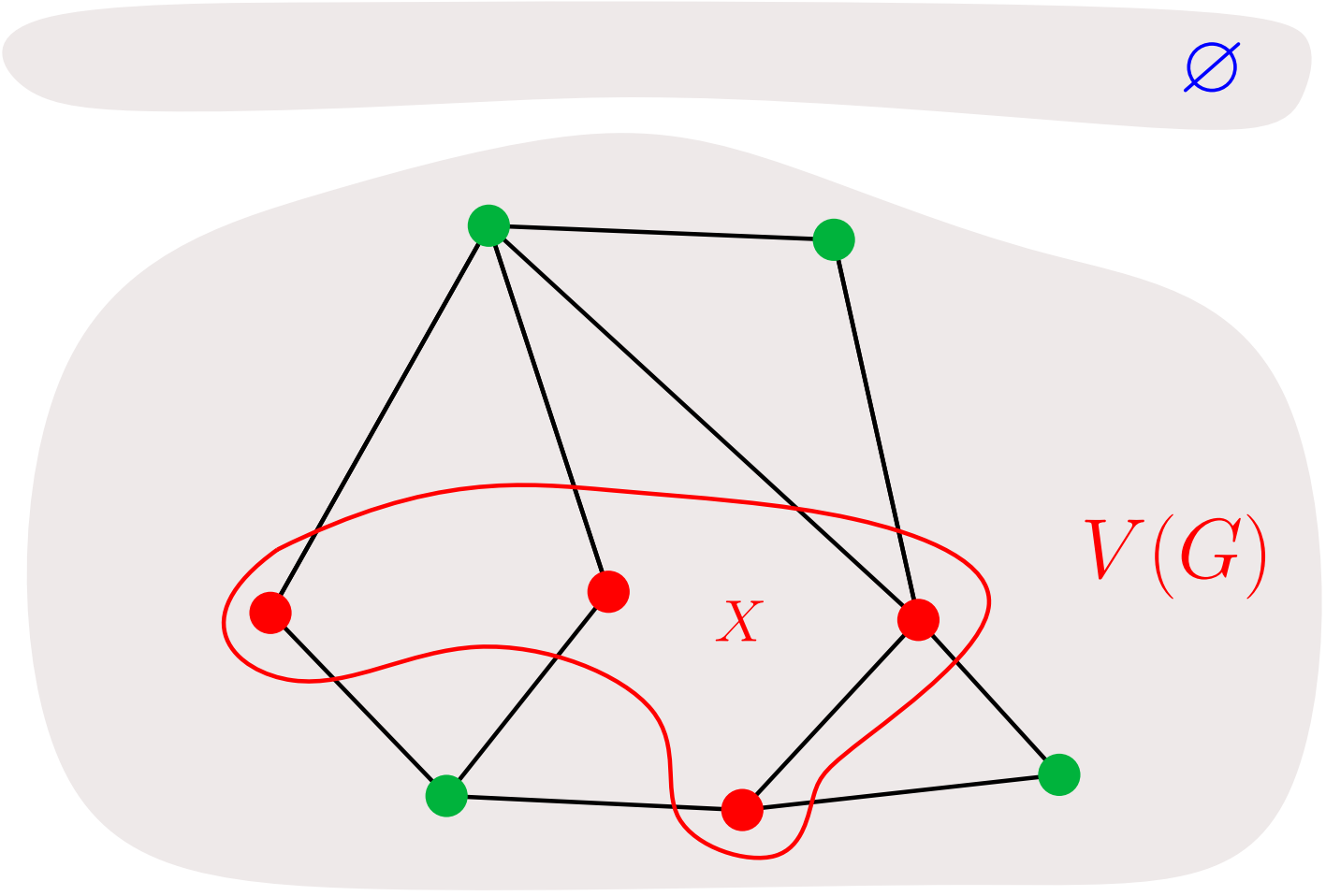
$\rightarrow X \subseteq A$ such that $X \cup Y$ dominates A, for (all) $Y \in R'$



Dealing with domination

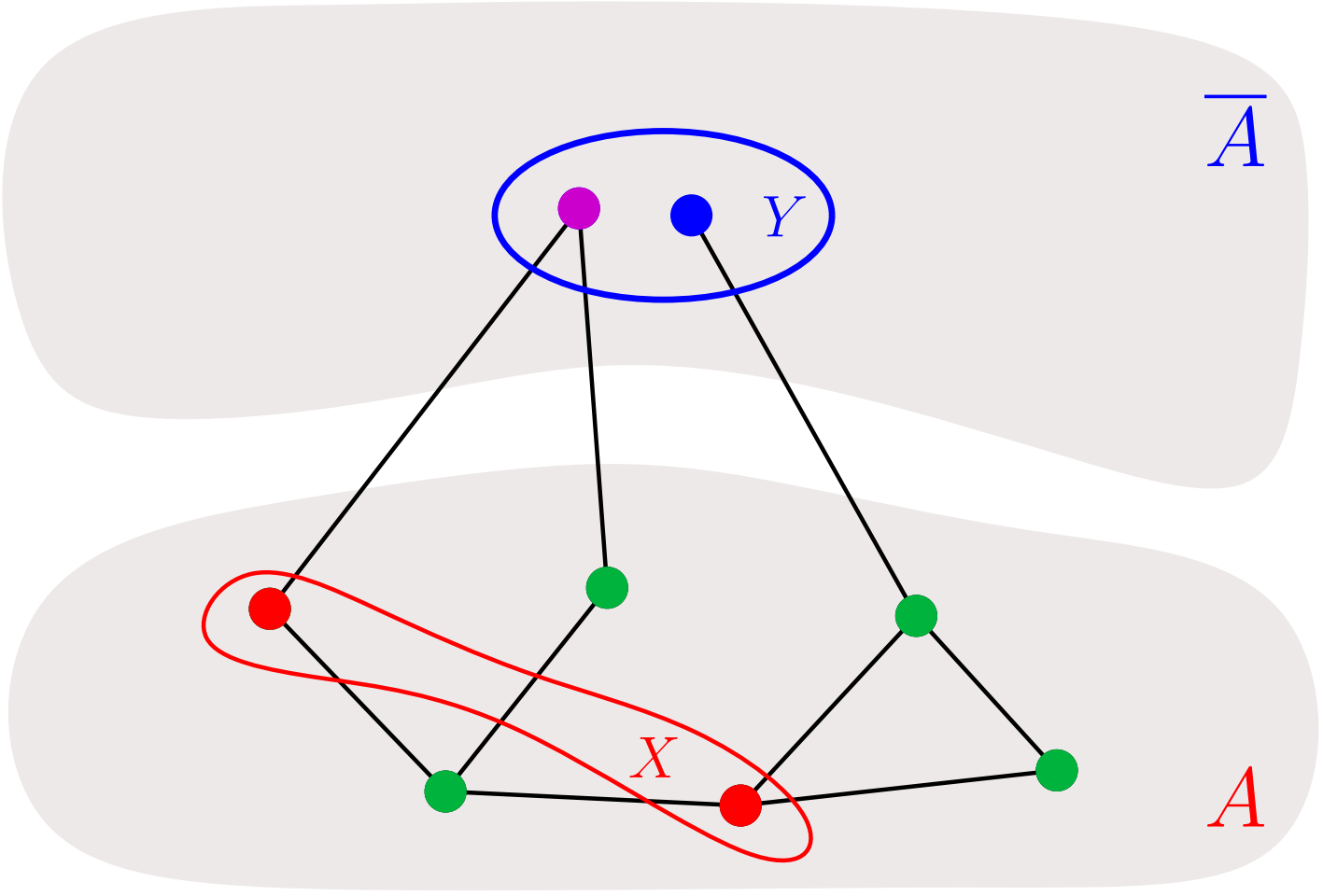
At the root of the decomposition, the cut is $(V(G), \varnothing)$:

$\rightarrow$ A partial solution is a dominating set.



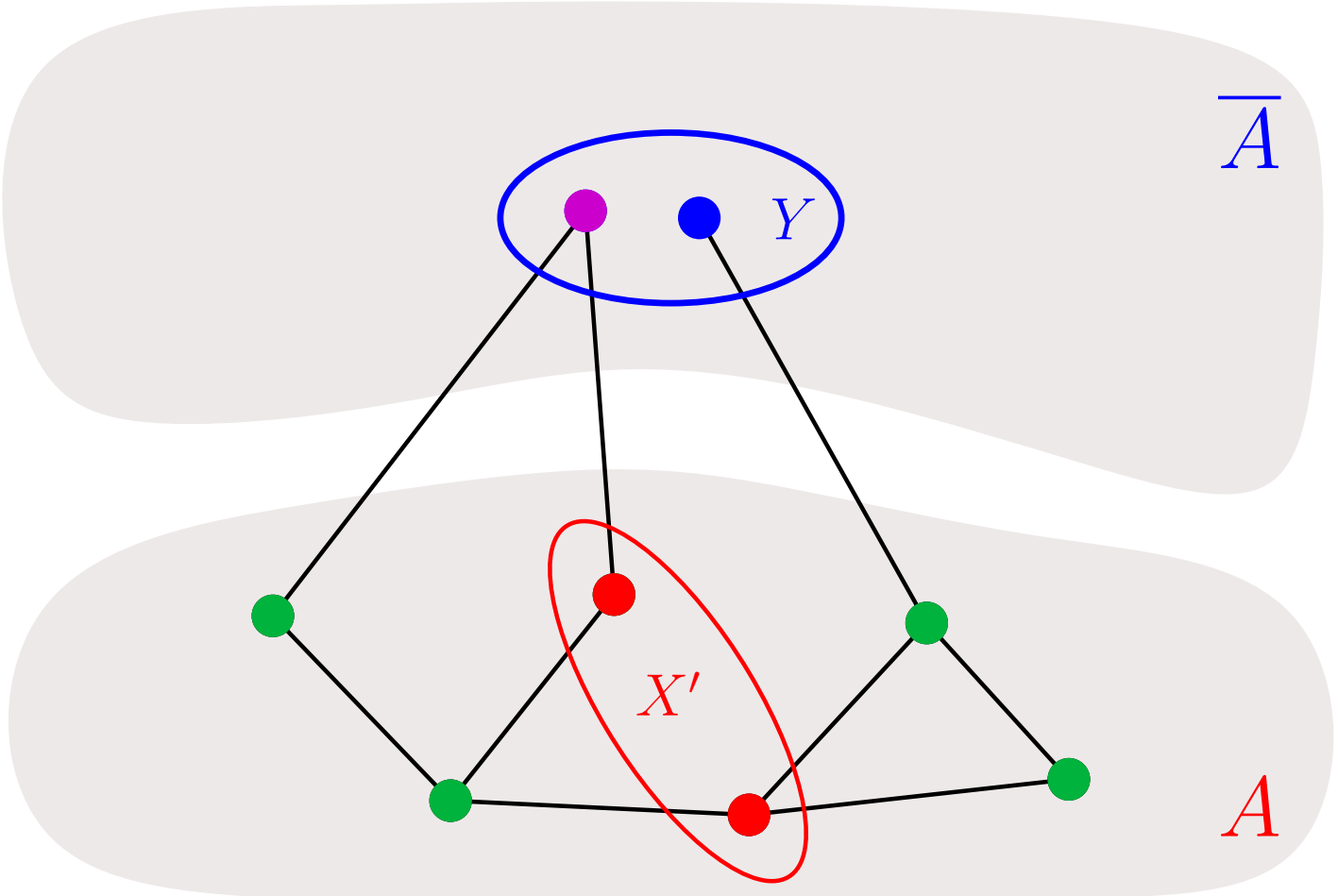
Dealing with domination

Two partial solutions associated with R' are equivalent for the domination if they are 1-neighbor equivalent!



Dealing with domination

Two partial solutions associated with R' are equivalent for the domination if they are 1-neighbor equivalent!



Lemma [Bui-Xuan, Telle and Vatshelle, 2013]

To solve Dominating set, it is enough to keep $\mathrm{nec}_1(A)^2$ partial solutions at each step.

One partial solution:

- for each 1-neighbor equivalence class R' of $\overline{A}$, and
- for each 1-neighbor equivalence class R of A

Dealing with connectivity

- We need an equivalence relation between sets of partial solutions.

Dealing with connectivity

- We need an equivalence relation between sets of partial solutions.

- For all $Y \in R'$ and for all sets of partial solutions $\mathcal{S}$, we define:

$$\text{best}(\mathcal{S}, Y) := \min\{\text{weight}(X) : X \in \mathcal{S} \text{ and } G[X \cup Y] \text{ is connected}\}.$$

R'-representativity

We say that $\mathcal{S}^\star$ R'-represents $\mathcal{S}$ if, for all $Y \in R'$, we have

$$\text{best}(\mathcal{S}, Y) = \text{best}(\mathcal{S}^\star, Y).$$

Representative set

- At the root $(V(G), \varnothing)$: an $\{\varnothing\}$-representative set of the set of all partial solutions must contain an optimal solution.

  $\rightarrow$ best$(\mathcal{S}, \varnothing) :=$ min$\{$weigth$(X) : X \in \mathcal{S}$ and $G[X]$ is connected$\}$.

Representative set

- At the root $(V(G), \varnothing)$: an $\{\varnothing\}$-representative set of the set of all partial solutions must contain an optimal solution.

  $\rightarrow$ best$(\mathcal{S}, \varnothing) := \min\{\text{weigth}(X) : X \in \mathcal{S}$ and $G[X]$ is connected$\}$.

Theorem

There exists a function reduce:

- Input: a set of partial solutions $\mathcal{S}$.
- Output: $\mathcal{S}^\star \subseteq \mathcal{S}$ such that $|\mathcal{S}^\star| \leq \text{nec}_1(\mathcal{L})^2$ and $\mathcal{S}^\star$ R'-represents $\mathcal{S}$.
- Running time: $|\mathcal{S}| \cdot \text{nec}_1(\mathcal{L})^{O(1)} \cdot n^2$.

Sketch of proof

Inspiration [Bodlaender et al. 2013]

Rank based approach: technique to obtained $2^{O(\mathsf{tw})} \cdot n$ time algorithms for many connectivity problems.

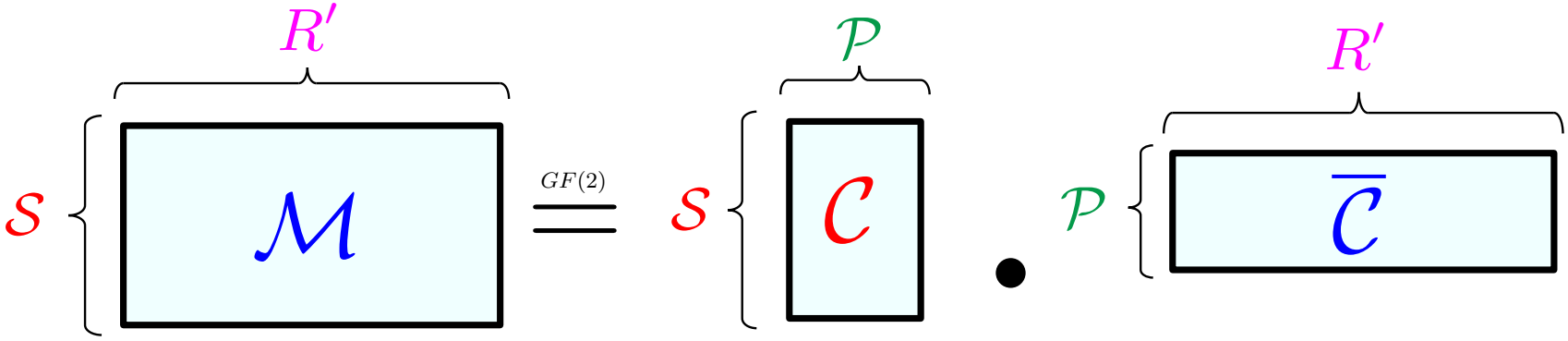
Let $\mathcal{M}$ be the $(\mathcal{S}, R')$-matrix:

$$\mathcal{M}[X, Y] := \begin{cases} 1 & \text{if } G[X \cup Y] \text{ is connected,} \\ 0 & \text{otherwise.} \end{cases}$$

- In $GF(2)$: a basis of the row space of $\mathcal{M}$ of minimum weight R'-represents $\mathcal{S}$.
- But $\mathcal{M}$ is too big to be computed.

Sketch of proof

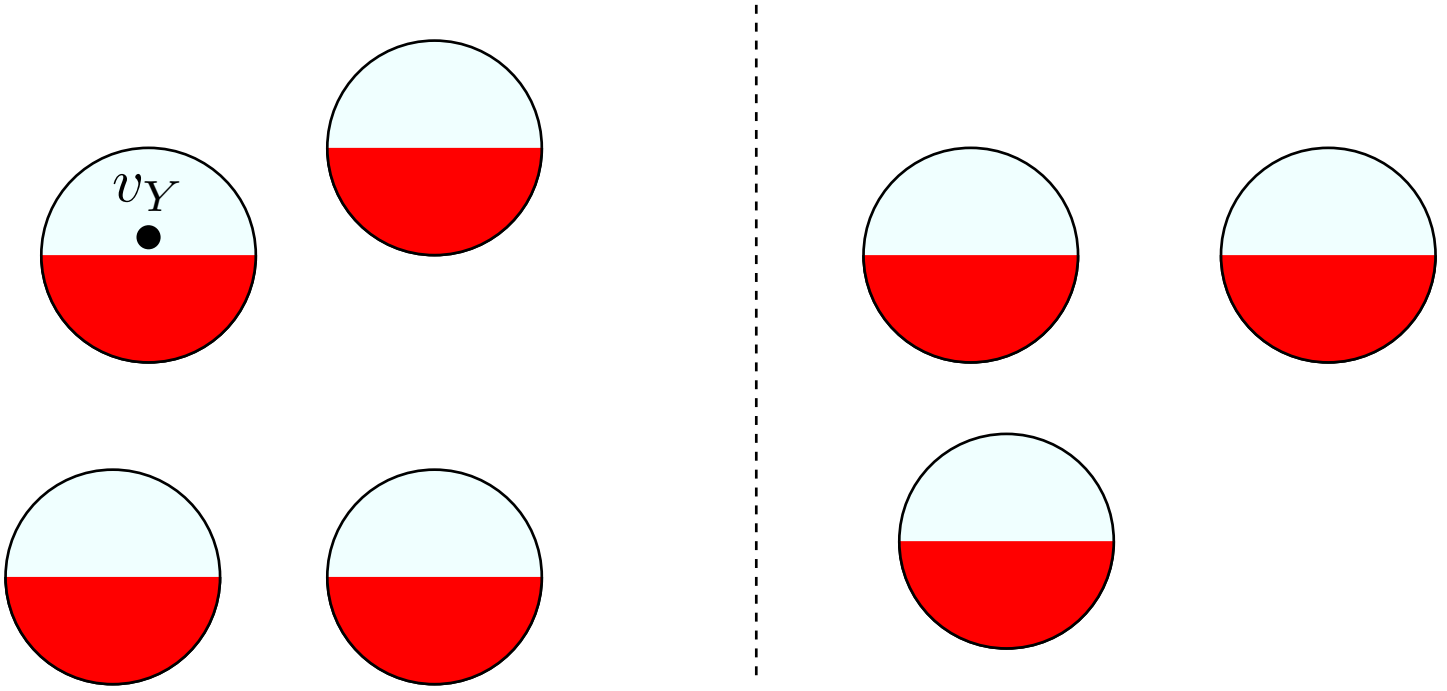
$\mathcal{P}$: the set of pairs (R'_1, R'_2) of 1-neighbor equivalence classes in $\overline{A}$.



- A basis of $\mathcal{C}$ is also a basis of $\mathcal{M}$.
- $|\mathcal{P}| = \mathsf{nec}_1(A)^2$.
- $\mathcal{C}$ is computable in time $|\mathcal{S}| \cdot |\mathcal{P}| \cdot n^2$.

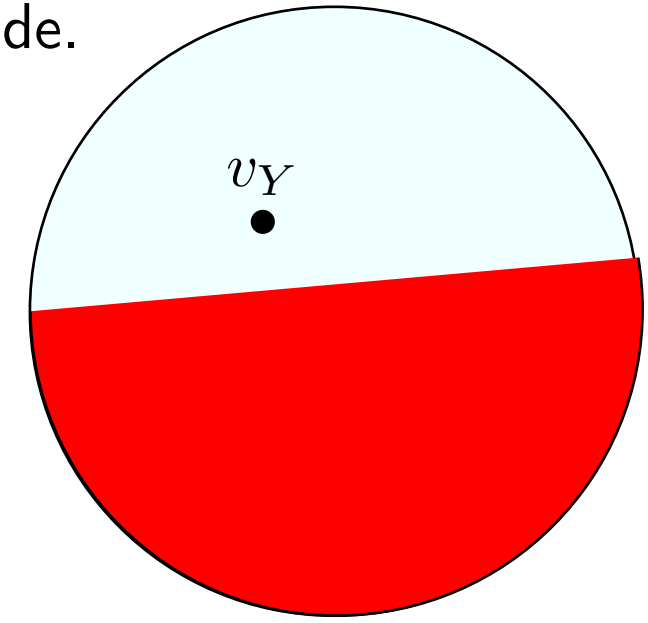
Trick

There is $2^{|CC(X \cup Y)|-1}$ ways of dividing $X \cup Y$ such that v_Y is on one side.



Trick

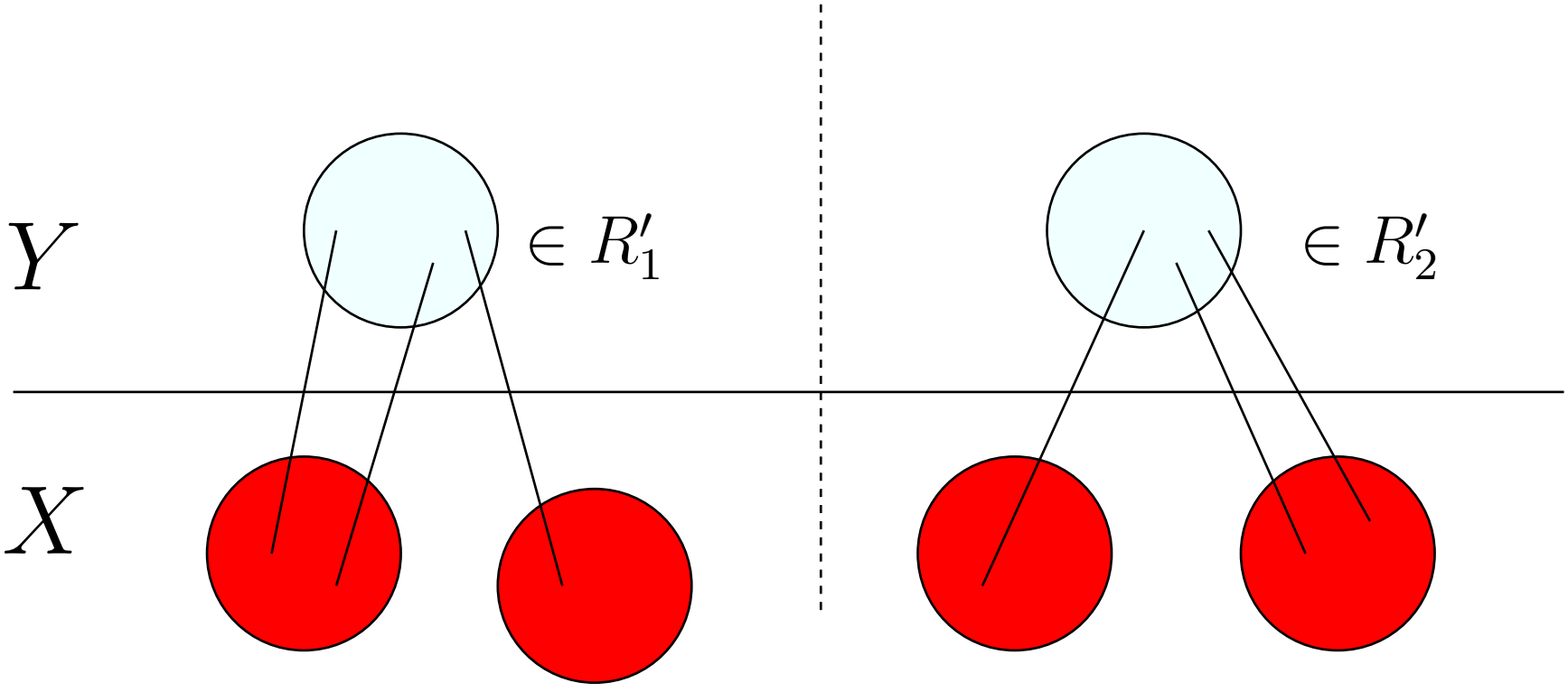
There is $2^{|CC(X \cup Y)|-1}$ ways of dividing $X \cup Y$ such that v_Y is on one side.



$$\mathcal{C} \cdot \overline{\mathcal{C}}[X, Y] = 2^{|CC(X \cup Y)|-1}$$

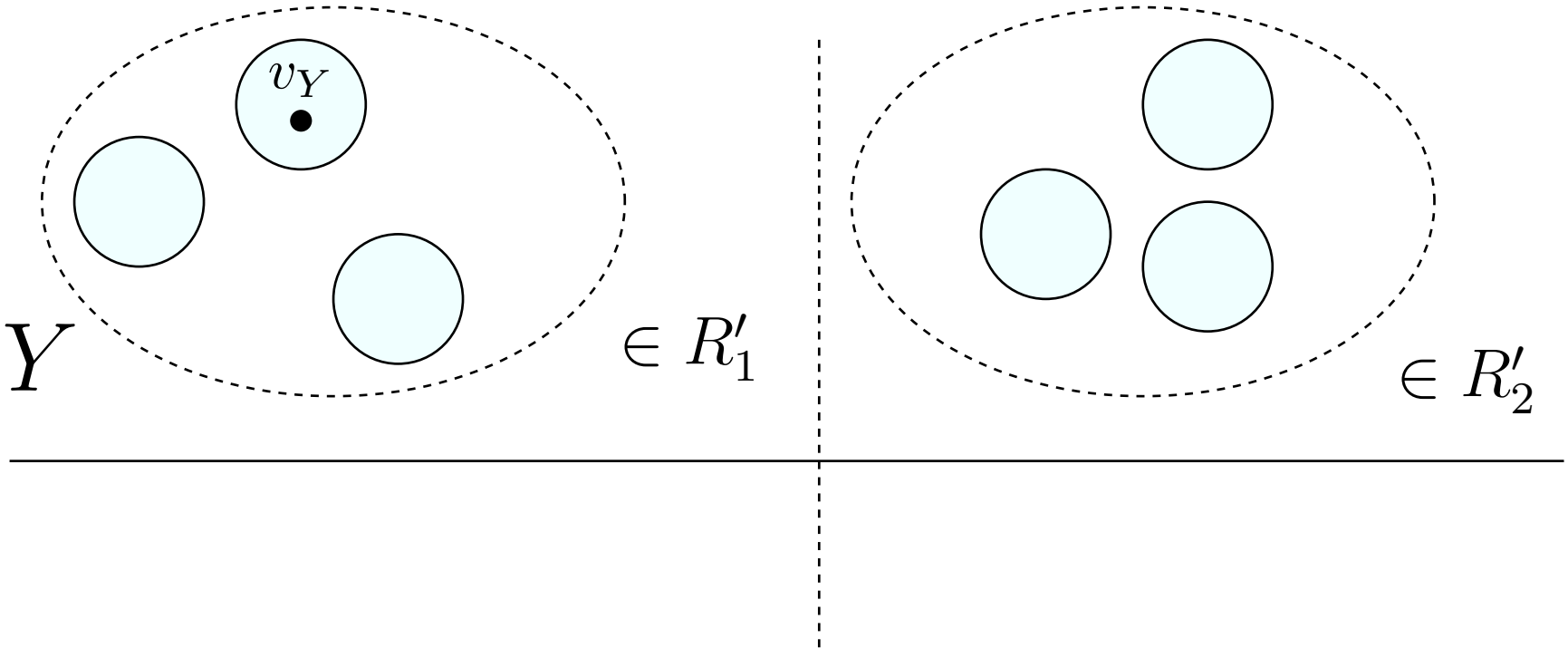
Trick

$\mathcal{C}[X, (R'_1, R'_2)]$=1 if and only if



Trick

$\overline{\mathcal{C}}[(R'_1, R'_2), Y]$=1 if and only if



Trick

$$\mathcal{C} \cdot \overline{\mathcal{C}}[X, Y] = 2^{|CC(X \cup Y)|-1}$$

$$\mathcal{C} \cdot \overline{\mathcal{C}} \overset{GF(2)}{=} \mathcal{M}$$

Generalization

We can generalize this to other local constraints.

→ d-neighbor equivalence: nec_d with $d \in \mathbb{N}$.

Theorem [B. and Kanté 2018]

We can solve the connected variants of (σ, ρ)-Dominating Set problems in time $\mathsf{nec}_d(\mathcal{L})^{O(1)} \cdot n^{O(1)}$.

→ Connected Vertex Cover, Node-Weighted Steiner Tree,....

Lemma

A $\mathsf{nec}_d(\mathcal{L})^{O(1)} \cdot n^{O(1)}$ time algorithm ⇒ we can solve the problem in time $2^{O(\mathsf{tw})} \cdot n$, $2^{O(\mathsf{cw})} \cdot n$, $2^{O(\mathsf{rw}^2)} \cdot n^3$ and $n^{O(\mathsf{mim})}$.

Acyclicity

Theorem [B. and Kanté 2018]

We can solve in time: $2^{O(\mathsf{tw})} \cdot n$, $2^{O(\mathsf{cw})} \cdot n$, $2^{O(\mathsf{rw}^2)} \cdot n^3$ and $n^{O(\mathsf{mim})}$ the acyclic and the acyclic $\wedge$ connected variants of any (σ, ρ)-Dominating Set problem

→ Feedback Vertex Set, Maximum Induced Linear Forest,...

→ Maximum Induced Tree, Longest Induced Path,...

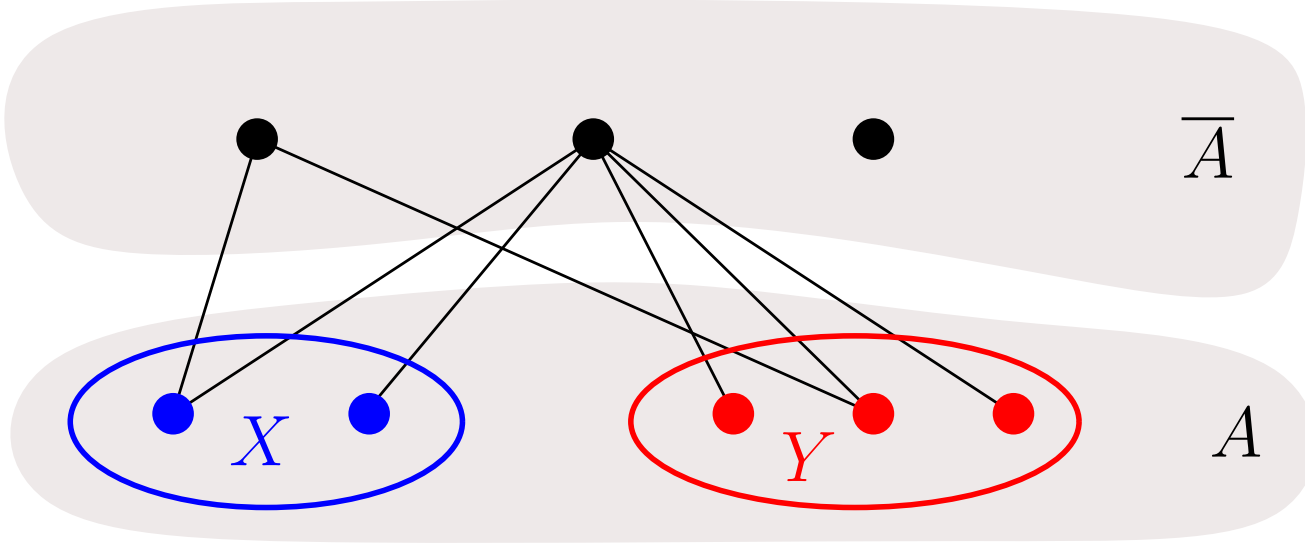
- No $\mathsf{nec}_d(\mathcal{L})^{O(1)} \cdot n^{O(1)}$ time algorithms (open question).

  → At some point, we had to use the specificity of the different width measures...

d-neighbor equivalence relation

$X, Y \subseteq A$ are d-neighbor equivalent in A if for all $v \in \overline{A}$, we have

$$\min(d, |N(v) \cap X|) = \min(d, |N(v) \cap Y|).$$

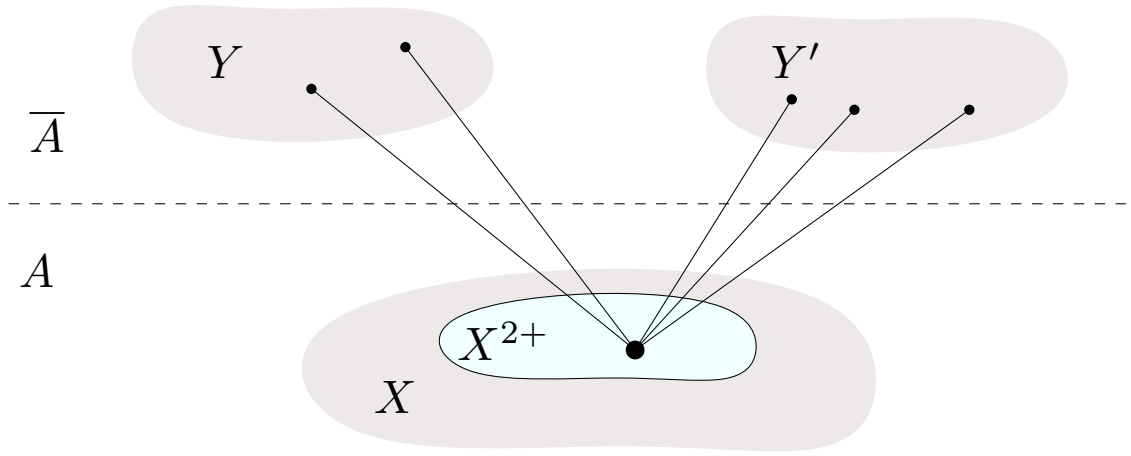


d-neighbor width

Defined from the function $\mathrm{nec}_d(A)$:= maximum between the number of equivalence classes of the d-neighbor equivalence over A and $\overline{A}$.

Maximum Induced Tree vs 2-neighbor equivalence

We compute a set of partial solutions for every 2-neighbor equivalence class R'.



We know the vertices that will have at least two neighbors.

Lemma

If X leads to a solutions, then $|X^{2+}|$ is upper bounded by $2\mathsf{mim}(A)$, $2\mathsf{rw}(A)$, $\mathsf{mw}(A)$.

New notion of representativity

Let R' be a 2-neighbor equivalence class.

- For all $Y \in R'$ and for all sets of partial solutions $\mathcal{S}$, we define:

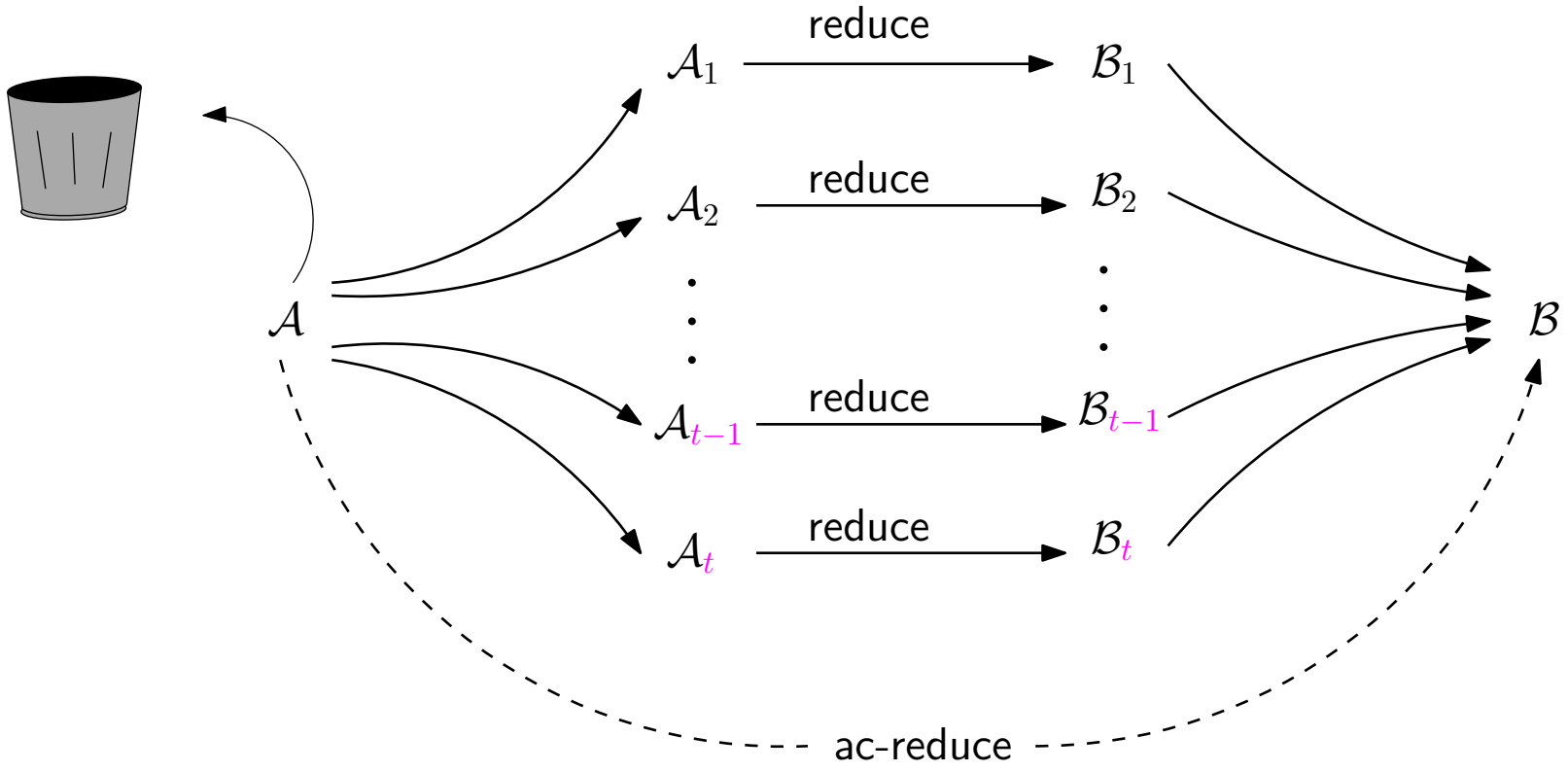
$$\text{ac-best}(\mathcal{S}, Y) := \min\{\text{weight}(X) : X \in \mathcal{S} \text{ and } G[X \cup Y] \text{ is a tree}\}.$$

Definition R'-ac-representativity

We say that $\mathcal{S}^\star$ R'-ac-represents $\mathcal{S}$ if for all $Y \in R'$, we have

$$\text{ac-best}(\mathcal{S}, Y) = \text{ac-best}(\mathcal{S}^\star, Y).$$

Manage acyclicity



- With $t \leq 2^{\mathsf{mw}+1} \cdot n$ or $2^{2\mathsf{rw}^2+1} \cdot n$ or $2n^{2\mathsf{mim}+1}$.

Overview

Thanks to the d-neighbor equivalence, we obtained the best algorithms:

- for many problems

  → (σ, ρ)-Dominating Set problems and their variants.

- for many width-measures.

  → clique-width, rank-width, mim-width, $\mathbb{Q}$-rank-width and tree-width...

The algorithms we get for clique-width and tree-width are optimal under ETH.

→ What about rank-width: $2^{O(\mathsf{rw}^2)} \cdot n^{O(1)}$ is optimal?

Hamiltonian cycle, Max Cut and Edge Dominating Set

Can we use the d-neighbor equivalence for W[1]-hard problems parameterized by clique-width?

- We can solve these 3 problems in time $2^{O(\mathsf{tw})} \cdot n$ and $n^{O(\mathsf{cw})}$.
  $\rightarrow$ Optimal under ETH.
- Using the d-neighbor equivalence width d a constant is useless.

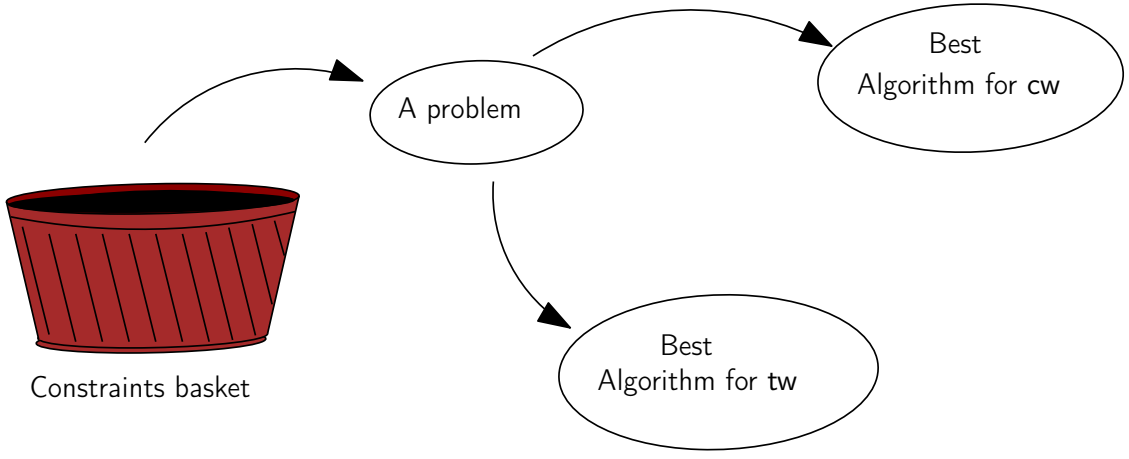
Lemma

If we can solve a problem in time $\mathsf{nec}_n(\mathcal{L})^{O(1)} \cdot n^{O(1)}$, then we can solve it in time $n^{O(\mathsf{cw})}$, $n^{O(\mathsf{rw}_{\mathbb{Q}})}$ and $n^{2^{O(\mathsf{rw})}}$.

- We can solve Max Cut in time $\mathsf{nec}_n(\mathcal{L})^{O(1)} \cdot n^{O(1)}$.

The dream

Towards a "Courcelle's theorem" which gives efficient algorithms?



Example:

We want a set of vertices X of minimum weight satisfying:

$$\mathsf{DominatingSet}(X) \ \wedge \ \mathsf{Acyclic}(X) \ \wedge \ \mathsf{Connected}(X).$$

Thm $\Rightarrow$ We can find one in time: $2^{O(\mathsf{tw})} \cdot n$, $2^{O(\mathsf{cw})} \cdot n$, $2^{O(\mathsf{rw}^2)} \cdot n^3$.